

AGRADECIMIENTOS

RESUMEN

PALABRAS CLAVE

Nodo de comunicaciones, Capa de Abstracción Hardware, HAL, Interfaz de Aplicación, Radio Cognitiva, CR, Redes de Sensores Inalámbricas, WSN, IEEE 802.15.4, Redes de Sensores Inalámbricas Cognitivas, CWSN, entornos inteligentes, flexibilidad, modos de bajo consumo, Microchip, PIC, MiWi™, sistemas empotrados.

ÍNDICE

AGRADECIMIENTOS	1
RESUMEN	2
PALABRAS CLAVE.....	3
ÍNDICE DE TABLAS	7
ÍNDICE DE FIGURAS.....	8
GLOSARIO	¡Error! Marcador no definido.
1. INTRODUCCIÓN.....	11
1.1. OBJETIVOS.....	12
1.2. FASES DE DESARROLLO	13
1.3. ESTRUCTURA DE LA MEMORIA	15
2. RESUMEN DEL ESTADO DEL ARTE.....	16
2.1. SISTEMA DE RADIO COGNITIVA Y REDES COGNITIVAS	16
2.2. REDES DE SENSORES INALÁMBRICAS.....	17
2.3. REDES DE SENSORES INALÁMBRICAS COGNITIVAS.....	18
3. FASE DE ANÁLISIS.....	23
3.1. MODELO DE UN NODO PARA REDES DE SENSORES INALÁMBRICAS COGNITIVAS.....	23
3.2. EVALUACIÓN DEL PROTOTIPO DEMOSTRADOR	24
3.3. ANÁLISIS DE REQUISITOS	27
3.3.1. Definición de nuevos requisitos.....	27
3.3.2. Capa de abstracción Hardware.	28
3.4. EVALUACIÓN DE ALTERNATIVAS DE DISEÑO.	30
3.4.1. - Microcontroladores	30
3.4.2. Transceptores.....	33
3.4.3. - Comparativa.....	36
4. FASE DE DISEÑO	38
4.1. DISEÑO HARDWARE	39
4.1.1. Microcontrolador	39
4.1.1.1. Arquitectura	39
4.1.1.2. Modos de bajo consumo.....	42
4.1.2. Interfaces radio	45
4.1.2.1. Transceptores.....	46
4.1.2.1.1. Módulo MRF24J40MA	46

4.1.2.1.2. MRF49XA PICtail Daughter Board.....	47
4.1.2.2. Protocolos de red.	48
4.1.2.2.1. Protocolo P2P.....	49
4.1.2.2.2. Protocolo MiWi.....	52
4.2. DISEÑO SOFTWARE.....	54
4.2.1. Diseño de la adaptación de la torre de protocolos MiWi.....	55
4.2.1.1. Decisiones de diseño	55
4.2.1.2. Estructura y diseño de la adaptación	56
4.2.2. Capa de abstracción hardware	58
5. FASE DE IMPLEMENTACIÓN.....	59
5.1. ENTORNO DE DESARROLLO Y HERRAMIENTAS.	59
5.2. LA TORRE DE PROTOCOLOS DE MICROCHIP.....	60
5.2.1. - Descripción general.....	60
5.2.2. Opciones de configuración.	61
5.2.2.1. Aplicación.....	61
5.2.2.2. Protocolos de red	63
5.2.2.3. Transceptor MRF49XA.....	64
5.2.2.4. Transceptor MRF24J40	64
5.2.3. Recursos necesarios.....	65
5.3. ADAPTACIÓN DE LA TORRE DE PROTOCOLOS.	65
5.3.1. Objetivos.....	65
5.3.2. - Estructura	66
5.3.3. La adaptación por niveles.....	67
5.3.3.1. - El nivel físico y los transceptores.....	67
5.3.3.2. - Los protocolos de red	68
5.3.4. - Opciones de depuración.....	76
5.4. – CAPA DE ABSTRACCIÓN HARDWARE.....	77
5.4.1. - Opciones de configuración.....	77
5.4.2. Interfaz de programación de alto nivel	78
5.4.2.1. Definición de variables.	78
5.4.2.2. - Códigos de Error.....	80
5.4.2.3. - Funciones de la interfaz de aplicación	81
6. TEST Y EVALUACIÓN	87
6.1. BREVE DESCRIPCIÓN DE LA PLATAFORMA DE PRUEBAS.....	87

6.2. PRUEBAS Y EVALUACIÓN	89
6.2.1. Pruebas funcionales.....	89
6.2.2. Pruebas de estrés y caracterización.....	91
6.2.2.1. Capacidad de procesamiento del microprocesador.	91
6.2.2.2. Máxima tasa binaria por cada interfaz	93
6.2.2.3. Tiempo de conmutación o reconfiguración de parámetros radio.....	98
6.2.2.4. Desconexión forzosa de un módulo MRF49XA.....	99
6.2.2.5. Medidas de consumo.....	100
7. CONCLUSIONES Y LÍNEAS FUTURAS	104
7.1. CONCLUSIONES	104
7.2. RECOMENDACIONES PARA LA IMPLEMENTACIÓN HARDWARE.....	105
7.3. LÍNEAS FUTURAS	107
BIBLIOGRAFÍA	109
ANEXO: FUNCIONES DE LA CAPA DE ABSTRACCIÓN HARDWARE.....	112

ÍNDICE DE TABLAS

<i>Tabla 1 : Comparativa de microcontroladores de 32 bits</i>	32
Tabla 2: Transceptores y módulos en la banda de 2,4 GHz	34
Tabla 3: Transceptores y módulos en bandas subGHz.....	35
Tabla 4: Tipos y roles de dispositivos en el protocolo MiWi P2P™	50
Tabla 5: Tipos y roles de dispositivos en la torre de protocolos de MiWi Mesh™	52
Tabla 6: Opciones de configuración de la pila de MiWi™ a nivel de aplicación.....	62
Tabla 7: Opciones de configuración de la pila de MiWi™ a nivel de protocolo	63
Tabla 8: Opciones de configuración adicionales del protocolo MiWi Mesh™	63
Tabla 9: Opciones de configuración del transceptor MRF49XA™	64
Tabla 10: Opciones de configuración del transceptor MRF24J40™	65
Tabla 11: Diferencias en la transmisión unicast entre la pila original y la pila adaptada.	72
Tabla 12: Opciones de configuración de la HAL	78
Tabla 13 : Flags de estado en la HAL	79
Tabla 14: Códigos de error de la HAL	81
Tabla 15: Tiempo de CPU requerido para el mantenimiento de la pila.	92
Tabla 16: Cadencia eficaz, envío y recepción de mensajes en P2P a 2,4 GHz.....	94
Tabla 17: Cadencia eficaz, envío y recepción de mensajes en MiWi a 2,4 GHz	95
Tabla 18: Cadencia eficaz, envío y recepción de mensajes en P2P a 434 y 868 MHz	96
Tabla 19: Cadencia eficaz, envío y recepción de mensajes en MiWi a 434 y 868 MHz	97
Tabla 20: Medidas de consumo.....	102

ÍNDICE DE FIGURAS

Figura 1: Foto a tamaño real del prototipo FCD diseñado en el laboratorio.....	24
Figura 2: Protocolos e interfaces del nodo FCD.....	26
Figura 3: Comparativa global de fabricantes	37
Figura 4: Protocolos, interfaces y periféricos del nodo NGD.....	38
Figura 5: Arquitectura interna de la familia PIC32MX	39
Figura 6 y Figura 7: Compatibilidad de módulos en PIC32MX 5xx/6xx/7xx para encapsulados de 64 pines (arriba) y 100 pines (abajo)	41
Figura 8 Esquema general de la pila de MiWi™	48
Figura 9: Simplificación del handshaking en el protocolo MiWi P2P™	49
Figura 10: Topologías de estrella (izq.) y P2P (dcha.)	51
Figura 11: Topologías en árbol (izq.) y en malla (dcha.)	53
Figura 12: Asignación de direcciones en el protocolo MiWi Mesh™	54
Figura 13: Diagrama de bloques software de una interfaz MiWi	56
Figura 14: Jerarquía de archivos de la pila de MiWi™	61
Figura 15: Jerarquía de archivos de la pila de Microchip adaptada	66
Figura 16: Adaptación del handshaking en P2P para compartir la tabla de conexiones.....	70
Figura 17: Mantenimiento de la torre de protocolos de MiWi P2P™	76
Figura 18: Selección de interfaces radio en la HAL	79
Figura 19: Foto de la plataforma de prueba o nodo FCD adaptado.	88
Figura 20: Frecuencia máxima de reconfiguración de parámetros radio.....	99
Figura 21: Montaje para la medida del consumo del nodo.....	101

LISTA DE ABREVIATURAS Y ACRÓNIMOS

ACK	<i>Acknowledgement</i>
ADC	<i>Analog-to-Digital Converter</i>
API	<i>Application Programming Interface</i>
AVR	<i>(Microcontroladores de Atmel)</i>
CCA	<i>Clear Channel Assessment</i>
CR	<i>Cognitive Radio</i>
CRC	<i>Cyclic Redundancy Check</i>
CPU	<i>Central Processing Unit</i>
CSMA-CA	<i>Carrier Sense Medium Access - Collision Avoidance</i>
CWSN	<i>Cognitive Wireless Sensor Networks</i>
DIE	<i>Departamento de Ingeniería Electrónica</i>
DMA	<i>Direct Memory Access</i>
DSA	<i>Dynamic Spectrum Access</i>
ED	<i>End Device</i>
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
EUI	<i>Extended Organizationally Unique Identifier</i>
FCD	<i>First Cognitive Device</i>
FFD	<i>Full Function Device</i>
GSM	<i>Global System for Mobile communications.</i>
HAL	<i>Hardware Abstraction Layer</i>
ICD	<i>In-Circuit Debugger</i>
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Electronics Engineering</i>
IETF	<i>Internet Engineering Task Force</i>
I^2C	<i>Inter-Integrated Circuit</i>
IP	<i>Internet Protocol</i>
ISA	<i>International Standards for Automation</i>
ISM	<i>Industrial, Scientific and Medical</i>
ITU	<i>International Telecommunications Union</i>
LNA	<i>Low Noise Amplifier</i>
LQI	<i>Link Quality Indicator</i>
LR-WPAN	<i>Low Rate WPAN</i>
LSI	<i>Laboratorio de Sistemas Integrados</i>
MAC	<i>Media Access Control</i>
MCU	<i>MicroController Unit</i>
MiApp	<i>Microchip Application Programming Interface</i>
MiMAC	<i>Microchip Media Access Control</i>
MiWi	<i>(Protocolos de Microchip) Microchip Wireless</i>
NGD	<i>Next Generation Device</i>
OSI	<i>Open System Interconnection</i>
P2P	<i>Peer To Peer</i>
PAN	<i>Personal Area Network</i>
PANC	<i>PAN Coordinator</i>
PANID	<i>PAN Identifier</i>
PC	<i>Personal Computer</i>
PHY	<i>Physical Layer (Capa física del modelo OSI)</i>

<i>PIC</i>	<i>(Microcontroladores de Microchip) Peripheral Interface Controller</i>
<i>PLL</i>	<i>Phase-Locked Loop</i>
<i>RAM</i>	<i>Random Access Memory</i>
<i>RISC</i>	<i>Reduced Instruction Set Computer</i>
<i>RFD</i>	<i>Reduced Function Device</i>
<i>RFID</i>	<i>Radio Frequency Identification</i>
<i>ROM</i>	<i>Read-Only Memory</i>
<i>RSSI</i>	<i>Received Signal Strength Indicator</i>
<i>RTC</i>	<i>Real-Time Clock</i>
<i>RTCC</i>	<i>Real-Time Clock and Calendar</i>
<i>SDR</i>	<i>Software Defined Radio</i>
<i>SMD</i>	<i>Surface-Mount Device</i>
<i>SPI</i>	<i>Serial Peripheral Interface</i>
<i>SRAM</i>	<i>Static Random Access Memory</i>
<i>SRD</i>	<i>Short Range Devices</i>
<i>SS</i>	<i>Spread Spectrum</i>
<i>UPM</i>	<i>Universidad Politécnica de Madrid.</i>
<i>TCP</i>	<i>Transmission Control Protocol</i>
<i>TRX</i>	<i>Transceiver</i>
<i>UDP</i>	<i>User Datagram Protocol</i>
<i>USB</i>	<i>Universal Serial Bus</i>
<i>VCO</i>	<i>Voltage-Controlled Oscillator</i>
<i>WDT</i>	<i>WatchDog Timer</i>
<i>WSN</i>	<i>Wireless Sensor Networks</i>
<i>WPAN</i>	<i>Wireless Personal Area Network</i>
<i>6LoWPAN</i>	<i>IPv6 over Low power WPAN</i>

1. INTRODUCCIÓN

En la última década hemos experimentado el despliegue de multitud de redes inalámbricas que han tenido una rápida adopción por parte de los usuarios. Este hecho ha supuesto un notable aumento en las interferencias entre sistemas y ha agravado la saturación del espectro radioeléctrico.

Estos problemas se achacan principalmente a la demanda creciente de tráfico, la heterogeneidad de redes y terminales inalámbricos existentes y al uso ineficiente que se hace del espectro. Pese a que las técnicas centradas en aumentar la eficiencia espectral en las comunicaciones y en reubicar o liberar zonas del espectro han dado sus frutos, hoy por hoy están prácticamente agotadas y se requieren soluciones consistentes a largo plazo.

Como respuesta a este panorama de múltiples redes heterogéneas interfiriéndose entre sí, surge la idea de desarrollar un nuevo paradigma de las comunicaciones inalámbricas en el que las redes compartan los recursos de manera natural, intercambien información y colaboren entre ellas para mejorar la eficiencia espectral y optimizar parámetros globales como el consumo, la fiabilidad o la seguridad.

Este nuevo paradigma se conoce como radio cognitiva (Cognitive Radio). Los sistemas de radio cognitiva deben de ser capaces de modificar sus parámetros de comunicación en función del estado de la red y los usuarios de manera dinámica. Como consecuencia, la red tiende a utilizar las zonas del espectro con menor ruido e interferencias en cada momento, redundando en una mayor fiabilidad en la comunicación. También puede suponer una reducción en el consumo, pues para mantener las prestaciones se puede reducir la potencia de transmisión, generando menor nivel de interferencias hacia otros sistemas. La aplicación de estas técnicas ha dado lugar al concepto de red cognitiva (Cognitive Network).

En definitiva, se trata de dotar a la red de flexibilidad y adaptabilidad y emplear técnicas que permitan mejorar el uso del espectro, reducir las interferencias y optimizar las prestaciones globales de la red. Las redes cognitivas recaban información del estado del espectro y la comparten entre los nodos que conforman la red para tomar decisiones y actuar de manera coordinada. Este hecho propicia aprovechar la disponibilidad local o temporal de mejores condiciones para la comunicación en diversas zonas del espectro radioeléctrico, predecir el comportamiento de la red, mejorar la fiabilidad o el consumo energético gracias a la colaboración entre nodos y reaccionar ante los cambios que surjan en el transcurso de la comunicación.

Las redes de sensores inalámbricas (WSN, por su acrónimo en inglés), en concreto, se plantean como un entorno idóneo para la implementación de estas estrategias ya que, en muchos casos, las restricciones impuestas sobre la red y las condiciones de trabajo requieren de comunicaciones inalámbricas eficientes, colaboración entre los nodos y adaptación frente a posibles incidencias. Estas redes conforman la infraestructura básica

para el desarrollo de toda una serie de aplicaciones y sistemas enmarcados dentro de los entornos inteligentes. La domótica, el control industrial, la prevención y detección de catástrofes naturales, o los sistemas sanitarios no invasivos y la telemedicina son solo algunas de las aplicaciones que se perfilan en este campo.

De la combinación de las técnicas mencionadas con las redes de sensores inalámbricas aparecen las redes inalámbricas de sensores cognitivas (Cognitive WSN). Uno de los principales desafíos en este campo es diseñar nodos de muy bajo coste y reducido tamaño con las mínimas prestaciones para incorporar el concepto de CR. El consumo energético de los nodos debe permitir implementar las mencionadas aplicaciones sin necesidad de estar conectados a la red eléctrica.

Este proyecto se enmarca en la línea de investigación de redes de sensores y radio cognitiva del Laboratorio de Sistemas Integrados (LSI), vinculado al Departamento de Ingeniería Electrónica (DIE) de la Universidad Politécnica de Madrid (UPM), donde en un proyecto anterior se diseñó un nodo demostrador que disponía de varias interfaces de comunicación inalámbrica y era capaz de intercambiar mensajes con otros nodos de la red^[1].

Este primer prototipo funcional ha permitido extraer datos y conclusiones para analizar qué elementos del diseño deben mantenerse, eliminarse o ser rediseñados. El presente proyecto recoge el testigo y pretende ir un paso más allá, rediseñando el nodo y dotándole de una interfaz de programación simple e intuitiva. De este modo, el nodo servirá como plataforma de desarrollo en proyectos futuros de redes de sensores inalámbricas cognitivas.

1.1. OBJETIVOS

El objetivo principal del proyecto es diseñar el nodo de una red de sensores con capacidades cognitivas y una capa de abstracción hardware (en inglés HAL, Hardware Abstraction Layer) como interfaz hacia el desarrollador, atendiendo a criterios de eficiencia, modularidad, flexibilidad y escalabilidad.

Debe suponer una mejora frente al prototipo demostrador optimizando el diseño, acotando claramente los requisitos e integrando las funcionalidades que ofrecen las interfaces de comunicación inalámbrica. La capa de abstracción hardware debe aislar a futuros desarrolladores de la complejidad hardware y software del nodo, ofreciendo funciones de control de alto nivel sencillas para el desarrollo de aplicaciones cognitivas.

A continuación se desgana el objetivo principal en subobjetivos más concisos:

- **Análisis de las conclusiones y resultados del prototipo demostrador.** Estudio de qué aspectos del diseño del prototipo demostrador son adecuados y cuáles no. Evaluación de las alternativas comerciales en cuanto a interfaces radio y microcontroladores se refiere.

- **Análisis de requisitos** y restricciones del nuevo prototipo (coste, tamaño, recursos, consumo...) con vistas al desarrollo de aplicaciones que hagan uso del dispositivo.
- **Estudio los elementos de las interfaces radio:** transceptores, torre de protocolos, opciones de configuración, características técnicas y recursos necesarios para su funcionamiento.
- **Implementación del firmware.** Gestión eficiente de las interfaces radio y otros módulos del nodo. Optimización del uso de los recursos disponibles (memoria, niveles de pila, modos de funcionamiento, etc.).
- **Desarrollo de una capa de abstracción hardware.** Integración del software de control de las interfaces inalámbricas facilitado por los fabricantes. Diseño de una capa de abstracción que permita a futuros desarrolladores utilizar el nodo de manera sencilla, con un conjunto de funciones específicas.
- **Pruebas generales del nodo.** Pruebas orientadas a verificar el funcionamiento del nodo y a demostrar el uso de las funciones definidas en la capa de abstracción hardware para el uso, configuración, control y gestión de las red.
- **Análisis de resultados y conclusiones.**

1.2. FASES DE DESARROLLO

El proyecto se ha desarrollado en las siguientes cinco fases:

Documentación y análisis del estado del arte.

Esta primera etapa ha consistido en recabar información y adquirir conocimientos específicos relativos a las redes inalámbricas de sensores cognitivas y familiarizarse con el entorno de desarrollo.

- Análisis del estado del arte en materia de redes de sensores y radio cognitiva.
- Documentación sobre protocolos y tecnologías disponibles para la implementación de redes de sensores.

Fase de análisis.

Antes de proceder a la implementación, en esta fase se han cubierto todos los objetivos referidos al estudio de los requisitos del nuevo nodo.

- Evaluación del prototipo demostrador realizado anteriormente en el laboratorio.
- Definición de requisitos generales del nodo. Limitaciones y criterios de diseño.
- Especificaciones de la capa de abstracción hardware a diseñar.

- Análisis de las alternativas comerciales para seleccionar el microcontrolador y las interfaces radio.

Fase de diseño

Esta etapa aborda el diseño del nuevo nodo, tratando de remarcar aspectos clave a tratar durante la fase de implementación.

- Selección de las interfaces inalámbricas y del microcontrolador que se emplearán en la plataforma. Decisiones de diseño.
- Familiarización con las librerías del fabricante, su estructura e implementación. Recursos necesarios de las torres de protocolos.
- Características del microcontrolador y estrategias de bajo consumo.
- Diseño de la capa de abstracción hardware para cubrir el mayor rango de funcionalidad. Planificación y diseño de la adaptación del firmware del nuevo diseño.
- Recomendaciones para la futura realización de un nuevo prototipo hardware.

Fase de implementación

Esta fase es el grueso del proyecto. En ella se explica la implementación de la interfaz de alto nivel que permite manejar el nodo de manera sencilla.

- Adaptación del firmware para controlar las interfaces radio presentes en el nodo. Principales diferencias con las librerías proporcionadas por el fabricante.
- Optimización del uso de los recursos del microcontrolador. Flexibilización del firmware para poder modificar o rediseñar la plataforma hardware.
- Volcado de trazas de depuración para la corrección de errores.
- Implementación de la capa de abstracción hardware y generación de documentación para los desarrolladores.

Pruebas, evaluación y conclusiones

Finalmente se ha procedido a evaluar el funcionamiento general del nodo y a analizar los resultados.

- Pruebas del nodo empleando las funciones de alto nivel definidas en las fases anteriores.
- Evaluación y validación de los resultados obtenidos. Documentación de problemas encontrados.
- Análisis de las conclusiones extraídas a lo largo de la realización del proyecto.
- Descripción de posibles mejoras o líneas futuras de investigación y desarrollo.

1.3. ESTRUCTURA DE LA MEMORIA

Tras la introducción y la presentación de los objetivos de este proyecto, se procede a explicar la estructura del resto del documento.

En el capítulo 2 se introducirán los conceptos de radio cognitiva, redes de sensores y redes de sensores cognitivas con mayor profundidad. Se evaluará el estado del arte en materia de la implementación de nodos para redes inalámbricas de sensores cognitivas.

En la sección 3 se ha realizado el análisis del prototipo demostrador, buscando extraer sus fortalezas y debilidades, para poder definir nuevos requisitos de diseño. De cara al rediseño se evalúan las principales alternativas en cuanto a transceptores y microcontroladores y se fijan unos requisitos para uno de los elementos clave del proyecto: el diseño de una capa de abstracción hardware.

En el capítulo 4 se desarrolla toda la fase de diseño. En ella se justificarán las decisiones de diseño que se han tomado para implementar el firmware. A continuación, se detallarán las principales características del microcontrolador y los transceptores elegidos, así como los protocolos de red que soportan. También se aborda el diseño de la capa de abstracción hardware y la adaptación del firmware.

Todos los detalles de la implementación están cubiertos en la sección 5. En ella se explica la adaptación de la torre de protocolos y la implementación de la capa de abstracción hardware contrastando las diferencias con las librerías originales del fabricante. También se detallan todas las opciones de configuración que buscan dotar al nodo de la mayor flexibilidad, modularidad y adaptabilidad posible para admitir mejoras y cambios en el diseño.

El capítulo 6 aglutina todas las pruebas que se han realizado con el nuevo firmware y la capa de abstracción hardware, incluyendo valoraciones sobre las cifras experimentales que se han obtenido durante la realización de las mismas. Se describe asimismo la plataforma donde se han realizado todas las pruebas y las condiciones de medida.

Por último, el capítulo 7 recoge las conclusiones de todas las etapas del desarrollo del nuevo nodo. Se bosquejan líneas futuras de trabajo o aspectos a mejorar dentro del diseño del nodo para desarrollar aplicaciones sobre el futuro prototipo.

2. RESUMEN DEL ESTADO DEL ARTE

Con la creciente proliferación de redes inalámbricas y dispositivos y su rápida adopción por parte de los usuarios, la saturación del espectro radioeléctrico y el nivel de interferencias entre sistemas es un problema cada vez más importante. Esta situación es especialmente crítica en las bandas de frecuencia que no requieren licencia, pues la heterogeneidad y el volumen de dispositivos que operan en ellas es mucho mayor.

Esta sección pretende ser un breve resumen del estado del arte en materia de radio cognitiva, técnica que plantea un nuevo paradigma en las comunicaciones para paliar estos problemas, y de su aplicación a WSN, dando lugar al concepto de CWSN.

2.1. REDES DE SENSORES INALÁMBRICAS

Una red de sensores inalámbrica es un conjunto de sensores autónomos distribuidos espacialmente para medir o monitorizar variables del entorno en el que operan. En determinados casos, además, los nodos cooperativamente transmiten la información hacia un nodo central o un centro de procesamiento, denominado comúnmente nodo sumidero.

Los nodos de una red de sensores son, generalmente, dispositivos electrónicos sencillos. Aparte de los sensores, suelen constar de un microcontrolador para procesar y almacenar temporalmente la información recabada por los sensores, un transceptor radio y una antena para transmitirla de manera inalámbrica a otros nodos o puntos de la red, y una fuente de energía para alimentar todos los subsistemas, típicamente baterías. Sus restricciones en coste y tamaño imponen limitaciones a la capacidad de procesamiento de los nodos, el almacenamiento en memoria, el consumo energético del mismo (ligado a su tiempo de autonomía) y al ancho de banda (tasa binaria de datos).

El rango de aplicaciones que cubren las redes de sensores es muy amplio, ya que se considera la principal tecnología sobre la que se desarrollan los entornos inteligentes. Algunos campos de aplicación son:

- **Control industrial:** las WSN se aplican fundamentalmente para el control de procesos, el mantenimiento o envejecimiento de las máquinas y la recopilación de datos evitando el despliegue de cableado.
- **Mantenimiento de estructuras:** los sensores distribuidos en infraestructuras como puentes, túneles y edificios ayudan a colectar datos para detectar problemas y riesgos (fatiga, exceso de carga, vibraciones, efectos de la meteorología, etc).
- **Salud y telemedicina:** es un campo muy en auge que plantea nuevos usos de las redes de sensores. Algunos ejemplos son la detección de accidentes

laborales o en el hogar, la observación de patrones de comportamiento o la monitorización de variables para uso sanitario.

- **Monitorización medioambiental:** las aplicaciones son muy diversas, desde la observación de magnitudes físicas en un área determinada (temperatura, humedad, concentración de gases) y el control de la calidad del aire o del agua, hasta aplicaciones orientadas a la prevención o detección de fenómenos naturales (incendios forestales, corrimientos de tierra...).
- **Domótica y hogar digital:** gracias a la disposición de pequeños dispositivos, surgen aplicaciones relacionadas con el control de la iluminación y la temperatura en las diversas estancias, la personalización en función del usuario, la seguridad y el ahorro energético dentro del hogar.
- **Localización y seguimiento:** aplicaciones para detectar dónde ha ocurrido un fenómeno y rastrear su paso por la zona vigilada.

Las tecnologías y protocolos más relevantes para WSN se suelen basar (o al menos compatibilizar) con el estándar IEEE 802.15.4^[11] para redes inalámbricas de área personal (WPANs). El énfasis del estándar está en la comunicación entre dispositivos cercanos con poca o ninguna infraestructura subyacente a muy bajo coste, con la intención de minimizar el consumo de energía. El estándar define los niveles de enlace (la implementación del subnivel de control de acceso al medio, MAC) y el nivel físico (PHY) del modelo OSI (Open Systems Interconnection), delegando los protocolos de red a las distintas aplicaciones. Algunos ejemplos basados en el 802.15.4 son las especificaciones de ZigBee^[12], WirelessHART^[13], ISA100.11^[14] y MiWi^[15].

Otro estándar de comunicaciones inalámbricas muy popular es en el que se basa Wi-Fi^[16], el IEEE 802.11^[17]. Se trata de un estándar más orientado al usuario final y que no está diseñado para cumplir con las restricciones de bajo consumo de las redes de sensores. No obstante, la posibilidad de disponer de conectividad con dispositivos IP (que soportan el protocolo de internet, IP) ha provocado que este estándar se incorpore en algunos nodos (típicamente los nodos pasarela o sumidero, que funcionan conectados a la red eléctrica). La convergencia hacia todo IP también ha fomentado el desarrollo de nuevos estándares como el 6LoWPAN^[18] (IPv6 Low Power WPAN), que permite transmitir paquetes IP sobre redes basadas en 802.15.4.

2.2. SISTEMA DE RADIO COGNITIVA Y REDES COGNITIVAS

En 1999, Mitola^[2] introdujo el concepto de radio cognitiva (CR) con vistas a hacer un uso más eficiente del espectro radioeléctrico. Inicialmente se planteó como una evolución de los sistemas de radio definida por software^[3] (Software Defined Radio), sistemas configurables que permitían alterar parámetros de la comunicación por radio (modulación, frecuencia de trabajo, formas de onda...) realizando un post-procesado de la señal recibida mediante software. Los sistemas de radio cognitiva parten de la idea de configurar los parámetros del nivel radio, pero con el objetivo de adaptarse

dinámicamente a las características del entorno radioeléctrico y las demandas de la red y sus usuarios, no buscando simplemente interoperabilidad.

Para ello, una característica fundamental en CR es la capacidad de los nodos para evaluar el estado del espectro. Los sistemas de radio cognitiva que se basan únicamente en este parámetro para tomar decisiones reciben el nombre de monitores del espectro^[4], frente a aquellos que emplean además otros indicadores, que se denominan radios cognitivas completas o "Radios de Mitola", al aproximarse más a la definición que plantea en origen^[2] el autor.

Desde entonces, motivada por los problemas incipientes en las redes inalámbricas, la investigación en este campo ha ido cobrando mayor interés. Enseguida hubo autores que, basándose en la compartición de información entre nodos, planteaban la implementación de un *plano de conocimiento* operando por encima de la red^[5]. Este nivel de conocimiento era mantenido con la información que recababan los nodos, extendida por toda la red para generar información contextual de la misma y ser la base de los algoritmos de decisión.

La integración de las técnicas de CR con el horizonte que se abre ante la compartición de información de estado entre los nodos de la red ha causado la aparición del término de Redes Cognitivas^[6] (Cognitive Networks). Las CN basan su funcionamiento en el aprendizaje autónomo de sus nodos y la mencionada colaboración para predecir el comportamiento de la red y mejorar sus prestaciones globales^[7]. En función de las necesidades de la red, se pueden tomar acciones para optimizar distintas variables como el consumo energético, la seguridad o la fiabilidad.

A raíz de la saturación del espectro, se busca flexibilizar su uso y mejorar su aprovechamiento basándose en técnicas de acceso dinámico al espectro^[8] (por sus siglas en inglés, DSA). Por lo general, el acceso al espectro suele recurrir a bandas no licenciadas, como las bandas para uso científico, médico o industrial (bandas ISM, en inglés). Las ventajas de implementar un uso oportunista del espectro son muy interesantes de cara a aprovechar huecos vacíos en el espectro (local o temporalmente libres) y evitar interferencias de otros sistemas operando en la misma banda.

Más recientemente, se están desarrollando nuevos estándares para redes inalámbricas basadas en radio cognitiva. Es el caso del IEEE 802.22^[9] para Redes Inalámbricas de Área Regional (Wireless Regional Area Networks) y el estándar Ecma 392^[10] que, empleando técnicas de radio cognitiva, se han diseñado para que los dispositivos de la red aprovechen los huecos espectrales en la banda de la televisión analógica.

2.3. REDES DE SENSORES INALÁMBRICAS COGNITIVAS

Tras una breve introducción a las redes inalámbricas de sensores, sus principales estándares y protocolos y la gran variedad de aplicaciones que se plantean con el uso de

ellas, a continuación estudiaremos el resultado de dotar a las WSN de capacidades cognitivas, dando lugar a las redes de sensores inalámbricas cognitivas (CWSN).

Podemos entender las CWSN como una particularización del concepto de red cognitiva sobre WSN. Las CWSN son una evolución natural de las redes inalámbricas de sensores, pues los estándares del IEEE utilizados en WSN ya contemplan el acceso a varias zonas del espectro. Además, los requisitos de fiabilidad y bajo consumo que imponen WSN también se persiguen en CR^[19], aunque CR no es la única técnica cognitiva empleada en CWSN.

Los problemas generales de las redes inalámbricas siguen estando presentes en CWSN. De hecho, la saturación del espectro en las zonas sin licencia es mucho más significativa y el acceso dinámico al espectro es cada vez más necesario. La coexistencia de los sistemas 802.15.4 y 802.11 en la banda de 2,4 GHz supone un mayor nivel de interferencias y una reducción de las prestaciones. Varias bandas de frecuencia en UHF¹ permitirían un mayor alcance con mejores cifras de eficiencia energética, lo que redundaría en la posible reducción del número de nodos y la consecuente simplificación de la topología de la red^[20].

A los problemas de escasez del espectro (fruto del uso ineficiente que se hace de él) y el mayor nivel de interferencias se le suman otros nuevos:

- La demanda creciente de tráfico que se ha observado en la última década, debida en gran parte al mayor número de redes y terminales. Una evidencia de este hecho es la vertiginosa penetración en el mercado de los teléfonos inteligentes (smartphones) y las tabletas.
- Tráfico notablemente más diversificado en un entorno que cada vez es más complejo y heterogéneo.
- Las técnicas enfocadas a la mejora de las comunicaciones y la eficiencia espectral que se han venido utilizando hasta la fecha se han explotado prácticamente hasta la extenuación, por lo que la necesidad de nuevas soluciones es un problema acuciante.

En ^[21] se señalan estas tendencias y se destacan las oportunidades que ofrecen la cooperación entre redes de diferente naturaleza, la colaboración entre nodos de la red para mejorar las prestaciones globales y la capacidad de adaptación al inherente dinamismo de las redes inalámbricas.

La arquitectura y la implementación de las técnicas cognitivas sigue siendo un tema fundamental de la investigación en este campo. Si bien existen diversos enfoques, todos se basan en el aprendizaje del entorno y la realimentación de otros nodos, el razonamiento y la predicción del comportamiento de la red y el mantenimiento de información contextual a nivel de red.

¹ Ultra-High Frequency - Corresponde a la banda del espectro electromagnético que ocupa el rango de frecuencias entre 300 MHz y 3 GHz.

El modelo que se bosqueja en ^[21] se basa en agentes que implementan funciones básicas y que establecen y mantienen un canal de control virtual. Este canal virtual es independiente de la ruta física que sigan los datos, y forma parte del plano de control que utilizan los agentes para intercambiar mensajes de gestión de la red.

En ^[22] se contemplan algunas propuestas de arquitecturas cognitivas hasta la fecha, basadas en la idea del Plano de Conocimiento planteado originalmente en ^[5] o en varios niveles de abstracción que desarrollan diferentes funciones cognitivas. Además, los autores de ^{[22][23]} resaltan problemas pendientes de resolver en CWSN, entre los que destacan:

- La necesidad de buscar nuevas formas de encaminamiento que sean sensibles a la eficiencia energética de la red.
- En línea con lo anterior, la necesidad de prolongar la vida útil de los nodos.
- La detección fiable de eventos y su notificación a otros nodos de la red.
- La posibilidad de optimizar la red y los nodos en función de la aplicación y las demandas de los usuarios o la red.

Con todo, CWSN es una tecnología joven con mucho camino que recorrer, pero que plantea un nuevo paradigma en las comunicaciones inalámbricas en el que las redes comparten de manera natural los recursos y negocian un compromiso con vistas a lograr mejores resultados frente a un objetivo concreto (mínimo consumo, seguridad o fiabilidad, entre otros).

Simuladores para CWSN

Para evaluar los diferentes modelos y arquitecturas, se ha invertido un esfuerzo importante en desarrollar simuladores y plataformas de prueba. A continuación comentaremos los más relevantes a día de hoy.

- **NS-2 / NS-3:** NS ^[24] es un simulador de redes basado en eventos discretos que soporta multitud de protocolos de redes tanto inalámbricas como cableadas, muy utilizado en la simulación de redes móviles ad-hoc. Inicialmente se diseñó para redes IP, aunque soporta también otras redes. Es un simulador libre desarrollado fundamentalmente en C++ bajo licencia GPL. La versión NS-2 ha sido relegada por la versión NS-3, que mejora el rendimiento e incorpora interfaces de programación (APIs) estándar. También incorpora algunos módulos desarrollados en Python. La última versión de NS-3 data de Diciembre de 2012 (v3.16).
- **SENDORA:** ^[25] Se basa en NS-2 y centra la simulación en la asignación dinámica del espectro y otros aspectos de radio cognitiva. Su uso se ha visto desplazado a un segundo plano por la popularidad de otros simuladores.
- **OMNeT++:** No es un simulador propiamente dicho, sino un entorno de simulación con una arquitectura abierta y modular para modelar eventos discretos ^[26]. No obstante, su principal uso es la simulación de redes y

protocolos. La mayor parte de los módulos se han desarrollado en C++, aunque para ciertos módulos se emplea el lenguaje NED. Es una herramienta muy potente y fuertemente orientada a objetos, gratuita para uso académico o no lucrativo. Última versión: Marzo 2012.

- **Castalia:** ^[27] Se trata de un simulador de WSN basado en OMNeT++ utilizado para probar algoritmos y protocolos en un modelo realista de canales radio. Desde el año 2007 es un proyecto de código abierto bajo licencia APL. La versión 3.2 fue lanzada en 2011.

Existen otros simuladores como SimPy o JiST/SWANS que emplean Python y Java respectivamente, aunque en general sus prestaciones en cuanto a tiempo de simulación y uso de memoria son peores que las de OMNeT++ o NS ^[28]. Además, existen simuladores específicos, como es el caso de TOSSIM y extensiones ^[29], un simulador para aplicaciones que utilicen TinyOS².

Bancos de prueba para CWSN

No existen muchas implementaciones de nodos para CWSN, y la mayoría de ellas están destinadas a formar parte de plataformas y escenarios de prueba. Estos sistemas de prueba (test-beds) suelen emplazarse en universidades y centros de investigación para recabar datos y probar algoritmos que realimentan el diseño tanto de los propios nodos de la red, como los simuladores de CWSN.

Las dos principales plataformas son:

- **TWIST:** ^[30] Es la plataforma de pruebas en interiores para WSN del grupo de investigación en redes de telecomunicación (TKN) de la Technische Universität Berlin (por su acrónimo en inglés, TWIST). Se trata de un banco de pruebas para experimentar con aplicaciones de redes de sensores inalámbricas que implementa servicios básicos como la configuración de los nodos o la obtención de datos de aplicación o trazas de depuración de toda la red.
Es uno de los mayores bancos de pruebas a nivel mundial. Consta de 102 nodos TmoteSky³ trabajando a 2,4 GHz y otros 102 nodos eyesIFX⁴ en 868 MHz repartidos en una extensión de más de 1500 m² siguiendo un patrón de rejilla bastante regular, con una distancia entre nodos de 3m. La capacidad de auto-configuración, el uso de hardware con interfaces estándar como USB y el software de código abierto hace que su arquitectura sea altamente escalable, replicable y con un coste asequiblemente bajo.
- **VT-CORNET Wireless @ Virginia Tech** ha desarrollado una plataforma con 48 nodos repartidos en 4 plantas para poner a prueba algoritmos, protocolos,

² TinyOS es un sistema operativo diseñado por la Universidad de California en Berkeley, en colaboración con Intel, para funcionar sobre sistemas empujados en WSN.

³ <http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>

⁴ www.infineon.com, eyesIFX datasheet.

aplicaciones y hardware específico en un entorno realista. Principalmente, los nodos utilizan radio definida por software para implementar técnicas de radio cognitiva, aunque también se desea desarrollar otras técnicas enfocadas a la auto-organización de los algoritmos y la mejora de la seguridad en la red. Uno de sus proyectos plantea desarrollar una arquitectura de código abierto para flexibilizar la plataforma.

Las plataformas mencionadas no incluyen nodos con acceso a varias zonas de frecuencia, sino varios tipos de nodos con distintas interfaces o nodos reconfigurables a nivel hardware. La flexibilidad de uso del espectro se limita a los canales definidos en la banda en los que puede trabajar cada transceptor.

Con todo, los nodos de las plataformas para CWSN son similares a la mayoría de dispositivos comerciales disponibles como nodos de redes de sensores, con la particularidad de que sobre ellos se han implementado o embebido simuladores, módulos y servicios añadidos para CR. Estos nodos típicamente incluyen un microcontrolador de bajo consumo y un transceptor comercial que emplea algún protocolo compatible con 802.15.4. y acceso a una banda sin licencia. No obstante, existen también algunos productos comerciales con varios transceptores que se comentan a continuación:

- **WaspMote:** Es una solución comercial de Libelium basada en un microcontrolador ATmega1281 de ATMEL a 8 MHz con 128 kB de Flash que permite incorporar un módulo transceptor comercial a elegir entre un abanico de posibilidades (ZigBee, Wi-Fi, Bluetooth, entre otros). Si se combina con una placa de expansión, se puede incluir un segundo módulo de RF adicional. Está orientado a ser un nodo válido para la mayoría de redes de sensores con protocolos comerciales.
- **Meshlium:** Es también un producto de Libelium más orientado a servir como nodo pasarela o "supernodo" en redes tipo malla. Se trata de un router Linux que puede contener 5 interfaces: Wi-Fi 2.4GHz, Wi-Fi 5GHz, 3G/GPRS, Bluetooth y ZigBee. También puede integrar un módulo GPS para aplicaciones móviles y en vehículos, y ser alimentado con placa solar o con baterías.

Una de las principales motivaciones para diseñar un nodo de CWSN dentro del laboratorio es el hecho de disponer de una plataforma de pruebas con varias interfaces radio sobre la que testear los algoritmos y estrategias cognitivas que se desarrollen en el Laboratorio de Sistemas Integrados, ya que existen muy pocas iniciativas a nivel internacional similares.

El desarrollo del firmware y software de control de las interfaces del nodo, así como una interfaz de aplicación sencilla es el siguiente paso para llegar a disponer de una plataforma de estas características.

3. FASE DE ANÁLISIS

En el capítulo anterior se han introducido los principales problemas que sufren hoy en día las comunicaciones inalámbricas y cómo ciertas técnicas, agrupadas bajo el calificativo de cognitivas, plantean un nuevo paradigma de comunicación para tratar de solventar o mitigar algunos de ellos.

En esta sección se hará hincapié los requisitos concretos que abordar en la fase de diseño e implementación del nodo. Para ello se va a analizar el prototipo demostrador anteriormente realizado en el laboratorio (en adelante primer dispositivo cognitivo o FCD, First Cognitive Device) buscando optimizar ciertos aspectos del mismo, suplir algunas de sus carencias e introducir funcionalidades deseables que en el diseño anterior no estaban presentes.

3.1. MODELO DE UN NODO PARA REDES DE SENSORES INALÁMBRICAS COGNITIVAS

En primer lugar se consideran qué características y capacidades deben incluir los nodos obligatoriamente para poder implementar una red de sensores inalámbrica cognitiva. De este modo se detectarán necesidades que permitirán sentar una base sobre la que empezar a trabajar en el diseño del nodo de nueva generación (NGD, Next Generation Device). A continuación, se recogen dichos requisitos:

- Un nodo de CWSN debe tener acceso a distintas bandas de frecuencia del espectro radioeléctrico. Este hecho es fundamental para implementar técnicas de acceso dinámico al espectro, aprovechar huecos espectrales locales o temporales y, en general, hacer un mejor uso de él.
- Los nodos de la red deben de ser capaces de reconfigurar parámetros del nivel radio, como por ejemplo el canal o la potencia de transmisión.
- Un nodo de CWSN debe poder monitorizar el estado del espectro, mediante técnicas tales como el escaneo del nivel de ruido para poder evaluar el entorno en el que opera.
- Un requisito inherente a las redes de sensores inalámbricas es que el diseño del nodo esté fuertemente orientado a bajo consumo. El nodo debe poder apagar módulos que no estén en uso, implementar estrategias de ahorro de energía y, bajo ciertas condiciones, funcionar con un sistema de alimentación autónomo.
- Debido a que el número de nodos en una red de sensores puede ser potencialmente grande, la propia naturaleza de las redes de sensores inalámbricas imponen requisitos de escalabilidad.

- Por el mismo motivo, se persigue que el coste de los nodos sea el mínimo posible.
- En CWSN la computación suele ser ubicua y estar distribuida entre todos los nodos que conforman la red. Los nodos recaban información, la comparten con otros miembros de la red y toman decisiones en mayor o menor medida. Para implementar estas capacidades, se requiere que los nodos tengan una capacidad de procesamiento moderada.

3.2. EVALUACIÓN DEL PROTOTIPO DEMOSTRADOR

Dentro del grupo de trabajo del laboratorio se realizó un primer nodo para CWSN cuyo principal objetivo era demostrar algunas funcionalidades y servir como diseño base para la actividad futura de esta línea de investigación. Este proyecto parte del nodo FCD diseñado por Fernando López Lara ^[1] y trata de evaluar sus fortalezas y debilidades para extraer conclusiones y opciones de mejora que puedan plasmarse en un nuevo diseño.

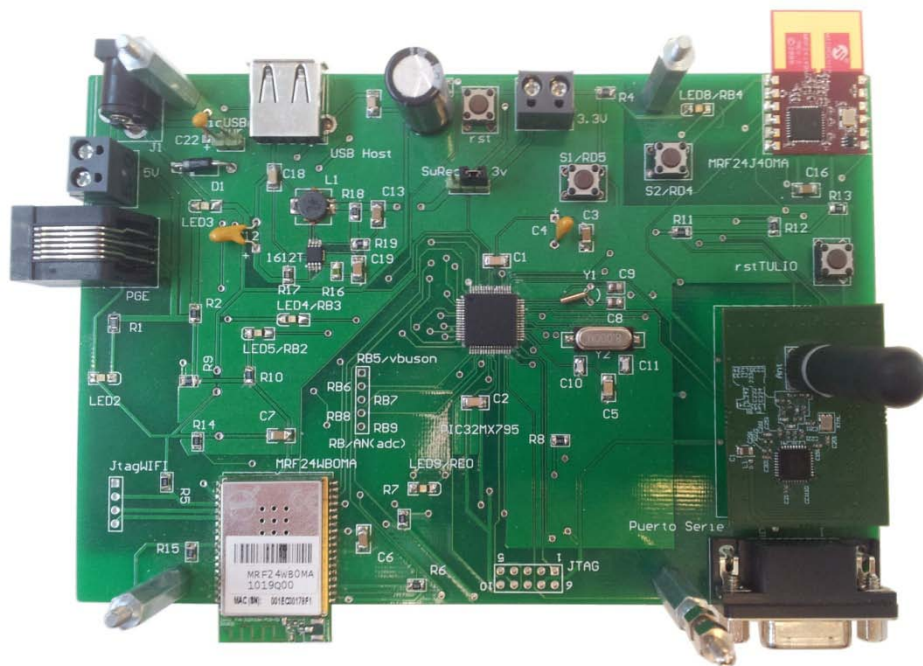


Figura 1: Foto a tamaño real del prototipo FCD diseñado en el laboratorio

El FCD dispone de un microcontrolador PIC de Microchip de alta gama, basado en un núcleo de 32 bits de MIPS Technologies: el MIPS M4K. Se trata de un procesador con un juego reducido de instrucciones (RISC) y un pipeline de cinco etapas cuya arquitectura es tipo Harvard, con bus de instrucciones y bus de datos independientes. El

procesador puede funcionar a una frecuencia de reloj de hasta 80 MHz y alcanzar un rendimiento de 1,65 DMIPS/MHz⁵.

El PIC lleva integrada memoria flash y memoria RAM de propósito general. Dentro de la familia de PICs de 32 bits se escogió el modelo con mayor cantidad de memoria, el PIC32MX795F512H, con 512 kB de flash y 128 kB de SRAM, para poder albergar aplicaciones complejas sin que la memoria fuese el recurso limitante.

En el diseño del FCD se incluyeron tres interfaces radio con acceso a las bandas ISM de 868 MHz y 2,4 GHz. En la banda de 868 MHz se empleó un módulo transceptor Tulio™ de AWD⁶ basado en un microcontrolador CC1110 de Texas Instruments, y para las interfaces en 2,4 GHz se eligieron dos módulos de Microchip: un módulo que emplea protocolos propietarios de Microchip (MiWi P2P / MiWi Mesh/ MiWiPRO)TM y también un módulo Wi-Fi (IEEE 802.11). El primero de ellos está orientado a redes inalámbricas de baja tasa binaria (LR-WPAN) y es compatible con el estándar 802.15.4 del IEEE, mientras que el módulo Wi-Fi se incluyó con vistas a aportar interconectividad.

El fabricante proporciona una pila de protocolos software para gestionar cada interfaz radio. En el caso del Tulio, el firmware del CC1110 venía cedido por AWD, y para la comunicación con el PIC se diseñó un protocolo de comunicación sencillo a través de la UART. Microchip ofrece gratuitamente la pila de MiWi y una pila TCP/IP propia para sus microcontroladores. La pila de protocolos de MiWi es ligera y no requiere tanta memoria como la de Wi-Fi, que es mucho más extensa y compleja, en función de las opciones que se habiliten. La comunicación con los módulos transceptores de Microchip se realiza a través de interfaces SPI.

Por último, el nodo consta de un puerto USB y un puerto serie RS232 para volcar trazas de depuración a un PC (o datos a una memoria externa en el caso de USB) y un oscilador secundario para poder implementar funcionalidades de sincronización y reloj de tiempo real.

La figura 2 resume todos los aspectos del diseño del prototipo FCD que se han comentado anteriormente.

Ahora, si bien el prototipo demostrador es un trabajo muy completo y próximo al concepto de nodo para CWSN que se ha introducido, se pueden mejorar ciertos aspectos y paliar algunos problemas que surgen de su evaluación y caracterización.

El módulo Tulio, aunque presenta una gran ventaja frente al resto de interfaces en cuanto a consumo y permite acceder a otra región del espectro, resulta ineficiente al incorporar otro microcontrolador, y añade complejidad en lo referente a la

⁵ Una de las muchas maneras de cuantificar el rendimiento de un procesador es utilizar el banco de pruebas Dhrystone y medir cuántos Millones de Instrucciones Por Segundo puede realizar, normalizando el resultado con la frecuencia de reloj del sistema (DMIPS/MHz).

⁶ Advanced Wireless Dynamics es una spin-off de la Universidad Politécnica de Madrid que surgió dentro del Laboratorio de Sistemas Integrados y cuya actividad radica en el diseño de soluciones de eficiencia energética, control industrial, entornos inteligentes y redes de sensores.

intercomunicación entre éste y el PIC. Además, este hecho, hace que el consumo general del nodo sea mayor.

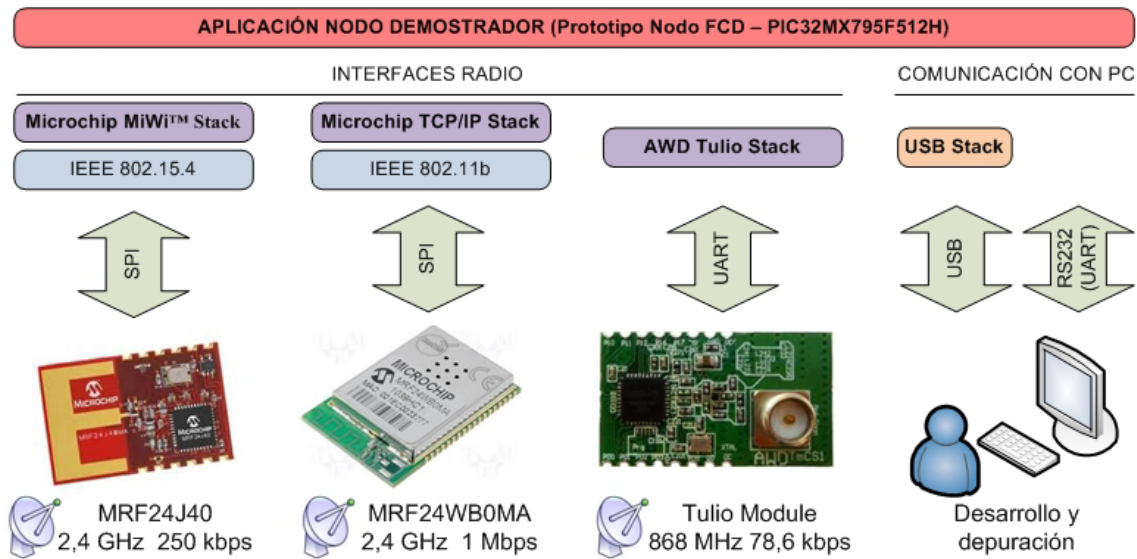


Figura 2: Protocolos e interfaces del nodo FCD

El módulo Wi-Fi puede enviar datos a una gran tasa binaria, pero ésta resulta exagerada dentro del marco de WSN, donde las necesidades pueden ser tan bajas como unas centenas de bits por segundo. Además su consumo es muy elevado y surgen problemas con la hibernación o suspensión del módulo, lo que supone un impedimento para que el sistema pueda funcionar con alimentación autónoma durante un periodo de tiempo aceptable. Aunque esta interfaz busca principalmente servir de pasarela de comunicaciones, merece la pena plantear nuevas alternativas y cotejar si se mantiene en el nuevo diseño del nodo.

La complejidad y el caos resultante de combinar pilas de protocolos diferentes, concebidas inicialmente por los fabricantes para trabajar de manera independiente, hace que el desarrollo de la capa cognitiva y las aplicaciones sea francamente complicado, por lo que es necesario desarrollar una capa de abstracción hardware que ofrezca una interfaz de aplicación sencilla.

En cuanto al hardware, se plantea mejorar módulos tales como el sistema de alimentación, reducir el tamaño general del nodo e introducir cambios que favorezcan un nuevo diseño más escalable y orientado a la realización de pruebas. No obstante, el rediseño hardware queda fuera del marco de este proyecto.

3.3. ANÁLISIS DE REQUISITOS

3.3.1. Definición de nuevos requisitos.

Después de comentar brevemente los requisitos que debe cumplir un nodo de CWSN y evaluar el primer dispositivo cognitivo del laboratorio, se procede a definir los requisitos adicionales del nuevo diseño. Éstos irán orientados a suplir carencias del nodo FCD, mejorar subsistemas y añadir nuevas funcionalidades deseables para el uso específico que se hará de la nueva plataforma en el futuro.

El principal problema del FCD probablemente sea su elevado consumo. Se debe invertir un gran esfuerzo en tratar de reducirlo para que, efectivamente, podamos hablar de un nodo de una red de sensores, manteniendo suficiente capacidad de procesamiento para WSN y bajo coste. El diseño del NGD debe implementar técnicas de ahorro de energía y estar orientado a poder funcionar, bien con baterías, o bien con una fuente de alimentación externa.

Al hilo de lo anterior, sería interesante que el nuevo nodo FGD sólo dispusiera de un único microcontrolador que gestione todos los transceptores e interfaces de comunicación del nodo. Este hecho redundaría en cierto ahorro energético al simplificar la comunicación y la sincronización entre los módulos, y en una mayor eficiencia del firmware embebido.

Es vital que el nodo pueda evaluar el estado de diferentes bandas del espectro. Por las limitaciones que surgen del uso del espectro radioeléctrico, las bandas que resultan más atractivas para el diseño son las ya mencionadas ISM, ya que no requieren licencia de uso. Las zonas del espectro donde típicamente trabajan dispositivos inalámbricos son:

- 433,05 - 434,79 MHz: Se trata de una banda ISM sólo disponible en la Región 1⁷, en la que principalmente operan dispositivos de corto alcance. En el diseño del FCD no se dispone de ningún transceptor en esta banda.
- 863 - 870 MHz: Se considera una banda ISM solo en Europa. En ella operan dispositivos de corto alcance (SRD) o de identificación por radiofrecuencia (RFID), y también, el módulo Tulio.
- 2,4 - 2,5 GHz: Es una banda ISM global en la que se ubican redes inalámbricas de área local (WLAN) y área personal (WPAN) tales como Wi-Fi, ZigBee, BlueTooth... En el nodo FCD, el módulo MiWi y el módulo Wi-Fi se encuentran en esta banda.

⁷ La ITU, en su sección de regulación de las radiocomunicaciones, define tres regiones para gestionar el espectro radioeléctrico a nivel mundial. La región 1 engloba Europa, África, Oriente Medio y parte de Asia; la región 2 cubre el continente americano, Groenlandia y algunas islas del Pacífico; la región 3 cubre el resto de territorios, principalmente China, el sureste asiático y Oceanía.

- 5,725 - 5,875 GHz: Es una banda menos utilizada, donde se hallan algunos canales de Wi-Fi. Ningún transceptor del nodo original utiliza estas frecuencias de trabajo.

La banda de 434 MHz resulta muy interesante para trabajar con redes de sensores y se desea incluir un transceptor que opere en ella para extender el acceso al espectro radioeléctrico del nodo. Habrá que seleccionar el transceptor que reúna las mejores características de cara a nuestro diseño.

Las bandas de 868 MHz y 2,4 GHz se desean mantener, si bien hay que evaluar si se conservan los transceptores del diseño original en ellas. El módulo Wi-Fi tiene un consumo de 154 mA en transmisión (un orden de magnitud mayor que el de otras interfaces) que es demasiado elevado para un nodo de CWSN, y además, se solapa espectralmente con el módulo MiWi.

En cuanto al Tulio, ya se ha comentado que el hecho de incorporar otro microcontrolador (System-on-Chip) supone cierta ineficiencia. No obstante, es una interfaz de bajo consumo que opera en 868 MHz y puede justificarse, tanto el hecho de conservarla, como el de reemplazarla por otra de similares características.

Sin embargo, la banda de 5 GHz se ha desestimado para el nuevo diseño, ya que un transceptor capaz de operar en esta banda gravaría fuertemente el consumo general del nodo. Incluir una interfaz en esta zona del espectro supone ir en contra del requisito de bajo consumo de WSN.

Por otro lado, ya que se pretende implementar un nodo que sirva a su vez como plataforma de desarrollo, es deseable que el nodo sea lo más versátil y flexible posible, admitiendo modificaciones y facilitando la depuración y el test de los diferentes módulos que se vayan incluyendo. Algunos requisitos que se desprenden de esto último son:

- Disponer de una tasa binaria mayor que la que típicamente se utiliza en WSN para explorar nuevos usos y aplicaciones.
- Mantener opciones de depuración para poder volcar trazas y verificar el buen funcionamiento de los módulos.
- Favorecer la modularidad y el acceso a los recursos no empleados por el microcontrolador para futuras extensiones y módulos externos.
- Implementar una capa de abstracción hardware que ofrezca una interfaz de programación al nivel de aplicación. Como este elemento es el núcleo del proyecto, se estudiará en detalle a continuación.

3.3.2. Capa de abstracción Hardware.

El dispositivo cognitivo de nueva generación pretende ser un prototipo demostrador orientado a la investigación, permitiendo dar soporte a múltiples

aplicaciones enmarcadas dentro de las redes inalámbricas de sensores cognitivas. No obstante, el diseño e implementación de estas aplicaciones no sería nada fácil si éstas tuvieran que lidiar de manera directa con todas las particularidades de cada uno de los protocolos y los transceptores del diseño. Este hecho conllevaría un incremento del tiempo de desarrollo muy notable y dificultaría la depuración de errores.

Por tanto, para el rediseño del nodo es fundamental implementar un nivel que permita integrar la heterogeneidad de las interfaces radio y abstraer de su complejidad a los niveles superiores, bien las propias aplicaciones, bien otras capas que doten al nodo de inteligencia o funcionalidades cognitivas.

La capa de abstracción hardware implícitamente requiere sencillez de uso y manejabilidad, pues en caso contrario estaríamos aportando complejidad en lugar de abstraer de ella, pero a su vez debe cubrir el mayor rango de funcionalidad posible. En consecuencia, debe existir un compromiso entre ambas fuerzas a la hora de implementar este nivel.

Como requisitos mínimos de la interfaz de aplicación se han definido los siguientes puntos:

- Disponer de una función de inicialización del sistema que cubra los siguientes aspectos:
 - Configuración del sistema (interrupciones, temporizadores y otros módulos internos, etc).
 - Inicialización de las interfaces de comunicación con periféricos.
 - Protocolos de red: autodescubrimiento o inicialización de la red, conexión con nodos vecinos, inicialización de variables de estado y tablas de rutado, etc.
- Funciones para cambiar el canal y la potencia de transmisión de una interfaz radio.
- Ofrecer la posibilidad de escanear canales para recabar información sobre el estado del espectro en todo momento.
- Implementar la posibilidad de desconectar módulos o que éstos pasen a estado de hibernación, en el que su consumo energético se reduzca drásticamente para poder reducir el consumo general del nodo. Particularmente, es primordial que los transceptores radio puedan conmutar de estado activo a dormido a merced de la aplicación.
- En línea con lo anterior, se deben implementar funciones para que el microcontrolador utilice sus modos de bajo consumo. Así, los intervalos de tiempo que no requieran procesamiento por parte del microcontrolador pueden suponer un ahorro energético, prolongando la autonomía del nodo.
- Disponer de mecanismos para habilitar las funciones de seguridad, tales como la encriptación de los paquetes.

- Una de las funcionalidades esenciales de un nodo de comunicación es poder enviar y recibir paquetes de datos. Se debe poder hacer uso de estas funciones en todas las interfaces del nodo y distinguir la interfaz por la que llegue tráfico.
- Permitir extraer y restaurar variables de estado del nodo, con vistas a que la aplicación pueda utilizarlas, enviarlas a través de la red, o restaurarlas en caso de errores.

3.4. EVALUACIÓN DE ALTERNATIVAS DE DISEÑO.

El diseño del nuevo nodo cognitivo pasa por la elección de ciertos componentes clave: el microcontrolador, encargado del procesamiento de la información y la coordinación de las distintas tareas y funciones del nodo; y los transceptores radio, que determinan el acceso al espectro radioeléctrico del nodo y su capacidad de recabar información sobre su estado.

Para tratar de tomar la mejor decisión de diseño se ha realizado un estudio comparativo de los diferentes productos comerciales disponibles.

3.4.1. - Microcontroladores

Existen multitud de microcontroladores en el mercado, con distintas arquitecturas y orientados a un gran rango de aplicaciones diferentes. Para el prototipo demostrador, su diseñador optó por un microcontrolador PIC de 32 bits tras evaluar las alternativas comerciales que ofrecían las compañías Atmel y Microchip. Se restringió el estudio a estas dos compañías por su gran variedad de productos y por su compromiso entre bajo coste y rendimiento. Finalmente se optó por Microchip porque sus microcontroladores permitían manejar mayor número de periféricos y dentro del laboratorio había mayor experiencia con los microcontroladores de Microchip, aunque ambas opciones hubiesen sido válidas.

No obstante, para el rediseño del nodo no se descarta cambiar el microcontrolador, así que se incluye un pequeño estudio más actual de los microcontroladores del mercado, centrado en aquellos más orientados a bajo consumo y redes de sensores. Para poder manejar diversos periféricos, soportar aplicaciones futuras y una carga de procesamiento media, se ha acotado la búsqueda a los microcontroladores con arquitectura de 32 bits que al menos dispongan de 128 kB de memoria flash.

La familia PIC32MX de Microchip.

La oferta de microcontroladores de esta compañía es muy variada y podemos encontrar desde microcontroladores muy básicos a otros con mayor rendimiento,

pasando por diferentes arquitecturas. En nuestro caso de estudio nos centraremos en la familia de microcontroladores de 32 bits: la familia PIC32MX.

En secciones anteriores, ya se han comentado brevemente las características técnicas del microcontrolador del nodo demostrador. En general, son aplicables a toda la familia PIC32MX, pues las diferencias entre modelos de esta gama radican en la cantidad de memoria y los periféricos que incorporan, manteniendo la misma CPU.

De la evaluación y utilización del prototipo ya diseñado en el laboratorio se desprende que el microcontrolador soporta perfectamente la ejecución del firmware del nodo y es una elección buena, si bien la cantidad de memoria del microcontrolador (tanto Flash como RAM) es suficiente, y puede ajustarse algo más, aun dejando cierto margen para desarrollos futuros. Esto puede redundar en un coste significativamente inferior, por lo que consideraremos otros modelos con menor memoria en la tabla resumen de microcontroladores.

La familia AVR UC3 de Atmel.

Podríamos decir que se trata de la familia equivalente a los PIC32MX de Microchip, ya que también agrupa los microcontroladores de 32 bits de la compañía y son dispositivos bastante similares a éstos. El procesador (de tipo RISC) puede funcionar con un reloj de hasta 66 MHz y lograr un rendimiento de 1,50 MIPS/MHz. En función del modelo, puede disponer de multitud de periféricos, interfaces de comunicación y conversores.

Atmel ofrece su propio entorno de desarrollo para sus microcontroladores y transceptores, favorece la integración de sus propios productos (y algunos productos de terceros asociados), librerías software para diversas aplicaciones y programas de ejemplo y documentación y soporte técnico. Con todo, es la mejor alternativa a Microchip y, por tanto, cotejaremos si incluir uno de sus microcontroladores que cumpla con los requisitos.

Otros microcontroladores basados en procesadores ARM Cortex-M3/Cortex-M4.

Aunque Microchip y Atmel son dos de los principales fabricantes y presentan ventajas para nuestra aplicación concreta, también destacan otros fabricantes de microcontroladores como Texas Instruments, NXP, Freescale...

Se ha detectado que muchos de estos fabricantes incluyen microcontroladores que emplean procesadores ARM dentro de su portfolio, principalmente los de la serie Cortex-M3 y Cortex-M4. Estos procesadores de 32 bits son muy populares ya que consiguen un buen equilibrio entre rendimiento y coste, su consumo es bastante reducido y son altamente versátiles, por lo que se plantean como alternativa para un gran número de aplicaciones.

En concreto, consideramos en el estudio algunos dispositivos de las series SAM3L y SAM4L de Atmel y la familia Stellaris™ de Texas Instruments.

Comparativa de microcontroladores de 32 bits														
	Microcontrolador	Frecuencia máx. (MHz)	Memoria (kB)		Periféricos y Conectividad						ADC	Pines	Consumo 3,3V, 25°C	€/ud**
			Flash	RAM	SPI	UART	I2C	USB*	CAN	Ethernet				
Microchip (PIC32MX)	PIC32MX795F512H PIC32MX795F512L	80	512	128	3 4	6	4 5		2	10/100 BaseT	16 canales (10 bits)	64 100	N.D.	9.47 9.57
	PIC32MX775F512H PIC32MX775F512L				3 4		4 5		2			64 100	Run: 85 mA Idle: 36 mA	8.75 9.47
	PIC32MX775F256H PIC32MX775F256L		256	64	3 4		4 5	Full Speed	2			64 100	Sleep: 41 µA	8.07 8.79
	PIC32MX675F256H PIC32MX675F256L				3 4		4 5	Host Dev. OTG	0			64 100		7.11 8.35
	PIC32MX764F128H PIC32MX764F128L		128	32	3 4		4 5		1			64 100	Run: 39 mA Idle: 17 mA	5.95 6.54
	PIC32MX664F128H PIC32MX664F128L				3 4		4 5		0			64 100	Sleep: 20 µA	5.80 6.41
ATMEL (AVR UC3)	AT32UC3C1512C	66	512	68	7	5	3	Full Speed Dev. OTG	2	1	16 canales (12 bits)	100	Run: 34 mA Idle: 17 mA Sleep: 43 µA	14.66
	AT32UC3C1256C		256											12.42
	AT32UC3C1128C		128	36										10.11
	AT32UC3A4256		256	128	6	4	2	High Speed	0	NO	8 canales		Run: 44 mA Idle: 24 mA Sleep: 54 µA	11.53
	AT32UC3A4128		128	128				Dev. OTG			(10 bits)			9.39
Basados en ARM Cortex-M3 / M4	ATSAM3X8C ATSAM3X4C (Atmel, Cortex-M3)	84	512 256	96 64	4	4	2	High Speed Host Dev.	2	1	16 canales (12 bits)	100	Run: 77 mA Idle: 29 mA Sleep: 27 µA	8.76 7.64
	ATSAM4LC4C ATSAM4LC2C (Atmel, Cortex-M4)	48	256 128	32	1	4	4	Full Speed Host Dev.	0	NO	15 canales (12 bits)	100	Run: 17 mA Idle: 8 mA Sleep: 6 µA	N.D.
	LM4F232H5QC (TI, Cortex-M4)	80	256	32	4	8	4 6	Full Speed OTG	2	NO	12 22 canales (12 bits)	64 100	Run: 30 mA Idle: 12 mA Sleep: 5 µA	10.21

Tabla 1 : Comparativa de microcontroladores de 32 bits

* USB2.0: Full Speed (12 Mbps máx.) High Speed (480 Mbps máx.). Implementaciones: Host, Device, OnTheGo (OTG)

** Precios orientativos. Mínimo precio unitario ofertado en España por los distribuidores RS Online, Farnell, DigiKey y Mouser (Enero 2013).

En la anterior tabla se puede ver una comparativa con las principales características de los microcontroladores. Como aclaración, en lo referente a consumo se han incluido tres medidas que se han denominado "Run", "Idle" y "Sleep"

indistintamente de como las denomine cada fabricante en su hoja de especificaciones. Estos datos de consumo reflejan las siguientes situaciones:

- Modo "Run": se trata del modo normal de ejecución. La CPU y el reloj del sistema están habilitados, ejecutando código desde la flash con la máxima frecuencia de reloj.
- Modo "Idle": se ha seleccionado el dato referente a la operación con la CPU detenida y el reloj del sistema habilitado a la frecuencia máxima. En este modo podrían seguir en funcionamiento la mayoría de periféricos.
- Modo "Sleep": se comparan las cifras de consumo con la CPU y el reloj del sistema detenido, lo que implica que, salvo algunas excepciones, los periféricos están deshabilitados.

3.4.2. Transceptores

Uno de los aspectos claves del nodo a diseñar es su capacidad para operar en diferentes bandas de frecuencia gracias a la inclusión de transceptores o módulos transceptores en el diseño. La elección de los mismos condiciona fuertemente las prestaciones del nodo, tanto eléctricas como funcionales, ya que los protocolos empleados repercuten en las capacidades del nodo dentro de la red (tasa binaria, monitorización del espectro...) e imponen sus limitaciones (roles de la red, máximo número de nodos, etc).

El estándar IEEE 802.15.4, orientado a redes inalámbricas de área personal con bajas tasas de datos, ha tenido una adopción bastante exitosa por parte de los fabricantes. Este hecho, unido a que el estándar solo define las capas de nivel físico y nivel de enlace, ha provocado la proliferación de múltiples protocolos de red, tanto propios de cada fabricante, como desarrollados por asociaciones de ellos (como es el caso de ZigBee™). Los protocolos compatibles con la especificación de 802.15.4 en Europa operan principalmente en la banda ISM de 2,4 GHz.

En ese sentido, dado que se desea que el nodo emplee protocolos lo más estandarizados y homogéneos posibles, solo se han considerado los transceptores de la banda de 2,4 GHz que cumplen con la especificación del estándar. En cambio, para los transceptores que operan en cualquiera de las bandas ISM por debajo de 1 GHz no se ha filtrado el estudio, dado que, además, muchos de ellos permiten trabajar con diferentes bandas simplemente alterando el circuito de adaptación de antena y la propia antena.

Las siguientes tablas incluyen una comparativa de los distintos transceptores y módulos de los principales fabricantes del mercado.

Transceptores y módulos en la banda de 2,4 GHz (IEEE 802.15.4)									
Fabr.	Transceptor	Frecuencias (GHz)	Protocolos y estándares	Máxima tasa binaria	Monitorización del espectro. Modos CCA*	Programas de ejemplo	MCU integrado	Consumo (typ.)	Precio** Unitario
Microchip	MRF24J40 (Transceptor)	2,405 - 2,48	ZigBee (pila SW de pago), P2P / MiWi / MiWiPRO (protocolos propietarios, IEEE 802.15.4 compatible)	250 kbps (IEEE 802.15.4) 625 kbps ("Modo Turbo")	Buena. Detección de nivel de energía.	Sí Ejemplos básicos de dos nodos (TX y RX).	NO	(Condiciones: Temp. = 25 °C, V _{DD} = 3.3 V) TX: (P _{máx} = 0 dBm) I = 23 mA RX: I = 19 mA, Sens. / P _{in máx} = -95 / +5 dBm SLEEP: I = 2 µA	3,41 €
	MRF24J40MA (Módulo)							(Condiciones: Temp. = 25 °C, V _{DD} = 3.3 V) TX: (P _{máx} = 0 dBm) I = 23 mA RX: I = 19 mA, Sens. / P _{in máx} = -94 / +5 dBm SLEEP: I = 2 µA	8,43 €
	MRF24J40MB (Módulo)							(Condiciones: Temp. = 25 °C, V _{DD} = 3.3 V) TX: (P _{máx} = 20 dBm) I = 130 mA RX: I = 25 mA, Sens. / P _{in máx} = -102 / -23 dBm SLEEP: I = 5 µA	15,02 €
	MRF24J40MC (Módulo)							(Condiciones: Temp. = 25 °C, V _{DD} = 3.3 V) TX: (P _{máx} = 19 dBm) I = 120 mA RX: I = 25 mA, Sens. / P _{in máx} = -108 / -23 dBm SLEEP: I = 12 µA	16,57 €
Atmel	AT86RF230 (Transceptor)	2,405 - 2,48	ZigBee (pila SW gratuita), IEEE 802.15.4, WirelessHart, SP100, 6LoWPAN, RF4CE	250 kbps (IEEE 802.15.4)	Muy buena. Detección de energía y detección de portadora.	Sí Diferentes escenarios. Guía para portar a diseños HW propios	NO	(Condiciones: Temp. = 25 °C, V _{DD} = 3 V) TX: (P _{máx} = 3 dBm) I = 16,5 mA RX: I = 15,5 mA, Sens. / P _{in máx} = -101 / +10 dBm SLEEP: I = 0,02 / 0,2 µA; IDLE: I = 1,5 / 7,8 mA	4,07 €
	AT86RF233 (Transceptor)	2,405 - 2,48 (802.15.4) 2,322 - 2,527		250 kbps (IEEE 802.15.4)				(Condiciones: Temp. = 25 °C, V _{DD} = 3 V) TX: (P _{máx} = 4 dBm) I = 13,8 mA RX: I = 11,8 mA, Sens. / P _{in máx} = -101 / +10 dBm SLEEP: I = 0,02 / 0,2 µA; IDLE: I = 0,3 / 5,2 mA	3,37 €
	ATmega128RFA1 (System-on-Chip)	2,405 - 2,48	ZigBee (pila SW gratuita), IEEE 802.15.4, IPv6 6LoWPAN, SP100, ISM, RF4CE, WirelessHART	2000 kbps			Sí	(Condiciones: Temp. = 25 °C, V _{DD} = 3 V) TX: (P _{máx} = 3,5 dBm) I = 18,6 mA RX: I = 16,6 mA, Sens. = -100 dBm SLEEP: I = 0,25 µA IDLE: I < 4,7 mA	7,62 €
Texas Instruments	CC2530 (System-on-Chip)	2,394 - 2,507	ZigBee, IEEE 802.15.4.	250 kbps (IEEE 802.15.4)	Buena. Detección de nivel de energía	Sí Ejemplos básicos de Zigbee y aplicaciones	Sí	(Condiciones: Temp. = 25 °C, V _{DD} = 3 V) TX: (P _{máx} = 4,5 dBm) I = 33,5 mA RX: I = 24,3 mA, Sens. / P _{in máx} = -97 / +10 dBm SLEEP: I = 1 µA	6,93 €
Freescall	MC13201 (Transceptor)	2,405-2,48	ZigBee, IEEE 802.15.4.	250 kbps (IEEE 802.15.4)	Media. Detección de energía.	Sí Uso de librerías y protocolos	NO	(Condiciones: Temp. = 25 °C, V _{DD} = 2.7 V) TX: (P _{TX} = 0 dBm) I = 30 mA RX: I = 37 mA, Sens. / P _{in máx} = -91 / +10 dBm SLEEP: I = 1 µA IDLE: I = 0,5 mA	3,12 €

Tabla 2: Transceptores y módulos en la banda de 2,4 GHz

Transceptores y módulos en bandas subGHz									
Fabr.	Transceptor	Frecuencias (MHz)	Protocolos y estándares	Máxima tasa binaria	Monitorización del espectro. Modos CCA*	Programas de ejemplo	MCU integrado	Consumo (typ.)	Precio** Unitario
Microchip	MRF49XA (Transceptor)	430 - 440 860 - 880 900 - 930 (a elegir)	P2P / MiWi / MiWiPRO (protocolos propietarios)	115,2 / 256 kbps	Media. Detección de energía sobre umbral.	Sí Ejemplos básicos de dos nodos.	NO	(Condiciones: Temp. = 25 °C, V _{DD} = 3.3 V) TX: (P _{máx} (f) = 5 - 7 dBm) I(f) = 22 - 24 mA RX: I(f) = 11 - 13 mA, Sens. / P _{in máx} = -110 / 0 dBm SLEEP: I = 0,3 µA	2.20
	MRF89XA (Transceptor)	863 - 870 902 - 928 950 - 960 (a elegir)		200 kbps	Buena. Detección de nivel de energía.			(Condiciones: Temp. = 25 °C, V _{DD} = 3.3 V) TX: (P _{máx} = 10 dBm) I = 25 mA RX: I = 3 mA, Sens. = -107 dBm SLEEP: I = 0,1 µA	2.30
	MRF89XAM8A (Módulo)	863 - 870		16 / 40 kbps	Buena. Detección de nivel de energía.			(Condiciones: Temp. = 25 °C, V _{DD} = 3.3 V) TX: (P _{TX} = 10 dBm) I = 25 mA RX: I = 3 mA, Sens. = -107 SLEEP: I = 0.1 µA	8.12
	MRF89XAM9A (Módulo)	902 - 928		28 / 200 kbps	Buena. Detección de nivel de energía.			(Condiciones: Temp. = 25 °C, V _{DD} = 3.3 V) TX: (P _{TX} = 10 dBm) I = 25 mA RX: I = 3 mA, Sens. = -105 dBm SLEEP: I = 0.1 µA	8.12
Atmel	AT86RF212 (Transceptor)	779 - 787 863 - 870 902 - 928 (a elegir)	ZigBee (pila SW gratuita), IEEE 802.15.4, 6LoWPAN	250 kbps (IEEE 802.15.4) 1000 kbps	Muy buena. Detección de nivel de energía y detección de portadora	Sí ZigBee y protocolo propio.	NO	(Condiciones: Temp. = 25 °C, V _{DD} = 3.0 V) TX: (P _{máx} = 10 dBm) I = 25 mA RX: I = 9,2 mA, Sens. / P _{in máx} = -101 / -5 dBm SLEEP: I = 0,2 µA / IDLE: I < 4,7 mA	4.35
Texas Instruments	CC1101 (Transceiver)	300 - 348 430 - 440 863 - 870 902 - 928	Protocolos propietarios	600 kbps	Muy buena. Detección de nivel de energía y detección de portadora	Sí Ejemplos básicos y aplicaciones	NO	(Condiciones: Temp. = 25 °C, V _{DD} = 3.3 V) TX: (P _{máx} = 12 dBm) I = 30 - 34 mA RX: I = 17 mA, Sens. / P _{in máx} = -95 / +10 dBm SLEEP: I = 0.5 µA	3.42
	CC1110 (System-on-Chip)				Media. Detección de energía sobre umbral.		Sí	(Condiciones: Temp. = 25 °C, V _{DD} = 3.0 V) TX: (P _{máx} = 10 dBm) I = 33 - 36 mA RX: I = 19 - 21 mA, Sens. / P _{in máx} = -94 / 10 dBm SLEEP: I = 0.5 µA	4.32

Tabla 3: Transceptores y módulos en bandas subGHz

*El estándar IEEE 802.15.4 exige que la capa de nivel físico implemente al menos una de las siguientes técnicas para comprobar si un canal está libre (conocidas como modos CCA, del inglés, Clear Channel Assessment): detección de energía por encima de un umbral, detección de portadora (cuyas características sean coherentes con las que define el estándar), una combinación de las dos técnicas anteriores, utilizando el protocolo ALOHA (acceso aleatorio con confirmación), o detectando el preámbulo de una trama dentro de un periodo de tiempo limitado. [Lo de debajo está sacado del estándar... por si quieres revisarlo]

** Precios orientativos. Mínimo precio unitario ofertado en España por los distribuidores RS Online, Farnell, DigiKey y Mouser (Enero 2013).

3.4.3. - Comparativa

En el caso de los microcontroladores PIC de Microchip, cabe destacar su amplia oferta de dispositivos y su precio, que en general, se encuentra por debajo del de sus competidores. Esta variedad nos permitiría ajustar los recursos del microcontrolador a las necesidades del nodo de manera más estricta. Por consiguiente, la compatibilidad pin a pin de muchos de sus microcontroladores nos permitiría cambiar el mismo sin alterar el diseño, lo que dotaría al nodo de gran versatilidad. En su contra juega un consumo superior a la media, posiblemente justificado al disponer de mayor rendimiento que otros microcontroladores funcionando a frecuencias de reloj menores.

Microchip además dispone de varios transceptores comerciales, módulos que incluyen también el circuito de adaptación de antena y la propia antena integrada, e incluso plataformas individuales que incorporar a tu diseño con un conector placa a placa. Los transceptores se comunican con el microcontrolador a través de SPI, aunque los protocolos propios de Microchip están preparados para funcionar eligiendo solo uno de ellos. En general tienen buenas características de consumo y monitorización del espectro, y los transceptores son bastante económicos (no tanto sus módulos o kits de desarrollo). Como inconvenientes, podemos resaltar que la pila ZigBee™ es de pago y sus protocolos propietarios no cumplen con el estándar 802.15.4 en las interfaces subGHz, por lo que el nodo sería menos interoperable.

La oferta de Atmel es también muy interesante, ya que tanto sus microcontroladores como sus transceptores destacan por sus excelentes prestaciones de bajo consumo. Los AVR de 32 bits tienen un precio notablemente superior al del resto de microcontroladores, siendo las demás características bastante similares a las de los PIC32 de Microchip, con matices en función de cada modelo. Los Atmel basados en ARM más eficientes energéticamente se quedan un poco cortos en cuanto a recursos y periféricos, y al contrario, los que disponen de mayor cantidad de memoria, capacidad de procesamiento y periféricos pierden (al menos parcialmente) sus buenas prestaciones de consumo.

El precio de sus transceptores radio está bastante más parejo al de otros fabricantes, pese a batirles claramente en aspectos como interoperabilidad (ZigBee, protocolos propietarios, IPv6...), tasa binaria y capacidad de monitorización del espectro. Sin embargo, una desventaja clara de Atmel de cara a nuestro diseño es que no plantean ningún transceptor de 434 MHz orientado a redes inalámbricas de área personal y bajo consumo (LR-WPAN). Otros fabricantes ofrecen transceptores versátiles que simplemente van equipados con distinta etapa final de radiofrecuencia (adaptación de antena y antena sintonizada en esa banda). El único transceptor comercial que disponen en 434 MHz está más orientado a control remoto y no se ha incluido en la comparativa por esta razón. Además, no disponen de módulos con antena integrada, por lo que el circuito de adaptación y la antena corren a cargo de la plataforma a la que se incorporan.

En cuanto a Texas Instruments, el microcontrolador que se ha incluido en la tabla tiene, en media, prestaciones muy buenas, pero sus productos de radio frecuencia están más orientados a soluciones integradas (system-on-chip) y dificulta su incorporación en un diseño con un único microcontrolador. Este hecho nos restringe mucho su gama de productos, pese a que algunos transceptores como el CC1101 son muy flexibles en su banda de trabajo y reúnen características buenas en la monitorización del espectro.

Al margen de las especificaciones técnicas de los dispositivos, también se ha tenido en cuenta la experiencia previa y el trabajo anterior desarrollado dentro del laboratorio, pues es un factor a tener en cuenta de cara al tiempo de desarrollo, y la calidad de la documentación y el soporte que ofrecen los fabricantes. En ese sentido, el siguiente gráfico muestra de manera cualitativa cuáles son los puntos fuertes y débiles de cada alternativa.

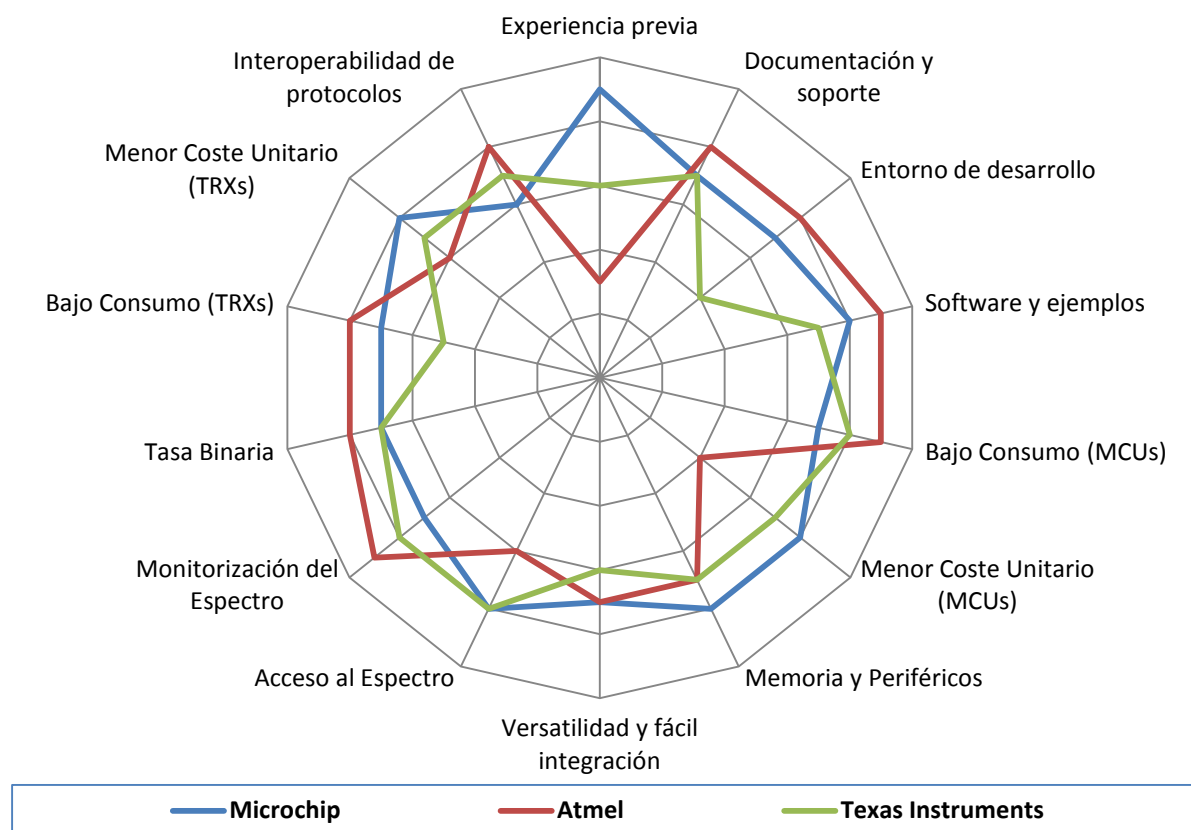


Figura 3: Comparativa global de fabricantes

4. FASE DE DISEÑO

A lo largo de este capítulo se tratan todos los temas referentes al diseño del dispositivo cognitivo de nueva generación, NGD.

Finalmente, se ha optado por mantener un microcontrolador PIC de 32 bits e incorporar un transceptor de Microchip en las bandas de 434 MHz, 868 MHz y 2,4 GHz. Los motivos que nos han llevado a tomar esta decisión son, principalmente, que con Microchip parece más viable, frente al resto de alternativas, lograr los siguientes aspectos:

- Integrar tres transceptores diferentes con un único microcontrolador y homogeneizar, en la medida de los posible, los protocolos que utiliza cada interfaz radio.
- Implementar interfaces radio orientadas a nuestros requisitos de aplicación: moderada capacidad de procesamiento, moderado ancho de banda, bajo consumo y bajo coste.
- Incorporar al menos una interfaz compatible con el estándar 802.15.4 para redes inalámbricas de área personal con requisitos de baja tasa de datos.

La siguiente figura presenta el diseño del NGD y las mejoras futuras que se plantean, así como resaltar el contraste con la equivalente figura del prototipo FCD incluida en la fase de análisis.

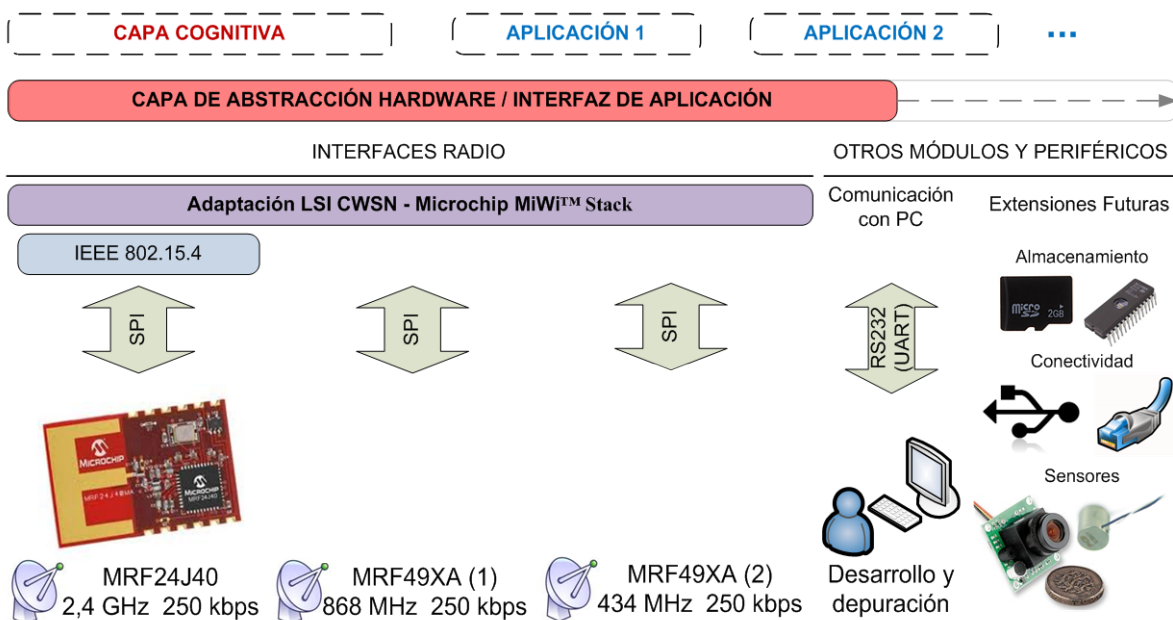


Figura 4: Protocolos, interfaces y periféricos del nodo NGD

4.1. DISEÑO HARDWARE

4.1.1. Microcontrolador

Para el nuevo diseño, el PIC32MX675F256L es un microcontrolador adecuado, ya que pese a que dispone de menos memoria de programa (flash) y memoria RAM, sigue siendo suficiente para albergar la pila de protocolos y futuras extensiones, reduciendo el coste y el consumo general del nodo. Además, al pasar de un encapsulado de 64 a 100 pines, permite conectar mayor cantidad de periféricos y módulos.

No obstante, como se desea que el nodo sea lo más flexible posible, se va a diseñar el firmware para que pueda testearse también sobre el nodo original con ligeras modificaciones y portarse de manera casi inmediata a la nueva plataforma hardware (y por extensión, a los microcontroladores compatibles pin a pin con ambos diseños). En las siguientes secciones vamos a profundizar un poco en los microcontroladores que atañen tanto al primer dispositivo cognitivo como al nuevo diseño del nodo NGD.

4.1.1.1. Arquitectura

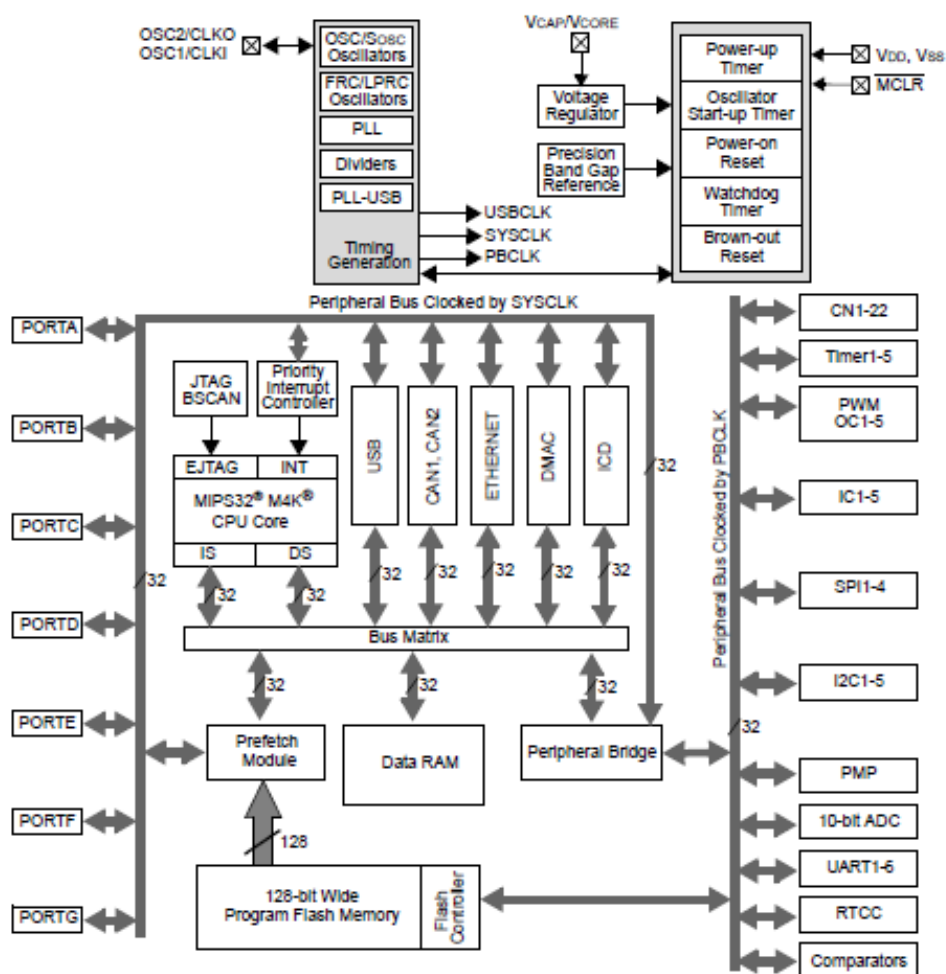


Figura 5: Arquitectura interna de la familia PIC32MX

La figura muestra un diagrama de la arquitectura interna de los PIC32MX pertenecientes a las sub-categorías 5XX/6XX/7XX. En ella podemos ver los diferentes módulos de los que se compone el microcontrolador y el flujo de datos entre ellos. Podemos estructurar todos los elementos de la arquitectura en cuatro niveles:

En primer lugar, como es lógico, el núcleo M4K (CPU) de MIPS Technologies del que ya se han comentado sus principales características. Es el encargado del procesamiento y control del hilo de ejecución, decodificando las instrucciones y gestionando el comportamiento del resto de módulos. Es también el único módulo que puede acceder a los registros internos de la CPU.

En un segundo nivel, podemos englobar al conjunto de módulos que, junto con la CPU, pueden gobernar el bus de sistema (Bus Matrix en la figura), que son: el módulo de Acceso Directo a Memoria (DMA por sus siglas en inglés), el controlador de la interfaz de depuración (ICD, In-Circuit Debugger) y el módulo USB. Este elemento de la arquitectura funciona como un conmutador de alta velocidad, siendo capaz de realizar transacciones a la misma velocidad que el reloj de la CPU (reloj de sistema). Los módulos que pueden gobernar el bus funcionan también a esta frecuencia y pueden iniciar transacciones con independencia del núcleo, e incluso puede haber concurrencia en los accesos. Por ejemplo, el módulo DMA puede iniciar una petición de lectura de la SRAM mientras la CPU accede a la caché para decodificar nuevas instrucciones.

A continuación encontramos el resto de periféricos que funcionan con el reloj del sistema pero no pueden gobernar el bus. En este grupo se encuentran tanto la memoria SRAM como la caché de la Flash, las interrupciones y los puertos de entrada y salida. El acceso a los periféricos que funcionan con la frecuencia de reloj del sistema típicamente se completa en un único ciclo de reloj. Algunas interfaces, como Ethernet o CAN, que pueden requerir un caudal de datos importante, se conectan al bus del sistema por motivos de velocidad.

Por último, a través del puente al bus de periféricos se accede a la mayoría de los periféricos. El bus de periféricos suele ir gobernado por un reloj más lento, fruto de dividir el reloj del sistema, dado que los periféricos conectados a él no necesitan tasas de transferencia elevadas. Las interfaces UART, SPI, e I2C, los canales del conversor analógico-digital (ADC), comparadores, etc., son ejemplos de algunos de los módulos que emplean el bus de periféricos.

Multiplexación de pines

Los microcontroladores PIC de las subfamilias 5xx/6xx/7xx se comercializan en encapsulados de 64 y 100 pines, por lo que no siempre todos los periféricos están disponibles fuera del chip. Además, los pines se multiplexan entre varios periféricos, impidiendo que se puedan utilizar simultáneamente.

Afortunadamente, muchos modelos comparten el mismo patillaje o presentan sutiles diferencias entre sí. Por ejemplo, hay modelos que no incorporan interfaces CAN y

simplemente los pines que emplea CAN en otros microcontroladores no tienen esa multiplexación disponible, pero el resto de pines son perfectamente compatibles con el modelo que sí las incorporan. Esto hace que, salvo excepciones, los periféricos estén multiplexados siempre de la misma forma en cada uno de los encapsulados. Las siguientes figuras muestran esa situación para encapsulados de 64 pines y de 100 pines.

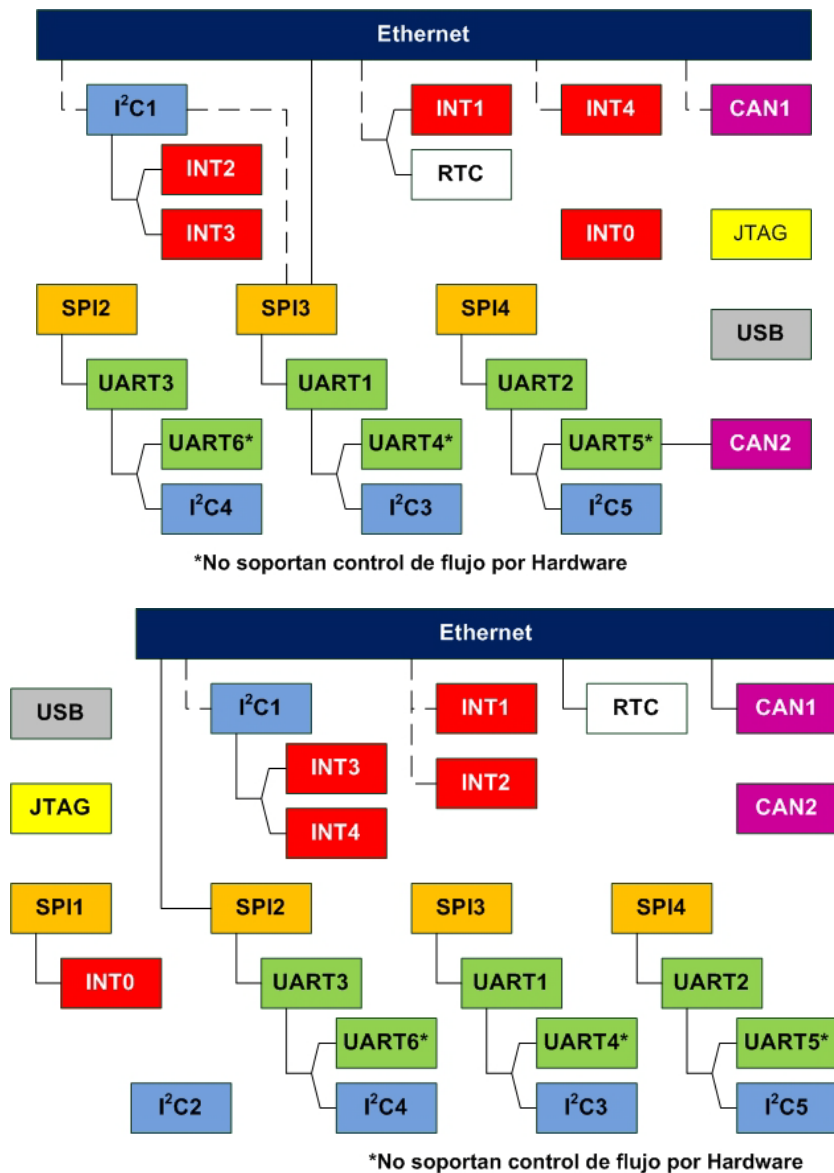


Figura 6 y Figura 7: Compatibilidad de módulos en PIC32MX 5xx/6xx/7xx para encapsulados de 64 pines (arriba) y 100 pines (abajo)

En los diagramas se puede observar que las interfaces SPI (que usan 4 líneas para la comunicación) suelen multiplexarse con las UART (2, o 4 líneas si emplean las funciones de control de flujo por hardware) y con las interfaces I2C (2 líneas). Esto quiere decir que, por ejemplo, en caso de emplear la SPI2 no podremos utilizar la UART3, y que, en caso de dejar libres ambos recursos, podremos disponer de las interfaces UART6 e I2C4.

Las UART 4, 5 y 6 merecen mención especial, ya que no soportan control de flujo por hardware y son compatibles con las UART 1, 2 y 3 respectivamente si éstas tampoco emplean el control de flujo por hardware.

La interfaz Ethernet supone la utilización de 10 pines para su implementación con funcionalidades reducidas, o 18 pines para disponer de todas las funcionalidades que ofrece el controlador. Su uso tiene incompatibilidades potenciales con muchos periféricos. Otros módulos como USB o JTAG, en cambio, suelen utilizar pines multiplexados exclusivamente con entradas de propósito general y no tienen conflicto con otros periféricos.

Aunque una de las máximas de diseño es que el nodo sea lo más flexible y configurable posible, es importante tener estas restricciones en mente cuando se vaya a desarrollar el prototipo hardware en el futuro.

4.1.1.2. Modos de bajo consumo

Para reducir el consumo energético, todos los microcontroladores modernos implementan modos de bajo consumo en los que se desactivan módulos y se reducen las prestaciones generales o incluso se detiene la CPU y la mayoría de los periféricos.

Generación de reloj

Para poder comprender los modos de bajo consumo, primeramente se explica el módulo de temporización interno de los microcontroladores PIC que nos atañen, que consta de:

- Dos osciladores internos de tipo RC: un oscilador de 8 MHz nominales (denominado FRC – Fast internal RC oscillator), que tiene cierta capacidad de ajuste mediante software; y un oscilador de 32 kHz (o LPRC - Low Power internal RC oscillator), que es el reloj utilizado por el módulo WatchDog.
- Entradas para conectar dos osciladores externos que actúen como osciladores principal y secundario, con ligeras diferencias de funcionamiento. El principal está pensado para generar el reloj del sistema (SYSCLK), mientras que el secundario típicamente es de 32.768 kHz, orientado a servir como reloj de tiempo real. Además, el reloj secundario puede emplearse con el único temporizador que funciona en modo "sleep".
- Módulos de síntesis de frecuencia basados en PLL para, opcionalmente, generar frecuencias de reloj de sistema más altas y un reloj de 48 MHz para el módulo USB.
- Un divisor de frecuencia seleccionable para el reloj del bus de periféricos (PBCLK)

El reloj de sistema puede generarse con cualquiera de las dos fuentes externas, cualquiera de las dos fuentes internas, mediante cualquier fuente como referencia para la síntesis con PLL (excepto LPRC) o como un submúltiplo de cualquiera de las fuentes

(excepto LPRC) gracias a un divisor seleccionable. El reloj del bus de periféricos se genera a partir del reloj del sistema con el divisor que se configure.

Modos de bajo consumo:

La familia PIC32MX implementa varios modos de bajo consumo seleccionables por software que se clasifican en 2 categorías: modos con CPU detenida (no decodifican nuevas instrucciones), y modos con CPU operativa. Todos los modos asumen como condiciones normales que el reloj del sistema se obtiene mediante el oscilador externo principal (con o sin síntesis mediante PLL) y la CPU operativa. Este modo de ejecución se suele denominar simplemente *"Run Mode"*.

Los modos con CPU operativa difieren en qué oscilador se escoge como fuente para generar el reloj de sistema:

- Oscilador FRC (*"FRC Run Mode"*): El reloj del sistema pasa a obtenerse mediante el oscilador interno FRC. Es decir, la frecuencia de reloj pasa a ser 8 MHz (sin aplicar divisores) o hasta 500 kHz (dividiendo la frecuencia en un factor máximo de 16). Este modo puede reducir considerablemente el consumo manteniendo buenas prestaciones.
- Oscilador LPRC (*"LPRC Run Mode"*): Es el modo análogo al anterior, empleando el otro oscilador interno (LPRC), con la diferencia de no poder escalar la frecuencia. El reloj del sistema sería de tan sólo 32 kHz y es un modo que no tiene mucho sentido utilizar salvo que los periféricos sean muy lentos.
- Oscilador Secundario (*"SOSC Run Mode"*): El oscilador secundario externo pasa a ser el reloj del sistema. Tiene el mismo problema que el modo anterior: una frecuencia de reloj muy baja para mantener todos los periféricos operativos no siempre es factible. Si no se dispone de este oscilador en el diseño, obviamente este modo no se puede emplear.

Además de cambiar la fuente de reloj del sistema para reducir la frecuencia de reloj, otra técnica que ofrece el PIC para reducir el consumo es ralentizar el reloj del bus de periféricos (PBCLK). Simplemente se trata de cambiar el divisor de frecuencia del reloj del bus de periféricos (PBDIV) a otro de mayor valor (en potencias de dos, desde 1 hasta un máximo de 8). Si se combina con algún modo anterior, se pueden mantener los periféricos a menor frecuencia sin perder prestaciones en la CPU, pero no es tan efectivo como cambiar a un modo con menor frecuencia o a un modo con CPU detenida.

Los modos con CPU detenida suponen mayor ahorro energético y podemos dividirlos en modos con reloj de sistema habilitado (denominados *"Idle"*) o deshabilitado (modo *"Sleep"*). De nuevo, los modos Idle difieren en la fuente que genera el reloj del sistema:

- Oscilador principal (*"POSC Idle Mode"*): El reloj de sistema sigue siendo el oscilador externo principal, pero la CPU se detiene.

- Oscilador FRC ("*FRC Idle Mode*"): El reloj del sistema pasa a ser el oscilador FRC interno o un submúltiplo si se emplea algún divisor.
- Oscilador secundario ("*SOSC Idle Mode*"): Únicamente si se dispone de oscilador secundario, éste puede emplearse como reloj de sistema.
- Oscilador LPRC ("*LPRC Idle Mode*"): El oscilador LPRC es ahora el reloj del sistema. Éste es el modo de menor consumo con algún reloj funcionando.

En general, la gran ventaja de los modos Idle es que permiten a los periféricos seguir funcionando al margen de la CPU, pero ésta solo se puede aprovechar cuando hay comunicación entre periféricos o acceso directo a memoria. Los periféricos pueden seguir funcionando o desactivarlos antes de entrar al modo Idle. Para nuestra aplicación no aportan demasiado, ya que los transceptores radio no tienen mucha capacidad de almacenar información antes de que la procese el microcontrolador y además, gran parte de su actividad requiere procesamiento por parte del microcontrolador.

- Modo dormido ("*Sleep Mode*"): Este es el modo de menor consumo de todos ya que detiene además el reloj del sistema y, en consecuencia, el reloj del bus de periféricos y los periféricos que funcionan a través de él. El microcontrolador deshabilita la fuente utilizada como reloj principal (salvo casos especiales). Sólo algunos periféricos y módulos pueden seguir funcionando, entre ellos el módulo WatchDog, el *Brown-Out Reset* (fluctuación/fallo en la alimentación), el reloj de tiempo real (RTC), interrupciones generadas por entradas sensibles a notificación de cambios y otros periféricos con fuentes de reloj externas. Los pines siguen drenando o inyectando corriente de la misma manera que lo hacían antes de pasar a modo dormido, por lo que conviene apagar los módulos antes de dormir el nodo para reducir más aún el consumo.

El microcontrolador puede despertar de los modos Idle y Sleep para volver al modo anterior de ejecución en diversas situaciones. Su comportamiento puede tener matices importantes en función del modo en el que se den dichas situaciones, y diferentes latencias en función del modo a restaurar. Los sucesos más relevantes que despiertan el microcontrolador son los eventos de reset (*Brown-Out*, *Master Reset*...), el desbordamiento del temporizador WatchDog, o rutinas de interrupción habilitadas durante los modos de bajo consumo.

El módulo WatchDogTimer

El temporizador WatchDog (WDT) es un contador que, en condiciones normales, si desborda provoca un evento de reset (WDT Reset). Se suele emplear para detectar comportamientos no deseados, reseteando la cuenta cuando todo va bien. El reloj que emplea es el LPRC, al que se le pueden aplicar divisores para fijar un periodo desde 1 ms hasta aproximadamente 17 minutos.

Sin embargo, cuando se pasa a un modo de CPU dormida, el contador sigue operativo y su desbordamiento provoca una interrupción no enmascarable que despierta al microcontrolador. Es una opción interesante para aplicaciones de redes de sensores

concretas que midan, envíen la medida y duerman hasta tener que realizar una nueva medida.

Temporizadores durante el modo sleep.

La mayor parte de periféricos utilizan el reloj del bus de periféricos y dependen de él para su funcionamiento. No obstante, como ocurre con el WDT, hay algunos casos especiales que hacen que determinados periféricos sigan operativos cuando el reloj de sistema está deshabilitado).

El Timer1 es el único temporizador del microcontrolador que puede funcionar en modos con CPU detenida si se configura adecuadamente. Básicamente, se trata de conmutar la fuente del reloj para que en vez de obtener el reloj del bus de periféricos, utilice el oscilador secundario. Este oscilador, recordemos, también es una entrada del módulo de reloj de tiempo real.

4.1.2. Interfaces radio

Un cambio muy relevante frente al nodo FCD es la eliminación del transceptor Wi-Fi (IEEE 802.11) de Microchip en el nuevo diseño. Tanto el módulo Wi-Fi de Microchip como los que ofrece Atmel para incorporar a sus microcontroladores (provistos por fabricantes colaboradores) tienen un consumo muy significativo para WSN. No obstante, se contempla que en futuras extensiones al diseño se pueda incorporar un transceptor de Wi-Fi (por ejemplo, vía SPI), o emplear una interfaz Ethernet en su defecto, en aquellos nodos que se alimenten a través de la red eléctrica.

Tampoco se mantiene el módulo Tulio, ya que incorpora un microcontrolador de Texas basado en el Intel 8051 y la comunicación y sincronización con el microcontrolador central añadía complejidad al diseño sin redundar en ningún beneficio. En su lugar, emplearemos el MRF49XA de Microchip.

El módulo MRF24J40MA de Microchip a 2,4 GHz es la única interfaz radio del prototipo demostrador que perdura en este rediseño, ya que se ha considerado que tiene un precio y unas prestaciones razonables, simplifica el diseño al ser de reducido tamaño y contar con antena integrada en PCB, y es compatible con el estándar 802.15.4 del IEEE.

El nuevo NGD dispondrá de hasta tres interfaces radio para operar en las bandas de 434 MHz, 868 MHz y 2,4 GHz. Para las interfaces de 868 y 434 MHz se emplearán dos transceptores MRF49XA, cada uno de ellos sintonizado en su banda de trabajo, dado que su precio y su consumo energético son bastante competitivos. Como Microchip no comercializa ningún módulo con antena integrada para este transceptor, y con el fin de reducir el tiempo de desarrollo, para desarrollar el firmware utilizaremos dos MRF49XA PICtail™/PICtail Plus Daughter Boards. Se trata de una solución comercial de Microchip para conectar al diseño una placa individual en la que consta el transceptor y la antena.

4.1.2.1. Transceptores

4.1.2.1.1. Módulo MRF24J40MA

El MRF24J40MA es uno de los módulos de radiofrecuencia que ofrece Microchip en la banda de 2,4 GHz con el transceptor MRF24J40 y la antena integrada en PCB. Su patrón de radiación es bastante omnidireccional y su potencia de salida puede alcanzar 1 mW (0 dBm). Otros módulos de esta gama de productos tienen antenas más directivas y con mayor ganancia, pero como contrapartida, la comunicación depende fuertemente de la orientación de la antena y la situación espacial del transmisor y el receptor.

El transceptor MRF24J40 es compatible con el estándar IEEE 802.15.4, cuyo propósito es ofrecer los fundamentos de las capas inferiores al nivel de red en redes inalámbricas de área personal (WPAN) enfocadas a bajo coste, baja velocidad y comunicación ubicua entre dispositivos con las mínimas infraestructuras posibles. El transceptor está diseñado para soportar diferentes protocolos de red compatibles con el estándar. En concreto soporta ZigBee y los protocolos propietarios de Microchip, denominados MiWi (Microchip Wireless), que se detallarán más adelante.

El estándar define 16 canales en la banda de 2,4 GHz separados 5 MHz y el uso de técnicas de modulación de espectro ensanchado (SS)⁸ para alcanzar una tasa binaria de 250 kbps a nivel radio. El transceptor permite alcanzar una tasa binaria de 625 kbps en lo que Microchip denomina como "*Turbo Mode*", pero activar esta funcionalidad supone no cumplir con el estándar.

El módulo es un componente de montaje superficial con un tamaño muy reducido (menos de 5 cm² incluyendo la antena), incluye un regulador de tensión, un VCO y un PLL interno para sintetizar las frecuencias necesarias a partir de un cristal de 20 MHz y el circuito de adaptación de antena. Es compatible con casi todas las familias de microcontroladores PIC y facilita su integración con estos microcontroladores mediante una interfaz SPI.

A modo resumen, se listan sus principales características:

- Tensión de alimentación: 2.4 - 3.6 V (3.3 V típico).
- Rango de temperaturas de operación: desde -40 hasta +85 °C.
- Margen dinámico RF: potencia de entrada entre -94 dBm (sensibilidad) y +5 dBm.
- Consumo: 23 mA transmisión (0 dBm); 19 mA recepción; 2 µA modo dormido.

⁸ Son técnicas que emplean más ancho de banda que el mínimo requerido para transmitir la información a determinada velocidad y, por tanto, ineficientes en ese sentido, pero con la contrapartida de poder coexistir con sistemas de banda estrecha trabajando en la misma banda de frecuencia, ya que para éstos solo suponen un pequeño incremento en el nivel de ruido. El receptor de espectro ensanchado no recibe las señales de banda estrecha al emplear un ancho de banda mucho más amplio en función de una secuencia de código preestablecido (conocido a priori por el transmisor y el receptor).

- Alcance: alrededor de 120 m máximo.
- Frecuencia de trabajo y ancho de banda: 2.405 - 2.480 GHz (ISM), 250 kbps.
- Funcionalidades: retransmisión automática de paquetes, opciones de encriptación en la capa MAC, Hardware Security Engine (AES-128) con diferentes modos, implementa CSMA-CA (acceso al medio mediante detección de portadora y elusión de colisión) a nivel hardware y modos CCA con indicadores RSSI/LQI.

4.1.2.1.2. MRF49XA PICTail Daughter Board

Para las interfaces de 434 y 868 MHz se ha utilizado una placa comercial de Microchip que contiene el transceptor MRF49XA y, además, los siguientes elementos:

- Un cristal de 10 MHz para sintetizar las frecuencias necesarias para la transmisión/recepción.
- El circuito de adaptación de antena y una antena tipo látigo para la banda de trabajo.
- Una memoria EEPROM (Electrically Erasable Programmable Read-Only Memory) que contiene una dirección de red para el dispositivo.
- Un par de pines para medir el consumo del módulo.
- Un conector placa a placa para incorporarlo en el diseño anfitrión.

El transceptor MRF49XA puede trabajar en las bandas de 434, 868 o 915 MHz con la antena y el circuito de adaptación adecuados, pero la banda de 915 MHz es una banda licenciada en Europa. Microchip solo ofrece la pila de sus propios protocolos de RF (MiWi) para utilizar estos transceptores. El transceptor utiliza una interfaz SPI para comunicarse con el microcontrolador.

Se trata de un transceptor que no es compatible con IEEE 802.15.4 debido a que el número de canales radio se configura a partir de la tasa binaria que se quiere emplear en ellos, mientras que el estándar define, en su versión de 2003, un canal en la banda de 868 MHz y diez en la de 915 MHz. En la revisión de 2006, el número de canales se ampliaron a tres y treinta respectivamente, pero la banda de 434 MHz siguió quedando fuera del estándar. No obstante, en la actualidad los grupos de trabajo del IEEE se plantean añadir nuevos canales en la banda de 315, 434 MHz, bandas subGHz disponibles en China y Japón, y nuevas técnicas de modulación.

Al igual que para el módulo de 2,4 GHz, se incluye un resumen de su hoja de características:

- Tensión de alimentación: 2.2 - 3.8 V (3.3 V típico).
- Rango de temperaturas de operación: desde -40 hasta +85 °C.
- Margen dinámico RF: potencia de entrada entre -110 dBm (máx. sensibilidad) y +5 dBm.
- Consumo: 24 mA transmisión (7 dBm); 12 mA recepción; 0.3 µA modo dormido.
- Alcance: varias centenas de metros.

- Frecuencia de trabajo y ancho de banda: 440 - 460 MHz / 860 - 880 MHz; 1,2 - 256 kbps
- Funcionalidades: Indicador RSSI, indicador DQI, detección de batería baja.

4.1.2.2. Protocolos de red.

En las siguientes páginas se incluye una breve introducción a los protocolos de red que ofrece Microchip con sus transceptores radio. Por ahora ha desarrollado tres protocolos con características diferentes, aunque todos agrupados bajo el nombre comercial MiWi™ (Microchip Wireless) y una torre de protocolos con la arquitectura común que muestra la figura.

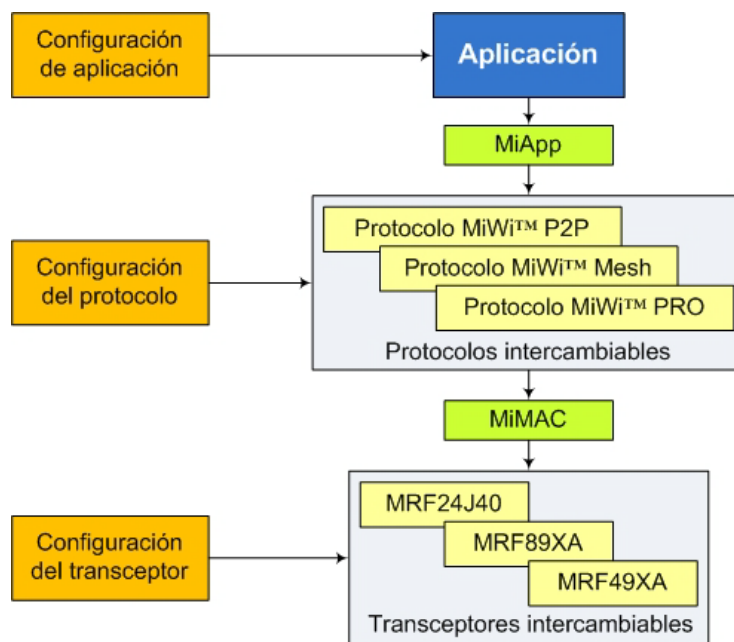


Figura 8 Esquema general de la pila de MiWi™

Como se observa en el esquema, los diferentes protocolos pueden manejar indistintamente cualquier transceptor de Microchip a través de la interfaz "MiMAC" (Microchip Media Access Control) y hacia el nivel superior ofrecen una interfaz de aplicación común "MiApp" (Microchip Application Programming Interface). La implementación de la capa MAC y la capa de nivel físico (PHY) toman como referencia el estándar 802.15.4, si bien algunos de sus transceptores no son compatibles con él.

Los protocolos y transceptores ofrecen cierta flexibilidad en función de los requisitos de la aplicación. El único impedimento es que la torre de protocolos solo permite emplear un único transceptor radio, por lo que requiere adaptar toda la pila para poder gestionar todas las interfaces desde el nivel de aplicación.

Los protocolos implementados por Microchip hasta el momento son:

- MiWi™ P2P (Peer-to-Peer): El protocolo P2P (punto a punto) es una simplificación del estándar 802.15.4 del IEEE. Soporta las topologías de estrella o

punto a punto y no implementa funciones de rutado, por lo que la comunicación se restringe al rango de cobertura del dispositivo.

- **MiWi™ Mesh:** Es un protocolo algo más complejo que el anterior orientado a redes tipo malla ("mesh" en inglés) o con agrupaciones de nodos anidados en árbol ("Cluster Tree"). Por comodidad, se suele denominar simplemente "MiWi". Las principales diferencias con el anterior son, por un lado, que implementa funciones de encaminamiento, y además, que soporta redes de mayor tamaño.
- **MiWi™ PRO:** este protocolo extiende las funcionalidades del protocolo MiWi, básicamente para poder disponer de más nodos capaces de encaminar datos a través de la red o disponer de una red de mayor tamaño.

Dado que las aportaciones de este último protocolo no resultan interesantes a priori para nuestro diseño, nos centraremos únicamente en los dos primeros protocolos.

4.1.2.2.1. Protocolo P2P

El protocolo P2P ofrece servicios de comunicación fiable entre dos dispositivos con funcionalidades de red muy sencillas, basadas en una simplificación del estándar IEEE 802.15.4. que ya se ha ido introduciendo a lo largo del documento. Las principales diferencias radican en el proceso de establecimiento del canal de comunicación y sus parámetros (en adelante, usaremos el término inglés "handshaking"), la liberación de conexiones y los saltos automáticos de canal, que son más sencillos.

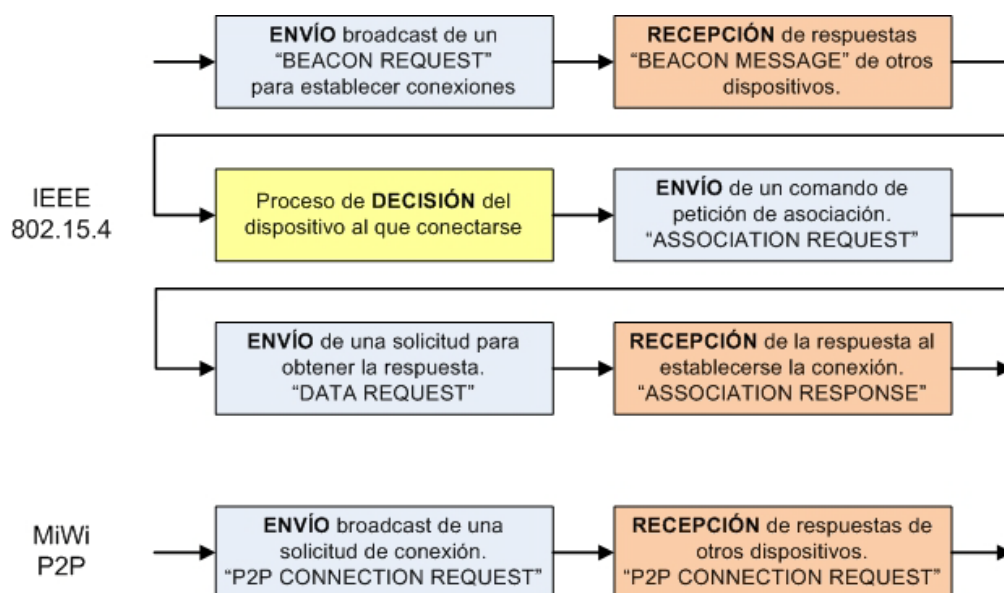


Figura 9: Simplificación del handshaking en el protocolo MiWi P2P™

El espacio en memoria que requiere el protocolo P2P puede ser de tan sólo 4 kB, con las opciones del protocolo más sencillas. Si se habilitan funcionalidades como mensajes indirectos o cifrado y seguridad, los requisitos aumentan. Aun así, la huella en memoria es bastante pequeña considerando un microcontrolador con al menos 256 kB de memoria flash.

Tipos de dispositivos

No todos los nodos cumplen las mismas funciones o ostentan las mismas capacidades dentro de la red. En ese sentido, existen dos tipos de dispositivos: dispositivos con funcionalidad completa (FFD, Full Function Device), y dispositivos con funcionalidad reducida (RFD, Reduced Function Device). Estas definiciones son coherentes con el estándar IEEE 802.15.4 e idénticas a las de otros protocolos que implementan el estándar, como por ejemplo ZigBee.

Los RFD son nodos que pueden dormirse y típicamente se alimentan a través de baterías, mientras que los FFD permanecen a la escucha siempre y requieren estar conectados a la red eléctrica. Los primeros, por tanto, pueden no recibir paquetes radio al estar en modo dormido y, al despertar, suelen sondear a los dispositivos FFD asociados para comprobar si han almacenado datos para ellos.

Por sus roles, podemos clasificar los nodos en dos categorías: coordinador de red de área personal (PANC, Personal Area Network Coordinator) o dispositivos finales (ED, End Device). El PANC es el nodo encargado de iniciar la red, quedando a la espera de nuevas conexiones, mientras que los dispositivos finales simplemente se conectan a la red y establecen las conexiones.

Rol en la red	Tipo de dispositivo	Descripción
Coordinador de red (PANC)	FFD	Un único PANC por red. Permanece siempre encendido a la espera de conexiones.
Dispositivos finales (ED)	FFD o RFD	Arrancan tras el PANC para establecer conexiones. Los RFD pueden dormirse y se conectan exclusivamente con el FFD del que primero reciben respuesta.

Tabla 4: Tipos y roles de dispositivos en el protocolo MiWi P2P™

Topologías de red

El protocolo P2P soporta las siguientes topologías:

- Topología en estrella: se trata de la configuración más básica. El coordinador (PANC) de la red es el núcleo de la red y el resto de dispositivos establecen una única conexión con él. Es un escenario idóneo para redes pequeñas con nodos próximos en torno al coordinador.
- Topología P2P: difiere de la anterior en que los dispositivos con funcionalidad completa (FFD) pueden establecer más de una conexión, sin que necesariamente se establezca conexión con el PANC. Los RFD siguen limitados a una única conexión con un FFD, pero ahora este nodo no tiene por qué ser el coordinador de la red.

Los diagramas muestran las mencionadas topologías de red en estrella y P2P.

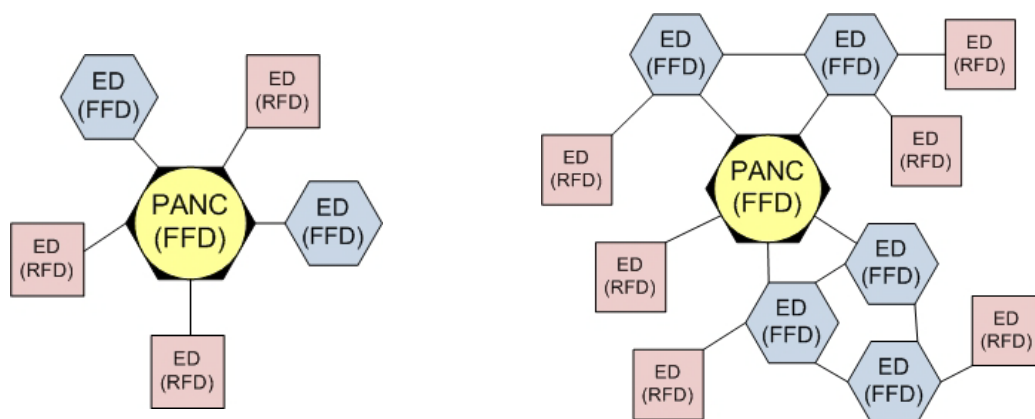


Figura 10: Topologías de estrella (izq.) y P2P (dcha.)

Direccionamiento en la red

El estándar IEEE 802.15.4 define dos tipos de direcciones:

- EUI (Extended Organizationally Unique Identifier): es una dirección de 8 bytes única a nivel mundial asignada al dispositivo. Los tres bytes más significativos los asigna el IEEE, mientras que los 5 restantes los asigna el fabricante a cada dispositivo que soporta el estándar. En ocasiones nos referiremos a esta dirección como dirección larga o dirección permanente, para distinguirla del siguiente tipo.
- "Short Address" (Dirección corta o dirección de red): esta dirección se asigna dinámicamente a los dispositivos a medida que se unen a la red existente. Las direcciones cortas también son únicas, pero solo dentro de la misma. Generalmente, la dirección de red asignada depende del orden de llegada a la red y la vecindad con otros nodos.

En el protocolo P2P sólo se puede emplear la dirección larga y las comunicaciones están restringidas a un único salto, es decir, un nodo solo puede comunicarse con otros con los que tiene visibilidad directa. Este hecho es consecuencia directa de no implementar funciones de encaminamiento, pues son las encargadas de determinar los saltos por la red. El uso de la dirección corta queda restringido para broadcast, ya que no existe dirección larga reservada para tal fin.

Microchip ofrece la posibilidad de reducir la extensión de las direcciones largas (hasta un mínimo de 2 bytes), pero en caso de utilizar una longitud distinta de 8 bytes se abandona la compatibilidad con el estándar y por consiguiente, la denominación de EUI. En el caso de utilizar direcciones más cortas la numeración y asignación de direcciones corre por cuenta propia.

Búsqueda de redes y escaneo de canales.

Dado que en una banda de frecuencia puede haber varias redes de área personal coexistiendo pero cada una de ellas debe operar en un único canal, el protocolo dispone de una capacidad para descubrir qué redes operan en su entorno local, qué canal emplean y cuál es su identificador de red de área personal (PANID) en el caso de redes

basadas en 802.15.4. La información se infiere de los mensajes recibidos de los miembros de la red al contestar a una petición broadcast realizada por el nodo que realiza la búsqueda. A esta característica se le denomina escaneo activo (en inglés "Active Scan").

Además, el protocolo P2P puede realizar un escaneo de energía ("Energy Scan") para evaluar qué canal presenta menor nivel de ruido. Esto es interesante de cara a la capa cognitiva, ya que estos datos aportan parámetros de decisión para optimizar las prestaciones en diversos escenarios, pero también resulta útil cuando, simplemente, el PANC quiere iniciar la red en las mejores condiciones que ofrezca el ambiente en el que opera la red.

4.1.2.2.2. Protocolo MiWi

En comparación con P2P, MiWi Mesh (por comodidad, simplemente MiWi) es un protocolo orientado a redes de mayor tamaño, ya que soporta que la comunicación se realice de manera directa o a través de varios saltos por la red. Es un protocolo más complejo, que incorpora funciones de encaminamiento y admite hasta un máximo de 1024 nodos.

Sus requerimientos en cuanto a memoria de programa varían en función de las opciones del protocolo habilitadas, el tamaño de la tabla de conexiones, el rol en la red, etc. Con 32 kB es más que suficiente para que el nodo con mayores requisitos pueda albergar el protocolo, sus funcionalidades y una aplicación de usuario sencilla.

Tipos de dispositivos

De nuevo, podemos dividir los dispositivos entre aquellos que tienen funcionalidad completa (FFD) y los de funcionalidad reducida (RFD) con las mismas características que se han comentado para P2P. Sin embargo, el protocolo MiWi (MiWi Mesh) incorpora un nuevo rol en la red: coordinador (similar a lo que en ZigBee se denomina "Router"). Los coordinadores, junto con el coordinador general de la red (PANC), son los encargados de llevar a cabo las funciones de encaminamiento.

Rol en la red	Tipo de dispositivo	Descripción
Coordinador de red (PANC)	FFD	Un único PANC por red. Permanece siempre despierto a la espera de conexiones y asigna direcciones de red.
Coordinador	FFD	Extiende la cobertura de la red y colabora en funciones de encaminamiento y gestión de la red. Permanece siempre despierto.
Dispositivos finales (ED)	FFD o RFD	Arrancan tras el PANC para establecer conexiones y sólo pueden dormirse si son RFD. Los RFD se conectan exclusivamente con el FFD del que primero reciben respuesta.

Tabla 5: Tipos y roles de dispositivos en la torre de protocolos de MiWi Mesh™

Topologías de red

Al disponer de funciones de rutado y soportar varios saltos en la red hasta alcanzar el nodo destino, MiWi permite otras topologías más complejas que P2P:

- Topología en árbol (Cluster Tree): en la que el PANC sería el tronco o centro de la red; los coordinadores, las diferentes ramificaciones; y los dispositivos finales, serían las hojas. Todos los paquetes radio que requieren más de un salto en la comunicación se encaminan a través de los coordinadores y el PANC.
- Topología tipo malla ("Mesh"): Se podría considerar una extensión de la topología P2P en la que otros FFD pueden enviar mensajes a otros FFD sin necesariamente pasar por los coordinadores o el PANC. Esta topología presenta ventajas en cuanto a latencia y fiabilidad.

Para comprender mejor ambas topologías, se incluye la siguiente figura con un ejemplo de la topología tipo árbol (izq.) y tipo malla (dcha.):

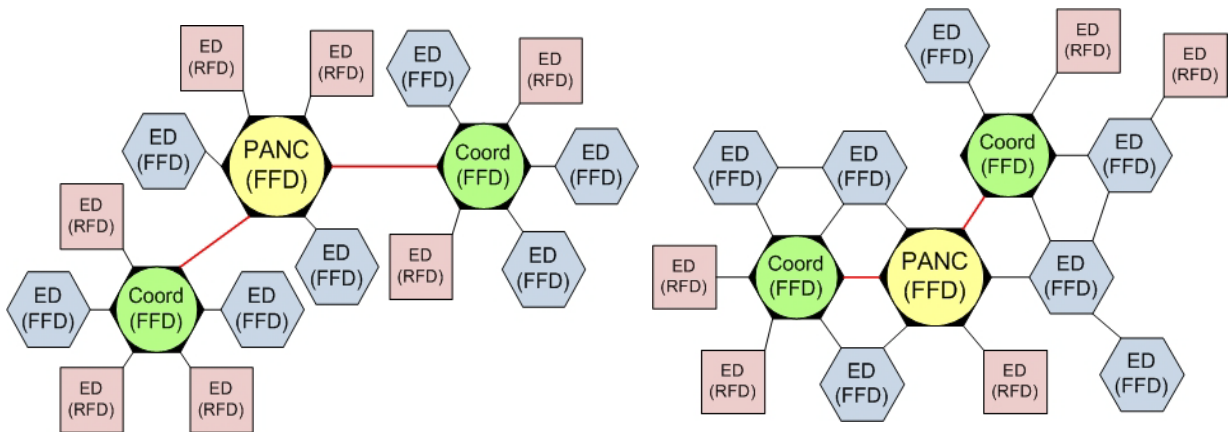


Figura 11: Topologías en árbol (izq.) y en malla (dcha.)

Direccionamiento en la red

Como en el caso de P2P, un nodo tiene asociadas dos direcciones: una dirección larga de entre 2 y 8 bytes que es fija (denominada EUI en el IEEE 802.15.4 y restringida a 8 bytes), y una dirección de red que se asigna al dispositivo al unirse a la red ya existente.

Una diferencia fundamental en la implementación del protocolo MiWi es el uso de la dirección corta o de red (2 bytes). La asignación se realiza por orden de llegada, con los siguientes criterios:

- El coordinador general de la red de área personal (PANC) siempre tiene la dirección 0. Él mismo se asigna esta dirección al crear la red.
- El resto de coordinadores (el protocolo soporta hasta 8 coordinadores incluyendo al PANC) tienen direcciones reservadas. El byte menos significativo de la dirección tiene el valor 0 por ser un coordinador. El byte más significativo toma

valores del 1 al 7 (pues el valor 0 corresponde al PANC) y es asignado por el PANC.

- Para los dispositivos finales, el byte más significativo coincide con el del coordinador que acojan como dispositivo asociado al conectarse a la red (pues pertenecen al "cluster" o subgrupo de ese coordinador). Del segundo byte, 1 bit se reserva para distinguir dispositivos que pueden dormirse de los que no, y los otros 7 bits, para numerar en orden creciente de llegada. La dirección de los dispositivos finales se la asigna su coordinador.

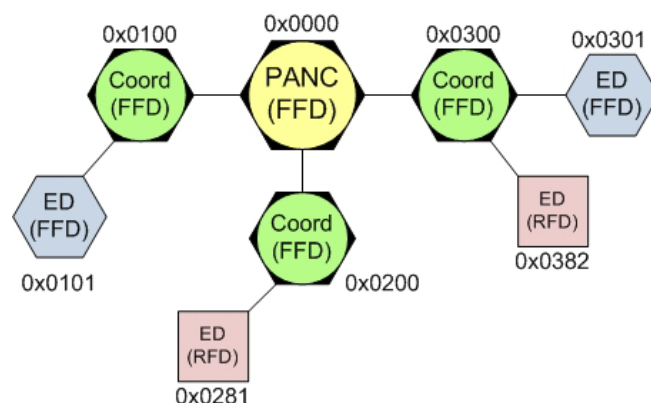


Figura 12: Asignación de direcciones en el protocolo MiWi Mesh™

Las funciones de encaminamiento se basan en estos criterios de asignación para poder hacer llegar el paquete radio a su destino. Este hecho también limita el número de nodos de la red a 1024 (hasta 8 coordinadores con 127 dispositivos finales asociados cada uno). El número de saltos permitido por defecto es cuatro.

Búsqueda de redes y escaneo de canales.

Las funcionalidades de búsqueda de redes y escaneo de canales son las mismas que para el protocolo P2P. Todos los miembros de una red comparten el mismo identificador de red de área personal (PANID).

4.2. DISEÑO SOFTWARE

La pila de protocolos que ofrece de manera gratuita Microchip está preparada para incorporar cualquiera de sus transceptores, pero no los tres simultáneamente. Como se pretende que el firmware sea lo más eficiente posible, el diseño del mismo debe adaptar el software facilitado por Microchip para que la torre de protocolos se comparta por los tres transceptores, en lugar de simplemente instanciar la pila por triplicado, una vez por cada interfaz radio.

Para abstraer de la complejidad que suponga esta adaptación y compartición de recursos y protocolos, se va a diseñar una capa de abstracción hardware que sirva como interfaz de aplicación para utilizar todas las funcionalidades radio. Este hecho, junto con las opciones de desarrollo y depuración, pretende simplificar el desarrollo de aplicaciones y

módulos software como, por ejemplo, la capa cognitiva que evalúe y optimice los parámetros del nodo y otras variables a nivel global de red.

4.2.1. Diseño de la adaptación de la torre de protocolos MiWi

4.2.1.1. Decisiones de diseño

Para la adaptación se han tomado algunas decisiones de diseño importantes relativas a la configuración del nodo que afectan a todas las interfaces MiWi presentes en él. Por su relevancia se comentan brevemente a continuación.

Configuración compartida de roles en la red.

Para la implementación del firmware del NGD se ha decidido que el tipo de dispositivo y su rol en la red sean comunes a todas las interfaces. Es decir, si es un dispositivo RFD (en ocasiones también denominados Sleeping Devices), podrá dormir todas y cada una de sus interfaces radio. Si es un dispositivo de tipo FFD, cabe la posibilidad de que sea un dispositivo final (End Device) o un coordinador, pero si es un coordinador, lo es para todas las interfaces.

La razón de ser de esta decisión es que si un nodo es coordinador, debe permanecer siempre a la escucha y requerirá alimentación ininterrumpida de la red eléctrica. Por tanto, merece la pena aprovechar que el nodo está conectado a la red eléctrica para que sea coordinador de todas sus interfaces y, así, tratar de minimizar el número de nodos que requieren alimentación de la red.

Direccionamiento en la red.

Tal y como se ha comentado en las páginas anteriores, se manejan dos tipos de direcciones en los protocolos de Microchip: una larga, que es única y fija, y otra corta, que se le asigna dinámicamente a un dispositivo al unirse a la red. Al disponer de más de una interfaz radio, surge la duda de cómo adaptar los protocolos para trabajar con las tres interfaces simultáneamente.

Dado que la dirección larga pretende ser única, se ha decidido que actúe como la dirección única de todo el dispositivo y se comparta en las tres interfaces MiWi. En el caso de incluir la interfaz de 2,4 GHz, compatible con el estándar IEEE 802.15.4, se recalca que las direcciones forzosamente son de 8 bytes y su asignación depende del fabricante y del IEEE.

En cambio, las direcciones cortas o direcciones de red dependen de la red en concreto, la vecindad de unos nodos con otros, el orden de llegada, etc. Además, son la principal herramienta en la que se apoya el encaminamiento en la red. En consecuencia, cada interfaz radio dispone de una dirección corta propia y no se comparte con todas las interfaces MiWi del nodo.

La ventaja de no compartir la dirección de red es que permite ahorrar espacio de direccionamiento si hay nodos que no disponen de todas las interfaces, permitiendo topologías más heterogéneas. Es decir, si por ejemplo a un nodo se le asigna la dirección 0x16 en sus dos interfaces de 434 y 868 GHz, la dirección 0x16 puede asignarse a otro dispositivo en la interfaz de 2,4 GHz.

Si por el contrario todos los nodos tienen las mismas interfaces, posiblemente la dirección de red de un nodo será la misma para todas sus interfaces, ya que se la asignará posiblemente el mismo coordinador al unirse a la red en cada una de las interfaces, yendo la numeración a la par en todas ellas. Este hecho puede suponer cierta ineficiencia, ya que se almacenan 3 direcciones de 2 bytes que pueden ser idénticas en muchos casos.

4.2.1.2. Estructura y diseño de la adaptación

Para poder integrar las tres interfaces MiWi deseadas en el diseño, se ha realizado un análisis de los principales bloques que componen la torre de protocolos de Microchip en los protocolos P2P y MiWi.

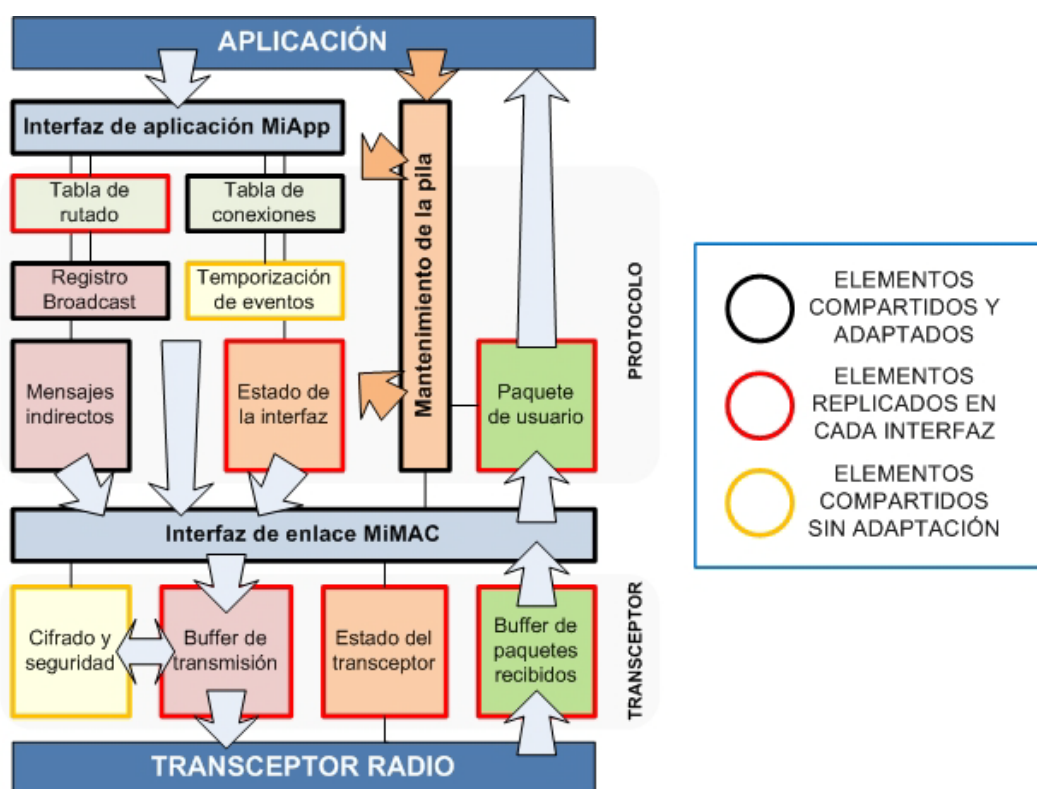


Figura 13: Diagrama de bloques software de una interfaz MiWi

Bloques del transceptor e interfaz de enlace

Comenzando por el más bajo nivel, el transceptor MiWi es un elemento que, lógicamente, necesita ser replicado para poder operar en distintas bandas de frecuencia. Además se utilizarán dos tipos de transceptores: el MRF49XA en las bandas de 434 y 868

MHz, y el MRF24J40 en 2,4 GHz. Como consecuencia, hay elementos intrínsecos a cada transceptor que no pueden compartirse.

El estado del transceptor depende del propio diseño del dispositivo (pues cada uno define sus estados) y cada transceptor del mismo tipo en general tiene su propio estado. Cada interfaz necesita también almacenar los datos del paquete que se van a enviar por ella, por lo que el buffer de transmisión debe ser un bloque propio de cada transceptor que se incluya en el diseño. También, como cada transceptor soporta ciertos mecanismos de seguridad e implementa sus rutinas de transmisión y recepción de mensajes se hace necesario replicar el almacenamiento de paquetes recibidos.

Sin embargo, los mecanismos de seguridad y cifrado de paquetes se pueden emplear para todos los transceptores. Las funciones que ofrecen dan servicio a todos los transceptores de Microchip, independientemente del que se escoja para el diseño. En nuestro caso, al incluir tres, codificarán y decodificarán los paquetes para todas las interfaces.

La interfaz de enlace de Microchip es un bloque formal que regula el servicio al protocolo de red. Ya que es un bloque que no implementa ninguna funcionalidad sino que define la interfaz entre niveles, es bastante sencillo adaptarlo manteniendo una estructura similar.

Bloques de los protocolos de red e interfaz de aplicación

A nivel de protocolo encontramos muchos más bloques funcionales, aunque no todos ellos están definidos para todos los protocolos. El protocolo P2P no incluye la tabla de encaminamiento y el registro de broadcast. El caso del bloque de temporización de eventos es especial, ya que es independiente del protocolo y el transceptor y simplemente da soporte a la pila.

Como en el caso de los transceptores, hay bloques que son específicos a cada interfaz radio y no se pueden adaptar fácilmente sin rediseñar la pila de protocolos. Es el caso del estado y las variables de cada interfaz y del espacio reservado para el paquete de usuario recibido y listo para que lo lea la aplicación.

La pila de protocolos en ocasiones almacena mensajes para dispositivos RFD asociados como mensajes indirectos y en el caso de MiWi, lleva control de los mensajes broadcast por la red. Estos bloques pueden ser compartidos por las interfaces siempre que se relacionen los mensajes a la interfaz radio por la que se recibieron o van a ser transmitidos.

La tabla de encaminamiento resulta difícil de adaptar, ya que al disponer de diferentes direcciones de red (una por cada interfaz), compartirla puede suponer problemas a la hora de determinar la ruta que debe seguir cada paquete. Sin embargo, la tabla de conexiones si puede compartirse gracias a que la dirección larga es única e idéntica para todas las interfaces. Es un elemento a adaptar que puede redundar en un ahorro de recursos.

Por último, la interfaz de aplicación y el mantenimiento de protocolos son los bloques con los que interactúa la aplicación en la torre de protocolos de Microchip. La interfaz de

aplicación es un elemento análogo a la de enlace, pero a más alto nivel. Ambos elementos se pueden adaptar para que existan tres interfaces, si bien el mantenimiento de la pila es mucho más complejo y conllevará mayor esfuerzo.

4.2.2. Capa de abstracción hardware

En el nuevo diseño del nodo NGD, la capa de abstracción hardware jugará el papel de intermediaria entre la aplicación y la pila de protocolos adaptada de Microchip. Para ello se definirán funciones en base a los requisitos fijados en el capítulo 3 que realizarán llamadas a la interfaz de aplicación de la pila adaptada y al mantenimiento de la torre de protocolos.

La HAL debe ser un elemento flexible más dentro del nuevo diseño, y en ese sentido, deja la puerta abierta a la inclusión de nuevos transceptores y protocolos. No es simplemente una extensión de la interfaz de aplicación de Microchip. Además, se añadirán funcionalidades deseables para desacoplar a las aplicaciones de la implementación de la torre de protocolos adaptada, como por ejemplo la posibilidad de que la pila realice el mantenimiento automáticamente. Las funciones de la HAL también permitirán acceder a los datos de usuario de manera sencilla sin necesidad de manejar las diversas estructuras de datos de cada interfaz.

Uno de los objetivos de la HAL es reducir la complejidad, por lo que el número de funciones de la interfaz de programación no debe ser muy elevado. Las funciones y parámetros de la HAL tendrán nombres representativos relativos a su funcionalidad y su uso para simplificar el desarrollo de las aplicaciones.

Para que las distintas situaciones anómalas que se den en tiempo de ejecución, se definirán códigos de error que informen del resultado de una invocación a la HAL. De esta manera la aplicación podrá tomar diferentes medidas en función del error notificado, o continuar normalmente si la acción realizada finalizó con éxito.

Dentro de la HAL se definen variables que monitorizan del estado general del nodo y sus interfaces radio. Estas variables deben permitir que la aplicación pueda saber qué interfaces tienen datos, cuáles están en modo dormido, si la seguridad está habilitada, etc.

El detalle de la implementación de las funciones de la HAL, las variables y los códigos de error se tratan en el siguiente capítulo.

5. FASE DE IMPLEMENTACIÓN

Antes de adentrarnos en la descripción detallada de la implementación, es importante recordar que uno de los requisitos principales del nodo es poder evaluar el estado y transmitir en distintas bandas del espectro radioeléctrico. Con tal fin, el NGD tiene la posibilidad de dar soporte a hasta tres interfaces MiWi que operen en distintas regiones del espectro. Dichas interfaces deben dar servicio a las estrategias cognitivas y a las aplicaciones concretas que se desarrollen sobre ellas.

Para tratar de hacer un mejor uso de los recursos del nodo y facilitar el diseño de aplicaciones, se ha invertido un gran esfuerzo en adaptar la torre de protocolos de MiWi, proporcionada por el fabricante, y en implementar una capa de abstracción hardware que dote a la aplicación de funciones de programación de alto nivel. El criterio que se ha tratado de seguir en la implementación software es el de mantener una interfaz lo suficientemente sencilla para ser manejable, cubriendo el mayor rango de funcionalidad posible.

En cuanto al hardware, antes de rediseñar el nodo completo, se ha preferido probar el nuevo software sobre el prototipo demostrador con ligeras modificaciones. Dentro del laboratorio se ha diseñado una placa de extensión que ha permitido probar la nueva implementación software. Este hecho ha supuesto grandes ventajas, pues no sólo ha servido para probar la viabilidad de la adaptación de la torre de protocolos, reduciendo así el coste y el tiempo de desarrollo en caso de errores, sino que también ha permitido detectar problemas y recabar nuevas ideas para un futuro rediseño de la plataforma.

5.1. ENTORNO DE DESARROLLO Y HERRAMIENTAS.

Para la realización de este proyecto hemos utilizado, principalmente, las herramientas que facilita Microchip para desarrollar aplicaciones en sus microcontroladores PIC y los medios de los que dispone el Laboratorio de Sistemas Integrados (LSI).

Microchip ofrece de manera gratuita un entorno de desarrollo integrado (IDE) denominado MPLAB-X, y un conjunto de compiladores para las diferentes arquitecturas de microcontroladores que comercializa. En el caso que nos concierne, sólo es necesario el compilador XC32, ya que el nodo utiliza un microcontrolador de la familia PIC32MX de 32 bits. Microchip también facilita librerías software que permiten manipular cómodamente los periféricos que incorpora el microcontrolador, cuyas funciones y macros son muy útiles a la hora de configurar y hacer uso de los recursos.

La combinación del entorno de desarrollo, las librerías y el compilador nos permite escribir código C de manera cómoda y sencilla, compilarlo y convertirlo a código máquina para poder ejecutarlo en el PIC. Existe mucha documentación al respecto del entorno, las librerías y los microcontroladores PIC accesible desde la web corporativa de Microchip.

Asimismo, se ofrecen programas a modo de demostración de diversos módulos y ejemplos de uso para aplicaciones concretas.

Para la depuración de errores en tiempo de ejecución, se ha empleado uno de los dispositivos de depuración de Microchip que soporta el microcontrolador utilizado, cuyo nombre comercial es ICD3. Esta herramienta ha sido indispensable para probar el funcionamiento del firmware, adaptar las torres de protocolos y testear las funcionalidades del nodo NGD.

Al margen de las herramientas de Microchip, también se ha hecho uso de las herramientas disponibles en el laboratorio, tales como el analizador de espectros *Agilent FieldFox™*, el osciloscopio y otros aparatos de medida.

5.2. LA TORRE DE PROTOCOLOS DE MICROCHIP

La adaptación software parte de la torre de protocolos de MiWi y las librerías que proporciona la compañía de manera gratuita. En concreto se ha utilizado la versión actualizada a fecha de 3 de Abril del 2012 (Microchip Solutions v2012-04-03). Para comprender las diferencias en la implementación, en las siguientes páginas se describe brevemente la pila original y su problemática antes de explicar los cambios que se han realizado.

5.2.1. - Descripción general

El código fuente está en lenguaje C, ya que el entorno y los compiladores facilitados por Microchip así lo precisan. Las librerías de Microchip incluyen, no sólo el entorno de desarrollo de MiWi con la respectiva torre de protocolos, sino también otros módulos que pueden ser útiles en determinadas plataformas o aplicaciones:

- Funciones de manejo de gráficos, pantallas táctiles capacitivas, sistemas de ficheros y tarjetas SD, entre otros.
- Torre de protocolos TCP/IP para implementar Wi-Fi. Esta pila de protocolos se incluía en el prototipo demostrador de Fernando López Lara para utilizar el transceptor MRF24WB0MA.
- Framework para USB con ejemplos para gestionar memorias USB, emular conexiones serie, etc. El prototipo demostrador utilizaba la interfaz USB para volcar trazas al PC o datos a una memoria externa.

En la adaptación software para el nuevo nodo, solo se ha empleado la sección relativa a MiWi, dejando la pila de USB y la de Wi-Fi para posibles mejoras o extensiones futuras. La propia torre de protocolos fija inherentemente una jerarquía entre los ficheros de MiWi que facilita la búsqueda de funcionalidades y opciones de configuración en el código. Todas las cabeceras se encuentran en el mismo subdirectorio clasificadas por módulos, mientras que los

archivos fuente relativos a MiWi se hallan principalmente en los subdirectorios de transceptores y protocolos inalámbricos.

5.2.2. Opciones de configuración.

En el capítulo de diseño veíamos que la pila de MiWi admitía configuraciones a diversos niveles de abstracción. Gracias a ello, la pila de Microchip es bastante versátil y puede ajustarse para un rango amplio de aplicaciones, no solo escogiendo el protocolo y el transceptor deseado, sino también modificando ciertas variables y opciones que influyen en la operación de los protocolos y en el uso de memoria de programa y otros recursos del microcontrolador. La figura muestra los ficheros de configuración, sombreados en azul oscuro en la columna izquierda, donde seleccionar las opciones descritas a continuación.

Es importante destacar que la mayoría de estas opciones se preseleccionan antes de programar el dispositivo y no se pueden cambiar de manera dinámica mediante software. Las opciones se habilitan o deshabilitan mediante directivas para el compilador. La selección de parámetros se hace de manera similar, modificando la definición de las constantes utilizadas en el código.

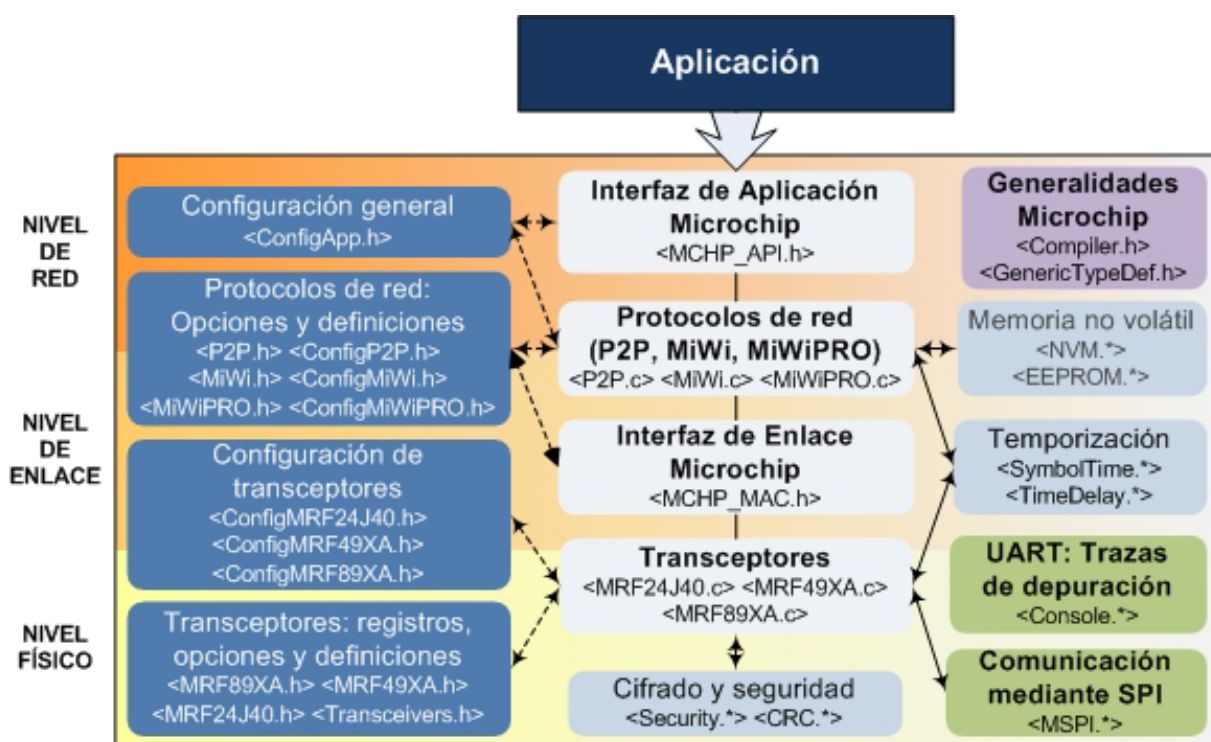


Figura 14: Jerarquía de archivos de la pila de MiWi™

5.2.2.1. Aplicación

En este nivel podemos seleccionar las generalidades de toda la torre de protocolos y parámetros a nivel de aplicación.

Opción de configuración	Descripción
ENABLE_CONSOLE	Habilita imprimir trazas a través de la UART en un hiperterminal.
ENABLE_NETWORK_FREEZER	Permite almacenar cierta información y variables de estado en memoria no volátil. Requiere disponer de almacenamiento no volátil.
HARDWARE_SPI	Se habilita para emplear los pines de la interfaz SPI del microcontrolador en lugar de realizar bit-banging ⁹ con pines de entrada/salida de propósito general
ENABLE_PA_LNA	Utiliza el amplificador de potencia y el amplificador de bajo ruido del módulo de RF (Sólo en módulos MRF24J40MB /MRF24J40MC).
ENABLE_HAND_SHAKE	Esta opción de P2P obliga a los dispositivos a realizar el proceso de hand-shaking antes de la comunicación. Si se deshabilita, los nodos en P2P sólo pueden enviar mensajes broadcast, o unicast con las direcciones destino que se hayan embebido manualmente en el código.
ENABLE_SLEEP	Indica que el nodo puede dormirse, por lo que a todos los efectos se le considera un dispositivo del tipo RFD.
ENABLE_ED_SCAN	Permite escanear los canales radio para determinar en cuál de ellos hay menos ruido
ENABLE_ACTIVE_SCAN	Esta característica permite buscar redes existentes en la banda de trabajo. Se puede regular el máximo número de redes de las cuales se almacena su información durante esta búsqueda en la configuración del protocolo.
ENABLE_SECURITY	Permite cifrar paquetes y disponer de los mecanismos de seguridad de los protocolos.
ENABLE_INDIRECT_MESSAGE	Como los dispositivos RFD pueden no estar activos en todo momento, los FFDs almacenan los mensajes para ellos hasta que el dispositivo RFD solicita los datos. La pila permite seleccionar cuántos mensajes se almacenan y el periodo de validez de los mismos en la configuración del protocolo.
ENABLE_BROADCAST	Guarda los mensajes broadcast como mensajes indirectos para que los dispositivos RFD que al despertar soliciten datos reciban el mensaje.

Tabla 6: Opciones de configuración de la pila de MiWi™ a nivel de aplicación

Además, en la configuración de aplicación se debe elegir el protocolo a utilizar, el rol del nodo en la red (si procede) y el transceptor utilizado (la pila original solo admite un único transceptor). Otros parámetros configurables son:

- La dirección permanente del dispositivo (EUI o dirección larga) y su longitud, salvo en el caso de emplear la longitud que exige 802.15.4 que queda restringida a 8 bytes y está bajo las políticas de asignación del IEEE y el fabricante.
- El identificador de red de área personal (PANID) para IEEE 802.15.4

⁹ Bit-banging es una técnica empleada en comunicaciones serie para emular con software las funciones que realiza el hardware dedicado. Los pines del microcontrolador se manipulan y comprueban para enviar y recibir datos con la correcta señalización y temporización mediante el software. Es una solución de bajo coste y flexible, pero requiere mayor procesamiento por parte del microcontrolador y es menos fiable que el hardware dedicado.

- El tamaño del buffer de transmisión y recepción: limita la carga útil máxima de los paquetes de aplicación.
- El tamaño de la tabla de conexiones, donde se registran las direcciones de los dispositivos asociados.
- El tamaño de la información adicional de aplicación que se almacena en la tabla de conexiones.
- El intervalo de tiempo que transcurre hasta que un dispositivo RFD se vuelve a despertar si se habilita la sincronización temporal.

5.2.2.2. Protocolos de red

Tanto si se emplea MiWi como P2P, los protocolos definen una serie de variables que permiten configurar ciertos aspectos del mismo y las siguientes opciones de configuración:

Opción de configuración	Descripción
ENABLE_DUMP	Habilita las funciones de volcado por pantalla de la tabla de conexiones.
ENABLE_TIME_SYNC	Para dar soporte a la sincronización temporal entre dispositivos. Si se habilita, a tal efecto se definen la frecuencia del reloj utilizado para el sincronismo (típicamente 32768 Hz) y el número de intervalos de sincronismo.
ENABLE_ENHANCED_DATA_REQUEST	Permite que los dispositivos RFD al despertar envíen una petición especial para ganar tiempo y poder ahorrar batería si el tiempo en activo es extremadamente reducido.

Tabla 7: Opciones de configuración de la pila de MiWi™ a nivel de protocolo

Adicionalmente, el protocolo MiWi dispone de las siguientes funciones opcionales:

Opción de configuración	Descripción
ENABLE_MIWI_ACKNOWLEDGEMENT	Activar esta funcionalidad implica asentar los paquetes a nivel de red para dar servicios de fiabilidad a la capa de aplicación. Se puede regular el tiempo de espera al ACK.
ENABLE_BROADCAST_TO_SLEEP_DEVICE	Permite almacenar mensajes broadcast para dispositivos RFD. El número máximo de paquetes a almacenar y el periodo de validez de dichos mensajes es regulable. Es redundante con la opción de ENABLE_BROADCAST.

Tabla 8: Opciones de configuración adicionales del protocolo MiWi Mesh™

Como MiWi implementa funciones de encaminamiento, existen variables que pueden ser alteradas en función de los requisitos del nodo para regular cuántos fallos se permiten en un salto antes de considerarlo como no válido y para gestionar conexiones indirectas ("sockets") entre dispositivos.

5.2.2.3. Transceptor MRF49XA

La configuración del transceptor permite seleccionar mediante la definición de constantes:

- La banda de frecuencias de trabajo: 915, 868 o 434 MHz. Debe ser coherente con el circuito de adaptación y la antena.
- La tasa binaria del canal radio: 1200, 9600, 19200, 38400, 57600 o 115200 kbps. Fija el ancho de banda de cada canal y el número de canales. A mayor ancho de banda, menor número de canales.
- Ganancia del amplificador de bajo ruido (LNA) interno y potencia de transmisión inicial (entre varias opciones).
- El umbral de potencia del indicador de intensidad de la señal recibida (RSSI) para su salida digital en el transceptor.
- Clave y número de secuencia para el cifrado. Elección entre varios modos de seguridad.
- Número de paquetes que se pueden recibir y almacenar hasta que la capa MAC los atienda.
- Intervalo de actualización del contador de tramas en memoria no volátil.

Además se pueden habilitar las siguientes opciones para añadir o reducir funcionalidades o alterar el comportamiento del transceptor:

Opción de configuración	Descripción
ENABLE_CCA	Habilitar el uso de técnicas Clear Channel Assessment (CCA). En ese caso se definen tres parámetros: el número de comprobaciones que se realizan en cada procedimiento, un umbral de cuántos fallos se permiten en él, y el número de veces que se reintenta el procedimiento si se excede dicho umbral.
ENABLE_ACK	Habilitar la transmisión automática de un paquete de asentimiento (ACK) cuando el paquete recibido así lo requiera. El número de ACKs que almacena el transceptor para evitar entregar duplicados a la capa MAC es regulable.
ENABLE_RETRANSMISSION	Permitir retransmisiones automáticas en caso de no recibir el ACK de un paquete enviado. Se puede regular cuántas retransmisiones realiza antes de notificar un fallo en el envío.
SOURCE_ADDRESS_ABSENT	No incluir la dirección del dispositivo que envía el paquete. Esta opción depende fuertemente de la aplicación, y sólo se habilita en aquellas en las que el remitente de un paquete radio no sea de interés.
INFER_DEST_ADDRESS	Inferir la dirección destino a partir del CRC en lugar de transmitir la dirección destino en el paquete.

Tabla 9: Opciones de configuración del transceptor MRF49XA™

5.2.2.4. Transceptor MRF24J40

Como en el transceptor MRF49XA, puede definir el número de paquetes que se pueden recibir y almacenar, la clave y el modo de seguridad y el intervalo de actualización

del contador de tramas en memoria no volátil. No obstante, al ser un transceptor compatible con IEEE 802.15.4 tiene menos opciones de configuración.

Opción de configuración	Descripción
TURBO_MODE	Esta opción permite alcanzar tasas teóricas de 625 kbps. Si permanece deshabilitada, se utiliza la tasa binaria que define el estándar de 250 kbps.
VERIFY_TRANSMIT	Comprueba que tras finalizar una transmisión, el transceptor cambia a un estado conocido. Esta opción sacrifica cierto rendimiento y repercute en la cadencia eficaz, tal y como se verá en el capítulo de pruebas.

Tabla 10: Opciones de configuración del transceptor MRF24J40™

5.2.3. Recursos necesarios.

La implementación de la torre de protocolos emplea algunos recursos del microcontrolador, como es lógico. La pila precisa indispensablemente de memoria de programa y dos temporizadores de 16 bits en modo 32 bits para poder temporizar eventos y esperas. Adicionalmente, en función de las opciones habilitadas en la torre de protocolos, el firmware requiere otros recursos del microcontrolador.

Algunas funciones utilizan interfaces de comunicación. Es el caso de las trazas por consola, que utilizan la UART2 por defecto, el transceptor radio, que emplea una interfaz SPI, y el volcado a memoria no volátil, para el que típicamente Microchip oferta una EEPROM por SPI que puede ser multiplexada con el transceptor o utilizar otro módulo por su cuenta. En función del diseño hardware, debe adaptarse correctamente la inicialización y utilización de estos periféricos.

Aparte, el transceptor radio requiere una línea de interrupción externa, salvo el MRF89XA que necesita dos. En las plataformas que no disponen de suficientes líneas de interrupción externa, se suele emplear algún pin de entrada que pueda generar interrupciones cuando detecte un cambio en la entrada (Change Notification).

La sincronización temporal utiliza el timer 1, el único que puede funcionar en modo dormido, y requiere que se incluya en el diseño hardware el reloj secundario a 32768 Hz.

5.3. ADAPTACIÓN DE LA TORRE DE PROTOCOLOS.

En esta sección se trata de sintetizar la adaptación que se ha hecho de la pila original de Microchip.

5.3.1. Objetivos

La principal motivación de realizar la adaptación es el hecho de incluir en el diseño los tres transceptores elegidos para disponer de las tres interfaces radio que permitan el acceso a sendas bandas de frecuencia del espectro.

Otros objetivos secundarios que se persiguen con la adaptación son, por un lado, buscar que la adaptación sea eficiente tanto en la utilización de recursos (memoria y módulos empleados) como en el procesamiento demandado a el microcontrolador, y por otro lado, facilitar la comprensión de los protocolos y la adaptación software, tratando que sea legible y fácil de depurar (pese al desarrollo de una capa de abstracción hardware por encima).

Como valor añadido, se quiere dotar al nodo de la mayor flexibilidad posible para que pueda admitir modificaciones o la adición de nuevos módulos que extiendan su funcionalidad. Para propiciar este hecho, se ha intentado que el firmware sea fácilmente configurable en distintas plataformas hardware y los módulos de las interfaces de comunicación se puedan reasignar sin necesidad de modificar el código.

5.3.2. - Estructura

En la adaptación se ha mantenido mayoritariamente la estructura de la pila original de MiWi, pues la abstracción por niveles resulta ventajosa, pero la gran mayoría de los ficheros se han alterado en mayor o menor medida.

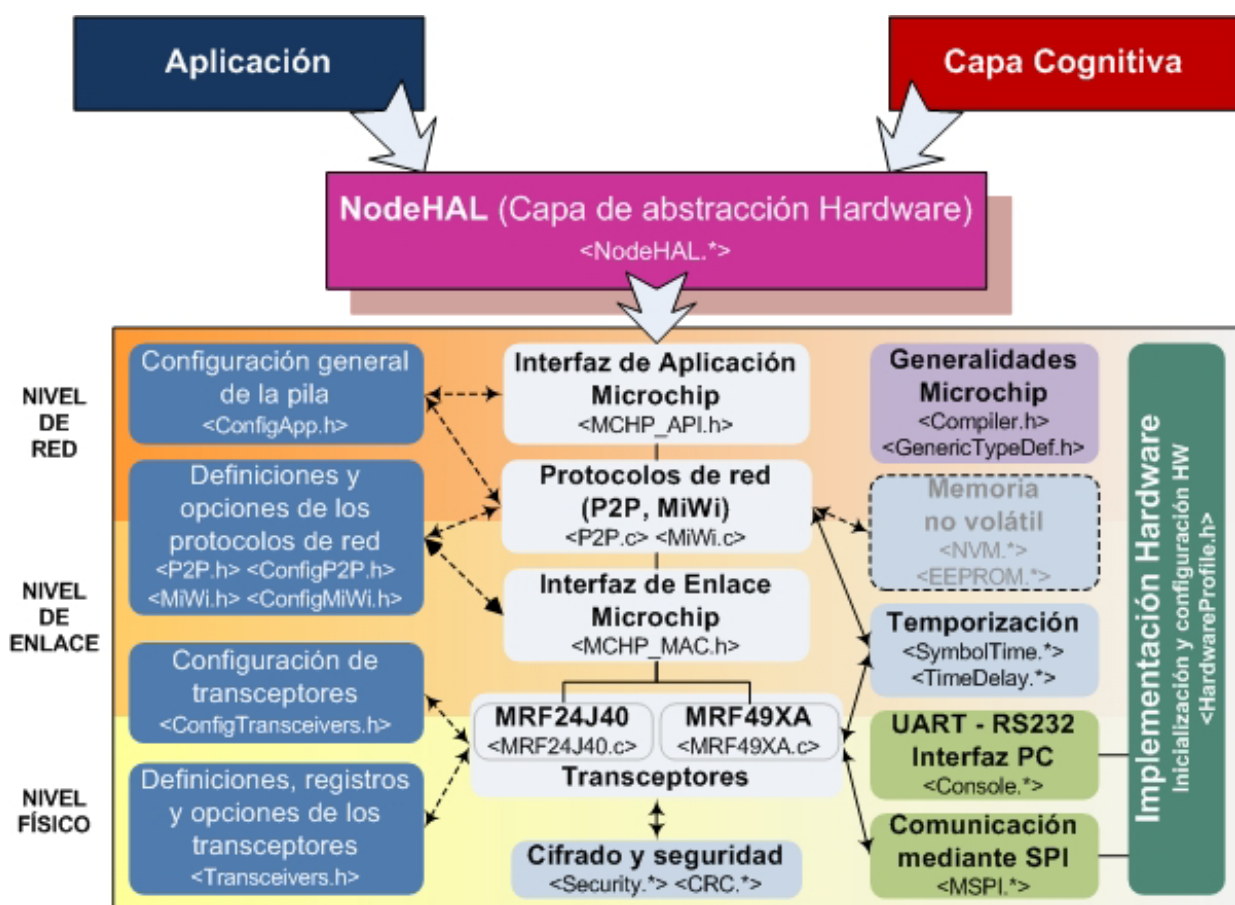


Figura 15: Jerarquía de archivos de la pila de Microchip adaptada

Para explicar los cambios, se seguirá un enfoque de abajo hacia arriba, de modo que se enlazaré con la descripción de la capa de abstracción hardware (HAL) en la siguiente sección.

5.3.3. La adaptación por niveles

5.3.3.1. - El nivel físico y los transceptores

Comenzando con el nivel más bajo de abstracción, encontramos los transceptores. Todos los transceptores originalmente disponían de una cabecera para configurar las opciones del transceptor, una cabecera con las propias definiciones para el mismo (direcciones de sus registros internos y tipos definidos para controlar su estado) y un archivo fuente donde se implementaba la misma interfaz MiMAC para todos.

Aunque la declaración de las funciones sea idéntica (pues MiMAC está diseñada para que se emplee cualquier transceptor), las diferencias intrínsecas de los transceptores hacen que el cuerpo de la función sea radicalmente diferente para cada uno de ellos. Por tanto, al no poder utilizar la misma función para manipular las distintas interfaces radio, y para evitar un conflicto de nombres durante la compilación, se ha duplicado la interfaz MiMAC, resultando un conjunto de funciones para el transceptor MRF24J40 (MiMAC_MRF24J40) y sus funciones homólogas para el MRF49XA (MiMAC_MRF49XA).

Para que la torre de protocolos pueda seleccionar uno de los dos transceptores MRF49XA del diseño se ha añadido un parámetro adicional a las funciones de MiMAC_MRF49XA. Por ahora, solo se contempla que el parámetro seleccione entre el MRF49XA_1 y el MRF49XA_2 del diseño (trabajando en las bandas de 434 y 868 MHz), aunque podría extenderse fácilmente a seleccionar entre más transceptores del mismo tipo si fuera de interés. Mediante dicho parámetro, se inicializan los punteros locales de cada función para acceder a las variables necesarias del transceptor elegido.

En aras de una mayor sencillez, las cabeceras de configuración se han reestructurado en dos únicos ficheros:

- **ConfigTransceivers.h:** Es donde se han incluido las opciones de configuración de todos los transceptores. Las opciones de un transceptor solo se están disponibles si el transceptor en cuestión está disponible en el diseño. Para tratar de reducir el consumo de espacio en memoria se ha forzado a que los transceptores compartan algunas variables como la clave de seguridad o el número de paquetes de usuario que puede almacenar cada interfaz. En el caso de los MRF49XA, se comparten también las mismas opciones de configuración, salvo la selección de la banda de trabajo y la tasa binaria de cada interfaz.
- **Transceivers.h:** Inicialmente en la pila de MiWi simplemente servía para incluir la cabecera del transceptor correspondiente. En el rediseño se ha empleado para reunir todas las definiciones de los registros de los transceptores y los tipos definidos que se emplean en la pila relativos a cada uno de ellos.

Además ha sido necesario adaptar las funciones lectura o escritura de las interfaces SPI. Microchip tenía la pila preparada para utilizar una interfaz para un único transceptor y, adicionalmente, una SPI para acceder a una EEPROM externa, que resultaban bastante caóticas.

En el nuevo firmware, en función de la configuración hardware de cada transceptor, se definen un par de macros para manejar la interfaz SPI a la que se conecta cada uno. Las macros invocan a las funciones de lectura y escritura sobre los registros de SPI, que ahora admiten como parámetro el módulo SPI a utilizar. Esto permite flexibilizar el uso de las interfaces SPI y disponer de una única función en lugar de varias para realizar la misma tarea. El único aspecto reprochable es que las interfaces SPI no se emplean en modo maestro-multiesclavo, lo que implica utilizar una interfaz SPI por cada periférico que requiera una interfaz de este tipo.

En cuanto a las funciones de cifrado y seguridad, el código fuente para calcular el código de redundancia cíclico (CRC) y cifrar o descifrar los paquetes radio es de las pocas partes del código que no han sufrido alteraciones respecto a la pila original.

5.3.3.2. - Los protocolos de red

Para el NGD solo se han adaptado dos de los tres protocolos que ofrece Microchip en su entorno de desarrollo: MiWi P2P™ y MiWi Mesh™. El protocolo MiWiPRO™ se deja como posible mejora en caso de que determinadas aplicaciones necesiten mayor número de nodos y mayor capacidad de encaminamiento.

Los protocolos implementan las funciones de la interfaz de aplicación MiApp para cubrir todas las funcionalidades del nivel de red: principalmente descubrir a otros dispositivos en la red, conectarse con ellos y tratar que los paquetes enviados alcancen su destino, bien de manera directa o bien siguiendo una ruta concreta dentro de la red. Para hacer uso de los transceptores y algunas otras funciones del nivel de enlace o del nivel físico, invocan a las funciones de la interfaz MiMAC.

Al haber dividido la interfaz MiMAC para controlar transceptores de dos tipos y un total de tres interfaces MiWi en distintas bandas de frecuencia, la adaptación de los protocolos es lo que más esfuerzo ha conllevado. Además, se han realizado algunas modificaciones a los protocolos para corregir ciertos aspectos que no eran deseables dados los propósitos con los que se diseña el nodo.

Quizás, lo primordial para adquirir una visión global de la nueva torre de protocolos es, de nuevo, la inclusión de un parámetro en las funciones de MiApp para referirse a cada una de las interfaces MiWi. De esta manera, la capacidad de seleccionar la interfaz sigue ascendiendo por la torre de protocolos hasta la HAL.

Para ello, se ha definido un tipo enumerado "miwi_band" que permite invocar a dichas funciones para emplear la interfaces en 434 MHz, 868 MHz y 2,4 GHz. La correspondencia entre la banda de trabajo y el transceptor que trabaja en ella se realiza en

este nivel. Para que esta asociación sea unívoca, no puede haber dos transceptores MiWi trabajando en la misma banda.

La estrategia que se ha seguido es similar a la de los transceptores. Con la selección de la interfaz MiWi en cada función se han inicializado los punteros locales pertinentes para acceder a las variables de dicho transceptor.

Tabla de conexiones.

Una característica muy importante de la adaptación es la compartición de la tabla de conexiones, cuya implementación depende del protocolo utilizado. La tabla de conexiones es el espacio que el nodo reserva en memoria para registrar los dispositivos asociados, sus direcciones y, opcionalmente información adicional para la aplicación.

En el caso del protocolo P2P, que sólo emplea la dirección larga, se ha añadido un flag por cada interfaz radio para indicar que una conexión está vinculada a esa interfaz. De este modo se evita añadir un registro en la tabla por cada interfaz radio que quiera registrar el mismo dispositivo o eliminar un registro de un nodo cuando una interfaz lo solicite pero aún sea válido para otras.

Para poder actualizar la tabla de conexiones correctamente en P2P y depurar posibles errores, se han modificado las funciones P2P que manejan la tabla. Esta modificación atañe básicamente a momentos concretos, principalmente al unirse a la red, restablecer conexiones anteriores o eliminar alguna conexión. Para el primero de ellos se muestra un diagrama explicativo en la figura 16, ya que es el caso más complejo y frecuente.

En cuanto a MiWi, como no es un protocolo tan sencillo y utiliza los dos tipos de conexiones, la tabla de conexiones dispone de más campos: un flag para indicar si es un RFD o sleeping device, la dirección larga (común para todas las interfaces), la dirección corta de cada interfaz presente en el nodo y un flag de si tiene conectividad directa con ese dispositivo a través de dicha interfaz, y finalmente, la información de aplicación que, como en el caso de P2P, es opcional.

Además, los coordinadores (incluido el PANC) disponen de una tabla de encaminamiento que indica el siguiente salto en la red para llegar a cada coordinador (en MiWi su número está limitado a 8) y su subred. Esta tabla se ha replicado, pues aunque lo más seguro es que todos los coordinadores tengan la misma dirección de red en todas sus interfaces (al asignársela el PANC al unirse a la red), evita errores de encaminamiento si no es así. Además, cada interfaz puede determinar rutas diferentes en función del alcance que se logre con cada una de ellas.

No obstante, es una medida prudente que los nodos coordinadores se conecten de manera progresiva, ya que iniciar todos los dispositivos simultáneamente podría provocar condiciones de carrera por asumir el rol de PANC.

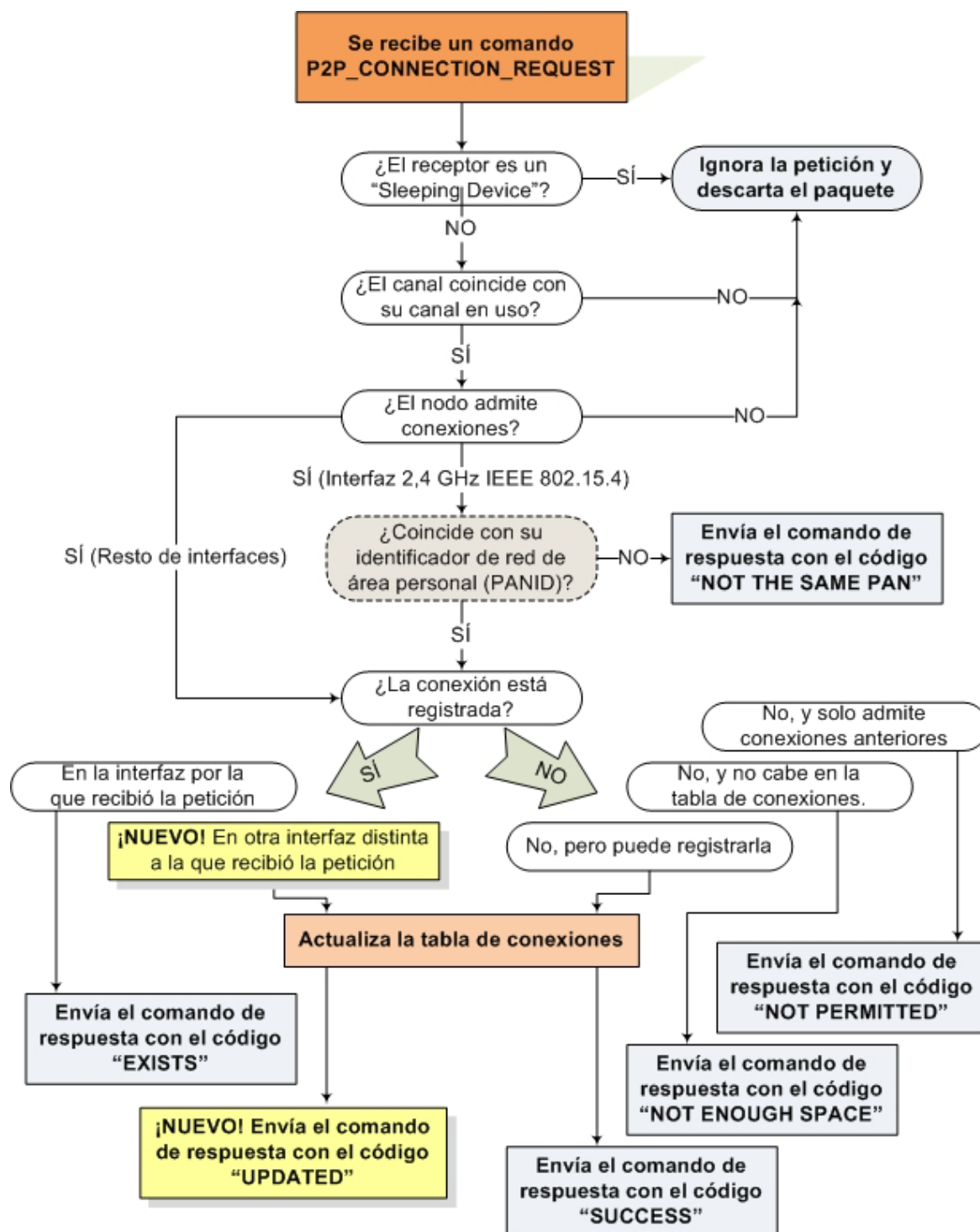


Figura 16: Adaptación del handshaking en P2P para compartir la tabla de conexiones.

Mensajes indirectos.

Una de las opciones de configuración de la torre de protocolos es permitir el uso de mensajes indirectos. Esta funcionalidad está orientada a un escenario bastante concreto, en el que los dispositivos RFD despiertan cada poco tiempo para solicitar sus mensajes pendientes, permaneciendo inactivos el resto del tiempo.

Relacionadas con esta característica, existían dos opciones de configuración redundantes (ENABLE_BROADCAST, opción a nivel de aplicación que solo se utilizaba en el protocolo P2P, y ENABLE_BROADCAST_TO_SLEEP_DEVICE, específica de MiWi), que forzaban a que además de enviar el paquete broadcast se almacenase como mensaje indirecto. De este modo, los dispositivos RFD que estuvieran dormidos podrían recibir el mensaje al solicitar datos pendientes. Se ha corregido esta duplicidad innecesaria, dejando una única opción a nivel de aplicación denominada como la opción específica de MiWi.

El hecho de almacenar paquetes indirectos para otros dispositivos consume memoria, por lo que en la adaptación se ha compartido el registro de mensajes indirectos para todas las interfaces. Para identificar a qué interfaz corresponde un mensaje indirecto, se ha añadido un campo en la estructura que almacena la información de los mismos. Para enviar los datos al RFD solicitante, debe pedirlos por la interfaz correcta, ya que sólo se enviarán por la interfaz que solicitó almacenar el paquete indirecto.

Como el hecho de que no se pueda enviar directamente un paquete a un dispositivo RFD (sino que tenga que pedirlo éste) se ha considerado que no es lo deseable, se ha modificado el comportamiento de los protocolos. La adaptación se explica en los apartados siguientes, que resumen los diferentes casos de transmisión broadcast y unicast en cada uno de los protocolos.

Envío de paquetes: protocolo P2P.

Broadcast

Aunque se han introducido modificaciones mínimas para reorganizar el código, el comportamiento frente a la pila original no se ha visto alterado. La transmisión broadcast implica las siguientes acciones a nivel de red:

1. (Solo si se habilitan los mensajes broadcast indirectos) Si en la tabla de conexiones hay registrado cualquier nodo que puede dormir sus interfaces (RFD), se almacena un mensaje indirecto.
2. Se envía el mensaje broadcast.
3. Se informa a la capa superior de si la transmisión tuvo éxito.

Unicast

Se distinguen dos casos:

- A) El nodo destino es un dispositivo que puede dormir sus interfaces radio (RFD).
 - A1. - Si los mensajes indirectos no están habilitados, el comportamiento es igual que en la pila original y simplemente se envía el mensaje, informando del resultado a la capa superior.
 - A2. - En cambio, si se habilitan, los comportamientos son diferentes:

Microchip MiWi Stack	Microchip LSI-CWSN Stack
<ol style="list-style-type: none"> 1. No se envía el paquete directamente. Si el dispositivo se encuentra en la tabla de conexiones, el paquete se intenta almacenar como mensaje indirecto. 2. El mensaje indirecto no se envía hasta que el nodo no recibe una petición de datos pendientes por parte del nodo destino. 3. Los paquetes indirectos tienen un tiempo de validez. Si no se envían durante ese periodo expiran para almacenar otros nuevos. 4. La capacidad de almacenar paquetes indirectos en memoria es limitada, por lo que se informa a la aplicación de si se pudo almacenar el paquete indirecto o no. 	<ol style="list-style-type: none"> 1. Primero se envía el paquete directamente. De este modo, si el nodo destino está despierto podrá recibir el paquete sin necesidad de solicitarlo. 2. Adicionalmente, si la transmisión falla y el nodo destino se encuentra en la tabla de conexiones se almacena un mensaje indirecto, siempre que quede espacio libre para hacerlo. 3. Se informa a la capa superior del resultado de la transmisión.

Tabla 11: Diferencias en la transmisión unicast entre la pila original y la pila adaptada.

B) El nodo destino es un dispositivo que nunca duerme sus interfaces radio.

En la adaptación, siempre se envía el paquete primero. En este caso, es indiferente que se habiliten los mensajes indirectos o no, porque el nodo destino debería recibir el mensaje al no poder dormir sus interfaces radio. El comportamiento es equivalente al de la pila original.

Envío y encaminamiento de paquetes: protocolo MiWi.

Como ya se ha introducido en secciones anteriores, el protocolo MiWi implementa funciones de encaminamiento por lo que en ocasiones, la transmisión de un paquete implica pasar por otros nodos de la red antes de llegar a su destino.

El protocolo MiWi presenta un error en su implementación por lo que los paquetes de usuario cuyo primer byte de carga útil es 0 son interpretados por el protocolo como paquetes de control. Para subsanar este problema se han implementado dos soluciones distintas, ofreciendo la posibilidad de elegir la que se considere oportuna:

- Reportar un error en la capa de abstracción hardware si la aplicación intenta escribir un paquete cuya carga útil (payload) comienza por cero. Ésta es la opción habilitada por defecto.
- Rellenar el primer byte de todos los paquetes con un valor distinto de cero y desechar este primer byte al recibir un paquete. Esta opción supone desaprovechar carga útil, pero es transparente para cualquier paquete de aplicación.

Broadcast

La transmisión broadcast en MiWi implica los mismos pasos que en el protocolo P2P, ya que los mensajes broadcast no necesitan ser encaminados al ser su destino toda la red.

Sí es importante recalcar que, a diferencia de P2P, los mensajes broadcast en MiWi son reenviados por los coordinadores para que se difundan por la red. Los coordinadores almacenan información del mensaje broadcast para detectar copias de ese mensaje en una tabla y así evitar inundar la red.

Unicast

Con el protocolo MiWi se presenta la posibilidad de emplear la dirección larga o la dirección de red. Si se usa la dirección larga, se busca en la tabla de conexiones si se dispone de la dirección corta de la interfaz elegida y si ésta es válida. En caso afirmativo, se enviará el mensaje empleando la dirección corta (que se explica a continuación). En caso contrario, o si la dirección larga no se encuentra en la tabla, se envía utilizando la dirección larga.

Al utilizar la dirección corta o dirección de red, la pila utiliza sus funciones de encaminamiento para tratar de que el paquete llegue a su destino. Los diferentes casos son:

- A) El nodo origen no es un coordinador (es un End Device). En ese caso le envía el mensaje al coordinador al que está asociado para que lo rute debidamente.
- B) El dispositivo origen es un coordinador, encargado del encaminamiento de los mensajes. Por cada salto en la red se pueden producir las siguientes situaciones:
 - B1. - El nodo destino está dentro de la subred del propio coordinador.
El coordinador enviará el mensaje al nodo destino de manera equivalente a una transmisión unicast de P2P, incluida la adaptación de los mensajes indirectos.
 - B2. - El nodo destino está en la subred de otro coordinador conocido.
El mensaje se envía al coordinador del nodo destino.
 - B3. - Otro coordinador conocido puede alcanzar el coordinador destino.
El mensaje se envía a través de dicho coordinador vecino, que hace de intermediario ("relay").
 - B4. - El coordinador no sabe como encaminar el mensaje.
El mensaje se envía al coordinador de red de área personal (PANC), que debe contener en su tabla de encaminamiento información sobre todos los coordinadores. Si el coordinador en cuestión es el PANC (algo que no debería ocurrir) se intenta enviar directamente al coordinador del nodo destino.

Capacidades no adaptadas.

Hay algunas capacidades de los protocolos que no se han adaptado, ya que en principio no se plantean necesarias para el nodo destino o requieren hardware específico que no dispone el prototipo original adaptado.

El volcado a memoria no volátil se corresponde con la opción de configuración de nivel de aplicación `ENABLE_NETWORK_FREEZER`, que debe permanecer deshabilitada. Para poder utilizar esta característica de la pila es necesario disponer de una memoria no volátil externa y adaptar las funciones de los protocolos que guardan y restauran las variables. Como contrapartida, se han incluido funciones en la HAL similares, aunque con los siguientes matices:

- No se requiere memoria no volátil. Simplemente se ofrecen los datos a la aplicación para que ésta los almacene donde considere oportuno.
- No se emplea un dispositivo externo y, en consecuencia, no se requiere el uso de una interfaz SPI para comunicarse con él. De este modo las SPI quedan libres para las interfaces radio.

En cuanto a la sincronización temporal (la opción `ENABLE_TIME_SYNC`), no se ha adaptado al considerar esta funcionalidad menos interesante y, al mismo tiempo, poder prescindir del reloj de tiempo real para reducir el coste y consumo del nodo. No obstante se deja la puerta abierta a incluir un segundo oscilador en el diseño, bien para la aplicación o bien para esta funcionalidad de la pila.

Por último, la opción `ENABLE_ENHANCED_DATA_REQUEST` no se ha probado y se considera no portada a la nueva pila a todos los efectos, pese a que el código sí se ha adaptado. Se trata de una opción muy específica de la pila que permite despertar una interfaz de los dispositivos RFD de manera que pida los mensajes indirectos pendientes de manera más rápida. No se ha empleado esta funcionalidad y se ha optado por reactivar las interfaces de manera normal, siendo la aplicación la que solicite los datos si lo considera oportuno.

Mantenimiento de la torre de protocolos.

Los protocolos de Microchip están diseñados para que realicen todas sus funciones de mantenimiento invocando una única función lo más frecuentemente posible. En dicha función se controlan todas las funcionalidades habilitadas y, lo más importante, se gestionan los paquetes de control y de usuario que reciben las distintas interfaces.

La adaptación de esta función ha sido laboriosa, por su gran extensión y su complejidad. Para poder reutilizar el código de la función de mantenimiento con varias interfaces, en general, se ha separado la parte común a todas las interfaces (mensajes indirectos, etc) de la parte específica de una única interfaz. La parte de cada interfaz se ha enmarcado dentro de una función de apoyo que se invoca desde la función de mantenimiento principal.

La función de apoyo realiza las tareas de mantenimiento para una sola interfaz y admite un parámetro para seleccionar cuál. La función se invoca tantas veces como interfaces radio se incluyen en el nodo, de manera consecutiva. La función emplea punteros que se inicializan al invocarla para acceder a las variables de la interfaz seleccionada.

Para ejecutar las tareas de mantenimiento, la nueva pila ofrece dos opciones:

- **Realizar el mantenimiento automáticamente, de manera periódica:**

Esta opción permite que la aplicación se despreocupe del mantenimiento mediante la habilitación de un temporizador en el microcontrolador. En concreto, se configura el timer1 para interrumpir con baja prioridad y realizar las labores de mantenimiento de la pila. Este temporizador coincide con el que se empleaba en la pila original de Microchip para implementar la sincronización temporal. Se ha elegido éste para que la aplicación pueda disponer de dos temporizadores de 16 bits que puedan combinarse para funcionar en modo 32 bits.

- **Delegar el mantenimiento de la pila en la aplicación:**

Esta opción implica que la propia aplicación es la que invoca a la función de mantenimiento con suficiente frecuencia para el buen funcionamiento de la pila. Se recomienda la primera opción salvo que sea muy crítico interrumpir a la aplicación con las tareas de mantenimiento.

Las figuras [A y B] muestran un esquema de las tareas de mantenimiento en los protocolos P2P y MiWi respectivamente, con línea discontinua aquellas tareas que son opcionales (dependen de la configuración) y sombreadas en gris las relacionadas con funcionalidades no adaptadas.

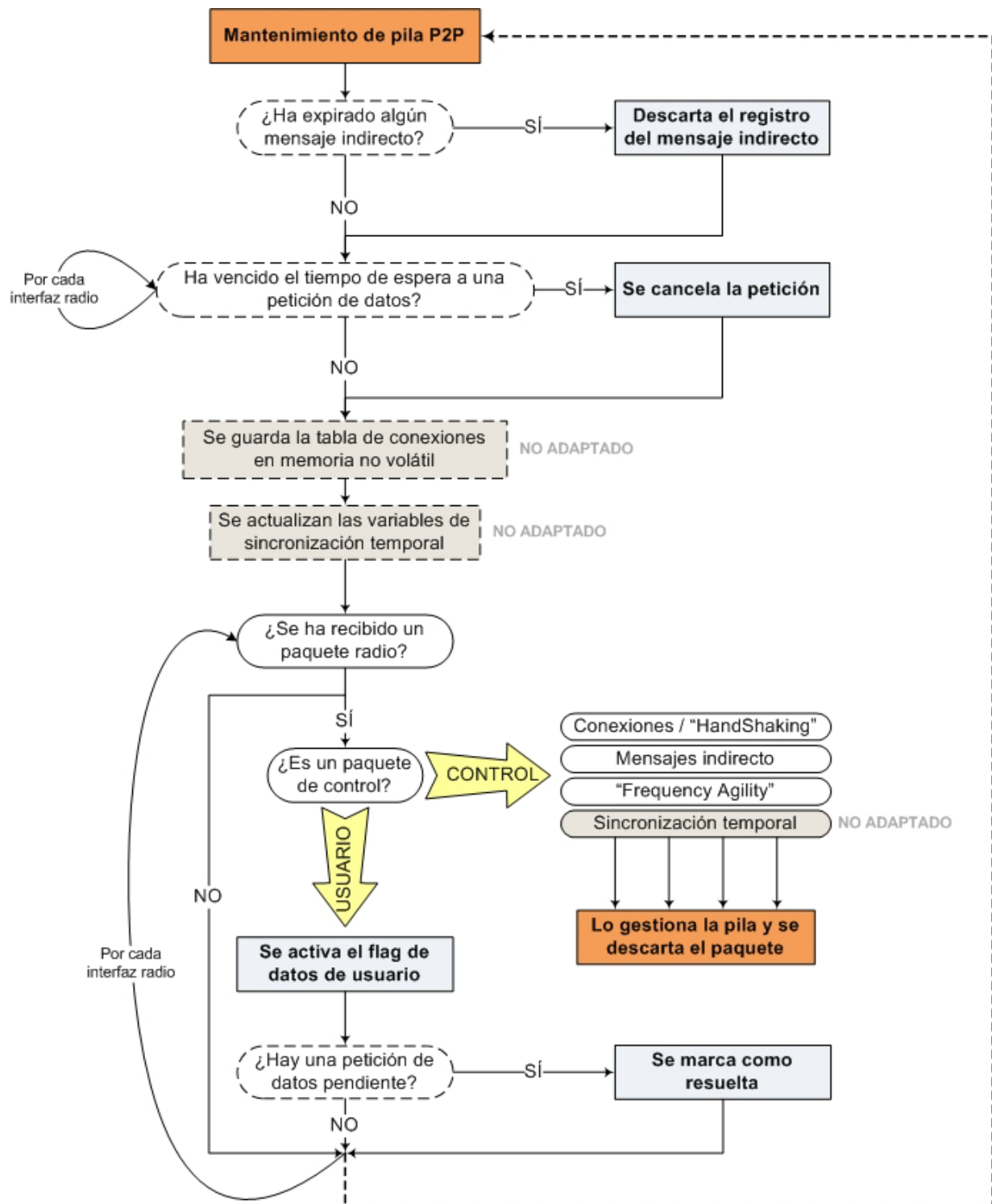


Figura 17: Mantenimiento de la torre de protocolos de MiWi P2P™

5.3.4. - Opciones de depuración

La torre de protocolos implementa funciones para volcar trazas e información por la UART a un hiperterminal. Esto ha sido de gran ayuda para probar la adaptación y depurar los errores que se han detectado durante el desarrollo de la nueva pila de protocolos y la capa de abstracción hardware.

Las funciones asociadas al envío de trazas se han reorganizado, definiendo algunas macros y flexibilizando la configuración del módulo UART a emplear de manera similar a como se hizo con las interfaces SPI de los transceptores. De este modo, el firmware no requiere ser modificado para manejar otra UART, sino simplemente la configuración.

5.4. – CAPA DE ABSTRACCIÓN HARDWARE

En la sección anterior se ha explicado con detalle la torre de protocolos de MiWi adaptada para admitir hasta tres transceptores en bandas diferentes, comenzando por el nivel físico hasta llegar al nivel de red. Siguiendo ese orden creciente en el nivel de abstracción, llegamos a la capa de abstracción hardware.

Con la implementación de la capa de abstracción hardware (HAL, del inglés, Hardware Abstracción Level) se pretende aislar al nivel de aplicación de la complejidad de los protocolos y transceptores utilizados y, al mismo tiempo, ofrecer una interfaz de programación de alto nivel que permita desarrollar aplicaciones de manera sencilla, aportando la máxima funcionalidad que ofrecen los niveles inferiores.

5.4.1. - Opciones de configuración

Al margen de las múltiples opciones que ofrecen los protocolos, la capa de abstracción ofrece algunas opciones adicionales fruto de la adaptación que se ha hecho de la torre de protocolos de MiWi. Además, otro de los objetivos es que el nodo sea flexible y reconfigurable para que admita modificaciones y extensiones fácilmente, por lo que existen opciones a tal efecto.

Opción de configuración	Descripción
MIWI_0434_RI MIWI_0868_RI MIWI_2400_RI	Estas son las opciones que definen las interfaces presentes en el nodo. Deben de ser coherentes con los transceptores definidos en ConfigApp.h y su configuración concreta en ConfigTransceivers.h. En caso contrario, el pre-compilador notificará un error.
MRF49XA_1_IN_SPIx MRF49XA_2_IN_SPIx MRF24J40_IN_SPIx	Mediante un conjunto de definiciones para cada transceptor, se puede seleccionar el módulo SPI (con 'x' igual a 1, 2, 3 o 4) que utiliza cada transceptor en el diseño hardware. El firmware configura entonces esos módulos debidamente.
MRF49XA_1_USES_INTx MRF49XA_2_USES_INTx MRF24J40_USES_INTx	De la misma manera que con la selección de la interfaz SPI, se puede seleccionar la línea de interrupción externa que utiliza cada transceptor (x entre 0 y 4). De esta manera se puede alterar el diseño sin tener que reconfigurar el firmware y la configuración de las interrupciones.
NODE_DOES_MAINTENANCE_TASKS	Habilita el mantenimiento automático de la pila por parte del nodo. El periodo es regulable desde decenas de microsegundos hasta casi un segundo. Si se deshabilita, la aplicación debe llamar a la función de mantenimiento con la mayor frecuencia posible.
WAKE_FROM_SLEEP_WDT	Para utilizar el WatchDog Timer (WDT) o el timer1 y el oscilador

y WAKE_FROM_SLEEP_SOSC_T1	secundario (SOSC_T1) para controlar el tiempo que el microcontrolador esté dormido. Hay que elegir una de las dos. Se comentará con más detalle al ver la función de la HAL de dormir el microcontrolador.
DEBUG_UARTx	Permite seleccionar la UART (1 - 6) que utiliza el diseño para mandar trazas de depuración al PC. Debe habilitarse la funcionalidad de la pila que permite generar las trazas para poder seleccionar un módulo.
ENABLE_DUMMY_BYTE	Es la opción para solventar el problema detectado en el envío de datos en el protocolo MiWi escribiendo un primer byte distinto de cero. Este primer dato se desecha inmediatamente al recibirlo. Si se deja deshabilitada, la HAL reporta un error al intentar escribir el primer byte de un paquete de usuario con valor cero.

Tabla 12: Opciones de configuración de la HAL

Todas estas opciones se encuentran en el archivo de configuración hardware (HardwareProfile.h), salvo la última de ellas, que se ha incluido en ConfigApp.h junto con las opciones de configuración de la torre de protocolos.

5.4.2. Interfaz de programación de alto nivel

En las siguientes páginas se explica toda la interfaz de aplicación que se ha desarrollado sobre la torre de protocolos para poder utilizar de manera sencilla todas sus funcionalidades, recogida en la cabecera "NodeHAL.h".

5.4.2.1. Definición de variables.

En este apartado se explican brevemente las variables que se han definido para gestionar todas las interfaces radio y ofrecer el servicio al nivel de aplicación.

Selección de interfaz radio

A lo largo de los diferentes niveles de abstracción se ha ido detallando cómo se exportaba la posibilidad de seleccionar la interfaz radio hacia los niveles superiores de la pila. En la capa de abstracción hardware, una vez más, se ofrece esa posibilidad a la aplicación, pero generalizando las interfaces presentes en el nodo por si se incluyen nuevas pilas de protocolos.

Para ello se ha definido un tipo enumerado "radioInterface" que se emplea en las funciones de la HAL para seleccionar la interfaz radio con la que se quiere realizar la operación que se solicite e interactuar con el nivel de aplicación. Las opciones que se han definido son: "NONE" (ninguna interfaz), "MIWI_0434", "MIWI_0868", "MIWI_2400" y "ALL_MIWI" (todas las interfaces MiWi disponibles). En caso de invocar alguna función refiriéndose a una interfaz que no esté incluida en el nodo, la pila notificará el error por pantalla (si se habilitan las trazas), devolverá un código de error y/o ignorará la invocación.

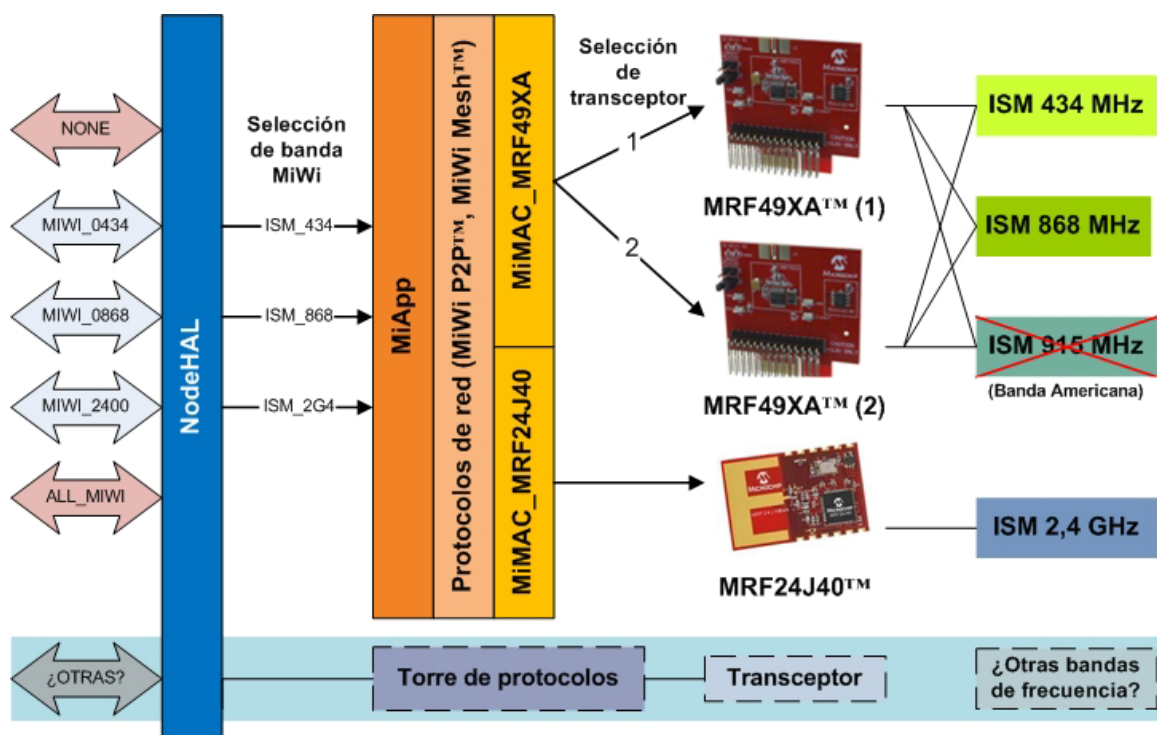


Figura 18: Selección de interfaces radio en la HAL

Estado del nodo

Se define una estructura de datos para recoger el estado del nodo. En este caso, la definición de esta estructura es diferente en función de las interfaces incluidas en el nodo. Se distinguen las siguientes partes:

- **Flags de estado**, unificados en el byte “StatusFlags”. Los flags indican si las interfaces están encendidas, si el nodo está dormido y si el cifrado de paquetes está habilitado. Los flags de interfaces no incluidas no se definen.

Flags de estado								
	Nibble de estado general				Nibble de interfaces radio			
Bits	7	6	5	4	3	2	1	0
Flags	SecEN	Asleep	Reservado	Reservado	Reservado	MiWi2G4	MiWi868	MiWi434

Tabla 13 : Flags de estado en la HAL

- **Estado de las interfaces.** Por cada interfaz se almacena el canal en el que se está operando, los resultados del escaneo (RSSI relativo al nivel del ruido) e indicadores de buffer de transmisión lleno y buffer de transmisión vacío.
- **Estadísticas.** El nodo registra el número de paquetes que se han enviado y recibido con éxito por cada interfaz radio. Estos parámetros han sido útiles en la fase de pruebas.

5.4.2.2. - Códigos de Error.

La gran mayoría de las funciones de la capa de abstracción Hardware (HAL) informan del resultado de su ejecución mediante un valor de retorno de 1 byte de extensión. Este hecho permite que la aplicación sea consciente de los errores que ocurran durante el transcurso de determinadas acciones y que pueda actuar en consecuencia.

Los códigos de error siempre tienen un '1' en el bit más significativo que les distingue de los valores válidos. Esto es, la HAL admite hasta un máximo de 128 códigos de error, correspondientes a los valores decimales desde 128 a 255, mientras que los valores de retorno desde 0 a 127 se interpretarán como válidos.

Aquellas funciones estipuladas para obtener el valor de una variable, bien facilitan una dirección para guardar dicha variable en ella, bien la variable nunca excede el valor 127 o queda limitada en este umbral para no interferir con los códigos de error. Por el contrario, si el valor de retorno de una función simplemente informa del resultado de la acción, el valor 0 siempre se utiliza para indicar que no han ocurrido incidencias durante la ejecución.

A continuación se muestran los códigos de error definidos en la capa de abstracción hardware y su significado. Los códigos de error no definidos se reservan para futuras extensiones de la HAL.

Error	Código de error		Descripción
	Hex.	Dec.	
NO_ERROR	00	0	Éxito.
ERRORES DE SELECCIÓN DE INTERFAZ RADIO			
UNKNOWN_INTERFACE_ERROR	80	128	No se reconoce la interfaz.
INVALID_INTERFACE_ERROR	81	129	La interfaz no es válida en la función invocada.
UNAVAILABLE_INTERFACE_ERROR	82	130	El nodo no dispone de la interfaz.
ASLEEP_INTERFACE_ERROR	83	131	La función requiere que la interfaz esté operativa.
ERRORES DE TRANSCÉPTORES			
TRANSCEIVER_PM_ERROR	90		Fallo general al dormir o despertar una interfaz.
DATA_REQUEST_TX_ERROR	91		Fallo al enviar una petición de datos.
DATA_REQUEST_RX_ERROR	92		No se recibió respuesta a una petición de datos.
INVALID_POWER_MODE_ERROR	9F		El modo de operación es inválido o desconocido.
ERRORES DE RADIOCOMUNICACIÓN , ENVÍO Y RECEPCIÓN DE DATOS			
TX_BUFFER_FULL_ERROR	A0		No se puede escribir en el buffer por estar lleno.
RX_BUFFER_EMPTY_ERROR	A1		No se puede leer datos del buffer al estar vacío.
BROADCAST_TX_ERROR	B0		Error en una transmisión de tipo broadcast.
UNICAST_MIWILONGADDR_ERROR	B1		Error en una transmisión unicast con una dirección larga de MiWi.
UNICAST_MIWISHORTADDR_ERROR	B2		Error en una transmisión unicast con una dirección corta de MiWi.
UNAVAILABLE_RX_ADDR_ERROR	BE		No se dispone de un paquete para leer su dirección origen.

INVALID_ADDR_MODE_ERROR	BF		La dirección no es de ningún tipo válido.
ERRORES CON PARÁMETROS DE RADIOFRECUENCIA			
INVALID_CHANNEL_ERROR	C0		No existe el canal o es un canal no permitido.
UNAVAILABLE_RSSI_ERROR	C1		No se puede leer el RSSI de un paquete recibido.
UNAVAILABLE_LQI_ERROR	C2		No se puede leer el LQI de un paquete recibido.
INVALID_SECLEVEL_ERROR	D0		Nivel de seguridad no definido.
ERRORES AL SALVAR Y RESTAURAR VARIABLES DE ESTADO			
CONNTABLE_EXCEEDED_ERROR	E0		Tamaño de la tabla de conexiones insuficiente.
ERRORES GENERALES DE MÓDULOS O DE LA PROPIA PILA			
MIWI0434_STACK_ERROR			Fallo de la pila de MiWi en la interfaz de 434 MHz.
MIWI0868_STACK_ERROR			Fallo de la pila de MiWi en la interfaz de 868 MHz.
MIWI2400_STACK_ERROR			Fallo de la pila de MiWi en la interfaz de 2,4 GHz.
WIFI_STACK_ERROR			Reservado para futuro uso
UART_ERROR			Reservado para futuro uso
USB_ERROR			Reservado para futuro uso
SPI_ERROR			Fallo de algún módulo SPI
I2C_ERROR			Reservado para futuro uso
INVALID_INPUT_ERROR			Algún parámetro de entrada no es válido.
ASLEEP_NODE_ERROR			No se puede realizar la operación. El nodo está dormido.
HAL_INTERNAL_ERROR			Error interno de la capa de abstracción hardware.
UNKNOWN_ERROR			Reservado para futuro uso. (Fatal Error).

Tabla 14: Códigos de error de la HAL

5.4.2.3. - Funciones de la interfaz de aplicación

En esta sección se explican las funciones de la HAL y algunos de sus detalles de funcionamiento, sin dar una explicación detallada del código. Para más información sobre la interfaz de programación, consulte el anexo.

Función de Inicialización

➤ **InitNode** - Inicializar todas las funcionalidades del nodo.

Esta función está pensada para ser ejecutada una única vez al comienzo del código de aplicación. Es la encargada de inicializar todas las variables internas de la capa de abstracción hardware, el estado del nodo, las interfaces de comunicación con los periféricos y los módulos que emplea el sistema (SPIs, UART, Timers, interrupciones...).

Una vez que el hardware y las interfaces de comunicación están listas, se habilitan las interrupciones para poder inicializar las interfaces radio presentes en el nodo. Esto implica inicializar el protocolo y los transceptores (sus variables de control), buscar redes existentes o crear una nueva red, conectarse a una red si procede (obteniendo una dirección de red en el caso de del protocolo MiWi) y establecer conexiones con otros dispositivos. Tras esta fase, se habilitan las tareas de mantenimiento de la pila de protocolos.

Funciones de Gestión de Energía

Con ellas podemos activar los modos de bajo consumo de los transceptores y el microcontrolador, estrategias fundamentales para reducir el consumo energético del nodo.

- ***SleepRadioInterface*** - *Dormir/Apagar una interfaz radio.*
- ***WakeUpRadioInterface*** - *Despertar/Encender una interfaz radio.*

Estas funciones permiten dormir y despertar las interfaces radio del nodo.

Al dormir una interfaz radio, el nodo no puede recibir datos a través de ella. Ni siquiera podrá detectar que llegan a la antena para generar una interrupción que avise al microcontrolador, salvo que sea un transceptor con capacidades para ello (a esta habilidad se le denomina Wake-On-Radio)¹⁰. Es el microcontrolador quien puede despertar a una interfaz dormida, bajo petición de la aplicación.

- ***SleepNode*** - *Dormir el microcontrolador y, opcionalmente, interfaces radio.*

La función SleepNode es la encargada de llevar al microcontrolador al modo dormido (Sleep), en el que se detiene el reloj del sistema y por tanto, la gran mayoría de los periféricos. Solo se ha implementado este modo de bajo consumo, pues es el de menor consumo de todos los posibles y en consecuencia, el más interesante para ahorrar energía.

Un detalle importante es que no existe la función de despertar al nodo. Esto es debido a que, durante el modo sleep, la CPU está detenida y no se decodifican nuevas instrucciones, por lo que la aplicación no puede hacer nada hasta que el microcontrolador despierte. Éste volverá a su estado anterior al producirse una interrupción que le saque del modo sleep. Al invocar la función, se indica qué interfaces mantener despiertas para que puedan despertar al nodo en caso de recibir datos.

Para dotar al nodo de la posibilidad de dormir el microcontrolador un intervalo de tiempo específico, se puede emplear el timer 1 o el WatchDogTimer (seleccionable mediante la opción de configuración de la HAL). Siempre se indica en la invocación un tiempo máximo. Transcurrido ese tiempo, el nodo sale automáticamente del modo sleep, salvo que otra fuente de interrupción anticipe este suceso. Si el tiempo máximo es muy largo y excede el desbordamiento del temporizador empleado, el nodo vuelve a dormir hasta alcanzar el límite impuesto por la aplicación.

Por la arquitectura del PIC, el timer1 sólo puede utilizar el reloj secundario externo como reloj, pues en modo sleep se detiene el reloj del bus de periféricos. En cambio, el WatchDog Timer (WDT) utiliza el oscilador interno de 32 kHz. Las ventajas de una alternativa son los inconvenientes de la otra, y viceversa.

El timer 1 permite regular mejor el tiempo, pero requiere disponer de un oscilador adicional, típicamente de 32768 Hz. Para esa frecuencia, la HAL puede configurar el periodo de desbordamiento del timer1 entre 1 ms y 512 segundos. Por contra, el WDT

¹⁰ NOTA AL PIE WAKE ON RADIO??

tiene un periodo de desbordamiento fijo (aunque configurable al cargar el programa en el PIC) desde 1 ms hasta casi 17,5 min (en potencias de 2).

Si se quiere dormir indefinidamente el nodo para que solo despierte en caso de producirse una interrupción, basta con indicar un valor extremo del tiempo en milisegundos. Al utilizar un entero de 32 bits para contar periodos de modo sleep, el rango temporal es desde 1 ms hasta decenas de años. No obstante, se ha utilizado un pulsador para poder abortar el modo sleep manualmente.

Funciones de Envío y Recepción (radiocomunicación)

- **PutTXData** - *Escribe un byte de datos en el buffer de transmisión de una interfaz.*
- **GetFreeTXBufSpace** - *Devuelve la capacidad libre del buffer de transmisión.*

Para poder escribir la carga útil de un paquete, se hace uso de estas funciones. Antes de que la aplicación trate de escribir datos nuevos, es conveniente que se cerciore de que hay espacio libre en el buffer en el buffer de transmisión invocando a la segunda función.

- **DiscardTXData** - *Descarta el paquete a transmitir.*
- **SendPckt** - *Envía el paquete guardado en el buffer de transmisión al nodo destino.*

Una vez que se ha comenzado a llenar el buffer, la aplicación puede lanzar la transmisión de un paquete radio (facilitando una dirección y el tipo de la misma) o descartar los datos del buffer para escribir un nuevo paquete.

Si se envía el paquete correctamente se descartarán los datos para preparar la transmisión de nuevos datos. En caso contrario, no se descartan, permitiendo retransmitir el paquete que se quería enviar inicialmente.

Para indicar el tipo de dirección, en la HAL se definen los siguientes modos:

- **BROADCAST_ADDRMODE**: Transmisión broadcast en la interfaz seleccionada.
- **LONG_MIWI_ADDRMODE**: Unicast en una interfaz MiWi con una dirección larga o permanente.
- **SHORT_MIWI_ADDRMODE**: Unicast en una interfaz MiWi con una dirección corta o de red.

El tamaño del buffer de transmisión lo fija la torre de protocolos en sus opciones de configuración

- **WhichRIHasData** - *Informa a la aplicación de qué interfaces tienen datos.*
- **GetPayloadToRead** - *Devuelve cuántos datos están pendientes de leer en una interfaz.*

La aplicación debe comprobar de manera frecuente si hay paquetes recibidos que procesar. Para ello, debe invocar a la primera de estas funciones, que informa de todas las

interfaces que tienen datos pendientes de ser leídos. Para que la aplicación pueda saber el tamaño de la información que queda por procesar, dispone de la segunda función.

- **GetRXData** - Lee un byte de datos del buffer de recepción de una interfaz
- **GetRXSourceAddr** - Obtiene la dirección de origen de un paquete recibido.

Para acceder a la carga útil de un paquete, la aplicación lee un byte mediante la función GetRXData. Corre a cargo de la aplicación almacenar los datos, pues al invocar la función repetidas veces se accede a los datos sucesivos, sin poder leer los anteriores. Una vez que se han leído todos los datos, se descarta el paquete para poder recibir nuevos paquetes por esa interfaz.

Mientras esté en proceso un paquete, la aplicación puede leer la dirección origen del paquete. Esta acción, por tanto, debe realizarse antes de leer el último byte de carga útil.

Funciones de Seguridad

- **SetSecurityLevel** - Cambia el modo de seguridad, habilitando o deshabilitando el cifrado de paquetes.

En función del nivel de seguridad, el comportamiento del nodo en determinadas acciones será diferente. Dicho nivel de seguridad se establece mediante esta función.

Por ahora, sólo se han implementado 2 modos, reservando la posibilidad de incluir otros nuevos como mejora. El modo por defecto es el modo sin seguridad (modo 0), en el que no se emplea ningún mecanismo del nodo. El modo 1 habilita el cifrado de los paquetes, tal y como se hayan configurado los transceptores para hacerlo.

La aplicación puede comprobar si el cifrado de paquetes está habilitado en el byte de flags de estado.

Funciones de Radiofrecuencia

- **SetChannel** - Establece el canal radio de una interfaz.
- **GetOpChannel** Establece el canal radio de una interfaz.

Estas funciones permiten cambiar o leer el canal de operación dentro de una interfaz radio. Los canales de una interfaz se numeran del cero en adelante por simplicidad, para que los canales válidos sean todos los enteros positivos menores que la constante que define el número de canales de cada interfaz.

Esto implica que los 16 canales de 2,4 GHz que define el estándar 802.15.4 se numeran del 0 al 15, y no desde el 11 al 26. Si se desea volver a la numeración que fija el estándar, basta redefinir la constante "MIWI2400ConfChannelOffset" con el valor cero en la cabecera de la HAL (lo que implica que los canales válidos serían del 11 al 26).

- ***SetTXPower*** - *Cambia el valor de la potencia de transmisión en una interfaz.*
- ***GetTXPower*** - *Obtiene el valor actual de la potencia de transmisión de la interfaz.*

Estas funciones permiten manipular y comprobar la potencia de transmisión de un interfaz radio. Como los transceptores solo admiten un conjunto discreto de valores, se han diseñado las funciones para comprimir un rango continuo de un byte en la HAL (256 valores) a los 8 y 32 valores posibles del MRF49XA y MRF24J40 respectivamente.

Los saltos de potencia están equiespaciados en la HAL, aunque no corresponden con saltos lineales en el nivel de potencia. Siempre el valor 0 es el valor máximo de potencia y el 255, el valor mínimo.

Al leer el valor de la potencia de transmisión de una interfaz se obtiene el valor de potencia escalado con los 256 valores posibles en la HAL, para que la aplicación sea consciente del margen que tiene disponible hasta el máximo o el mínimo permitido.

- ***DoChannelScanning*** - *Realiza el escaneo de nivel de ruido.*
- ***GetScanResult*** - *Obtiene el nivel de ruido del último escaneo en un canal.*

Estas funciones permiten realizar el escaneo del nivel de ruido en todos los canales de una determinada interfaz y obtener los resultados del último escaneo bajo demanda. Los valores del RSSI relativos al nivel de ruido de cada canal permanecen almacenados en la HAL y se sobrescriben en la siguiente invocación de la función de escaneo.

La función de escaneo devuelve el canal con menor ruido de la interfaz seleccionada y el valor del RSSI del nivel de ruido en dicho canal, aunque si se solicita un escaneo en todas las interfaces devuelve el canal de la interfaz radio con menor ruido global y el identificador de esta interfaz.

- ***GetRSSI*** - *Lee el valor del indicador de intensidad de señal recibida (RSSI) del paquete recibido.*
- ***GetLQI*** - *Lee el valor del indicador de calidad del enlace (LQI) del paquete recibido.*

Mientras se procesa un paquete, se puede acceder a los parámetros indicativos de la calidad del enlace (LQI) y la intensidad de la señal recibida (RSSI) mediante estas funciones.

Funciones de Restauración de variables

- ***SaveConnTable*** - *Ofrece a la aplicación guardar la tabla de conexiones.*
- ***RestoreConnTable*** - *Permite que la aplicación restaure la tabla de conexiones.*
- ***SaveRoutingTable*** - *Si se emplea el protocolo MiWi, permite guardar la tabla de encaminamiento.*
- ***RestoreRoutingTable*** - *Complementa a la función anterior, restaurando la tabla de encaminamiento.*

Estas funciones suponen la alternativa a la funcionalidad de la torre de protocolos de MiWi de almacenamiento no volátil. De esta manera, la aplicación puede acceder a información sensible de los protocolos, como es la tabla de conexiones y la tabla de encaminamiento (sólo disponible en el protocolo MiWi, ya que P2P no implementa funciones de encaminamiento) y restaurarla en caso de necesidad.

Modificar y restaurar las variables de los protocolos puede conducir a un mal funcionamiento de la red, especialmente si se alteran las direcciones del nodo, por lo que es preciso manejar estas funciones con cautela.

Función de Mantenimiento de la torre de protocolos.

- ***AllStacksTasks*** - Realiza el mantenimiento de la pila de protocolos.

Esta función solo está disponible si se habilita la opción de la HAL en la que la aplicación se hace cargo del mantenimiento de la torre de protocolos, en cuyo caso debe invocar la función con la mayor frecuencia posible.

Funciones auxiliares y de depuración

- ***DumpRXPckt***- Vuelca el contenido de un paquete radio por pantalla.
- ***DumpConnTable*** - Imprime la tabla de conexiones por pantalla.
- ***DumpSingleConn***- Imprime la conexión solicitada por pantalla.

Este grupo de funciones permiten mostrar por pantalla el contenido de algunas variables muy útiles de la torre de protocolos como los paquetes radio recibidos y la tabla de conexiones. Para ello es necesario que se habiliten las opciones de depuración: `ENABLE_CONSOLE`, para enviar trazas por la UART, y `ENABLE_DUMP`, para disponer de las funciones que muestran la tabla de conexiones de MiWi.

- ***SWDelay***- Introduce un retardo software.

Se recomienda no abusar de esta función, ya que consume ciclos de CPU.

- ***GetSentPkts*** - Lee el valor de paquetes enviados con éxito por una interfaz.
- ***GetProcPkts*** - Lee el valor de paquetes procesados con éxito por una interfaz.

Estas funciones permiten acceder al valor guardado internamente en la HAL de los paquetes enviados o procesador por una interfaz radio.

6. TEST Y EVALUACIÓN

Para probar la adaptación de la torre de protocolos y la implementación de la capa de abstracción hardware se ha realizado una batería de pruebas extensa. En esta sección se explica, en primer lugar, la plataforma de pruebas que se ha empleado, y a continuación se comentan las pruebas realizadas en detalle, justificando los resultados obtenidos.

6.1. BREVE DESCRIPCIÓN DE LA PLATAFORMA DE PRUEBAS

Para poder depurar errores y comprobar el correcto funcionamiento del firmware y la HAL, antes de diseñar y probar un prototipo del NGD se ha decidido emplear el nodo FCD, alterando su diseño ligeramente. Para ello, Guillermo Jara Luengos ha desarrollado paralelamente a la realización de este proyecto una placa de expansión al nodo FCD dentro de su proyecto fin de carrera, también enmarcado dentro de la temática de las redes de sensores inalámbricas cognitivas.

En ese sentido la gran mayoría de las características de la plataforma de prueba son las del nodo FCD que se analizaron en la sección 3.2. No obstante, gracias a la adaptación mencionada, se ha sustituido el módulo transceptor de Wi-Fi (MRF24WB0MA) del diseño del FCD por la placa de expansión. Ésta permite conectar, bien el transceptor eliminado, bien la placa comercializada por Microchip (MRF49XA PICtail™/PICtail Plus Daughter Board) que contiene el transceptor MRF49XA.

El transceptor MRF49XA permite transmitir y recibir paquetes en las banda de 434 MHz o en la banda de 868 MHz, en función del circuito de adaptación de antena y la antena que incorpore. La placa de expansión multiplexa las líneas de entrada y salida de propósito general y la interfaz de comunicación SPI del transceptor de Wi-Fi eliminado con las que emplea el módulo. La conexión del transceptor escogido se realiza mediante puentes ("jumpers", en terminología inglesa), evitando que varios dispositivos de la placa de expansión intenten hacer uso del mismo pin.

Para el desarrollo del firmware, el hecho de poder probar las funcionalidades de los protocolos con ambos transceptores en el diseño ha sido esencial, y gracias a esta extensión hardware, se ha reducido el tiempo y el coste de desarrollo.

Lamentablemente, debido a que el transceptor Wi-Fi solo empleaba una interfaz SPI, solo se puede incluir un único transceptor MRF49XA, no los dos que se pretenden incluir en el futuro prototipo del nodo NGD. Por este motivo, las pruebas del firmware se han realizado con dos interfaces, alternando el MRF49XA con la antena a 434 MHz con el de 868 MHz. Pese a todo, para probar la adaptación de la torre de protocolos, dos interfaces radio son suficientes, ya que permite comprobar la compartición de recursos y variables (tabla de conexiones, etc.).

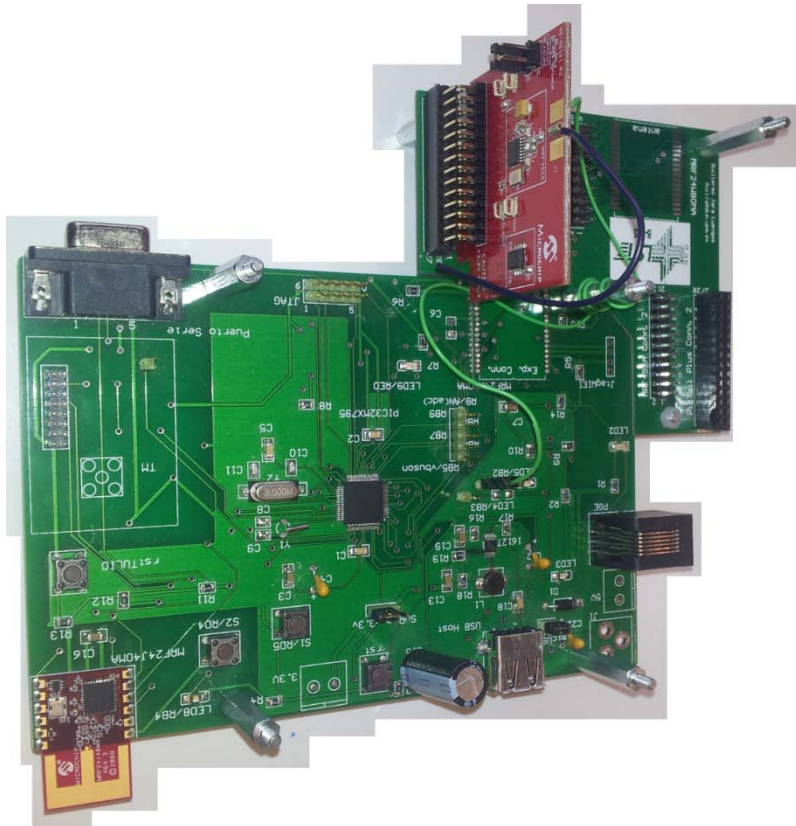


Figura 19: Foto de la plataforma de prueba o nodo FCD adaptado.

A modo resumen, se listan los principales elementos del diseño FCD adaptado:

- Microcontrolador PIC32MX795F512H con 512 kB de memoria flash y 128 kB de RAM.
- Dos cristales a 8 MHz y 32,768 kHz para generar el reloj del sistema a 80 MHz e implementar funciones de reloj de tiempo real.
- Módulo transceptor MRF24J40MA con antena integrada en PCB para la interfaz de MiWi™ a 2,4 GHz, compatible con el estándar IEEE 802.15.4 para LR-WPANs.
- Placa de extensión en sustitución del transceptor de Wi-Fi para conectar un módulo MRF49XA PICTail™/PICTail Plus Daughter Board.
- Módulo transceptor Tulio a 868 MHz. Este módulo ha sido desconectado y no ha sido portado al nuevo firmware.
- Posibilidad de volcar trazas a un PC a través de la UART mediante un conversor a RS-232 MAX3232 de Maxim incluido en el diseño.
- Conector USB y microUSB con vistas a volcar trazas o alimentar el nodo mediante USB.
- Regulador de Microchip para convertir una tensión no regulada de 5 V a una tensión regulada de 3,3 V necesaria en el circuito.
- Puerto de programación para utilizar un depurador de Microchip y cargar los programas.

- Pulsadores de reset y de propósito general conectados, éstos últimos conectados a entradas del microcontrolador que soportan las funciones de detección de cambios en la misma.

6.2. PRUEBAS Y EVALUACIÓN.

6.2.1. Pruebas funcionales

Se trata de un conjunto de pruebas unitarias del uso de las funciones de la HAL sobre la torre de protocolos adaptada que evalúan la ejecución de las acciones solicitadas o la notificación de códigos de error. Se comentan las pruebas por los bloques de las funciones de alto nivel presentadas en la sección 5.4.2.3.

Función de Estado

Para la función de leer los flags de estado (GetStatusFlags), se ha comprobado que la HAL registra satisfactoriamente los cambios en cada una de las situaciones que implican una actualización de esta variable de control.

Función de Inicialización

La función InitNode ha sido probada para inicializar las interfaces del nodo en diversas situaciones, cubriendo los dos protocolos de red, las tres interfaces radio (alternando las interfaces de 434 y 868 MHz), las distintas combinaciones dos a dos de roles y tipos de nodos, y diferentes anchos de banda del transceptor MRF49XA. Esta función incluye:

- El autodescubrimiento de redes válidas o la creación de la red en el caso del PANC.
- El proceso de handshaking para unirse a una red existente y la obtención de una dirección de red en el caso del protocolo MiWi.
- La actualización de la tabla de conexiones una vez se ha obtenido respuesta del dispositivo asociado.

Funciones de Gestión de Energía

En cuanto a las funciones de dormir y despertar una interfaz, se ha comprobado el funcionamiento de las mismas y se han realizado algunas pruebas unitarias, como por ejemplo invocar repetidas veces la función de dormir una interfaz cuando ésta ya está dormida, invocarla con una interfaz no incluida en el nodo, etc.

No obstante, la función SleepNode ha tenido un peso mayor en las pruebas, por su complejidad e importancia en el diseño. Se han realizado pruebas para despertar al nodo de manera automática con el Timer1 y con el WatchDog (según la opción de configuración de la HAL) y de manera manual mediante un pulsador conectado a una entrada de propósito general (generando una interrupción por cambio en la entrada).

En el caso del WDT se han testado diferentes configuraciones del divisor del módulo Watchdog (influye en el rango de tiempos que puede dormir el nodo) con diferentes tiempos, comprobando que, si es necesario, nada más despertar el nodo puede volver a dormirse para cubrir el tiempo indicado por la aplicación.

Por el contrario, el modo sleep haciendo uso del Timer1 ha dado problemas. Tras un intenso proceso de depuración, se ha descubierto que es un problema hardware relacionado con el oscilador secundario a 32,768 kHz y las capacidades empleadas en el diseño. Al medir con la sonda del osciloscopio el reloj se introducía una capacidad parásita que hacía que oscilase correctamente, no obstante, el funcionamiento del timer 1 seguía siendo extraño. En el prototipo hardware del nodo NGD habría que probar esta funcionalidad.

Funciones de Envío y Recepción (radiocomunicación)

La transmisión de paquetes se ha testado con todo tipo de detalles, tanto dirigida a dispositivos RFD como FFD. Se han utilizado todos los modos y direcciones disponibles en cada protocolo para comprobar la utilización de las mismas, así como otras funcionalidades de la pila (tabla de conexiones, etc). También se ha probado el caso más básico de encaminamiento en MiWi (transmisión de un coordinador a un dispositivo asociado).

Se ha evaluado el funcionamiento de las funciones de control del buffer de transmisión enviando paquetes con la máxima y mínima carga útil. También se ha comprobado la opción de configuración `ENABLE_DUMMY_BYTE` que corregía el problema del protocolo MiWi comentado en el capítulo 5.

En cuanto a las funciones de la HAL relacionadas con la recepción, se han realizado pruebas para comprobar que se recibía la cantidad transmitida de datos por la interfaz utiliza. Se ha comprobado la transmisión sin errores y la veracidad de la dirección de origen.

Funciones de Seguridad

El modo de seguridad del nodo se configura correctamente mediante la función definida en la HAL y permite la transmisión y recepción de mensajes cifrados a través de todas las interfaces.

Funciones de Radiofrecuencia

Aparte de las múltiples pruebas funcionales que se han realizado, las pruebas de estas funciones contemplan muchos casos de error notificados por la HAL, como el cambio a un canal no válido, la lectura del valor de LQI o del RSSI cuando no se ha recibido un paquete, etc.

En el caso de las funciones que manipulan la potencia de transmisión se han realizado pruebas de alcance para comprobar que, a medida que se transmiten paquetes con menor potencia, dos nodos que están separados dejan de recibir los mensajes que se envían.

Las capacidades de escaneo del espectro del transceptor MRF49XA son algo mediocres, ya que los valores del RSSI se mapean a un conjunto reducido de valores discretos.

Restauración de variables

Para las funciones de salvar y restaurar variables de estado se han realizado pruebas enfocadas a comprobar únicamente su funcionamiento, ya que a priori son funciones que se emplearán con muy poca frecuencia, en condiciones especiales.

Mantenimiento de la torre de protocolos

No se han realizado pruebas como tal, sino que la adaptación de la torre de protocolos ha ido requiriendo paulatinamente modificaciones y la evaluación de la función de mantenimiento de la torre de protocolos. Otras funciones de la HAL, como la recepción de paquetes, depende del comportamiento de esta función.

Funciones auxiliares y de depuración

Estas funciones han sido utilizadas como herramienta en el proceso de prueba y depuración del firmware y la capa de abstracción hardware. La mayoría de ellas son simples macros que invocan a funciones de la torre de protocolos de MiWi, por lo que prácticamente no ha sido necesario probarlas.

6.2.2. Pruebas de estrés y caracterización.

A continuación se explican una serie de pruebas enfocadas a evaluar algunos casos límite en el funcionamiento del nodo o medir prestaciones generales del mismo.

6.2.2.1. Capacidad de procesamiento del microprocesador.

Descripción

Esta prueba pretende medir cuánta capacidad de procesamiento demanda el mantenimiento de la torre de protocolos para evaluar si puede obstaculizar la ejecución de otras tareas de nivel de aplicación. Para ello, se determinará cuanto tiempo de ejecución requieren las tareas de mantenimiento "en vacío", esto es, sin actividad en la red y sin tareas de aplicación (simplemente, un bucle infinito). Se realizará a la frecuencia nominal de trabajo (80 MHz).

Procedimiento

Para poder medir el tiempo de ejecución se empleará el temporizador interno del microcontrolador (Core Timer), intercalando instrucciones de captura del valor del temporizador en lugares concretos del código. Este temporizador (de 32 bits) se incrementa automáticamente cada 2 ciclos del reloj del sistema, ofreciendo mayor precisión que los módulos temporizadores.

Se medirá el tiempo de mantenimiento de las interfaces de la plataforma de prueba por separado, así como el tiempo común a todas las interfaces (fruto de la adaptación de la

pila de protocolos). Se realizarán dos medidas por cada protocolo testado: una con el MRF49XA en la banda de 434 MHz y otra en la banda de 868 MHz.

Dado que el futuro prototipo NGD incorporará dos interfaces con el transceptor MRF49XA, se extrapolarán los resultados para estimar el tiempo total que requerirán las tres interfaces radio.

Condiciones de medida

Para la realización de las pruebas se ha escogido un periodo de mantenimiento de las pila de protocolos de 25 ms, ya que tras todas las pruebas realizadas a lo largo de la adaptación del firmware se ha determinado que es un valor adecuado para el buen funcionamiento del nodo en condiciones normales.

No obstante, la elección del periodo de mantenimiento repercute fuertemente en las prestaciones de la pila. Un periodo demasiado largo hace que la HAL no actualice el estado de las interfaces rápidamente, retrasando el procesamiento de los paquetes por parte de la aplicación y pudiendo suponer mayor pérdida de paquetes. Además, los periodos muy largos comprometen que ciertos paquetes de control que tienen periodos de validez (asentimientos, respuestas a conexiones entrantes, etc.), no se traten con celeridad. Por el contrario, periodos demasiado cortos "roban" ciclos de CPU a la aplicación sin redundar en una mejora en las prestaciones.

En esta prueba se ha configurado el nodo como FFD (no puede dormir sus interfaces) con el mayor número de funcionalidades habilitadas, salvo aquellas no adaptadas, para requerir mayor tiempo de mantenimiento. Esto implica emplear el rol de coordinador en el protocolo MiWi, ya que el mantenimiento de estos nodos tiene mayor complejidad.

La tabla de conexiones se configura para admitir hasta 16 entradas en la tabla de conexiones. Este tamaño duplica el valor por defecto en la pila de MiWi original.

Resultados

Tarea	Protocolo P2P	Protocolo MiWi
Mantenimiento común a todas las interfaces.	15,1 μ s	34,4 μ s
Mantenimiento de la interfaz a 434 MHz	32,8 μ s	33,4 μ s
Mantenimiento de la interfaz a 868 MHz	32,8 μ s	33,4 μ s
Mantenimiento de la interfaz a 2,4 GHz	11,2 μ s	10,4 μ s
Medida del periodo de mantenimiento de la pila.	25011,1 μ s	25011,2 μ s
ESTIMACIÓN TOTAL (3 interfaces)	92 μs	112 μs

Tabla 15: Tiempo de CPU requerido para el mantenimiento de la pila.

Los resultados muestran que en ausencia de actividad de la red, el mantenimiento de las pilas supone una fracción de tiempo despreciable frente a un periodo de

mantenimiento de 25 ms (menor de un 0,5%). Esto permite reducir el periodo de mantenimiento sin mermar gravemente la capacidad de procesamiento ofrecida al nivel de aplicación.

Se recomienda, por tanto, escoger un periodo de mantenimiento de entre 5 y 30 ms, en función de la complejidad de la aplicación, la topología de la red y la carga de tráfico que se prevea. En caso de que la aplicación realice el mantenimiento, estos tiempos son un indicativo de la frecuencia con la que debe invocar a la función de mantenimiento.

6.2.2.2. Máxima tasa binaria por cada interfaz

Descripción

Se trata de caracterizar la cadencia eficaz (C_{eff}) que se ofrece al nivel de aplicación a través de cada una de las interfaces radio presentes en el nodo y observar las capacidades de envío y recepción del nodo en situaciones con gran carga de trabajo a nivel radio.

Mediante el uso de las funciones de la HAL, se comprobará cuántos paquetes pueden ser enviados por segundo y cuántos de ellos ser procesados en recepción para estimar la tasa binaria real que ofrece el nodo.

Las diferentes pruebas pretenden testear también:

- La influencia de la opción `VERIFY_TRANSMIT` en el transceptor MRF24J40.
- El efecto de disponer de mayor o menor espacio en memoria para almacenar paquetes y procesarlos en cuanto sea posible.
- Las prestaciones que presenta el transceptor MRF49XA seleccionando varias tasas binarias (anchos de banda a nivel radio).
- Los casos con mínima y máxima carga útil (payload), pues la cadencia eficaz depende fuertemente del tamaño de los paquetes enviados.

Procedimiento

Por cada protocolo de red e interfaz radio se realizarán pruebas de envío y recepción de ráfagas de paquetes. Las ráfagas se transmiten dejando un tiempo suficientemente largo entre ellas para que el nodo receptor pueda procesar los paquetes almacenados antes de que llegue una nueva ráfaga.

Se utilizarán variables internas para contabilizar los paquetes enviados y procesados por cada una de las interfaces.

Para el cálculo de la cadencia eficaz se miden los tiempos con el *Core Timer* antes y después de la ráfaga, se normaliza el número de paquetes de una ráfaga (N) con la diferencia de tiempo en segundos (t_r) y se multiplica por el tamaño en bits del *payload* (u), obteniendo la cadencia en bits por segundo (bps).

$$C_{eff}(bps) = \frac{N * u}{t_r}$$

Condiciones de medida

Para esta prueba, el escenario es el siguiente:

- Los nodos transmisor y receptor se encuentran separados 20 cm y transmiten la máxima potencia. De este modo se evita, en la medida de lo posible, pérdida de paquetes que desvirtúen la medida.
- En MiWi se configuran los nodos como coordinadores, ya que la recepción de paquetes requiere procesamiento adicional.
- El nodo transmisor escribe el payload, lo envía al nodo receptor y descarta el paquete para cada uno de los paquetes que envía en un ráfaga. El payload se utiliza para numerar los paquetes e identificarlos en recepción. Las ráfagas empleadas son de 255 paquetes en P2P, y 254 en MiWi para evitar el problema del comienzo del payload por 0 comentado en el capítulo anterior.
- En recepción, se simula que la aplicación guarda los datos para procesar el paquete. Se examina cuántos paquetes han llegado de cada ráfaga y se realiza una media con varias ráfagas para determinar cuántos paquetes se procesan, redondeando el resultado al entero más próximo.
- Se repite cada prueba configurando el nodo para poder almacenar 2, 5 y 10 paquetes por cada interfaz radio (mediante la definición BANK_SIZE de los transceptores). En os resultados, corresponden a las columnas BS 2, BS 5, BS 10.
- El tamaño máximo de un paquete lo determina el protocolo empleado (poseen distinto tamaño de cabecera), la longitud de las direcciones de los nodos y las opciones de cifrado de paquetes. Se ha empleado el tamaño máximo de paquete de acuerdo a la utilización de direcciones de 8 bytes, seguridad habilitada y modo de cifrado CCM (Counter CBC MAC) de 32 bits para el MRF24J40 y de 16 bits para el MRF49XA.

Resultados en 2,4 GHz

Protocolo P2P a 2,4 GHz - MRF24J40 (250 kbps)						
Modo de transmisión	Payload (bytes)	Cadencia eficaz (kbps)	Enviados	Recibidos		
				BS 2	BS 5	BS 10
BROADCAST	90	57,5 / 64,0*	255 (99,9%)	38 (14,9%)	42 (16,5%)	45 (17,6%)
	1	2,21 / 2,32*	255 (99,9%)	39 (15,3%)	42 (16,5%)	46 (18,0%)
UNICAST (DIRECCIÓN LARGA)	90	52,6 / 57,7*	255 (99,9%)	139 (54,5%)	142 (55,7%)	147 (57,6%)
	1	1,68 / 1,71*	255 (99,9%)	51 (20,0%)	53 (20,8%)	57 (22,4%)

Tabla 16: Cadencia eficaz, envío y recepción de mensajes en P2P a 2,4 GHz

Protocolo MiWi a 2,4 GHz - MRF24J40 (250 kbps)						
Modo de transmisión	Payload (bytes)	Cadencia eficaz (kbps)	Enviados	Recibidos		
				BS 2	BS5	BS 10
BROADCAST	79	53,5 / 59,1*	254 (99,9%)	4 (1,6%)	4 (1,6%)	5 (2,0%)
	1	2,03 / 2,11*	254 (99,9%)	2 (0,8%)	2 (0,8%)	2 (0,8%)
UNICAST (DIRECCIÓN LARGA)	79	50,2 / 55,4*	254 (99,9%)	129 (50,8%)	132 (52,0%)	135 (53,1%)
	1	1,65 / 1,75*	254 (99,9%)	49 (19,3%)	51 (20,1%)	56 (22,0%)
UNICAST (DIRECCIÓN CORTA)	79	50,5 / 55,3*	254 (99,9%)	129 (50,8%)	131 (51,6%)	133 (52,4%)
	1	1,70 / 1,75*	254 (99,9%)	50 (19,7%)	52 (20,5%)	56 (22,0%)

Tabla 17: Cadencia eficaz, envío y recepción de mensajes en MiWi a 2,4 GHz

* Deshabilitando la opción `VERIFY_TRANSMIT`

En los resultados podemos ver claramente que el cuello de botella en 2,4 GHz se produce en la recepción de los paquetes. Esto parece lógico, ya que en recepción hay más carga de trabajo que en transmisión y existe una latencia inherente a la actualización del estado en la HAL y el procesamiento de los datos por parte de la aplicación. El caso de broadcast en MiWi merece una mención especial y e abordará más adelante.

La cadencia eficaz depende fuertemente del tamaño de la carga útil, pues por cada paquete siempre se envían los bits de cabecera de los protocolos utilizados. La sobrecarga que imponen éstos (en este escenario) es de 37 octetos en el caso de P2P y 48 octetos en el de MiWi, por lo que los paquetes de longitud mínima tienen muy bajo rendimiento. No obstante, calculando la cadencia a nivel físico en todos los casos se obtiene un valor en torno a los 90 kbps, lo que permite concluir que ése es el límite que impone el microcontrolador y el firmware embebido.

La diferencia en el tamaño de los paquetes también supone una variación en su tiempo de transmisión. Los paquetes pequeños presentan peores cifras, ya que al llegar antes al receptor éste debe almacenarlos (y atenderlos la aplicación) en un periodo de tiempo más corto. En pocas palabras, podríamos decir que se pierden más paquetes por la saturación del receptor.

Se observa que el hecho de disponer mayor espacio en memoria no juega un papel tan determinante en la recepción de los paquetes, aunque mejora ligeramente. Como comentario adicional (no recogido en las tablas de resultados), el hecho de almacenar más paquetes hace que la aplicación los reciba más desordenados, pues los primeros paquetes pueden sufrir esperas mientras se reciben y se tratan paquetes posteriores. A todo esto hay que añadir que este hecho también consume más memoria.

La opción `VERIFY_TRANSMIT` del transceptor MRF24J40 no presenta demasiadas ventajas para paquetes con poco payload, pero en el caso opuesto puede suponer una mejora de la cadencia eficaz de hasta un 11,3%. No obstante también tiene su

contrapartida, ya que al enviar paquetes más rápido y no alterar el proceso de recepción se acusa la saturación del receptor un poco más, obteniendo peores métricas en el lado del receptor.

Resultados en 434 y 868 MHz

Dada la gran similitud de resultados obtenidos (diferencias nimias en todos los resultados), se agrupan los resultados de las interfaces de 434 y 868 en las siguientes tablas.

Ya que el transceptor MRF49XA permite configurar la tasa binaria del canal radio se han realizado medidas con tres configuraciones distintas.

Protocolo P2P a 434 y 868 MHz - MRF49XA (9,6 kbps)						
Modo de transmisión	Payload (bytes)	Cadencia eficaz (kbps)	Enviados	Recibidos		
				BS 2	BS 5	BS 10
BROADCAST	90	8,42	255	255	255	255
	1	0,47	255	173	174	175
UNICAST (DIRECCIÓN LARGA)	90	6,78	255	255	255	255
	1	0,22	255	255	255	255
Protocolo P2P a 434 y 868 MHz - MRF49XA (38,4 kbps)						
Modo de transmisión	Payload (bytes)	Cadencia eficaz (kbps)	Enviados	Recibidos		
				BS 2	BS 5	BS 10
BROADCAST	90	28,7	255	255	255	255
	1	1,29	255	65	67	73
UNICAST (DIRECCIÓN LARGA)	90	20,5	255	255	255	255
	1	0,27	255	255	255	255
Protocolo P2P a 434 y 868 MHz - MRF49XA (115,2 kbps)						
Modo de transmisión	Payload (bytes)	Cadencia eficaz (kbps)	Enviados	Recibidos		
				BS 2	BS 5	BS 10
BROADCAST	90	62,0	255	119	121	125
	1	2,13	255	40	43	48
UNICAST (DIRECCIÓN LARGA)	90	23,1	255	255	255	255
	1	0,30	255	255	255	255

Tabla 18: Cadencia eficaz, envío y recepción de mensajes en P2P a 434 y 868 MHz

Protocolo MiWi a 434 y 868 MHz - MRF49XA (9,6 kbps)						
Modo de transmisión	Payload (bytes)	Cadencia eficaz (kbps)	Enviados	Recibidos		
				BS 2	BS 5	BS 10
BROADCAST	79	6,75	240 - 252	20	20	20
	1	0,30	254	7	7	7
UNICAST (DIRECCIÓN LARGA)	79	5,50	254	254	254	254
	1	0,17	254	254	254	254
UNICAST (DIRECCIÓN CORTA)	79	5,44	254	254	254	254
	1	0,17	254	254	254	254
Protocolo MiWi a 434 y 868 MHz - MRF49XA (38,4 kbps)						
Modo de transmisión	Payload (bytes)	Cadencia eficaz (kbps)	Enviados	Recibidos		
				BS 2	BS 5	BS 10
BROADCAST	79	22,6	251 - 253	7	7	7
	1	0,92	254	3	3	3
UNICAST (DIRECCIÓN LARGA)	79	18,9	254	254	254	254
	1	0,32	254	254	254	254
UNICAST (DIRECCIÓN CORTA)	79	16,6	254	254	254	254
	1	0,32	254	254	254	254
Protocolo MiWi a 434 y 868 MHz - MRF49XA (115,2 kbps)						
Modo de transmisión	Payload (bytes)	Cadencia eficaz (kbps)	Enviados	Recibidos		
				BS 2	BS 5	BS 10
BROADCAST	79	47,8	253 - 254	4	4	4
	1	1,71	254	2	2	2
UNICAST (DIRECCIÓN LARGA)	79	15,8	254	254	254	254
	1	0,33	254	254	254	254
UNICAST (DIRECCIÓN CORTA)	79	21,1	254	254	254	254
	1	0,32	254	254	254	254

Tabla 19: Cadencia eficaz, envío y recepción de mensajes en MiWi a 434 y 868 MHz

Como ocurre en la interfaz de 2,4 GHz, para las interfaces de 434 MHz y 868 MHz también se observa que el cuello de botella es la recepción y el procesamiento de los mensajes, que la carga útil de los paquetes está relacionada con la cadencia eficaz y que, por lo general, los paquetes con menos payload transmitidos con mayor ancho de banda presentan peores resultados. No obstante, las tablas nos permiten extraer conclusiones adicionales.

La recepción de paquetes unicast presenta muy buenos resultados debido a que antes de transmitir un paquete nuevo se espera a recibir el ACK del anterior. Como contrapartida, las cifras de la cadencia eficaz se ven afectadas, especialmente cuando se reduce el tiempo de transmisión de los paquetes gracias al hecho de disponer de mayor ancho de banda a nivel radio. El envío y recepción de asentimientos solo se produce si se habilita la opción `ENABLE_ACK` del transceptor MRF49XA.

En cuanto a la cadencia eficaz, se observa un fenómeno de saturación o estancamiento. Para anchos de banda pequeños la cadencia eficaz está limitada por la tasa binaria que ofrece el canal radio, mientras que para anchos de banda muy grandes, la cadencia eficaz pasa a estar limitada por las capacidades del microcontrolador. Es decir, que el microcontrolador no es capaz de escribir los paquetes radio, las cabeceras de los niveles de la pila y activar la transmisión para lograr exprimir el ancho de banda máximo que ofrecen los canales de mayor ancho de banda.

Haciendo la conversión a la tasa binaria en el canal radio a partir de la cadencia eficaz medida en los canales de 115,2 kbps, se estima que el punto de inflexión se encuentra en torno a un ancho de banda a nivel radio de 85 kbps. Se supone que hasta alcanzar ese valor, el comportamiento de la cadencia eficaz es aproximadamente lineal.

Por último se aborda el caso especial de Broadcast en el protocolo MiWi. El hecho de que un nodo coordinador reciba un paquete broadcast por cualquier interfaz puede desencadenar varias acciones: reenviarlo para que el mensaje tenga mayor difusión en la red, almacenarlo en una tabla especial para detectar duplicados del mensaje e incluso guardar un mensaje indirecto para dispositivos RFD asociados.

El hecho de retransmitir el mensaje supone un procesamiento adicional en el que, además, se incluye un retardo pseudoaleatorio (jitter) de 20 ms de media. Es la manera que tiene la torre de protocolos de dar cierto margen en la retransmisión y evitar que el mismo paquete se solape con sus réplicas en la red. Este retardo inducido por la pila echa por tierra las métricas de paquetes recibidos, especialmente en los casos de mayor ancho de banda y menor carga útil, pues los paquetes llegan en instantes de tiempo más próximos y no pueden ser atendidos.

6.2.2.3. Tiempo de conmutación o reconfiguración de parámetros radio

Descripción

En esta prueba se evaluará cuántos cambios por segundo pueden realizarse en parámetros de las interfaces radio mediante las funciones de la HAL. En concreto se desea medir con qué frecuencia pueden realizarse las siguientes acciones:

- Cambiar la potencia de transmisión de una interfaz.
- Cambiar el canal de transmisión.
- Dormir y despertar una interfaz radio.

Procedimiento

De nuevo se empleará el temporizador interno del sistema para medir la frecuencia con la que se realizan las acciones. Se realizarán 200 operaciones consecutivas y se calculará el tiempo medio.

Condiciones de medida

Las pruebas se realizan a la frecuencia nominal de trabajo en ausencia de actividad en la red. El periodo de mantenimiento de las pilas se mantiene en 25 ms.

Resultados

Operación	Protocolo P2P			Protocolo MiWi		
	434 MHz	868 MHz	2,4 GHz	434 MHz	868 MHz	2,4 GHz
Cambio del canal de transmisión	33 s ⁻¹	33 s ⁻¹	6800 s ⁻¹	33 s ⁻¹	33 s ⁻¹	6818 s ⁻¹
Cambio de la potencia de transmisión	17933 s ⁻¹	17931 s ⁻¹	14753 s ⁻¹	17808 s ⁻¹	17816 s ⁻¹	14753 s ⁻¹
Dormir y despertar una interfaz	65 s ⁻¹	65 s ⁻¹	2822 s ⁻¹	65 s ⁻¹	65 s ⁻¹	2809 s ⁻¹

Figura 20: Frecuencia máxima de reconfiguración de parámetros radio

En la tabla de resultados podemos ver como la operación de cambiar la potencia de transmisión es muy rápida en todas las interfaces y protocolos, ya que básicamente consiste en una serie de comprobaciones y la escritura en los registros del transceptor utilizado.

Sin embargo, las cifras en las interfaces de 434 y 868 MHz para el resto de operaciones son muy mediocres. Esto se debe a que el firmware del transceptor MRF49XA facilitado por Microchip introduce retardos software, ralentizando mucho la ejecución. En el caso del cambio de canal, se espera 20 ms para estabilizar el transceptor, mientras que al despertar una interfaz se introduce un retardo de 10 ms ya que el oscilador interno del transceptor tiene una latencia de arranque de entre 2 y 7 ms.

6.2.2.4. Desconexión forzosa de un módulo MRF49XA

Descripción

Aprovechando que el transceptor MRF49XA no se encuentra soldado en la plataforma de prueba, se determinarán los efectos de retirar el módulo transceptor cuando la pila de protocolos esté configurada para manejarlo. De este modo, sabremos si la HAL detecta este suceso, y por consiguiente, la aplicación.

Procedimiento

Se configura el firmware para utilizar, al menos, una interfaz radio que emplee el MRF49XA, cargando como programa un banco de pruebas general. Una vez que el nodo esté preparado se desconecta el módulo y mediante las trazas de depuración se evalúa el comportamiento de la torre de protocolos.

Condiciones de medida:

Se realiza la prueba para ambos protocolos y tanto para un FFD como para un RFD. En MiWi, se configura la prueba para que en el caso del FFD se adopte el rol de coordinador.

Resultados:

- **Inicialización de la interfaz.**
La operación termina aparentemente bien, solo que el dispositivo no se une a la red ni obtiene una dirección de ella (si procede). Si el nodo puede crear una nueva red lo hace, al no poder recibir respuestas por esa interfaz. El escaneo de los canales reporta el valor máximo en todos los canales, por lo que siempre crearía la red en el canal 0.
- **Cambio de canal.**
La pila cree que el cambio de canal se ha producido correctamente y la HAL modifica sus variables internas.
- **Dormir la interfaz (o el nodo completo) y despertarla:**
Duerme y despierta teóricamente a la interfaz en cuestión y se actualizan los flags.
- **Escribir datos en el buffer de transmisión:**
No reporta error hasta alcanzar (teóricamente) el límite de capacidad del buffer de transmisión, en el que la HAL reporta el error de buffer lleno.
- **Mandar un paquete broadcast.**
No se envía el paquete y la HAL devuelve el código de error correcto.
- **Mandar un paquete unicast.**
De nuevo, no se envía y la HAL devuelve el código de error correcto.
- **Leer datos del buffer de recepción.**
No se recibirán nuevos datos de recepción y, si se intenta leer el buffer cuando está vacío, la HAL devuelve el error pertinente.
- **Escaneo de canales.**
Se realiza el escaneo y se obtiene el mismo valor en cada canal.

En general, todas las operaciones que implican simplemente escrituras en los registros del transceptor no detectan la desconexión del módulo. La comunicación por SPI se produce, pero el transceptor no se encuentra en el otro extremo para recibir las órdenes del microcontrolador. No ocurre lo mismo cuando se espera una interacción por parte del transceptor o se lee el valor de alguno de sus registros para realizar comprobaciones. Es entonces cuando se producen fallos que detecta la torre de protocolos.

6.2.2.5. Medidas de consumo.

Descripción

Se trata de medir el consumo general del nodo en diversas situaciones:

- En modo sleep con todas las interfaces apagadas, para evaluar el correcto funcionamiento del modo de bajo consumo y observar el mínimo consumo de la plataforma testada.
- En modo normal con todas las interfaces apagadas, para estimar el consumo del microcontrolador en este modo.
- En modo normal, despertando y durmiendo las interfaces para medir el consumo de las mismas.

Procedimiento

Se realizará el sencillo montaje de la figura para medir la corriente de alimentación que demanda el dispositivo en cada una de las situaciones.

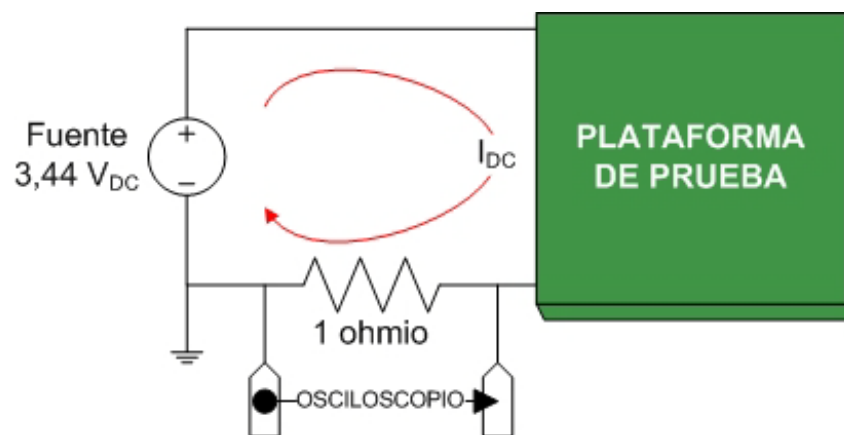


Figura 21: Montaje para la medida del consumo del nodo.

La corriente de alimentación se monitoriza mediante un osciloscopio que mide la caída de tensión en la resistencia de 1 ohmio. De este modo, la lectura del osciloscopio en mV equivale al mismo valor de la corriente en mA (aplicando la ley de Ohm). También se mide la corriente en el indicador de la fuente, si bien la lectura que ofrece la aguja es más bien cualitativa.

Condiciones de medida

Al conectar la plataforma de prueba, se ajusta la tensión de la fuente para que en bornes del conector de alimentación del nodo la tensión sea de 3,25 V aproximadamente, sin sobrepasar el umbral de los 3,5 V por seguridad. De este modo, los datos de consumo se deberían parecer a los de las hojas de especificaciones.

En el osciloscopio se activa la función de promediado para leer el valor medio en cada una de las situaciones y despreciar componentes de ruido.

Resultados:

Actividad del nodo	Osciloscopio	Fuente de alimentación
Modo normal con todas las interfaces encendidas.	128 mA	120 mA
Modo normal sólo con la interfaz a 2,4 GHz encendida.	116 mA	108 mA
Modo normal con todas las interfaces radio dormidas.	92,2 mA	85 mA
Modo sleep con todas las interfaces apagadas.	9,3 mA	6 mA

Tabla 20: Medidas de consumo

La primera medida corresponde a la situación normal del nodo realizando un escaneo de energía en una de sus interfaces, obteniendo un valor promedio de 128 mA. En la fuerte se observan ligeras fluctuaciones de pocos mA debida a la transmisión de paquetes. Es un consumo importante, pero dentro de lo que cabe esperar por las especificaciones de los transceptores radio y los microcontroladores PIC.

La diferencia entre las dos primeras situaciones es simplemente que el transceptor MRF49XA ha pasado a estar en modo dormido. Observamos que la reducción en el consumo ha sido de 12 mA, un valor que coincide exactamente con el consumo medio del transceptor según especificaciones. Por tanto, podemos concluir que el transceptor efectivamente ha entrado en modo de bajo consumo y que el valor real observado es muy similar al teórico.

Para caracterizar el consumo del MRF24J40 y la interfaz a 2,4 GHz nos fijamos en las situaciones 2 y 3. La disminución en el consumo es ligeramente mayor que los 19 mA que indica el fabricante en la hoja de características como consumo en recepción del MRF24J40.

En el modo sleep observamos que el consumo cae a tan solo unos pocos miliamperios. Este hecho resulta chocante a priori, pues el consumo del microcontrolador debería de ser tan solo de unas decenas de microamperios. Sin embargo, tras comprobar el consumo teórico de algunos componentes de la plataforma hardware, podemos achacar gran parte del consumo observado al conversor integrado MAX3232 para volcar trazas por RS-232, que tiene un consumo típico de 8 mA.

El consumo del microcontrolador se puede estimar como la diferencia entre las dos últimas situaciones, unos 81 mA. El valor es muy similar al valor que ofrecen las especificaciones de otros PIC muy similares, como el PIC32MX775F512H (85 mA), por lo que podemos pensar que aunque la hoja de características no explicita el consumo del microcontrolador de la plataforma de prueba, se agrupa con los microcontroladores más potentes de la gama.

Para concluir, algunos comentarios finales:

- La resistencia utilizada tiene un 5% de tolerancia, lo que justifica las diferencias observadas entre la fuente, el osciloscopio y las hojas de características. Al medir la resistencia con un polímetro se obtuvo 1,1 ohmios, por lo que los valores reales de consumo posiblemente sean un punto medio entre la lectura del osciloscopio y la de la fuente. Las medidas no deben interpretarse como un valor exacto, sino como un valor aproximado que permite esbozar el consumo en cada situación.
- El consumo en modo sleep puede deberse a otros módulos hardware, a pines del microcontrolador en colector abierto, o elementos del diseño que propicien fugas de corriente. Se debe invertir un esfuerzo en rediseñar el nodo para minimizar este consumo parásito de los componentes.

7. CONCLUSIONES Y LÍNEAS FUTURAS

7.1. CONCLUSIONES

El presente proyecto tiene como objetivo principal diseñar un nodo para una red de sensores inalámbrica con capacidades cognitivas que incorpore una capa de abstracción hardware como interfaz de programación de alto nivel para el desarrollo de aplicaciones.

En primer lugar, se ha estudiado el estado del arte en materia de radio cognitiva y redes de sensores. Se han presentado algunos estándares para WSN, simuladores y plataformas de prueba utilizados a nivel internacional, así como las características de algunos nodos de WSN.

Como punto de partida, dentro del laboratorio se disponía de un primer prototipo de nodo cognitivo (First Cognitive Device). Gracias al análisis realizado en el capítulo 3, se ha rediseñado el nodo para suplir algunas de sus carencias, incorporar nuevos elementos al diseño y eliminar aquellos otros que, por los requisitos impuestos, no se ha estimado oportuno mantener.

En ese sentido, se ha implementado una nueva versión del firmware para el prototipo cognitivo de nueva generación (Next Generation Device) con acceso a tres regiones distintas del espectro electromagnético. Dicho acceso se basa en tres transceptores de Microchip y la pila de protocolos MiWi™ que ofrece la mencionada compañía, que ha sido adaptada y modificada para permitir el uso de más de un transceptor.

Para abstraer de la complejidad hardware y software del diseño se ha desarrollado una capa de abstracción hardware sencilla que permite manejar las interfaces radio fácilmente. En ese sentido, se ha mejorado notablemente la usabilidad del nodo cognitivo, pues basta conocer un conjunto reducido de llamadas a la HAL para manejar las funcionalidades que ofrece el nodo. Este hecho es importantísimo para reducir coste y tiempo de desarrollo de las aplicaciones que se desarrollen sobre él.

Finalmente se han realizado una serie de pruebas sobre el nodo enfocadas a comprobar el funcionamiento del mismo y la implementación de la HAL, así como caracterizar sus prestaciones más importantes. El análisis de los resultados ha permitido extraer datos y conclusiones que serán de gran utilidad de cara a las líneas futuras.

Gracias a la implementación del nuevo firmware podemos concluir además que:

- El nodo es mucho más versátil y reconfigurable, admitiendo modificaciones hardware, la inclusión de nuevos módulos y la utilización de otros microcontroladores PIC de 32 bits con distinto encapsulado realizando mínimas modificaciones en el firmware.

- Las interfaces radio son menos heterogéneas y su consumo energético, en conjunto, es mucho menor que el del prototipo FCD. Además, la eliminación del módulo de Wi-Fi hace que el dispositivo se ajuste más a los requisitos de baja tasa de datos y prestaciones que imponen las WSN.
- La implementación de un modo de bajo consumo en el que se detiene el reloj del sistema y la CPU del microcontrolador es vital para paliar los problemas de consumo del nodo FCD. La posibilidad de disminuir drásticamente el consumo del nodo es el primer paso para hacer viable que el dispositivo pueda funcionar de manera autónoma con baterías. Una de las claves para lograr un consumo medio reducido es hacer uso, siempre que las circunstancias lo permitan, del modo sleep del microcontrolador, ya que la reducción en el consumo instantáneo es muy significativa.
- La adaptación de la torre de protocolos de MiWi™ permite hacer un uso más eficiente de los recursos del microcontrolador, evitando duplicidades mediante la compartición de recursos y reduciendo la complejidad general al disponer de niveles en la pila en común.
- El nuevo diseño del nodo NGD se mantienen las prestaciones del nodo FCD en cuanto a capacidad de procesamiento al optar por microcontroladores de la gama PIC32, aumentando la capacidad de entrada y salida para incluir nuevos módulos.
- El coste de fabricación del dispositivo puede verse notablemente reducido al eliminar ciertos componentes, elegir microcontroladores con menos memoria flash y RAM, reducir el tamaño del nodo o eliminar algunos componentes como el reloj de tiempo real, cumpliendo los requisitos de bajo coste.
- El mantenimiento de la torre de protocolos puede delegarse a la propia HAL, siendo necesario únicamente que la aplicación compruebe la llegada de nuevos paquetes radio y procese la información de aplicación.

7.2. RECOMENDACIONES PARA LA IMPLEMENTACIÓN HARDWARE

Por otro lado, el análisis del nodo FCD y las pruebas que se han realizado con el nodo FCD adaptado han permitido extraer información muy valiosa para desarrollar el prototipo hardware del nodo NGD. A continuación se marcan algunas recomendaciones importantes de cara al futuro diseño de aplicaciones y el prototipo hardware del nodo NGD.

Diseño hardware orientado a bajo consumo.

Se debe prestar especial cuidado a posibles fugas de corriente en el diseño del prototipo, reduciendo el consumo estático en la medida de lo posible y gestionando adecuadamente los pines de entrada y salida de propósito general.

Módulo USB sólo en dispositivos conectados a la red.

A raíz de la implementación del modo de bajo consumo, se ha observado que la generación de relojes mediante síntesis con PLL grava fuertemente el consumo. El PIC dispone de un módulo PLL adicional para generar el reloj utilizado por el módulo USB que, además, puede evitar que el microcontrolador entre en modo de bajo consumo. Se recomienda por tanto no incluir la pila de USB en el firmware salvo en los nodos que utilicen la red eléctrica como fuente de alimentación.

Selección de módulos y periféricos con mayor interoperabilidad.

Tras evaluar el prototipo demostrador que se realizó en el laboratorio, estudiar qué interfaces y qué microcontrolador incluir en el nuevo diseño, y qué limitaciones imponen los recursos de los que dispone el microcontrolador, a continuación se incluyen recomendaciones para la futura implementación hardware que se realice en el laboratorio.

Para la conexión de los interfaces radio se recomienda dejar libre la interfaz SPI2 y las interrupciones externas INT0 (incompatible con la SPI1) e INT2. De este modo se aprovecha mejor el hecho de dejar la I²C1 inhabilitada.

En caso de requerir una interfaz I²C, se recomienda utilizar la I²C2. Lo mismo ocurre con el bus CAN2 (si el microcontrolador dispone de este periférico), ya que el CAN1 se multiplexa con Ethernet.

Para mandar trazas al PC se puede utilizar la UART3 o la UART6 si la SPI2 no está siendo utilizada, o la interfaz USB en caso contrario. Si no se emplean las funciones de control de flujo por hardware es más beneficioso utilizar la UART6 pues habilita el uso de la I²C4.

Versatilidad y modularidad para admitir mejoras

Como ya se ha comentado a lo largo de este documento, se quiere favorecer que el nodo admita mejoras y nuevos módulos en el futuro, por lo que es interesante disponer de zócalos o receptáculos para conectores para acceder a los siguientes recursos:

- Las interfaces multiplexadas SPI2 / UART3 / UART6 / I²C4 (4 líneas)
- Los módulos USB (4 líneas) e I²C2 (2 líneas)
- La línea de interrupción externa INT2 (o en su defecto, otra que haya quedado libre).
- Los 10 pines básicos de Ethernet.
- Pines de entrada/salida de propósito general libres o multiplexados con otros periféricos (ADC, captura de entrada, comparación de salida, comparadores...)

Por último, se recomienda mantener el cristal de 8 MHz como oscilador principal y la posibilidad de incluir un segundo oscilador por si se quiere implementar el reloj de tiempo real.

Reducción del coste de fabricación

Hay varias estrategias que pueden reducir notablemente el coste global del nodo. La primera de ellas es que el diseño en PCB tenga un tamaño menor que el del prototipo FCD y sea modular. Con ello no sólo se reduce directamente el coste de fabricación del soporte hardware, sino que además se aprovecha el hecho de que ciertos módulos sólo se incluyan en algunos nodos (transceptores, mejoras, nuevas interfaces de comunicación, etc) o en momentos concretos (por ejemplo, las baterías).

Otro enfoque para reducir el tamaño y el coste del nodo es realizar un diseño hardware propio de la placa MRF49XA PICTail Daughter Board™, ya que Microchip facilita una lista con los componentes necesarios y referencias al fabricante de cada elemento empleado en su diseño comercial. El nuevo diseño del módulo transceptor MRF49XA puede incorporar una antena con mayor ganancia o más robusta que la del producto comercial que ofrece Microchip.

Optimización de la pila en función de la aplicación.

El alto grado de flexibilidad y configuración que presentan los transceptores, los protocolos y la HAL debe jugar en favor de la aplicación a la que se le da soporte. Por eso, es importante experimentar distintas configuraciones que permitan ajustarse a las necesidades de la aplicación en busca de mejor rendimiento y menor consumo de memoria flash y RAM.

7.3. LÍNEAS FUTURAS

Por último, a continuación se mencionan posibles líneas futuras que desarrollar para solventar algunos problemas o añadir mejoras al diseño actual del nodo NGD:

- **Implementación del prototipo hardware.** Se trata de la continuación lógica de este proyecto, ya que la implementación hardware ha quedado fuera del marco del mismo. La caracterización y prueba del nuevo prototipo hardware mostrará las características que mejorar o en las que invertir un mayor esfuerzo.
- **Utilización de modos de consumo intermedio.** En los momentos de baja actividad en la red puede ser interesante recurrir a modos de consumo intermedios que, en combinación con el modo sleep, reduzcan el consumo general del nodo más aún.
- **Implementación de nuevos modos de seguridad.** Por ahora solo se contempla el cifrado de paquetes dentro del modo de seguridad, pero la HAL está preparada para admitir nuevos modos de seguridad que impliquen acciones adicionales.
- **Adaptación del protocolo MiWiPRO™.** Si las necesidades de la red demandan mayor número de nodos o capacidades de rutado, puede portarse el protocolo con mayor funcionalidad y complejidad de Microchip.
- **Desconexión de la alimentación por módulos.** Esta es otra estrategia muy sencilla para reducir el consumo del nodo que puede aplicarse, por ejemplo, al

módulo MAX3232 (que demanda 8 mA). Bastaría incluir un interruptor controlado por software que actuara como bypass de la tensión de alimentación. Su utilización con otros módulos es interesante, pero puede requerir cierta adaptación del firmware o, en el caso de los transceptores, la necesidad de reiniciar los mismos tras la desconexión.

- **Wake-On-Radio en los transceptores del diseño.** Uno de los principales problemas que presenta el nuevo firmware es que los transceptores de Microchip no disponen de la posibilidad de despertar cuando haya presencia de datos en la antena. Es decir, al dormir una determinada interfaz, el nodo pasa a ser totalmente inconsciente de la actividad de la red en esa interfaz. Por ello, la opción de implementar Wake-On-Radio en los transceptores es muy atractiva, ya que permitiría dormir el nodo completo y que sólo fuese despertado en caso de recibir datos por una de sus interfaces.
- **Nodos pasarela con conectividad Ethernet.** Ya que se ha eliminado la interfaz de Wi-Fi del nodo, se plantea la utilización del controlador de ethernet del que dispone el microcontrolador para aumentar la conectividad de determinados nodos que estén conectados a la red eléctrica. Éste podría ser el caso del coordinador de red (PANC) u otros nodos que actúen como nodo sumidero para enviar datos a través de esta interfaz.
- **Diseño de un protocolo que dé servicios añadidos.** Uno de los problemas encontrados es que en momentos de mucha actividad en la red, el almacenamiento de paquetes puede provocar desorden en los datos ofrecidos al nivel de aplicación. Se podría implementar un protocolo sobre la HAL que dote de servicios de fiabilidad y permita separar el tráfico de distintas aplicaciones embebidas en el nodo.
- **Diseño de la capa cognitiva.** La implementación de estrategias cognitivas es crucial para lograr un nodo de CWSN y, a tal efecto, se hace necesario el diseño de un nivel cognitivo. Mediante la monitorización del estado de la red, el intercambio de información entre los nodos, el escaneo del nivel de ruido en las zonas del espectro y la toma de decisiones conjunta se podrán investigar nuevas estrategias para optimizar las prestaciones de la red. La capa de abstracción hardware dispone de funciones específicas para realizar esas tareas, por lo que la implementación de dicha capa es ahora mucho más sencilla.

BIBLIOGRAFÍA

- [1] López, F.; *Diseño e implementación de un nodo para redes de sensores con capacidades cognitivas*, PFC, Laboratorio de Sistemas Integrados, ETSIT-UPM, 2011.
- [2] Mitola, J. III; Maguire, G.Q. Jr; *Cognitive radio: making software radios more personal*. Personal Communications, IEEE, V. 6, Issue 4, pp. 13-18, 1999.
- [3] Mitola, J. III; *Cognitive Radio An Integrated Agent Architecture for Software Defined Radio*, PhD. Dissertation, Royal Institute of Technology, Kista, Sweden, 2000.
- [4] Haykin, S. ; *Cognitive radio: brain-empowered wireless communications*. Selected Areas in Communications, IEEE Journal 23, pp. 201-220, 2005.
- [5] Clark, D. D.; Partridge, C.; Ramming, J. C.; Wroclawski, J. T.; *A knowledge plane for the internet*, Proceedings of the 2003 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, pp. 3-10, Karlsruhe, Germany, 2003.
- [6] Thomas, R.W.; DaSilva, L.A.; MacKenzie, A.B.; *Cognitive networks*. Proceedings of the First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005.
- [7] Thomas, R. W.; Friend, D. H.; Da Silva, L. A.; Mackenzie, A. B.; *Cognitive networks: adaptation and learning to achieve end-to-end performance objectives*, Communications Magazine, Personal Communications, IEEE, V.44, Issue 12, pp. 51-57, 2006.
- [8] Berlemann, L.; Mangold, S.; *Cognitive Radio and Dynamic Spectrum Access*, Wiley, 2009, ISBN 9780470511671.
- [9] IEEE; *IEEE 802.22 Working Group on Wireless Regional Area Network*, <http://ieee802.org/22/>
- [10] Ecma International; *Ecma 392 Standard. MAC and PHY for Operation in TV White Space*. <http://www.ecma-international.org/publications/standards/Ecma-392.htm>, 2009.
- [11] IEEE; *IEEE 802.15 WPAN™ Task Group 4*, <http://www.ieee802.org/15/pub/TG4.html>
- [12] ZigBee Alliance; *ZigBee Alliance Home Page*, <http://www.zigbee.org/>
- [13] HART Communication Foundation; *WirelessHART Technology*, <http://www.hartcomm.org/>
- [14] ISA, Wireless Compliance Institute; *ISA100 Wireless Systems for Automation*, <http://www.isa100wci.org/>
- [15] Microchip Technology Inc.; *MiWi Development Environment*
- [16] Wi-Fi Alliance; *Wi-Fi Alliance Home Page*, www.wi-fi.org
- [17] IEEE; *IEEE 802.11™ Wireless Local Area Networks*, <http://www.ieee802.org/11/>
- [18] IETF; *IETF IPv6 over Low power WPAN*, <https://datatracker.ietf.org/wg/6lowpan/>
- [19] Shenai, K.; Mukhopadhyay, S.; *Cognitive sensor networks*, 26th International Conference on Microelectronics, MIEL 2008, pp. 315 – 320, 2008.
- [20] Cavalcanti, D.; Das, S.; Wang, J.; Challapali, K.; *Cognitive Radio based Wireless Sensor Networks*, Proceedings of 17th International Conference on Computer Communications and Networks, 2008.
- [21] Rabaey, J.; Wolisz, A.; Ercan, A.O.; Araujo, A.; Burghardt, F.; Mustafa, S.; Parsa, A.; Pollin, S.; IHSiang, W.; Malagón, P.; *Connectivity Brokerage - Enabling Seamless Cooperation in Wireless Networks : A White Paper*, BWRC Publications, 2011.

- [22] Vijay, G.; Ben Ali Bdira, E.; Ibnkahla, M.; *Cognition in Wireless Sensor Networks: A perspective*, Sensors Journal IEEE, V. 11, Issue 3, pp. 582–592, 2011.
- [23] Zahmati, A.S.; Hussain, S.; Fernando, X.; Grami, A.; *Cognitive Wireless Sensor Networks: Emerging topics and recent challenges*, Science and Technology for Humanity (TIC-STH), IEEE Toronto International Conference, pp. 593-596, 2009.
- [24] Yunjiao, X.; Ho Sung, L.; Ming, Y.; Kumarawadu, P.; Ghenniwa, H.H.; Weiming, S.; Performance Evaluation of NS-2 Simulator for Wireless Sensor Networks, Canadian Conference on *Electrical and Computer Engineering (CCECE)*, pp. 1372-1375, 2007.
- [25] Pescosolido, L.; *Performance of SENDORA networks*. 2011.
- [26] Pongor, G.; *OMNeT: Objective Modular Network Testbed*, Proceedings of the International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems (MASCOTS), 1993.
- [27] Rastegarnia, A.; Solouk, V. ; *Performance evaluation of Castalia Wireless Sensor Network simulator*, 34th International Conference on Telecommunications and Signal Processing (TSP), pp. 111 - 115, 2011.
- [28] Weingartner, E.; vom Lehn, H.; Wehrle, K. ; A performance comparison of recent network simulators, IEEE International Conference on Communications (ICC), pp. 1-5, 2009.
- [29] Lahmar, K.; Cheour, R.; Abid, M.; *Wireless Sensor Networks: Trends, Power Consumption and Simulators* , 6th Asia Modelling Symposium (AMS), pp. 200-204, 2012.
- [30] Handziski, V.; Köpke, A.; Willig, A.; Wolisz, A.; *TWIST: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks*, Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality (REALMAN), 2006.
- [31] Newman, T.R.; An, H.; Gaeddert, J.; Hilbun, B.; Bose, T.; Reed, J.H.; *Virginia Tech cognitive radio network testbed and open source cognitive radio framework*, 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities and Workshops (TridentCom), pp. 1-3, 2009.
- [32] López, F.; Romero, E.; Blesa, J.; Villanueva, D.; Araujo, A.; *Cognitive Wireless Sensor Network Device for AAL Scenarios*, Lecture Notes in Computer Science, Ambient Assisted Living, Springer Berlin, V. 6693, pp. 116-121, 2011.
- [33] Microchip Technology Inc.; *PIC32MX Family Reference Manual*
- [34] Microchip Technology Inc.; *PIC32MX5XX/6XX/7XX Family Data Sheet*
- [35] MIPS Technologies Inc.; *MIPS32® M4K® Processor Core Datasheet*
- [36] Microchip Technology Inc.; *AN833 The Microchip TCP/IP Stack*
- [37] Microchip Technology Inc.; *MRF24WB0MA/MRF24WB0MB Data Sheet*
- [38] Advanced Wireless Dynamics; *Tulio RF Modem. 868 & 915 MHz (ISM Bands) Radio Module*
- [39] Microchip Technology Inc.; *AN1284 - Microchip Wireless (MiWi™) Application Programming Interface - MiApp*
- [40] Microchip Technology Inc.; *AN1204 - Microchip MiWi™ P2P Wireless Protocol*
- [41] Microchip Technology Inc.; *AN1066 - Microchip MiWi™ Wireless Networking Protocol Stack*
- [42] Microchip Technology Inc.; *AN1371 - Microchip MiWi™ PRO Wireless Networking Protocol*
- [43] Microchip Technology Inc.; *AN1283 - Microchip Wireless (MiWi™) Media Access Controller - MiMAC*
- [44] Microchip Technology Inc.; *MPLAB X User's Guide*

- [45] Microchip Technology Inc.; *MPLAB ICD3. In-Circuit Debugger User's Guide for MPLAB X IDE*
- [46] Microchip Technology Inc.; Tranceptores MiWi
- [47] Microchip Technology Inc.; Tranceptores MiWi
- [48] Microchip Technology Inc.; Tranceptores MiWi
- [49] Microchip Technology Inc.; Tranceptores MiWi
- [50] Microchip Technology Inc.; Tranceptores MiWi
- [51] Atmel
- [52] Texas Instruments
- [53] Freescale
- [54] Farnell
- [55] RSOnline
- [56] Mouser
- [57] DigiKey
- [58] El lenguaje C
- [59]

ANEXO: FUNCIONES DE LA CAPA DE ABSTRACCIÓN HARDWARE

- **BYTE InitNode ()**

Descripción: Inicializa el sistema y las variables utilizadas. Inicializa las pilas de protocolos presentes en el nodo. Es una función que está pensada para ejecutarse una única vez, antes de ejecutar código específico de aplicación.

Parámetros: Ninguno.

Valor de retorno: “NO_ERROR” o código de error.

- **BYTE GetStatusFlags()**

Descripción: Devuelve (lee) el byte de flags de estado.

Parámetros: Ninguno.

Valor de retorno: Byte de flags de estado.

- **BYTE GetOpChannel (radioInterface ri)**

Descripción: Devuelve el canal en el que opera la interfaz radio pasada como parámetro.

Parámetros: **ri** – Interfaz radio de la que se quiere obtener el canal que está en uso. Las opciones NONE, ALL_MIWI y ALL son opciones no válidas en esta función.

Valor de retorno: Canal en el que se está operando en caso de éxito o código de error.

- **BYTE GetScanResult (radioInterface ri, BYTE channel, BYTE *storeItHere)**

Descripción: Lee el valor del RSSI guardado al evaluar el ruido de un canal (mediante la función de escaneo de canales). Los resultados se inicializan a cero al invocar la función InitNode. El valor del RSSI del canal e interfaz solicitadas se almacena en la dirección pasada como parámetro.

Parámetros: **ri** – Interfaz radio elegida. Las opciones NONE, ALL_MIWI y ALL no son válidas.

channel – Canal elegido.

***storeItHere** – Byte donde se almacenará el valor del RSSI en caso de éxito.

Valor de retorno: “NO_ERROR” o código de error.

- **BYTE GetPayloadToRead (radioInterface ri)**

Descripción: Esta función devuelve la cantidad de bytes pendientes de leer en una interfaz radio. En caso de invocar ALL_MIWI o ALL devuelve la suma de bytes pendientes de leer de las distintas interfaces.

Parámetros: **ri** – Interfaz radio seleccionada. La opción NONE no es válida en esta función.

Valor de retorno: En caso de éxito, devuelve el número de bytes pendientes de leer. Si los datos pendientes superan los 127 bytes se devolverá 127 (0b01111111), ya que es el

máximo valor que no se interpreta como código de error. Si se produce un error se notifica el correspondiente código.

- ***BYTE GetRSSI (radioInterface ir, BYTE *storeItHere)***

Descripción: Lee el valor del RSSI del paquete recibido por la interfaz especificada y lo almacena en la dirección pasada como parámetro.

Parámetros: **ri** – Interfaz radio del paquete recibido. Las opciones NONE, ALL_MIWI y ALL no son válidas.

***storeItHere** – Byte donde se almacenará el valor del RSSI en caso de éxito.

Valor de retorno: “NO_ERROR” o código de error.

Nota: No confundir el valor del RSSI del paquete recibido (que indica la intensidad de la señal al recibir un paquete radio y solo está disponible mientras se procesa dicho paquete) con el de los resultados del escaneo de canales (que miden el ruido presente en un canal específico y se almacenan en la variable de estado general del nodo).

- ***BYTE GetLQI (radioInterface ir, BYTE *storeItHere)***

Descripción: Lee el valor del LQI del paquete recibido por la interfaz especificada, indicador de la calidad del enlace en el canal empleado, y lo almacena en la dirección pasada como parámetro.

Parámetros: **ri** – Interfaz radio del paquete recibido. Las opciones NONE, ALL_MIWI y ALL no son válidas.

***storeItHere** – Byte donde se almacenará el valor del LQI en caso de éxito.

Valor de retorno: “NO_ERROR” o código de error.

- ***BYTE GetFreeTXBufSpace (radioInterface ri)***

Descripción: Informa a la aplicación de cuántos bytes quedan libres en el buffer de transmisión de la interfaz radio seleccionada.

Parámetros: **ri** – Interfaz radio seleccionada. NONE, ALL_MIWI y ALL son opciones no válidas.

Valor de retorno: En caso de éxito, devuelve el número de bytes que se pueden escribir en el buffer de transmisión. Si el valor supera el umbral de 127 bytes, devolverá 127 (0b01111111), ya que es el máximo valor que no se interpreta como código de error (aunque los buffers de MiWi tienen capacidades menores que 127 bytes). Si se produce un error se notifica el correspondiente código.

- ***BYTE SleepRadioInterface (radioInterface ri)***

Descripción: Esta función sirve para cambiar el modo de operación de un transceptor radio a modo SLEEP. En caso de invocar la función con ALL, todos los transceptores presentes en el nodo intentarán pasar a estado dormido.

Parámetros: **ri** – Interfaz radio seleccionada. Emplear la opción “NONE” devolverá un error.

Valor de retorno: “NO_ERROR” o código de error. En el caso de seleccionar todas las interfaces disponibles con “ALL”, comienza intentando dormir las interfaces a frecuencias más bajas y termina durmiendo el módulo WiFi (si el nodo dispone de él). Si alguna interfaz falla, devolverá el código de error de intentar dormir esa interfaz y abortará el proceso. La aplicación podrá averiguar qué interfaces se han dormido consultando el byte de flags de estado. En caso de invocarla con ALL_MIWI, es análogo a ALL pero excluyendo la interfaz de WiFi.

- **BYTE SleepNode(*radioInterface forceWakeUp*, *UINT32 slpTime_ms*)**

Descripción: Esta función se invoca para poner el microcontrolador en modo de bajo consumo, deteniendo el reloj del sistema y durmiendo las interfaces que no puedan despertar prematuramente al nodo. En caso de que ninguna fuente de interrupción despierte al nodo, se despierta automáticamente transcurrido el tiempo indicado en el segundo parámetro.

Parámetros: *forceWakeUp* – Interfaces radio que pueden despertar al nodo.

slpTime_ms – Tiempo máximo que dormirá el nodo, en milisegundos.

Valor de retorno: “NO_ERROR” o código de error.

- **BYTE WakeUpRadioInterface (*radioInterface ri*)**

Descripción: Es la función complementaria a SleepRadioInterface. Despierta el transceptor de la interfaz solicitada en caso de que estuviera dormido. Si se invoca con “ALL”, se activan todas las interfaces que estuvieran dormidas.

Parámetros: *ri* – Interfaz radio seleccionada. La invocación con “NONE” devuelve error.

Valor de retorno: “NO_ERROR” o código de error. En el caso de seleccionar todas las interfaces disponibles con “ALL”, comienza intentando despertar las interfaces a frecuencias más bajas y termina durmiendo el módulo WiFi (si el nodo dispone de él). Si alguna interfaz falla, devolverá el código de error de intentar despertar esa interfaz y abortará el proceso. La aplicación podrá averiguar qué interfaces se han despertado consultando el byte de flags de estado. En caso de invocarla con ALL_MIWI, es análogo a ALL pero excluyendo la interfaz de WiFi.

Nota: Las interfaces MiWi pueden despertarse invocando un “DATA REQUEST”. A priori, esta opción se ha obviado y se despiertan de manera “sencilla”.

- **BYTE SetTXPower(*radioInterface ri*, *BYTE power*)**

Descripción: Esta función permite cambiar la potencia de transmisión de las interfaces radio disponibles., indicando un nivel de potencia relativo entre 0 (máximo) y 255 (mínimo).

Parámetros: *ri* – Interfaz radio seleccionada. NONE y ALL son opciones no válidas.

power – Valor relativo de potencia a fijar en el transceptor. Cada transceptor solo permite un conjunto de valores reducido. El rango de 256 se trunca a los valores permitidos por “pasos equidistantes”, por lo que pequeños incrementos en el parámetro de entrada pueden no producir un cambio en la potencia de salida. Además, en

algunos transceptores, sucesivos pasos en el nivel de potencia suponen distintas variaciones en el nivel de salida.

Valor de retorno: “NO_ERROR” o código de error.

Nota: Los valores de potencia teóricos del MRF24J40 varían entre 0dBm (máx.) a -36,3 dBm (mín.) en 32 pasos irregulares. Los valores de potencia teóricos del MRF49XA varían entre 0 dBm y -17,5 dBm en 8 pasos de 2,5 dBm. La ganancia de la antena o la desadaptación del circuito de antena pueden suponer variaciones a dichos valores. Falta implementarlo en WiFi.

- **BYTE GetTXPower(radioInterface ri, BYTE *storeItHere)**

Descripción: Esta función permite leer el valor de ajuste de la potencia de transmisión de la interfaz radio seleccionada.

Parámetros: **ir** – Interfaz radio seleccionada. Las opciones NONE, ALL_MIWI y ALL no son válidas.

***storeItHere** – Byte donde se almacenará el valor de ajuste de potencia actual de la interfaz radio seleccionada (en caso de éxito).

Valor de retorno: “NO_ERROR” o código de error.

Nota: La potencia puede fijarse con SetTXPower a cualquier valor de 8 bits, siendo 0 el valor de potencia máximo y 255 el mínimo. Ese valor se trunca internamente a un valor que acepte el transceptor y se almacena en la HAL. Éste último es el que se lee mediante GetTXPower, por lo que puede suceder que se escriba un valor y se lea otro distinto. Por la misma razón se recuerda que, pequeñas variaciones al cambiar el ajuste de potencia pueden no suponer una variación en la potencia de salida. Los valores de ajuste de potencia son equidistantes pero no suponen variaciones lineales en términos generales.

- **BYTE PutTXData(radioInterface ri, BYTE data)**

Descripción: Escribe un dato en el buffer de transmisión de la interfaz indicada salvo que el nodo esté dormido, la interfaz esté dormida o no esté disponible, o el buffer de transmisión esté lleno. Si se llena el buffer al escribir el dato, activa el indicador de buffer lleno. Si se invoca con ALL_MIWI o ALL se intenta escribir el dato en todas las interfaces seleccionadas, y si falla en alguna de ellas, devuelve el código de error correspondiente.

Parámetros: **ri** – Interfaz radio seleccionada. NONE no es una opción válida en esta función.

data – Proporciona el dato a escribir en el buffer.

Valor de retorno: “NO_ERROR” o código de error.

- **BYTE DiscardTXData(radioInterface ri)**

Descripción: Descarta los datos del buffer de transmisión de la interfaz seleccionada para poder escribir un nuevo paquete.

Parámetros: **ri** – Interfaz radio seleccionada. NONE no es una opción válida en esta función.

Valor de retorno: “NO_ERROR” o código de error.

- **BYTE SendPckt(radioInterface ri, BYTE AddrMode, BYTE *Address)**

Descripción: Envía un paquete radio con todo lo que se ha escrito en el buffer de transmisión de la interfaz radio seleccionada. El segundo parámetro indica qué tipo de dirección es la que se pasa como puntero en el tercer parámetro. Una vez enviado el paquete, se resetea el puntero del buffer de transmisión para poder enviar un nuevo paquete. Si falla, puede intentar retransmitirse, pero en caso de éxito la retransmisión implica escribir los datos de nuevo.

Parámetros: **ri** – Interfaz radio seleccionada. La opción “NONE” devolverá error.
 AddrMode – tipo de dirección empleada. Se definen máscaras para simplificar la interfaz.

***Address** – Puntero a la dirección del nodo destino.

Valor de retorno: “NO_ERROR” o código de error.

- ***BYTE WhichRIHasData()***

Descripción: Informa a la aplicación sobre qué interfaces tienen datos pendientes de ser leídos.

Parámetros: Ninguno.

Valor de retorno: *Un byte que informa de las interfaces que tienen datos en caso de éxito, o código de error. Se recomienda hacer uso de las máscaras de interfaces radio para comprobar cada interfaz.*

- ***BYTE GetRXData(radioInterface ri, BYTE *storeItHere)***

Descripción: Recoge el siguiente dato (byte) del buffer de recepción de la interfaz indicada como parámetro y lo almacena en la dirección suministrada en el segundo parámetro. Si el nodo está dormido, la interfaz está dormida o no está disponible, o se invoca cuando no hay datos que leer, devolverá un código de error.

Parámetros: **ri** – Interfaz radio seleccionada. Las opciones NONE, ALL_MIWI y ALL no son válidas.

***storeItHere** – Byte donde se almacenará el payload leído.

Valor de retorno: “NO_ERROR” o código de error.

Nota: Una vez leído un dato, no se puede volver a leer. Se va actualizando una variable para leer el dato siguiente y cuando no quedan datos por leer se descarta el paquete completo del buffer de recepción de la interfaz para poder recibir el próximo. Es responsabilidad de la aplicación almacenar y/o procesar los datos leídos.

- ***BYTE GetRXSourceAddr(radioInterface ri, BYTE *storeItFromHere)***

Descripción: Obtiene la dirección origen del paquete radio recibido por la interfaz especificada.

Parámetros: **ri** – Interfaz radio seleccionada. Las opciones NONE, ALL_MIWI y ALL no son válidas.

***storeItFromHere** – Byte a partir del cual se copiará la dirección de origen del paquete recibido.

Valor de retorno: “NO_ERROR” o código de error.

Valor de retorno: devuelve un byte con un valor entero no negativo informando del número de conexiones almacenadas o con un código de error para identificar la causa del problema.

Nota: Las interrupciones se deshabilitan temporalmente para acceder a los datos de manera segura. Esta función requiere un espacio en memoria menor o igual al número de conexiones de la tabla (CONNECTION_SIZE – configurable) por el número de bytes que ocupa cada conexión (MIWI_CONN_ENTRY_SIZE – constante dependiente del protocolo empleado).

- ***BYTE RestoreConnTable(BYTE *takeItFromHere, BYTE numConn)***

Descripción: Restaura las conexiones válidas almacenadas a partir de la dirección facilitada. Se escribe en las primeras entradas de la tabla de conexiones la información relativa a el número de conexiones que indica el parámetro numConn, y el resto de entradas se marcan como conexiones no válidas.

Parámetros: ***takeItFromHere** – Puntero a la dirección base de memoria a partir de la cual se leerán las conexiones a restablecer.

numConn – número de conexiones almacenadas que serán restauradas.

Valor de retorno: devuelve “NO_ERROR” o un código de error.

Nota: Las interrupciones se deshabilitan temporalmente para acceder a los datos de manera segura. Es importante recalcar que los datos restaurados, en caso de no ser válidos, estar modificados o ser incongruentes con variables y definiciones del nodo en cuestión o de algún otro nodo de la red (direcciones, rol del nodo, estado...), pueden causar problemas de funcionamiento en la red.

- ***BYTE SaveRoutingTable(radioInterface ri, BYTE *storeItFromHere)***

Descripción: Esta es una función exclusiva de los nodos coordinadores y el protocolo MiWi. En caso contrario no se define. Guarda la tabla de rutado de la interfaz MiWi seleccionada a partir de la dirección de memoria pasada como parámetro.

Parámetros: **radioInterface ri** – Interfaz radio seleccionada. Solo valen las opciones de MiWi.

***storeItFromHere** – Puntero a la dirección base de memoria a partir de la cual se almacenarán los datos.

Valor de retorno: devuelve “NO_ERROR” o un código de error.

Nota: Las interrupciones se deshabilitan temporalmente para acceder a los datos de manera segura. La tabla de rutado de cada interfaz MiWi son 8 bytes. En caso de emplear la opción de interfaz ALL_MIWI, podrían almacenarse hasta 24 bytes consecutivamente.

- ***BYTE RestoreRoutingTable(radioInterface ri, BYTE *takeItFromHere)***

Descripción: Esta es una función exclusiva de los nodos coordinadores y el protocolo MiWi. En caso contrario no se define. Restaura la tabla de rutado de la interfaz MiWi seleccionada almacenada a partir de la dirección de memoria pasada como parámetro.

Parámetros: **radioInterface ri** – Interfaz radio seleccionada. Solo valen las opciones de MiWi.

***takeItFromHere** – Puntero a la dirección base de memoria a partir de la cual se leerán los datos.

Valor de retorno: devuelve “NO_ERROR” o un código de error.

Nota: Las interrupciones se deshabilitan temporalmente para acceder a los datos de manera segura. Es importante recalcar que los datos restaurados, en caso de no ser válidos, estar modificados o ser incongruentes con variables y definiciones del nodo en cuestión o de algún otro nodo de la red (direcciones, rol del nodo, estado...), pueden causar problemas de funcionamiento en la red.

- ***void SWDelay(UINT16 ms)***

Descripción: Esta es una función de depuración. Genera un retardo software del valor deseado.

Parámetros: **UINT16 ms** – Valor de 16 bits que indica el retardo en milisegundos.

Valor de retorno: ninguno.

- ***BYTE DumpRXPckt(radioInterface ri)***

Descripción: Esta es una función de depuración. Vuelca el contenido de un paquete radio por pantalla (hiperterminal).

Parámetros: **radioInterface ri** – Interfaz radio seleccionada. ALL_MIWI, ALL y NONE son opciones no válidas.

Valor de retorno: devuelve “NO_ERROR” o un código de error.

- ***void DumpConnectionTable()***

Descripción: Esta es una función de depuración. Vuelca el contenido de la tabla de conexiones por pantalla (hiperterminal).

Parámetros: ninguno.

Valor de retorno: ninguno.

- ***void DumpSingleConn(BYTE index)***

Descripción: Esta es una función de depuración. Vuelca el contenido de una conexión de la tabla de conexiones por pantalla (hiperterminal). Si es un índice no válido vuelca toda la tabla de conexiones.

Parámetros: ninguno.

Valor de retorno: ninguno.

- ***void AllStacksTasks()***

Descripción: Con esta función se ofrece la posibilidad de que sea la propia aplicación la que ejecute las tareas de mantenimiento de las interfaces presentes en el nodo, evitando así ser interrumpida por el nodo para realizar esta tarea. Esta función únicamente está disponible si se habilita la opción “APP_DOES_MAINTENANCE_TASKS”, en cuyo caso es importante que la aplicación ejecute frecuentemente la función de mantenimiento para garantizar el correcto funcionamiento de las pilas.

En caso contrario se declara “*static*” y el nodo se encarga de ejecutar la función periódicamente habilitando un temporizador con muy baja prioridad de interrupción.

Parámetros: Ninguno.

Valor de retorno: Ninguno.

