

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

Resumen	IV
Resumen	V
Agradecimientos	VI
Índice General	VII
Índice de Figuras	IX
Índice de Tablas	X
Lista de acrónimos	XII
1. Introducción y Objetivos	1
1.1. Estado del arte	1
1.2. Redes WSN	3
1.2.1. Redes LPWAN	4
1.2.2. LoRa	5
1.3. Objetivos y motivaciones	7
1.4. Metodología	7
2. Diseño hardware	9
2.1. Herramientas de diseño	9
2.2. Elección del hardware	9
2.2.1. Integrado CMWX1ZZABZ	9
2.2.1.1. Microcontrolador STM32L072	10
2.2.1.2. Transceptor SX1276	10
2.2.2. Conversor analógico/digital AD7194	11
2.2.3. Sensor de temperatura MCP9808T	12
2.2.4. Componentes activos	12
2.2.5. Componentes pasivos	13
2.2.5.1. Termopares	13
2.3. Diseño las tarjetas	14
2.3.1. Diseño del nodo básico	14
2.3.1.1. Esquemáticos	15
2.3.1.2. Layout	16
2.3.2. Diseño de la tarjeta de expansión	17
2.3.2.1. Medida de los termopares	17

2.3.2.2.	Medida de entradas analógicas	18
2.3.2.3.	Media del nivel de batería	19
2.3.2.4.	Alimentación de los componentes	19
2.3.2.5.	Esquemáticos	21
2.3.2.6.	Layout	22
2.4.	Lista de componentes	23
2.5.	Fabricación y montaje	23
3.	Diseño software	25
3.1.	Diseño del protocolo	25
3.1.1.	Formato de trama	26
3.1.1.1.	Trama de conexión	26
3.1.1.2.	Trama de configuración	27
3.1.1.3.	Trama de datos	28
3.1.2.	Comportamiento del protocolo	28
3.1.2.1.	Conexión de un nodo a la red	28
3.1.2.2.	Envío y recepción de datos	28
3.1.2.3.	Tratamiento de errores	29
3.2.	Implementación	30
3.2.1.	Fundamentos teóricos	30
3.2.1.1.	Sistemas embebidos	30
3.2.1.2.	Sistemas operativos de tiempo real (RTOS)	30
3.2.1.3.	Máquinas de estados extendidas (xFSM)	30
3.2.1.4.	Colas circulares	30
3.2.2.	Descripción del entorno	30
3.2.2.1.	Drivers, Librerías y útiles	30
3.2.2.2.	Herramientas de edición, compilación y depuración	31
3.2.3.	Modelado del comportamiento de los nodos	32
3.2.4.	Modelado del comportamiento del nodo maestro	33
3.2.5.	Estructura del código	34
3.2.6.	Proceso de diseño e implementación	34
3.2.7.	Resultados obtenidos	34
4.	Conclusiones y líneas futuras	35
4.1.	Conclusiones	35
	Bibliografía	37

Índice de figuras

1.1. Parámetros fundamentales de un sistema inalámbrico	2
1.2. Relación de consumo, velocidad y alcance de las diferentes tecnologías [Friedli (2016)]	3
1.3. Topologías de red más usadas	3
1.4. [(SEMTECH, 2015)]	5
1.5. [Telkamp (2015)]	6
2.1. Componentes de Altium DPX (CAPEL, 2008)	9
2.2. Esquema de bloques del chip CMWX1ZZABZ (Murata, 2018)	10
2.3. Consumo y modos de funcionamiento del microcontrolador STM32L072 (STMicroelectronics, 2014)	11
2.4. Esquema de bloques de la familia de trancéptores SX127x (SEMTECH, 2016)	11
2.5. Diagrama funcional del AD7194 (Analog Devices, 2017)	12
2.6. Curvas de tensión termoeléctrica	13
2.7. Castellated holes	14
2.8. Esquemáticos de la tarjeta de expansión.	15
2.9. Estructura de la PCB	16
2.10. Circuito acondicionador de entrada de los termopares.	17
2.11. Circuito acondicionador de las entradas analógicas.	18
2.12. Circuito de medida de la batería.	19
2.13. Circuito de alimentación de la tarjeta de expansión.	20
2.14. Filtro de la alimentación	20
2.15. Esquemáticos de la tarjeta de expansión.	21
2.16. Estructura de la PCB	22
3.1. Cronograma de conexión de un nodo	28
3.2. Esquema de modelado de los nodos mediante máquinas de estado	32
3.3. Modelado del comportamiento mediante máquinas de estado extendidas	33

Índice de tablas

1.1. Ejemplo teórico para un canal con 125 KHz de ancho de banda	6
2.1. Componentes del nodo básico	23
2.2. Componentes de la placa de expansión	23

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 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, 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, así 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 ocasiones 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 sí,

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.

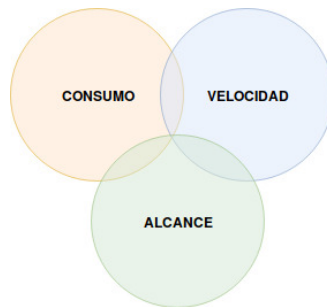


Figura 1.1: Parámetros fundamentales de un sistema inalámbrico

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 alcanza velocidades de transmisión de hasta 848Kbps, dependiendo del entorno en el que se lleve a cabo la comunicación [Shuang Song (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): 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 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.

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

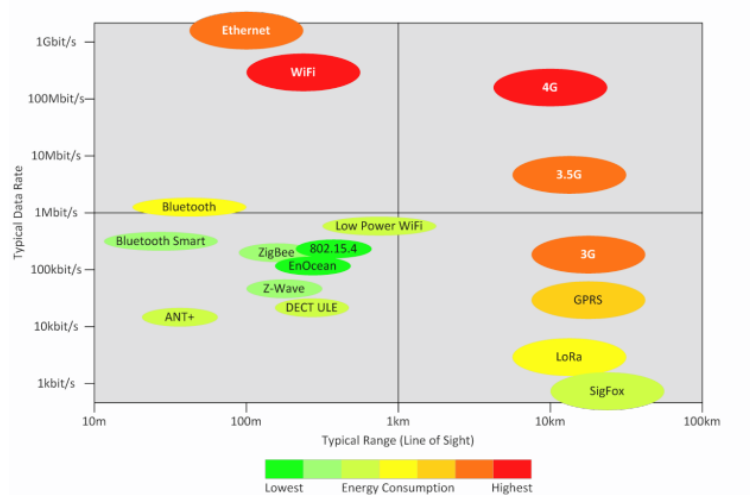


Figura 1.2: Relación de consumo, velocidad y alcance de las diferentes tecnologías [Friedli (2016)]

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 interactúan 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.

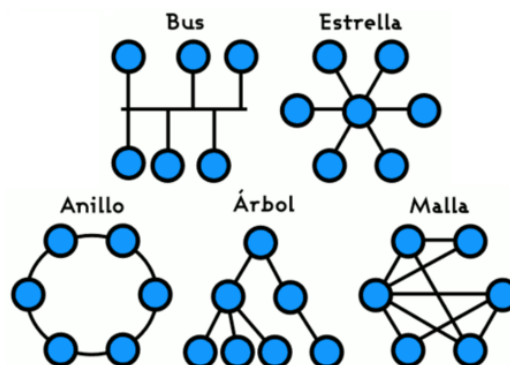


Figura 1.3: Topologías de red más usadas

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

en extensas áreas geográficas. Estos factores, hacen 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 utilización de diferentes técnicas de transmisión, con las utilizadas en las redes LPWAN.

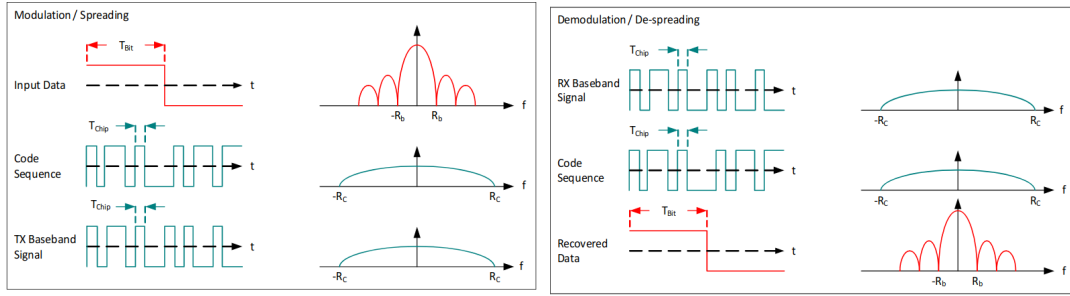
1.2.1. Redes LPWAN

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 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 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.

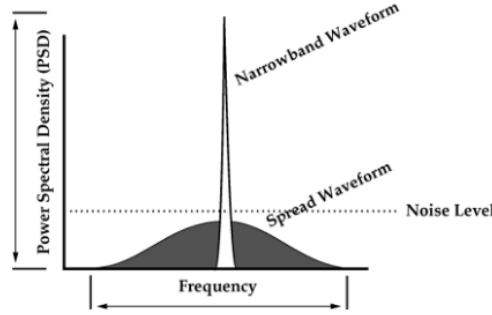
Para conseguir estas características, las redes LPWAN utilizan diferentes técnicas como:

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

Un chirp es una señal sinusoidal cuya frecuencia varía 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 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:

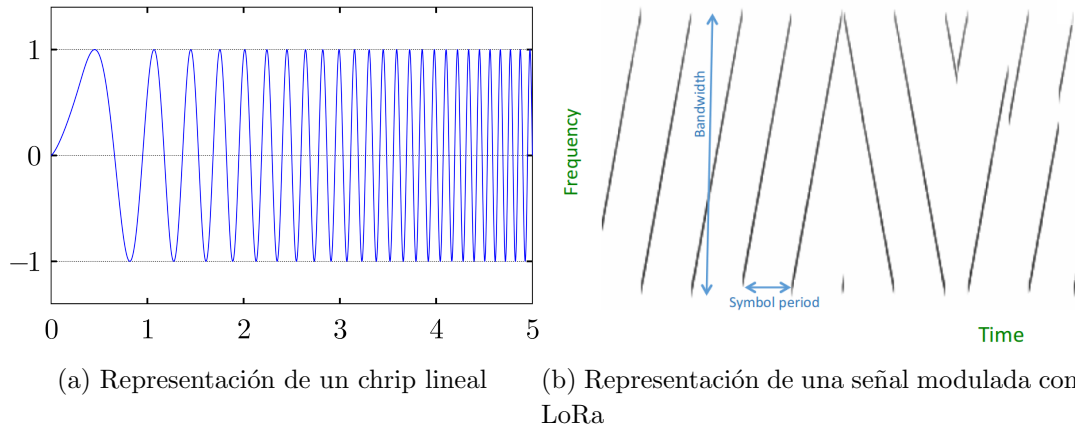


Figura 1.5: [Telkamp (2015)]

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

Como se puede 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.

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:

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

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

1.3. 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 específico y que, en futuras implementaciones, soporten un mayor número de arquitecturas de red.

Seguidamente, se propondrá un 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 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.

1.4. 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 Designer*: 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.

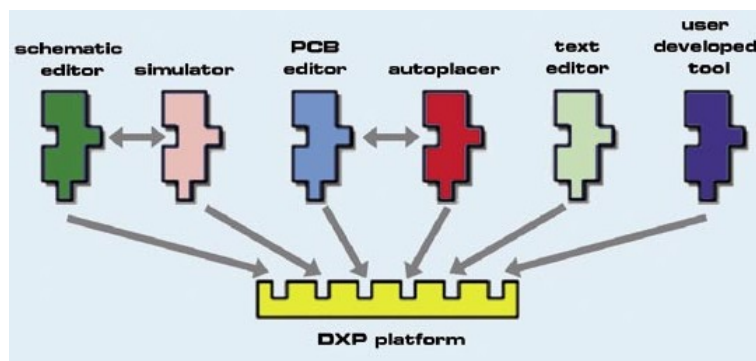


Figura 2.1: Componentes de Altium DPX (CAPEL, 2008)

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 esquemáticos, 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 trancceptor 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 .

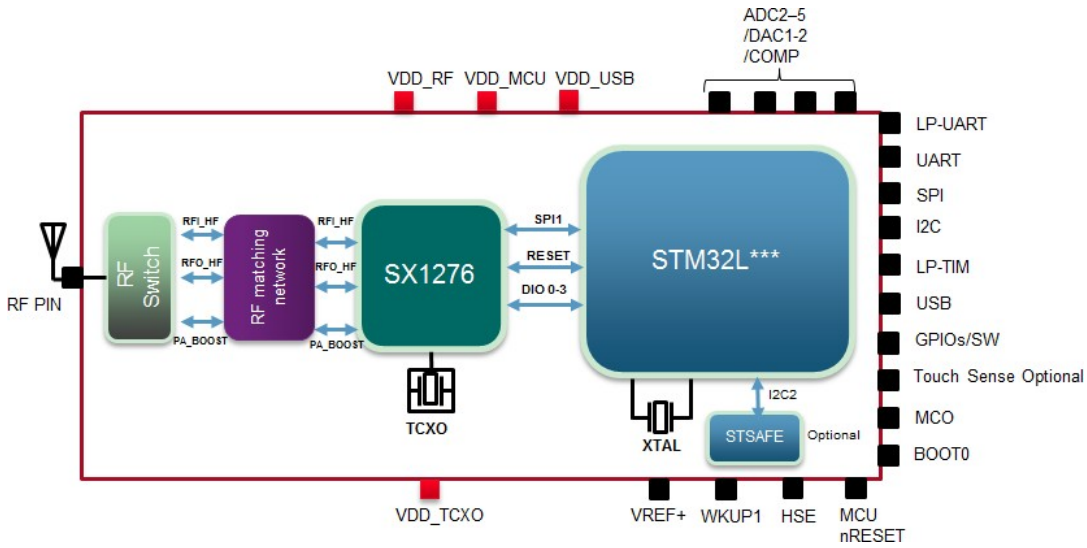


Figura 2.2: Esquema de bloques del chip CMWX1ZZABZ (Murata, 2018)

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 (STMicroelectronics, 2014). Este microcontrolador tiene un amplio abanico de periféricos 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 transceptor 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 caracteriza por trabajar en un rango de frecuencias desde 137 MHz hasta los 1020 MHz, con un "spreading factor" o factor de enanchado 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 (SEMTECH, 2016).

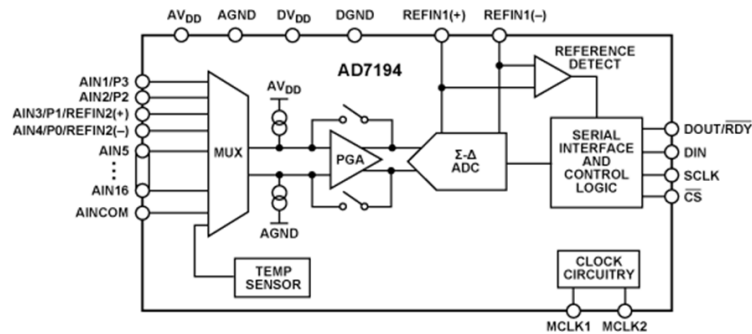


Figura 2.5: Diagrama funcional del AD7194 (Analog Devices, 2017)

La motivación de usar este ADC, es su aplicación: La medida de termopares. En esta aplicación, se hace necesario una gran precisión (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 temperaturas entre $-20\text{ }^{\circ}\text{C}$ y $100\text{ }^{\circ}\text{C}$ con una precisión de $\pm 0,25^{\circ}\text{C}$. 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 μA en funcionamiento (Microchip, 2011).

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 unión de cobre + estaño de los conectores genera una diferencia de potencial parásita que influye en las medidas). A este fenómeno se le llama

2.2.4. Componentes activos

Además de los ya mencionados, también se ha hecho uso de otros componentes activos, de los que podemos destacar:

- **Regulador TLV755P:** Es un regulador LDO (*Low-droput regulator* de sus siglas en inglés) de tensión fija a 3.3 V, capaz de dar a su salida hasta 500 mA. Al ser un LDO solo necesita una diferencia de potencial de 238 mV entre sus terminales de entrada y salida, lo que nos permite alimentarlo con una batería de 3.7 V. Otro detalle importante es que su corriente de fugas en reposo es de solo 25 μA , lo que es ideal para mantener el bajo consumo.
- **Regulador AD1582BRTZ:** Este regulador es capaz de dar una tensión fija de 2.5 V con una precisión muy alta y con una corriente de fugas muy baja. En el proyecto se usará como nivel de referencia del ADC.
- **Operacional MCP6001T:** Se trata de un amplificador operacional de uso general. Funciona con una tensión de alimentación entre 1.8 V y 6 V y tiene un consumo típico de 100 μA . Se utilizará para dar alta impedancia de entrada al circuito de medida de la batería.

2.2.5. Componentes pasivos

En este apartado englobamos las resistencias, condensadores e inductancias. De manera general, se ha procurado elegir componentes con un encapsulado SMD de tamaño 0603, el cual es lo suficiente grande para poder soldarlo manualmente sin mucha dificultad.

En cuanto a la resistencias, se han elegido de película de carbono, con una tolerancia del 1%. Por otro lado, los condensadores son, en su mayoría, cerámicos con un dieléctrico X5R ó X7R y una tolerancia $\leq 5\%$.

2.2.5.1. Termopares

Un termopar es un transductor de temperatura formado por la unión de dos metales diferentes. La variación de temperatura en el punto de unión de los dos metales provoca una variación proporcional de la tensión en el punto de contacto, lo que a su vez genera una corriente eléctrica a través de los conductores. A este efecto termoeléctrico se le conoce como efecto Seebeck.

La relación entre la tensión y la temperatura en el termopar se mide mediante el coeficiente de Seebeck, el cual suele tener unidades de $\mu V/K$. Este coeficiente dependerá de los metales utilizados en la unión, de los cuales dependerán también los rangos de temperatura que será capaz de medir el termopar. Las diferentes tipos uniones se distinguen mediante letras como K (NiCr-Ni), E (NiCr-CuNi) o J (Fe-CuNi). En la gráfica 2.6 podemos observar las variaciones de tensión frente a temperatura de los diferentes tipos de termopares.

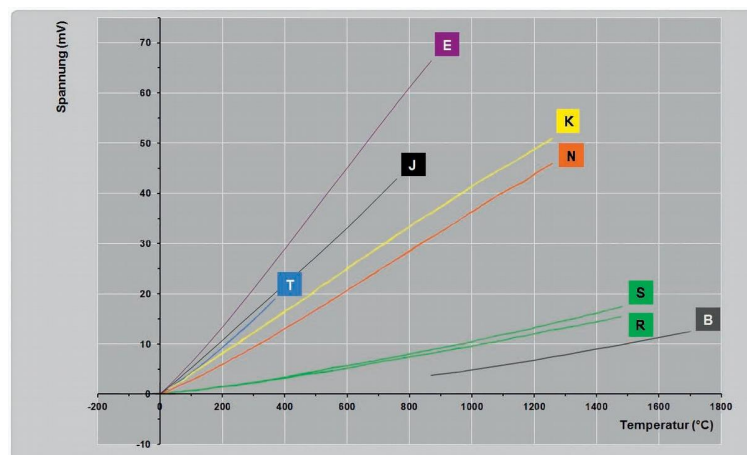


Figura 2.6: Curvas de tensión termoeléctrica

Dado que las medidas se realizan en temperatura ambiente y que la unión cobre-estaño de los conectores forman también un termopar, la medida realizada no se corresponderá con el valor exacto de temperatura. Por ello, para obtener la temperatura absoluta, es necesario realizar una compensación conocida como "compensación de la zona fría".

2.3. Diseño las tarjetas

2.3.1. Diseño del nodo básico

En esta parte se describirá el proceso de diseño un nodo general (sin aplicación específica definida) que contendrá lo necesario para un correcto funcionamiento del microcontrolador y la radio. Uno de los objetivos principales será conseguir un módulo de tamaño reducido que pueda ser fácilmente montado en otro circuito impreso. Por esto, se procurará reducir el tamaño de los conectores y pistas y se intentará reducir al máximo el número de componetes.

Para poder utilizar los módulos en placas de expación, se ha optado por un montaje *placa a placa* mediante conectores de borde con *castellated holes* (Agujeros almenados por su traducción al español). Los *castellated holes* son vias (agujeros metalizados que conectan dos capas de un circuito) colocadas en los borde de un circuito impreso las cuales son cortadas para crear medios agujeros metalizados. Estos medios agujeros generalmente son utilizados para conectar de manera sencilla pequeños módulos como si fuesen un componente mas. La fabricación de este tipo de conectores conlleva un proceso especial, lo que eleva el precio de fabricación de los prototipos.

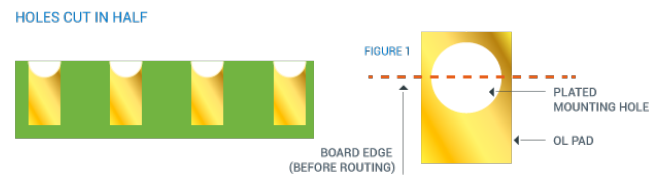


Figura 2.7: Castellated holes

El módulo también cuenta con algunos elementos de depuración, como los diodos LED D1 y D2 (sección 2.3.1.1), conectados a 3.3V y al GPIO PA0 del microcontrolador respectivamente. También cuenta con botón el cual, al ser pulsado, conecta el pin NRESET a tierra reiniciando el microcontrolador (el pin es activo a nivel bajo según la especificación del fabricante 2.2).

Para su programación y depuración, se han añadido 5 pads, con una separación de 1.27 mm, conectados a la alimentación y a la interfaz SWD (*Serial Wire Debug* por sus siglas en inglés) del microcontrolador. Mediante estos pads y un el programador STLINK, podemos crear una conexión con la herramienta de depuración GDB, la cual veremos en la sección 3.2.2.2 de este documento.

La antena va conectada al nodo mediante un conector miniatura del tipo U.LF. Los conectores U.LF, trabajan en un amplio rango de frecuencias (hasta los 6 GHz), por lo que es habitual verlos en tarjetas miniPCI WiFi o en módulos GPS. Ante la imposibilidad de conseguir una buena adaptación de impedancias entre el microcontrolador y la antena (lo que causará pérdidas de retorno en la línea de transmisión), se ha procurado reducir la longitud de la pista que los une y se han añadido los pads para conectar una red de adaptación en el caso de que fuese necesario (componentes L1, L2 y C6 de esquemático de la sección 2.3.1.1).

2.3.1.1. Esquemáticos

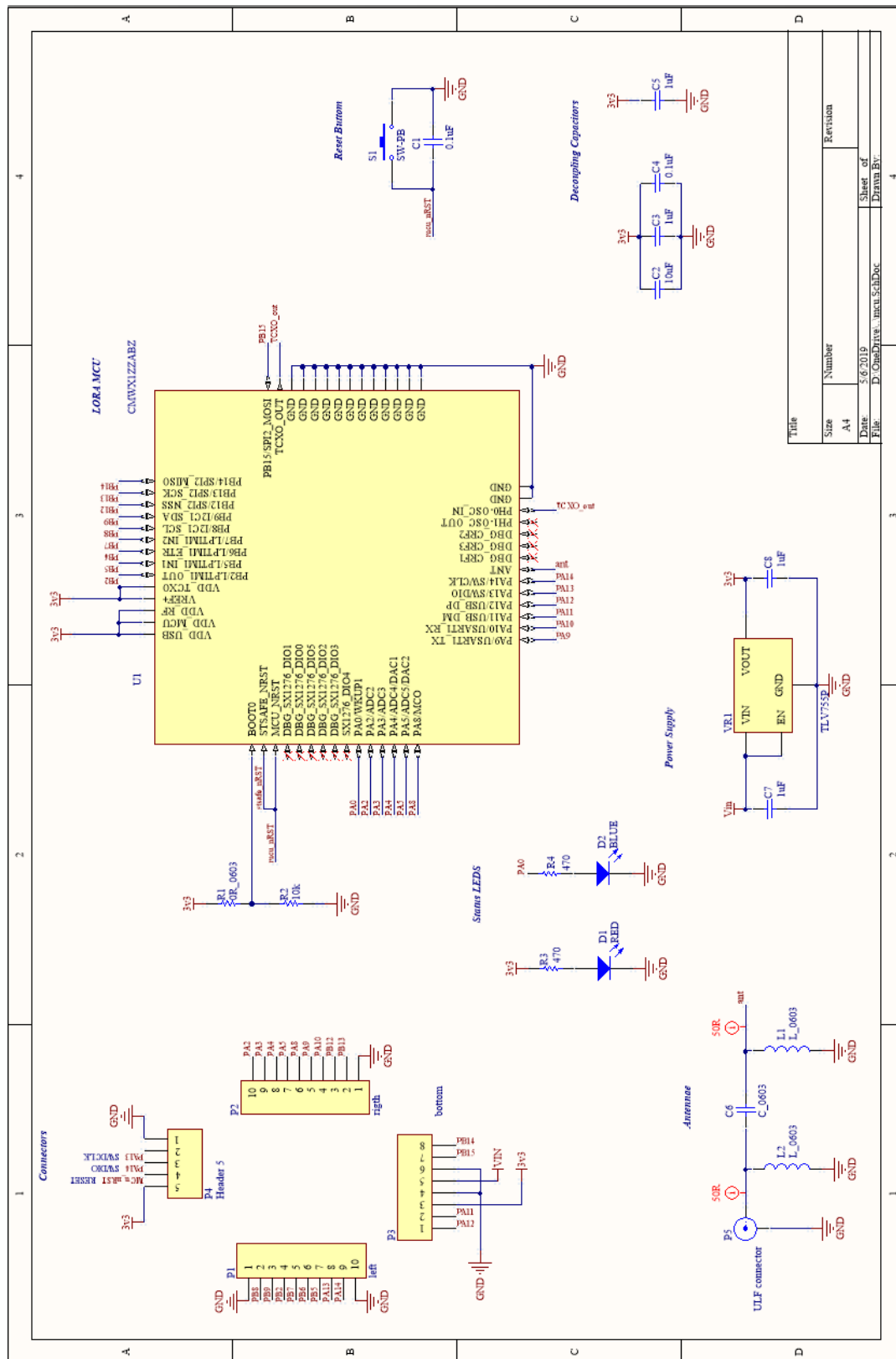
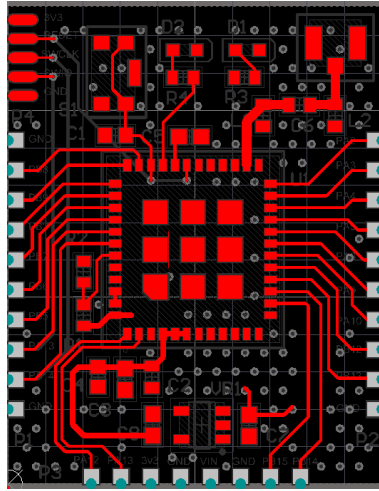
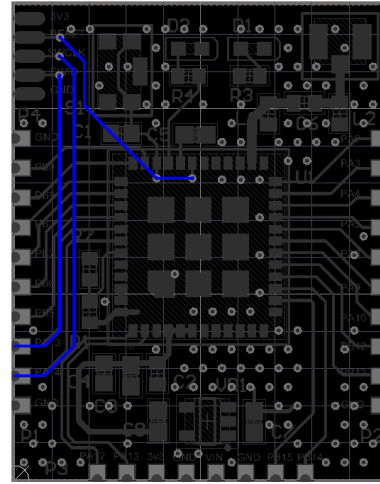


Figura 2.8: Esquemáticos de la tarjeta de expansión.

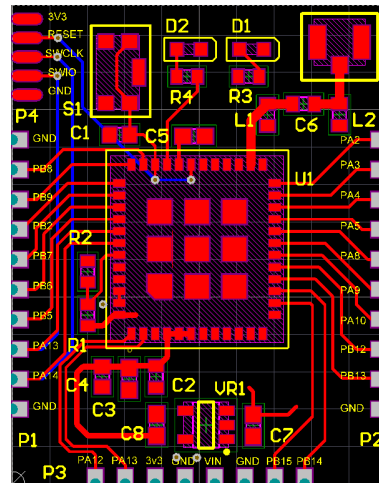
2.3.1.2. Layout



(a) Elementos de la capa superior.



(b) Elementos de la capa inferior.



(c) PCB con todas las capas

Figura 2.9: Estructura de la PCB

El resultado obtenido es una tarjeta de 32x25 mm de longitud como la mostrada en la figura 2.9, la cual se ha diseñado con solo dos capas para reducir los costes y tiempo de fabricación. Como se puede apreciar, todos los componentes se encuentran situados en la capa superior de la PCB, esto se ha hecho con el fin de poder usar la PCB como un módulo de montaje superficial.

Para reducir los efectos de capacitancias parásitas y autoinductancias entre las pistas, ambas capas cuentan con un plano de masa que cubre todas las zonas libres del circuito. Ambos planos están conectados mediante un gran número de vias (también llamado *via stitching*), las cuales ofrecen una línea de retorno rápido para las corrientes, evitan bucles en la masa y evitan que los planos funcionen como antenas. Estos planos se han ocultado en la figura 2.9 para facilitar la visualización de las pistas y componentes.

2.3.2. Diseño de la tarjeta de expansión

Con el nodo básico funcionando, el objetivo ahora es darle una aplicación específica. Para ello, se diseñará una tarjeta de expansión con sensores con las siguientes características:

- 6 entradas para termopares
- 3 entradas analógicas
- Sensor de temperatura cerca de los conectores
- Conectores enchufables con clemas
- Sensor de batería
- ADC de alta precisión
- Posibilidad de cortar la alimentación

Como ADC se ha utilizado un AD7194 como el descrito en la sección 2.2.2, el cual cuenta con el suficiente número de canales para soportar los 6 termopares, las 3 entradas analógicas y el sensor de la batería. Para medir la temperatura de unión se ha utilizado un sensor de temperatura MCP9808T, descrito en la sección 2.2.3. En los siguientes apartados se describirán de forma más detallada los esquemas utilizados para la medida de los termopares, entradas analógicas, batería y alimentación de la placa.

2.3.2.1. Medida de los termopares

Para acondicionar la entrada se ha utilizado la estructura mostrada en la figura 2.10.

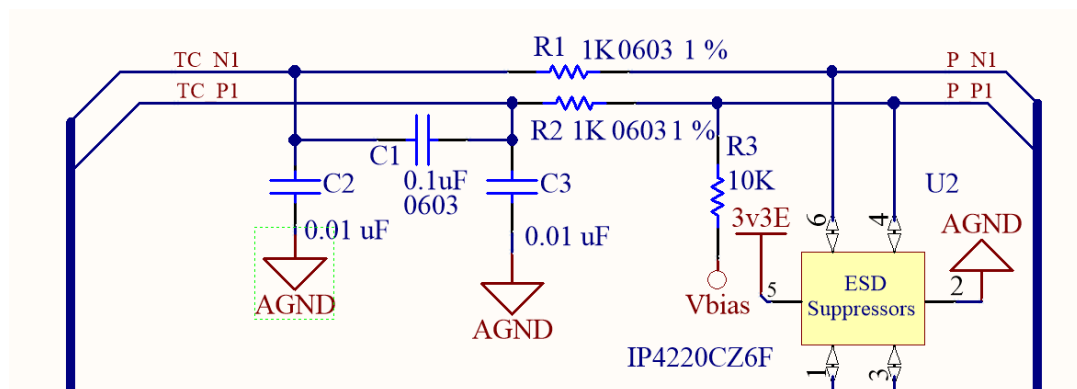


Figura 2.10: Circuito acondicionador de entrada de los termopares.

Los pares R1-C2 y R2-C3, forman dos filtros paso bajo, que filtrarán las entradas en como común del termopar, eliminando el ruido presente en el termopar y evitando el alisasing en el ADC. Sabiendo que $R1 = R2 = R$ y que $C1 = C2 = C$, obtenemos que la frecuencia de corte del filtro viene dada por la expresión:

$$f_c = \frac{1}{2 \times \pi \times R \times C} \approx 16KHz \quad (2.1)$$

Al no ser ideales, los componentes son iguales, lo que genera una perturbación en el modo diferencial de la señal, lo que a su vez desemboca en una disminución del ratio de rechazo al modo común o CMRR (*Common Mode Reject Ratio* por sus siglas en inglés). Como la medida de termopares requiere un alto CMRR, se ha añadido un filtro diferencial, formado por C1, R1 y R2, que compensa los efectos de los filtros en modo común. La frecuencia de corte de este filtro viene dado por la ecuación:

$$f_c = \frac{1}{2 \times \pi \times (R_1 + R_2) \times C_1} \approx 16KHz \quad (2.2)$$

2.3.2.2. Medida de entradas analógicas

Para las entradas analógicas, se ha utilizado el circuito acondicionador de la entrada de la figura 2.11.

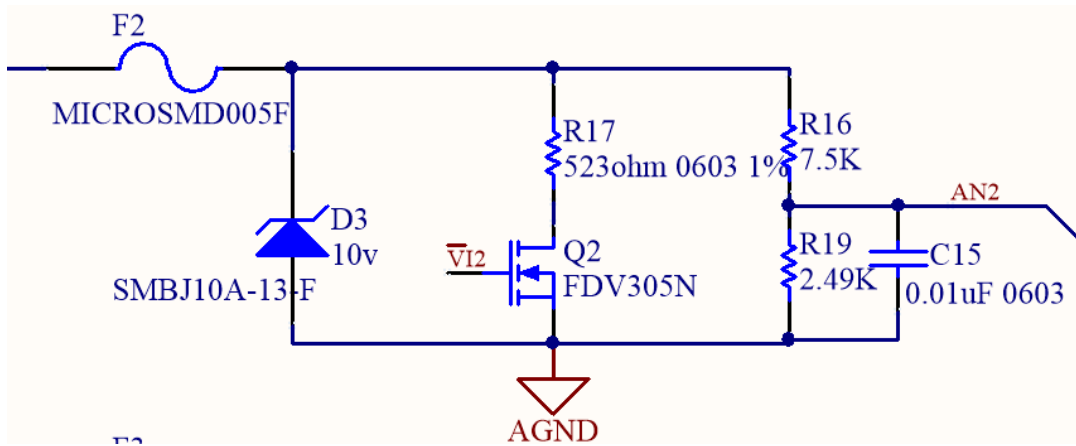


Figura 2.11: Circuito acondicionador de las entradas analógicas.

las resistencias R16 y R19, conforman un divisor resistivo con una atenuación de 1/4 de la tensión de entrada. Teniendo en cuenta que utilizamos un regulador AD1582BRTZ a 2.5 V (como el especificado en la sección 2.2.4), conseguimos un margen dinámico a la entrada de 10 V.

Por otro lado, con la conmutación del transistor Q2, conseguimos que la impedancia de entrada sea aproximadamente de 523ω , lo que permite la medida de entradas en corriente de hasta 20 mA. La expresión de la corriente a la entrada se puede expresar de la forma:

$$i_e \approx \frac{4V_m - V_{DS}}{R_{17}} \quad (2.3)$$

siendo:

- V_m tensión medida en el ADC
- V_{DS} tensión *Drain-Source* del transistor

El circuito tambien cuenta con protección a su entrada: El fusible F2 limita la corriente de entrada del circuito a un máximo de 50 mA. A su vez, el diodo zener D3 limita la tensión máxima a la entrada de 10 v. En el caso de que la entrada superase los 50 mA, el fusible mantendría una corriente contante de 50 mA hasta superar el tiempo de ruptura T_r (aproximadamente 100 ms según las especificaciones del fabricante).

2.3.2.3. Media del nivel de batería

Para medir el nivel de tensión de la batería, se ha utilizado en circuito de la figura 2.12.

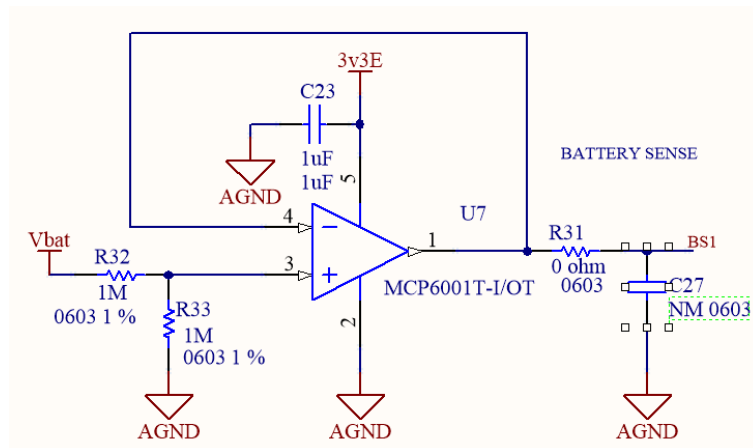


Figura 2.12: Circuito de medida de la batería.

Las resistencias R32 y R33 del circuito forma un divisor resistivo con una atenuación de 1/2. El amplificador operacional, el cual se encuentra con una configuración de seguidor de tensión, proporciona una entrada en alta impedancia (evitando así derivas de corriente). Como las resistencias estan el rango de los mega ohmios, la corriente que circula desde la batería tomará valores de micro amperios, disminuyendo el consumo de batería.

Por último, la resistencia R31 y condensador C27 dan la posibilidad de poner un filtro paso bajo si fuese necesario. Este filtro funcionaría como filtro anti-aliasing, evitando errores en las medidas del ADC.

2.3.2.4. Alimentación de los componentes

Para alimentar la tarjeta, se ha utilizado un regulador TLV755P como el detallado en la sección 2.2.4, cuya activación se puede controlar desde microcontrolador

mediante el pin de *ENABLE*. Mediante las resistencias R34 y R35 se puede elegir entre tener el regulador siempre activo o controlarlo desde el microcontrolador.

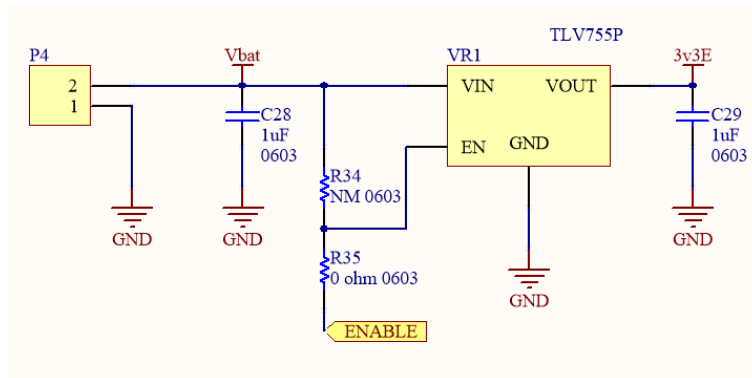


Figura 2.13: Circuito de alimentación de la tarjeta de expansión.

Con el fin de eliminar el ruido de conmutación presentes en la alimentación y mejorar la precisión del ADC, se han separado la alimentación analógica de la digital mediante el filtro de la figura 2.14.

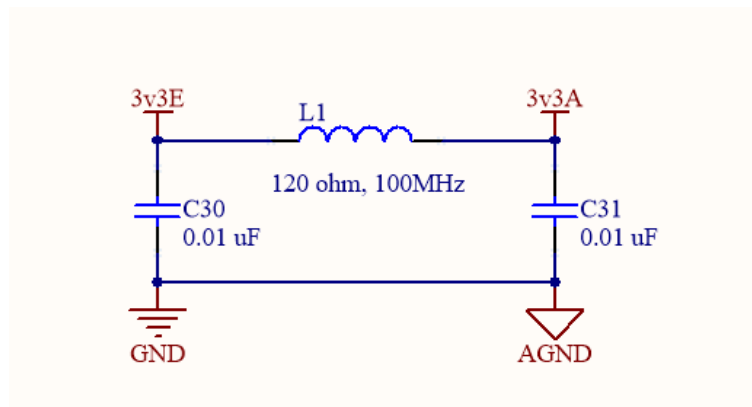


Figura 2.14: Filtro de la alimentación

2.3.2.5. Esquemáticos

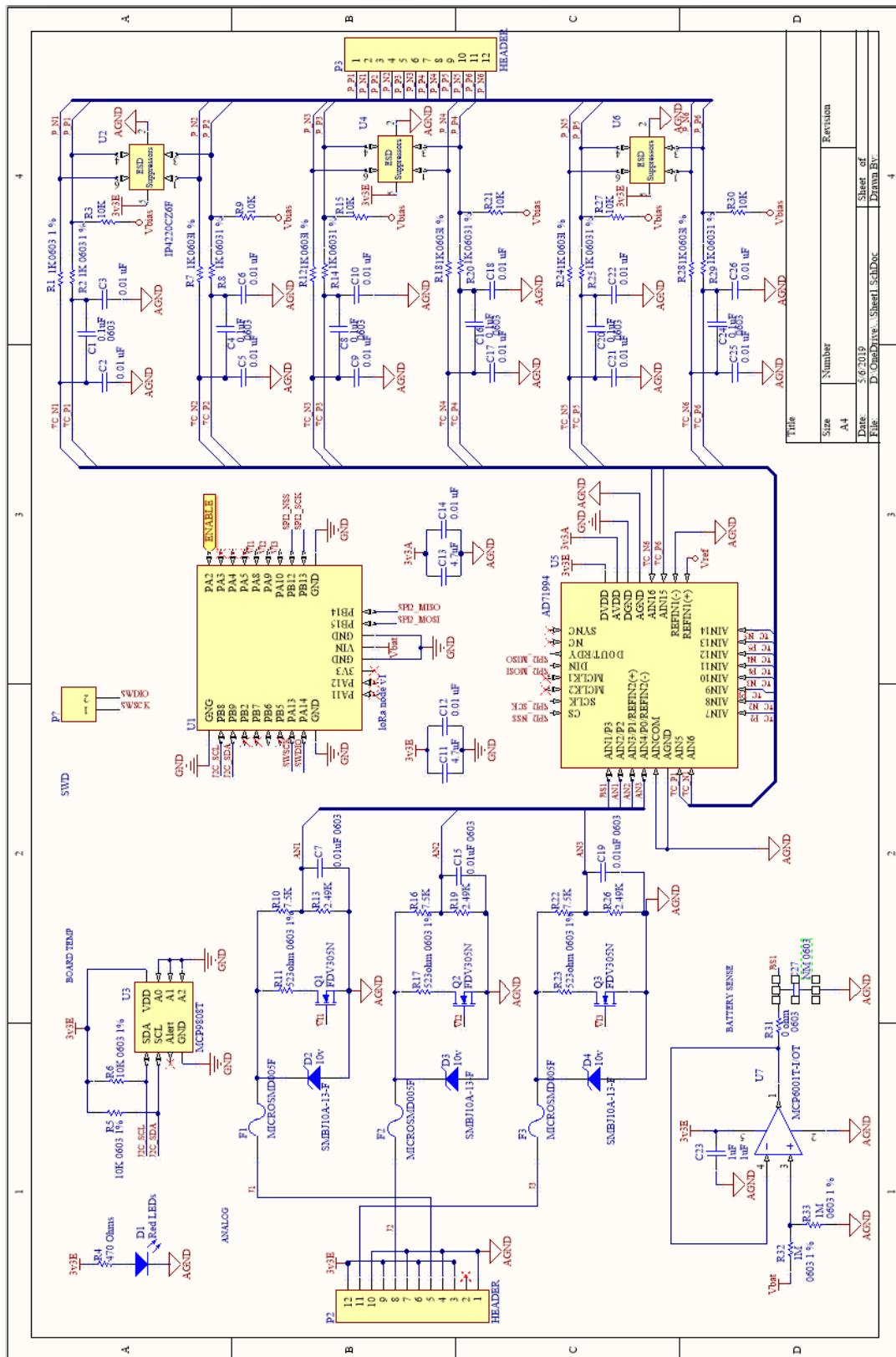
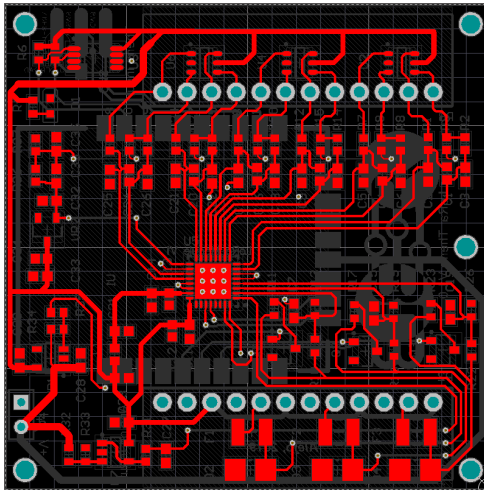


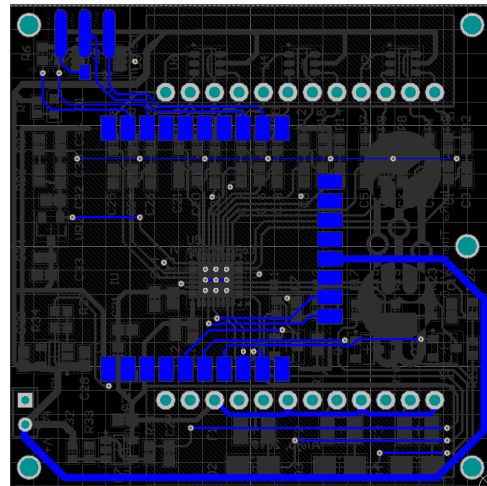
Figura 2.15: Esquemáticos de la tarjeta de expansión.

2.3.2.6. Layout

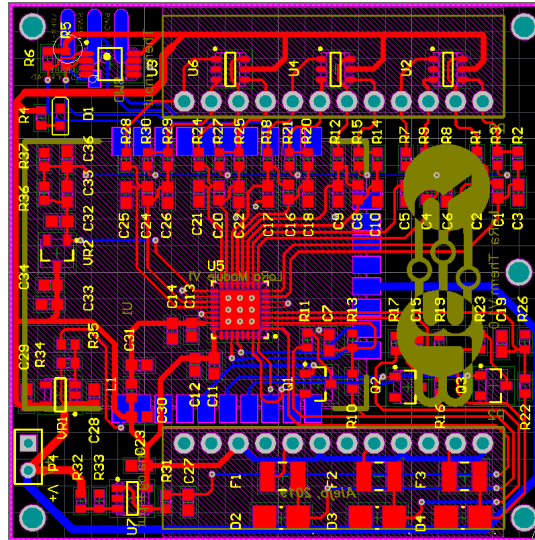
A la hora de diseñar la PCB, se han colocado los componentes en bloques funcionales, procurando acortar el tamaño de las pistas. La disposición de los bloques funcionales en la PCB se ha hecho intentando reducir al máximo el espacio ocupado e intentando utilizar pistas simples y con conexión directa entre componentes. El sensor de temperatura MCP9808T se ha situado en una zona cercana a los conectores de los termopares para medir con mayor precisión la temperatura de unión.



(a) Elementos de la capa superior.



(b) Elementos de la capa inferior.



(c) PCB con todas las capas

Figura 2.16: Estructura de la PCB

Al igual que en los nodos vistos en la sección 2.3.1.2, esta placa está diseñada a dos capas y cuenta con dos planos de masa, en ambas caras del circuito, con el fin de evitar efectos parásitos en las líneas de transmisión. El resultado, es una PCB de 50x50 mm, cuya disposición de componentes se encuentra en la figura 2.16.

2.4. Lista de componentes

Description	Designator	Part Number	Value
Capacitor	C1, C4	C0603C104J3RAC	0.1uF
Capacitor	C2	GRM188R60J106KE47D	10uF
Capacitor	C3, C5, C7, C8	885012206052	1uF
INFRARED GaAs LED	D1	150060RS75000	Red
INFRARED GaAs LED	D2	150060BS75000	NM
BNC Elbow Connector	P5	CONUFL001-SMD	
Resistor	R2	RC0603FR-0710KL	10K
Resistor	R3, R4	AC0603FR-13470RL	470ohm
Switch	S1	TL6330AF200Q	
RF MODULE	U1	CMWX1ZZABZ-078	
LDO Voltage Regulators	VR1	TLV75533PDBVR	

Tabla 2.1: Componentes del nodo básico

Description	Designator	Part Number	Value
Capacitors	C1, C4, C8, C16, C20, C24	C0603C104J3RAC	0.1uF
Capacitors	C2, C3, C5, C6, C9, C10, C12, C14, C17, C18, C21, C22, C25, C26, C30, C31, C33	885012206089	0.01 uF
Capacitors	C7, C15, C19	885012206089	0.01uF
Capacitors	C11, C13, C34	JMK107BJ475KA-T	4.7uF
Capacitors	C23, C28, C29, C32, C35, C36	885012206052	1uF
INFRARED GaAs LED	D1	150060RS75000	10V
ESD Suppressors	D2, D3, D4	SMBJ10A-13-F	
Resettable Fuses	F1, F2, F3	MICROSMD005F-2	
Ferrite Beads	L1	2506031217Z0	120 ohm, 100MHz
MOSFET	Q1, Q2, Q3	FDV305N	
Resistors	R1, R2, R7, R8, R12, R14, R18, R20, R24, R25, R28, R29, R36, R37	CRCW06031K00FKEAC	1K

Tabla 2.2: Componentes de la placa de expansión

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:

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).

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.

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

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:

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).

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:

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

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 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.

3.1.2.3. 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

3.2. Implementación

3.2.1. Fundamentos teóricos

3.2.1.1. Sistemas embebidos

Un sistema embebido es un sistema, que a diferencia de los sistemas de propósito general, está diseñado para realizar unas pocas funciones específicas. Estas funciones, generalmente, tienen requisitos de ejecución de tiempo real.

Estos sistemas suelen tener recursos limitados y requisitos estrictos. Por este motivo, existe un gran compromiso entre el hardware y el software (la manipulación del hardware se hace de forma directa), lo que limita su programación a lenguajes de bajo nivel.

En la mayoría de los casos, estos sistemas no pueden tener intervención humana, por lo que sus principales características deben ser:

- **Fiabilidad:** El sistema debe tener una tasa de fallos baja.
- **Robustez frente a fallo:** En caso de fallo, el sistema debe ser capaz de recuperarse de manera eficiente.
- **Disponibilidad:** El sistema debe estar disponible en el menor tiempo posible.
- **Seguridad:** En muchos casos, deben contar con funciones y protocolos que aseguren la integridad de los datos y eviten su acceso a terceros.

3.2.1.2. Sistemas operativos de tiempo real (RTOS)

Un sistema operativo de tiempo real (también conocido como RTOS por sus siglas en inglés), es un sistema operativo ligero, diseñado para sistemas embebidos con requisitos temporales estrictos. Los RTOS permite la ejecución de código organizado en forma de tareas independientes y jerarquizadas.

Un RTOS se puede describir como un "planificador de tiempo real", ya que su función principal es organizar tareas según su prioridad y ejecutarlas garantizando tiempos de ejecución. También, administran los accesos a memoria y los recursos compartidos mediante la implementación de mutex, semáforos, colas, etc. Estos sistemas son especialmente útiles en microcontroladores con un único núcleo, los cuales solo pueden ejecutar una tarea de manera simultánea.

La utilización de un RTOS tiene grandes ventajas que facilitan el desarrollo, depuración y mejora de un sistema, entre ellas podemos encontrar:

- **Abstracción del manejo del tiempo:** Un sistema operativo de tiempo real contiene las herramientas necesarias para controlar el tiempo de ejecución.
- **Escalabilidad y modularidad:** La organización del código en forma de tareas independientes, permite añadir, quitar o modificar módulos de manera rápida y sin comprometer el funcionamiento global del sistema.
- **Reutilización del código:** La utilización de tareas independientes, también permite migrar módulos a otros proyectos.

- **Control del consumo:** Los RTOs permiten tener un control estricto de los tiempo de ejecución, lo que se traduce en un mejor control del consumo del sistema.
- **Fácil depuración:** Las tareas independientes pueden ser depuradas de forma individual y aislada.

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 (*Application Program 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.

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.

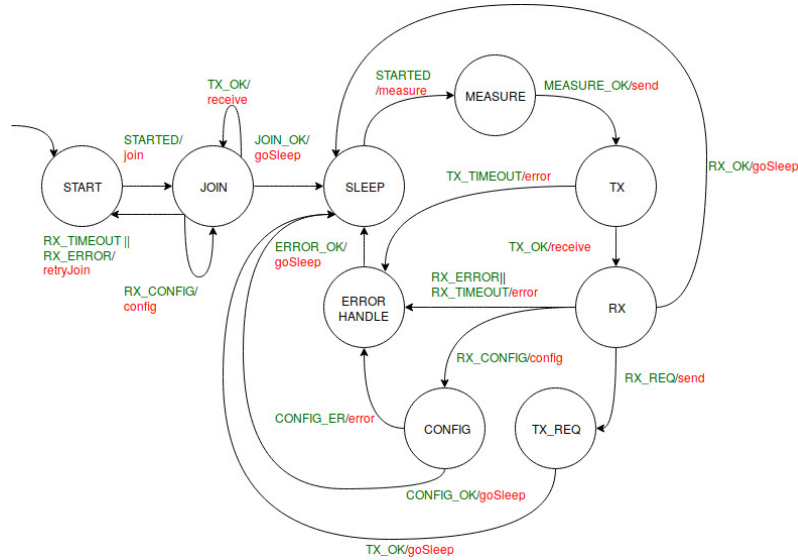


Figura 3.2: Esquema de modelado de los nodos mediante máquinas de estado

- **SLEEP:** El sistema entrará en bajo consumo durante el tiempo definido. En este estado, se deshabilitarán 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 errores, 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 procesamiento de datos, de esta forma, evitamos la pérdida de

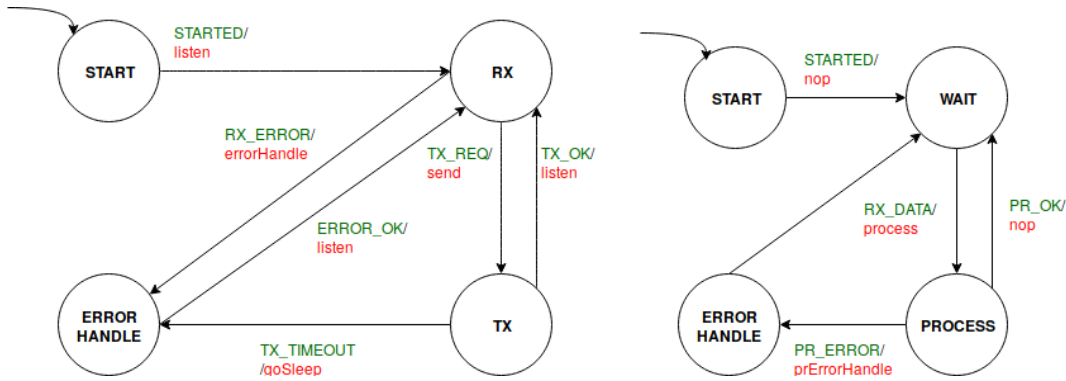


Figura 3.3: Modelado del comportamiento mediante maquinas de estado extendidas

datos durante procedentes de los nodos. Todo esto es posible gracias a las posibilidades que nos ofrece 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 presencia 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 procesamiento de datos son los siguientes:

- **WAIT:** Se mantiene 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

- Analog Devices (2017). AD7194 - 8-Channel, 4.8 kHz, Ultralow Noise, 24-Bit Sigma-Delta ADC with PGA. Technical Report Rev. B, Analog.
- CAPEL (2008). ¿Qué es Altium Designer? *CAPEL*.
- 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.
- Microchip (2011). MCP9808 - Maximum Accuracy Digital Temperature Sensor. Technical Report DS25095A, Microchip.
- Minihold, R. (2014). Analisis of wifi and wimax and wireless network coexistence. Technical report, School of Computing, Teesside University.
- Murata (2018). Sub-GModule Data Sheet. Technical Report BP-ABZ-C, Murata.
- SEMTECH (2015). Lora modulation basics. Technical Report 2, Semtech Corporation.
- SEMTECH (2016). SX1276 - 137 MHz to 1020 MHz Low Power Long Range Transceiver. Technical Report Rev. 5, SEMTECH.
- Shuang Song, B. (2011). Near field communication (nfc) technology and measurements. Technical report, Rohde and Schwarz.
- STMicroelectronics (2014). STM32L0xx ultra-low power features overview . Technical Report DocID025944 Rev 1, STMicroelectronics.
- 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*.