

**ESCUELA TÉCNICA SUPERIOR DE  
INGENIERÍA DE TELECOMUNICACIÓN**

**UNIVERSIDAD DE MÁLAGA**



**PROYECTO FIN DE CARRERA**

*PROGRAMACIÓN DE UN NODO CONMUTADOR PARA  
LA GESTIÓN REMOTA DE REDES 802.15.4/ZIGBEE*

**INGENIERÍA DE TELECOMUNICACIÓN**

MÁLAGA, 2011

SERGIO ARÉVALO BARAZAS

**ESCUELA TÉCNICA SUPERIOR DE  
INGENIERÍA DE TELECOMUNICACIÓN**

**UNIVERSIDAD DE MÁLAGA**

**Titulación: Ingeniería de Telecomunicación**

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

D./D<sup>a</sup>. \_\_\_\_\_

D./D<sup>a</sup>. \_\_\_\_\_

D./D<sup>a</sup>. \_\_\_\_\_

para juzgar el Proyecto Fin de Carrera titulado:

**PROGRAMACIÓN DE UN NODO CONMUTADOR PARA LA  
GESTIÓN REMOTA DE REDES 802.15.4/ZIGBEE**

del alumno D. *Sergio Arévalo Barazas*

dirigido por D. *Eduardo Casilari Pérez*

ACORDÓ POR \_\_\_\_\_ OTORGAR LA  
CALIFICACIÓN DE \_\_\_\_\_

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

Málaga, a \_\_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_

El/La Presidente/a

El/La Vocal

El/La Secretario/a

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

**ESCUELA TÉCNICA SUPERIOR DE  
INGENIERÍA DE TELECOMUNICACIÓN**

**UNIVERSIDAD DE MÁLAGA**

**PROGRAMACIÓN DE UN NODO CONMUTADOR PARA LA GESTIÓN  
REMOTA DE REDES 802.15.4/ZIGBEE**

**REALIZADO POR:**

*Sergio Arévalo Barazas*

**DIRIGIDO POR:**

*Eduardo Casilari Pérez*

**DEPARTAMENTO DE:** *Tecnología Electrónica*

**TITULACIÓN:** Ingeniería de Telecomunicación

**PALABRAS CLAVE:** ZigBee, 802.15.4, TIMAC, CC2530

**RESUMEN:** En este proyecto se lleva a cabo la programación de una plataforma para la configuración de nodos CC2530 de Texas Instruments bajo la norma IEEE 802.15.4, permitiendo la creación de redes balizadas y su posterior estudio.

# Agradecimientos

Me gustaría comenzar dando las gracias a mis padres, Francisco y María Pilar, por su cariño, paciencia y apoyo constante durante todos estos años. Gracias por estar ahí tanto en los momentos “buenos” como en los “malos”. Gracias por darme esta oportunidad y tantas otras, compartiendo mis proyectos e ilusiones. Sin vosotros nada de esto habría sido posible.

Agradecer a mi tutor, Eduardo Casilari, por su paciencia durante estos dos últimos años y por su atención e inestimable ayuda.

No quisiera olvidar a mis hermanos, cuñadas y sobrinos, por aportar una nota de color incluso en los días más grises.

Gracias a esas personas que, de un modo u otro, han influido de manera positiva en mi vida y me han hecho crecer como persona. Gracias a vosotros, amigos, por tan buenos momentos (y que no cambie la cosa!).

*Last but not least...* Gracias Cris, gracias por emprender a mi lado esta aventura que no tiene fronteras, gracias por confiar siempre en mí.



# ACRÓNIMOS

ADC: *Analog-to-Digital Converter*

AES: *Advanced Encryption Standard*

AP: *Access Point*

APS: *Application Support*

ASK: *Amplitude Shift Keying*

BE: *Back-off Exponent*

BI: *Beacon Interval*

BO: *Beacon Order*

BPSK: *Binary Phase Shift Keying*

BSN: *Beacon Sequence Number*

CS: *Carrier Sense*

CCA: *Clear Channel Assessment*

CSMA-CA: *Carrier Sense Multiple Access with Collision Avoidance*

CW: *Contention Window length*

BSS: *Basic Service Set*

BTLE: *Bluetooth Low Energy*

ED: *Energy Detection*

FCS: *Frame Check Sequence*

FHSS: *Frequency Hop Spread Spectrum*

## ACRÓNIMOS

FFD: *Full Function Device*

FIFO: *First In First Out*

GTS: *Guaranteed Time Slot*

HAL: *Hardware Abstraction Layer*

IEEE: *Institute of Electrical and Electronics Engineers*

ISM: *Industrial, Scientific & Medical*

ISO: *International Standards Organization*

LAN: *Local Area Network*

LCD: *Liquid Crystal Display*

LQI: *Link Quality Indicator*

LR-WPAN: *Low Rate Wireless Personal Area Network*

MAC: *Medium Access Control*

MFR: *MAC Footer*

MHR: *MAC Header*

MIMO: *Multiple Input and Multiple Output*

MPDU: *MAC Protocol Data Unit*

MSA: *MAC Sample Application*

NB: *Number of Back-offs*

NCP: *Network Co-Processor*

NIC: *Network Interface Card*

NWK: *Network*

O-QPSK: *Orthogonal Quadrature Phase Shift Keying*

OSAL: *Operating System Abstraction Layer*

PC: *Personal Computer*

PCI: *Peripheral Component Interconnect*

PCMCIA: *Personal Computer Memory Card International Association*

PDA: *Personal Digital Assistant*

PHR: *Physical Header*

POS: *Personal Operating Space*

PPDU: *PHY Protocol Data Unit*

PWM: *Pulse Width Modulator*

RFD: *Reduced Function Device*

RSS: *Received Signal Strength*

SD: *Superframe Duration*

SFR: *Special Function Register*

SHR: *Synchronization Header*

SIG: *Special Interest Group*

SMA: *SubMiniature version A*

SNR: *Signal to Noise Ratio*

SO: *Superframe Order*

SoC: *System on Chip*

SPI: *Serial Peripheral Interface*

SRAM: *Static Random-Access Memory*

UART: *Universal Asynchronous Receive Transmit*

USART: *Universal Synchronous/Asynchronous Receiver/Transmitter*

USB: *Universal Serial Bus*

WBAN: *Wireless Body Area Network*

Wi-Fi: *Wireless Fidelity*

WLAN: *Wireless Local Area Network*

WPAN: *Wireless Personal Area Network*

WSN: *Wireless Sensor Network*

ZDO: *ZigBee Device Object*



*ACRÓNIMOS*

# Índice General

<b>1. Introducción .....</b>	<b>1</b>
1.1. Marco tecnológico .....	2
1.1.1. Redes inalámbricas WLAN. Redes Wi-Fi .....	2
1.1.2. Redes inalámbricas WPAN, LR-WPAN y WSN.....	4
1.1.2.1. Bluetooth .....	5
1.1.2.2. ZigBee .....	7
1.2. Objetivos del proyecto .....	9
1.3. Estructuración de la memoria .....	9
<b>2. Descripción del estándar 802.15.4/ZigBee .....</b>	<b>11</b>
2.1. Introducción .....	11
2.2. Tipos de dispositivos .....	12
2.3. Topologías de red .....	13
2.3.1. Topología en estrella .....	13
2.3.2. Topología “Peer-to-Peer” .....	14
2.4. Capa física .....	16
2.4.1. Selección de canal .....	16
2.4.2. Detección de energía (ED) .....	17
2.4.3. Indicación de la calidad del enlace .....	18
2.4.4. Evaluación del canal libre (CCA) .....	18
2.4.5. Transmisión y recepción de datos .....	19
2.5. Capa MAC .....	20
2.5.1. Modos de funcionamiento de red .....	20
2.5.2. Algoritmo CSMA-CA .....	23
2.5.3. Inicio y mantenimiento de redes de área personal .....	25

2.5.4. Formato de trama MAC .....	27
2.6. Implementación del estándar ZigBee .....	29
2.6.1. Capa de red .....	29
2.6.2. Capa de aplicación .....	32
2.6.3. Seguridad .....	33
<b>3. Elección de tecnologías .....</b>	<b>35</b>
3.1. Introducción .....	35
3.2. Elección del sistema de desarrollo .....	37
3.2.1. Kit de desarrollo CC2520DK de Texas Instruments .....	37
3.2.2. Kit de desarrollo MC1322X de Freescale .....	38
3.2.3. Kit de desarrollo CC2530DK de Texas Instruments .....	39
3.3. Kit de desarrollo CC2530ZDK .....	39
3.3.1. Módulo CC2530EM .....	41
3.3.2. Módulo CC2531 USB Dongle .....	41
3.3.3. Placa de desarrollo SmartRF05EB .....	42
3.3.4. Circuito integrado CC2530 .....	48
3.4. Pila de protocolos TIMAC .....	51
3.4.1. Capa OSAL .....	52
3.4.2. Capa HAL .....	53
3.4.3. Capa MAC .....	54
3.4.4. Capa de aplicación .....	55
3.5. Herramientas software .....	56
3.5.1. IAR Embedded Workbench EW8051 .....	56
3.5.2. LabWindows/CVI .....	57
3.5.3. Packet Sniffer .....	58
<b>4. Desarrollo software .....</b>	<b>61</b>
4.1. Introducción .....	61
4.2. Especificaciones .....	62
4.3. Fase de desarrollo .....	63
4.3.1. Configuración en placas de desarrollo .....	64
4.3.2. Plataforma de configuración vía puerto serie .....	72
4.3.2.1. Interfaz gráfica de usuario .....	72
4.3.2.2. Modificación de la pila de protocolos TIMAC .....	79

<b>5. Fase de pruebas .....</b>	<b>85</b>
5.1. Aplicación SMARTRF05EB .....	88
5.2. Aplicación de configuración remota.....	94
5.3. Funcionamiento de redes balizadas .....	98
5.3.1. Formación de redes .....	98
5.3.2. Sincronización en redes balizadas .....	106
<b>6. Manual de usuario .....</b>	<b>109</b>
6.1. Configuración de la placa SmartRF05EB.....	109
6.2. Aplicación SMARTRF05EB .....	112
6.2.1. Funcionalidad de los periféricos de la placa .....	112
6.2.2. Configuración de motas CC2530 .....	115
6.3. Aplicación de configuración remota .....	117
6.3.1. Instalación .....	117
6.3.2. Configuración de motas CC2530 .....	120
<b>7. Conclusiones y líneas futuras de trabajo .....</b>	<b>125</b>
7.1. Conclusiones .....	125
7.2. Líneas futuras de trabajo .....	127
<b>Bibliografía .....</b>	<b>129</b>

*ÍNDICE GENERAL*

# Índice de Figuras

Figura 1.1. Red WLAN funcionando en modo de infraestructura. Fuente: [2] .....	4
Figura 1.2. Espacio de operación de WPAN y WLAN. Fuente: [3] .....	5
Figura 1.3. Duración de las baterías en función del ciclo de trabajo. Fuente: [7] .....	8
Figura 2.1. Capas del protocolo ZigBee. Fuente: [10] .....	11
Figura 2.2. Topologías en redes 802.15.4. Fuente: [11] .....	14
Figura 2.3. Topología en Árbol .....	15
Figura 2.4. Formato de trama PPDU. Fuente: [11] .....	19
Figura 2.5. Estructura de una Supertrama. Fuente: [10] .....	21
Figura 2.6. Diagrama de Flujo del Algoritmo CSMA-CA. Fuente: [11] .....	24
Figura 2.7. Formato de la Trama de Baliza de la capa MAC. Fuente [10] .....	27
Figura 2.8. Formato de la Trama de Datos de la capa MAC. Fuente: [10] .....	28
Figura 2.9. Formato de la Trama ACK de la capa MAC. Fuente: [10] .....	28
Figura 2.10. Formato de la Trama de Comando de la capa MAC. Fuente: [10] .....	28
Figura 2.11. Capa de Aplicación del protocolo ZigBee .....	32
Figura 3.1. Kit de desarrollo CC2530ZDK. Fuente: [14] .....	40
Figura 3.2. Módulo de evaluación CC2530EM.Fuente: [14] .....	41
Figura 3.3. Módulo CC2531 USB Dongle. Fuente: [18] .....	41
Figura 3.4. Placa de Desarrollo SmartRF05EB de Texas Instruments. Fuente: [14] ....	42
Figura 3.5. Arquitectura de la placa SmartRF05EB. Fuente: [16] .....	43
Figura 3.6. Jumper de selección de método de alimentación de la placa SmartRF05EB. Fuente: [16] .....	43
Figura 3.7. Interconexión de dispositivos mediante UART RS232. Fuente: [16] .....	44
Figura 3.8. Esquema de conexión entre el EM y cabeceras de jumpers. Fuente: [16] ...	47
Figura 3.9. Diagrama de bloques de la familia de chips CC253x. Fuente [15] .....	48

Figura 3.10. Estructura de la pila de protocolos TIMAC .....	52
Figura 3.11. Interfaz de usuario de IAR EW8051 v7.60.1 .....	56
Figura 3.12. Interfaz de usuario de LabWindows/CVI .....	58
Figura 3.13. Ejemplo de captura de datos con Packet Sniffer de Texas Instruments.....	59
Figura 4.1. Diagrama de flujo de funcionamiento básico del sistema .....	64
Figura 4.2. Ejemplo de ventana de edición de un control del tipo Ring Slide .....	74
Figura 4.3. Layout de la aplicación de configuración de nodos 802.15.4 .....	75
Figura 4.4. Ejemplo de datos de configuración codificados .....	78
Figura 5.1. Sistema de pruebas .....	86
Figura 5.2. Diagrama de flujo para las pruebas de “caja blanca” .....	93
Figura 5.3. Mensaje de error al producirse fallo de envío de datos .....	97
Figura 5.4. Configuración del coordinador con parámetros del Escenario 1 .....	99
Figura 5.5. Captura de tramas de los procesos de: a) Inicialización de red balizada y asociación de dispositivo final, b) Transmisión de mensaje de dispositivo final a coordinador de red y c) Transmisión de mensaje de coordinador de red a dispositivo final. (Escenario 1) .....	100
Figura 5.6. Intercambio de mensajes durante el proceso de asociación a una red .....	101
Figura 5.7. Proceso de asociación simultánea a red de varios dispositivos finales (Escenario 2) .....	102
Figura 6.1. Switch de selección de módulo de evaluación .....	109
Figura 6.2. Selectores para la configuración de la placa SmartRF05EB. a) Jumper para la selección del modo de alimentación. b) Switch de encendido de la placa SmartRF05EB. c) Switch para la habilitación del puerto de comunicaciones RS232 .....	110
Figura 6.3. Ubicación de los periféricos en la placa SmartRF05EB .....	111
Figura 6.4. Instalación de la aplicación (pantalla inicial) .....	118
Figura 6.5. Instalación de la aplicación (selección de ruta de instalación) .....	118
Figura 6.6. Instalación de la aplicación (acuerdo de licencia) .....	119
Figura 6.7. Instalación de la aplicación (software a instalar) .....	119
Figura 6.8. Instalación de la aplicación (final de la instalación) .....	120
Figura 6.9. Interfaz gráfica de la aplicación de configuración remota .....	122
Figura 6.10. Mensaje de ayuda para la configuración del parámetro Beacon Order ...	122
Figura 6.11. Mensaje de error del puerto de comunicaciones .....	123

# Índice de Tablas

Tabla 1.1. Comparativa de estándares 802.11. Fuente: [1] .....	3
Tabla 1.2. Comparativa de tecnologías de comunicación inalámbricas .....	8
Tabla 2.1. Asignación de canales .....	17
Tabla 4.1. Lista de parámetros de configuración .....	65
Tabla 4.2. Estructura de datos para el almacenamiento del menú Potencia de Transmisión .....	66
Tabla 4.3. Opciones de impresión para la función HalLcdWriteString .....	66
Tabla 4.4. Correspondencia entre el número de canal y su frecuencia asociada .....	67
Tabla 4.5. Codificación hexadecimal de la potencia de transmisión en los módulos CC2530 .....	68
Tabla 4.6. Flags de interrupción asociados a los botones Joystick y S1 .....	70
Tabla 4.7. Modos de funcionamiento de los controles en LabWindows/CVI .....	73
Tabla 4.8. Parámetros de configuración del puerto serie en la placa SmartRF05EB ....	80
Tabla 4.9. Eventos asociados a la UART de la placa SmartRF05EB .....	80
Tabla 5.1. Características de los equipos que integran el subsistema de configuración	87
Tabla 5.2. Características de los equipos que integran el subsistema de monitorización .....	87
Tabla 5.3. Test de funcionamiento de los botones de la placa SmartRF05EB .....	88
Tabla 5.4. Test de los distintos modos de alimentación de la placa SmartRF05EB .....	89
Tabla 5.5. Test de los diodos LED .....	89
Tabla 5.5. (Continuación) Test de los diodos LED .....	90
Tabla 5.6. Test LCD (menús de configuración) .....	90
Tabla 5.7. Test LCD (mensajes en ejecución) .....	91
Tabla 5.8. Comprobación de los rangos de valores asignados a los parámetros .....	92



Tabla 5.9. Test de compatibilidad con diversos sistemas operativos .....	94
Tabla 5.10. Test de funcionamiento de los controles de la aplicación .....	95
Tabla 5.10. (Continuación) Test de funcionamiento de los controles de la aplicación...	96
Tabla 5.11. Direcciones de red asignadas por el coordinador (Escenario 2) .....	103
Tabla 5.12. Asignación real de direcciones de red (Escenario 2) .....	104
Tabla 5.13. Estudio de la pérdida de sincronización en redes balizadas .....	107
Tabla 5.14. Comparación entre el tiempo teórico entre balizas y el tiempo real medio .....	108
Tabla 6.1. Selección del modo de alimentación mediante el jumper P11 .....	110
Tabla 6.2. Menús de configuración mostrados en la pantalla LCD .....	112
Tabla 6.3. Mensajes de estado e instrucciones mostrados en la pantalla LCD .....	113
Tabla 6.4. Funcionalidad de los botones de la placa SmartRF05EB .....	114
Tabla 6.5. Funcionalidad de los diodos LED de la placa SmartRF05EB .....	114
Tabla 6.5. (Continuación) Funcionalidad de los diodos LED en la placa SmartRF05EB .....	115

# CAPÍTULO 1

## Introducción

Actualmente nos hallamos inmersos y formamos parte de la sociedad de la información, y uno de los mayores retos de la Ingeniería moderna es poner a disposición del usuario dispositivos que faciliten el acceso a esta.

Desde los primeros ordenadores hasta las más modernas PDA (*Personal Digital Assistant*) y teléfonos móviles, la Ingeniería ha evolucionado de manera vertiginosa en los últimos años, logrando crear dispositivos cada vez más rápidos, pequeños y versátiles. Sin embargo, el desarrollo de estos dispositivos carecería de sentido de no ser por la implantación de infraestructuras (tanto físicas como lógicas) que permitan su interconexión. Así, de manera paralela a la creación de los dispositivos finales de usuario, se han ido desarrollando toda clase de redes de interconexión, desde las primeras redes cableadas LAN (*Local Area Network*), hasta las redes inalámbricas WLAN (*Wireless Local Area Network*), WPAN (*Wireless Personal Area Network*), LR-WPAN (*Low Rate Wireless Personal Area Network*) o WSN (*Wireless Sensor Network*).

Son este último tipo de redes, las inalámbricas, las que más están evolucionando en las dos últimas décadas, pues dan respuesta a una de las necesidades más demandadas por los usuarios: la movilidad, poder deshacerse de los cables y acceder a la información en cualquier lugar.

Hoy en día casi nadie es ajeno a palabras como Wi-Fi (*Wireless Fidelity*) o *Bluetooth*, dos de las tecnologías sin cables más extendidas hasta el momento, y que han ocupado un importante papel en el desarrollo de las telecomunicaciones a nivel mundial. Estas tecnologías forman parte de un amplio catálogo capaz de proporcionar soluciones

específicas para prácticamente cualquier necesidad de cobertura, velocidad de transmisión, coste o consumo. En nuestro proyecto nos centraremos en las redes inalámbricas de área personal y de sensores, concretamente en las de baja tasa de transmisión y bajo consumo.

## 1.1. MARCO TECNOLÓGICO

### 1.1.1. REDES INALÁMBRICAS WLAN. REDES WI-FI.

Las redes inalámbricas WLAN surgen como una extensión sin cables de las redes LAN, ofreciendo como principal ventaja la posibilidad de conexión inalámbrica de alta velocidad en entornos domésticos y de negocio a redes locales. Otra ventaja muy importante que ofrecen las WLAN, es la reducción de costes de instalación ya que, en general, no se requieren obras o reformas. El área de cobertura varía aproximadamente entre los 20 y los 200 metros, dependiendo de la potencia transmitida y de los obstáculos que haya en el entorno.

El origen de las redes WLAN se remonta al año 1979 con la publicación, por parte del IEEE (*Institute of Electrical and Electronics Engineers*), de los resultados satisfactorios obtenidos por un grupo de ingenieros de IBM, que conectaron varios ordenadores mediante enlaces de infrarrojos para formar una red local en una fábrica de Suiza.

Es en el año 1990 cuando se reúne el comité IEEE 802 y crea un nuevo grupo de trabajo, dedicado en exclusiva al desarrollo de estándares para redes WLAN, el grupo IEEE 802.11, pero tuvieron que pasar 7 años para que se ratificara la norma 802.11, que en su primera versión alcanzaba tasas de transmisión de hasta 2 Mbps. En la tabla 1.1 podemos ver una comparativa de los estándares 802.11 más relevantes, concretamente algunos de los certificados por la “*Wi-Fi Alliance*”.

El estándar IEEE 802.11, comúnmente conocido como “Wi-Fi”, contempla dos tipos de elementos de red: las estaciones inalámbricas (o STA) y los AP (*Access Point*). Los STA son, en general, ordenadores personales, PDA o teléfonos móviles equipados con un interfaz de red inalámbrico (NIC: *Network Interface Card*). Inicialmente estas tarjetas de red se adquirirían de manera independiente a las estaciones de trabajo, con interfaces tan conocidas como PCI (*Peripheral Component Interconnect*), PCMCIA (*Personal Computer Memory Card International Association*) o USB (*Universal Serial Bus*). Sin embargo,

debido a la aceptación universal de esta tecnología y al abaratamiento de los procesos de fabricación de componentes electrónicos, existe una gran variedad de dispositivos que integran directamente en sus chips dicho interfaz inalámbrico. Por su parte, los AP son nodos que suelen formar parte de una red cableada (aunque también pueden estar interconectados por Wi-Fi o Wimax), que cuentan con un interfaz inalámbrico y son capaces de dar acceso a la red a múltiples estaciones móviles.

<i>Estándar</i>	<i>Descripción</i>	<i>Año</i>	<i>Frecuencia (GHz)</i>	<i>Tasa máx. (Mbps)</i>	<i>Alcance máx.(m)</i>
802.11	Estándar WLAN original	1997	2.4	2	100
802.11a	Estándar WLAN de alta velocidad en la banda de 5 GHz	1999	5	54	120
802.11b	Mejora velocidad respecto al 802.11 original. Adopta como nombre Wi-Fi	1999	2.4	11	140
802.11g	Establece modulación adicional para la banda de 2.4 GHz, aumentando velocidad	2003	2.4	54	140
802.11n	Mejora del alcance y velocidad en las bandas de 2.4 y 5GHz (uso de MIMO)	2009	2.4 y 5	600	200

**Tabla 1.1. Comparativa de estándares 802.11. Fuente: [1].**

Dependiendo de la forma de conectar los elementos de red anteriores se pueden definir hasta tres configuraciones para las redes WLAN:

- *Infraestructura*: en esta configuración se emula un sistema celular, en el que cada celda, o BSS (*Basic Service Set*), es el área de cobertura de un AP que forma parte de una red cableada. Los dispositivos inalámbricos que se encuentran dentro de la celda se conectan a la red o entre ellos mismos a través del AP. En la figura 1.1 se muestra un ejemplo de WLAN funcionando bajo esta configuración.
- *Configuración Ad hoc*: en este caso dos o más dispositivos 802.11 se conectan directamente entre ellos sin tener que canalizar previamente la información a través de un AP. A esta configuración también se la conoce como “*peer to peer*”.
- *Configuración mallada*: en esta configuración los AP se comunican mediante conexiones inalámbricas multisalto.

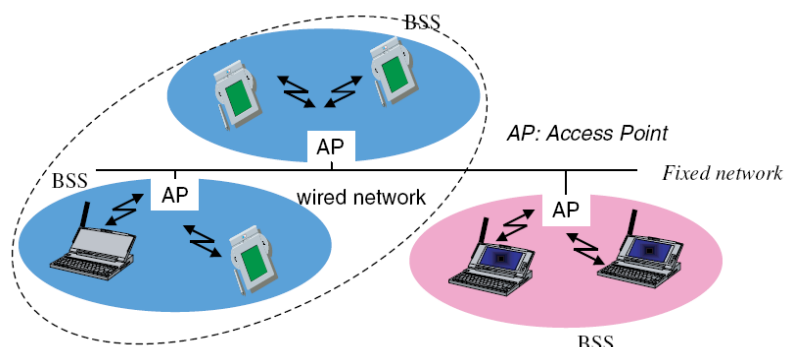


Figura 1.1. Red WLAN funcionando en modo de infraestructura. Fuente: [2].

### 1.1.2. REDES INALÁMBRICAS WPAN, LR-WPAN Y WSN

Dentro del catálogo de redes inalámbricas, las WPAN y las WSN enfocan su mercado en escenarios donde es prioritario reducir tanto el consumo energético como la complejidad de diseño, y por tanto, las tecnologías WLAN no ofrecen soluciones eficientes. Además este tipo de redes proporciona baja velocidad de transmisión y un área de cobertura reducida. En la figura 1.2 se muestra una pequeña comparativa de consumo, complejidad y velocidad de transmisión para diferentes tecnologías WLAN y WPAN.

El grupo de trabajo 802.15 del IEEE se encarga de regular toda la normativa referente a redes WPAN, y define 3 tipos en función del ámbito de aplicación de las mismas:

- *WPAN de alta velocidad:* responden a la norma IEEE 802.15.3, y se usan principalmente para aplicaciones en tiempo real y multimedia, proporcionando tasas de transmisión de hasta 55 Mbps.
- *WPAN de media velocidad:* fueron las primeras redes de área personal, usadas principalmente para sustituir cables por enlaces inalámbricos en conexiones de corta distancia. Están reguladas bajo la norma IEEE 802.15.1 (*Bluetooth*), que en su segunda versión alcanza hasta 3 Mbps.
- *WPAN de baja velocidad:* También conocidas como LR-WPAN, estas redes responden al estándar IEEE 802.15.4 y ofrecen tasas de transmisión de hasta 250 Kbps. Dan respuesta a aplicaciones muy específicas donde no es necesaria una gran velocidad de transmisión, o aplicaciones en las que los nodos de red estén

alimentados por baterías y el consumo de potencia sea un parámetro crítico, por ejemplo las redes de área corporal o WBAN (*Wireless Body Area Network*).

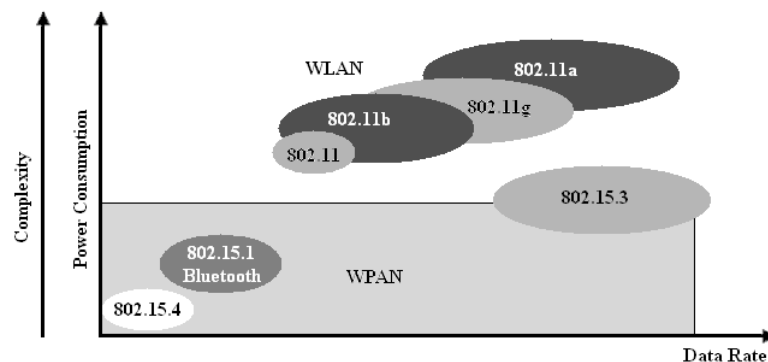


Figura 1.2. Espacio de operación de WPAN y WLAN. Fuente: [3].

Es conveniente destacar que, pese a que en la propia definición del estándar 802.15.4/ZigBee, éste está enmarcado dentro del ámbito de las redes LR-WPAN, son las redes inalámbricas de sensores (WSN) las que sacan más partido de esta tecnología.

Las WSN tratan de proporcionar conectividad inalámbrica a conjuntos de sensores y actuadores que se encuentren distribuidos en un determinado espacio, realizando tareas tan diversas como medición de fenómenos atmosféricos, seguimiento de objetos, monitorización de estructuras y detección de eventos, entre otros. Debido a que estos sensores pueden estar colocados en lugares de difícil accesibilidad, es necesario minimizar su consumo, de forma que puedan funcionar durante el mayor tiempo posible sin reemplazar las baterías que los alimentan.

### 1.1.2.1. Bluetooth

La idea de *Bluetooth* nace en el año 1994, cuando la compañía sueca Ericsson publicó un estudio sobre un interfaz radio de bajo coste y bajo consumo para la comunicación entre teléfonos móviles y sus accesorios. Debido a lo novedoso de esta idea y a las ventajas que aportaba, en el año 1998 las empresas IBM, INTEL, Nokia y Toshiba se unieron a Ericsson y formaron el SIG (*Special Interest Group*) de *Bluetooth* [5]. En la actualidad hay más de

2500 empresas unidas a este grupo, con la finalidad de desarrollar y promover esta tecnología.

Las primeras tecnologías de red WPAN se basaron en la especificación de *Bluetooth*, dando lugar a la formación del grupo de trabajo 802.15 del IEEE. Así, el estándar 802.15.1 se diseñó para tener total compatibilidad con la primera versión de *Bluetooth*, que respondía a las necesidades de bajo consumo, bajo coste de implementación y alcance reducido.

*Bluetooth* está basado en una arquitectura maestro-esclavo. Los dispositivos se organizan en celdas denominadas “*piconets*”, en las que puede haber hasta un máximo de 8 nodos, de los cuales uno es el maestro, encargado de organizar la comunicación entre los demás integrantes de la red. La banda de frecuencias de trabajo es la de los 2.4 GHz, con 79 canales de 1 MHz. Debido a que la banda ISM (*Industrial Scientific & Medical*) de 2.4 GHz (de uso libre) es muy utilizada por una gran cantidad de tecnologías, tales como redes 802.11 u hornos microondas, *Bluetooth* utiliza una técnica de modulación muy robusta, denominada FHSS (*Frequency Hop Spread Spectrum*), que la hace prácticamente inmune a las interferencias. Esta técnica divide la información a enviar en pequeños paquetes, los cuales se transmiten siguiendo una secuencia determinada a través de los 79 canales disponibles. La velocidad de transmisión que se alcanza en la primera versión del estándar es de 712 Kbps obteniendo un consumo de menos de 1mW para transmisiones a menos de 10 metros de distancia.

Una evolución de *Bluetooth*, todavía en una fase muy temprana de desarrollo, que trata de abrirse hueco en el mercado de las tecnologías de muy bajo consumo es “*Bluetooth Low Energy*” (BTLE), definido en la versión V.4 del estándar, aprobada en Julio de 2010. La principal ventaja que aporta es la optimización del consumo respecto a las versiones anteriores de *Bluetooth*, pudiendo así funcionar con una simple pila de botón durante varios años, lo que es una característica esencial para su implementación en redes de sensores. Esta reducción de consumo se consigue a partir de la modificación de la capa física y el formato de paquete, de forma que, para conseguir la compatibilidad entre los nuevos dispositivos BTLE y los antiguos, es necesario fabricar chips duales que soporten los dos modos de trabajo.

### 1.1.2.2. ZigBee

*ZigBee* es un estándar de comunicaciones inalámbricas desarrollado y promovido por la *ZigBee Alliance* [6], que trata de dar respuesta a las aplicaciones en las que el coste y el consumo de potencia son parámetros críticos de diseño. *ZigBee* está basado en la norma IEEE 802.15.4, que implementa las capas física (PHY) y de control de acceso al medio (MAC: *Medium Access Control*). Esta tecnología está enmarcada dentro de un subgrupo de las redes de área personal, las LR-WPAN (WPAN de baja velocidad de transmisión), y está orientada a la implementación de redes de sensores inalámbricos y actuadores, comunmente conocidas como WSN.

Dentro de las características más importantes de esta tecnología, se puede destacar que alcanza tasas de transmisión de hasta 250 Kbps y tiene un rango de cobertura de entre 10 y 75 metros, operando en las bandas ISM a las frecuencias de 2.4 GHz, 868 MHz (Europa), y 915 MHz (Estados Unidos).

Otra característica esencial de *ZigBee* es la optimización del consumo de los elementos que componen la red. Son los nodos finales (típicamente sensores) los elementos con menor consumo, ya que permiten ser configurados de forma que permanezcan inactivos hasta que ocurra un determinado evento y tengan que enviar información. Por su parte, en topologías de red complejas, en las que los nodos intermedios (*routers*) deben estar alimentados con baterías, existe la posibilidad de que éstos funcionen en modo balizado, consiguiendo que permanezcan inactivos la mayor parte del tiempo y sólo se despierten periódicamente para sincronizar sus relojes o recibir y retransmitir la información procedente de otros niveles de la red. Esta posibilidad, combinada con una baja velocidad de transmisión y corto alcance, hace que el consumo de potencia esté muy optimizado. De esta forma se consigue que un nodo pueda estar alimentado por el mismo par de baterías durante varios años. En la figura 1.3 se observa la evolución de la duración de las baterías de un dispositivo *ZigBee* en función de su ciclo de trabajo.

La posibilidad de crear redes que sigan una topología en malla hace que, ante la caída de un nodo, todo el sistema se pueda recuperar encontrando un nuevo camino de comunicación.



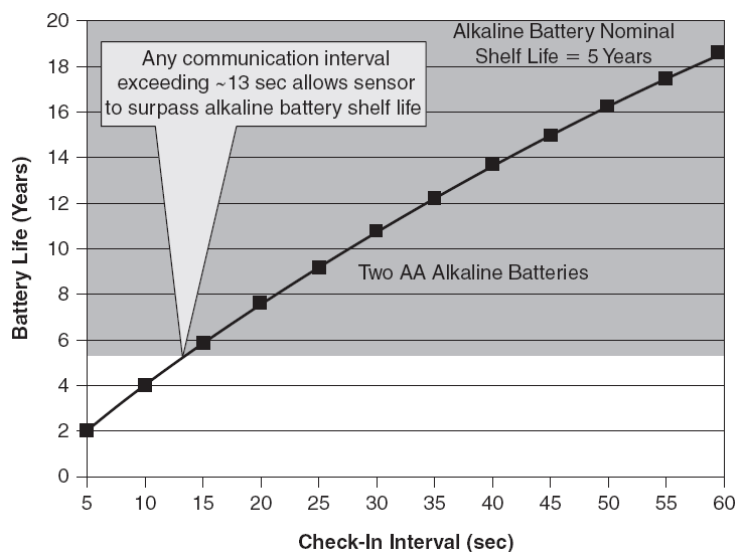


Figura 1.3. Duración de las baterías en función del ciclo de trabajo. Fuente: [7].

Para finalizar, en la siguiente tabla se muestra una comparativa entre redes LR-WPAN y dos de las tecnologías inalámbricas más implantadas hasta el momento, Wi-Fi y *Bluetooth*. Cabe destacar la conveniencia de uso de *ZigBee* en el despliegue de redes o aplicaciones en las que no es necesaria una gran velocidad de transmisión pero sí es importante minimizar la complejidad, el consumo y el coste.

	802.11b WLAN	Bluetooth WPAN	LR-WPAN
<b>Alcance</b>	~100 m	~10 – 100 m	10 m
<b>Tasa de Transmisión</b>	~2 – 11 Mbps	1 Mbps	< 0.25 Mbps
<b>Consumo</b>	Medio	Bajo	<b>Muy Bajo</b>
<b>Tamaño</b>	Grande	Pequeño	Muy Pequeño
<b>Coste/Complejidad</b>	Alto	Medio	Muy bajo

Tabla 1.2. Comparativa de tecnologías de comunicación inalámbricas.

## 1.2. OBJETIVOS DEL PROYECTO

El proyecto se centrará en el estudio de redes inalámbricas de área personal y baja tasa de transmisión que siguen la norma 802.15.4/ZigBee.

Una de las características más importantes que trata de impulsar a esta tecnología es la posibilidad de trabajar en modo balizado, minimizando el ciclo de trabajo de los nodos de red y aumentando así su autonomía. Este modo de funcionamiento ha sido objeto de estudio desde los inicios de *ZigBee*, y hoy día contamos con una gran cantidad de trabajos teóricos que lo optimizan hasta la saciedad.

Sin embargo llevar la teoría a la práctica no es tan sencillo, y es una realidad que, dentro del gran número de compañías que fabrican esta tecnología, son muy pocas las que ofrecen la posibilidad de habilitar el denominado modo balizado en sus dispositivos. Además, la documentación suministrada por los propios fabricantes es en muchos casos confusa, pues ofrecen dispositivos que según ellos son totalmente compatibles con las especificaciones del estándar 802.15.4, cuando, por el contrario, gran parte de las funcionalidades propias de la norma no están incluidas en las pilas de protocolo.

Por tanto, el objetivo principal del proyecto será programar una aplicación para la configuración remota de un nodo conmutador *ZigBee*, que permita habilitar el modo de funcionamiento balizado, y modificar los parámetros que lo regulan. De manera complementaria se realizará un estudio sobre la sincronización de los nodos en redes balizadas, y la formación de las mismas.

## 1.3. ESTRUCTURACIÓN DE LA MEMORIA

Para concluir este apartado, a continuación se muestra la estructura básica de la memoria, organizada por capítulos, resumiendo los aspectos más importantes que se van a tratar en cada uno de ellos.

- **Capítulo 1: Introducción.** En este apartado se han comentado los aspectos básicos de las tecnologías de red inalámbricas que más éxito están teniendo en el mercado, como son Wi-Fi, *Bluetooth* y *ZigBee*.

- **Capítulo 2: Descripción del estándar 802.15.4.** En este capítulo se profundiza en el estudio de la norma que implementa las capas física y de control de acceso al medio de *ZigBee*, haciendo énfasis en todos los parámetros que regulan el modo de funcionamiento balizado.
- **Capítulo 3: Elección de tecnologías.** En este capítulo se justifica la elección del *hardware* de entre las diferentes opciones ofrecidas por los fabricantes. Además se realiza un estudio detallado de los entornos de programación y análisis necesarios para la elaboración de la aplicación.
- **Capítulo 4: Desarrollo software.** Aquí se realiza un análisis pormenorizado de la fase de desarrollo del *software* de configuración de los nodos de red.
- **Capítulo 5: Fase de pruebas.** En este apartado se muestran las principales pruebas realizadas para la verificación de la aplicación elaborada. Además, se realiza un estudio sobre la pérdida de sincronización entre nodos de redes que estén funcionando en modo balizado.
- **Capítulo 6: Manual de usuario.** En este capítulo se explica el manejo de la aplicación realizada para la configuración remota de los nodos de red, aclarando la funcionalidad de todos y cada uno de los elementos que aparecen en ella.
- **Capítulo 7: Conclusiones y líneas futuras de trabajo.** Finalmente se realiza una pequeña reflexión sobre los resultados obtenidos durante la elaboración del proyecto, dejando abiertas posibles líneas de investigación que den continuidad al trabajo realizado.

# CAPÍTULO 2

## Descripción del estándar 802.15.4/ZigBee

### 2.1. INTRODUCCIÓN

El estándar 802.15.4 está enmarcado dentro del ámbito de las redes inalámbricas de área personal y baja velocidad de transmisión. Este tipo de redes se caracteriza por su simplicidad y bajo coste, proporcionando conectividad inalámbrica en aplicaciones que requieren un consumo de potencia muy reducido.

Los términos “ZigBee” y “802.15.4” son, en general, usados de manera indistinta. Sin embargo este uso es incorrecto, ya que la norma IEEE 802.15.4 únicamente define las capas física y de control de acceso al medio para redes LR-WPAN. Como se muestra en la figura 2.1, *ZigBee* usa las capas PHY (*Physical*) y MAC (*Medium Access Control*) del estándar 802.15.4 e implementa una solución propia para las capas de red, aplicación y seguridad.

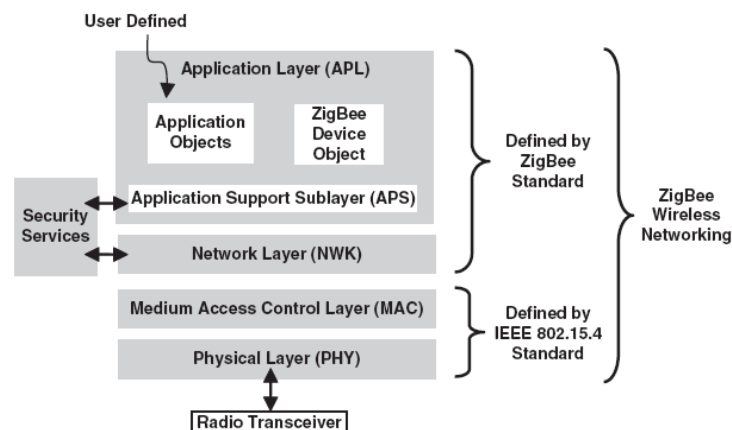


Figura 2.1. Capas del protocolo ZigBee. Fuente: [10].

El estándar 802.15.4 fue creado de manera independiente de *ZigBee*, dando opción a que los usuarios puedan desarrollar sus propias capas de red y aplicación, orientadas y optimizadas hacia las aplicaciones para las que se diseñan. Una ventaja que ofrece esta posibilidad es la de reducir la cantidad de memoria necesaria para cargar el protocolo, lo que puede implicar una reducción significativa de costes. Por el contrario, implementar una solución específica para estas capas hace que se pierda la interoperabilidad con redes y dispositivos que sí implementan el protocolo *ZigBee* completo.

Para la elaboración de este proyecto se va a utilizar la pila de protocolos TIMAC, una solución de Texas Instruments que implementa las capas PHY y MAC del estándar 802.15.4 y una pequeña aplicación que permite la formación de redes y transmisión de información.

En los siguientes apartados se va a tratar de dar una visión general de la norma IEEE 802.15.4, para así poder entender el funcionamiento del *software* TI-MAC, y adquirir los conocimientos necesarios para llevar a cabo la programación de la aplicación de que es objeto este Proyecto de Fin de Carrera. Adicionalmente, se destacarán los aspectos y funcionalidades más importantes de las capas que implementa el estándar *ZigBee*.

## 2.2 TIPOS DE DISPOSITIVOS

El estándar 802.15.4 define dos tipos de dispositivos, los de funcionalidad reducida o RFD (*Reduced-Function Devices*), y los de funcionalidad completa o FFD (*Full-Function Devices*).

Los RFD suelen ser los dispositivos finales (generalmente sensores), responsables de recolectar datos y retransmitirlos para que puedan ser posteriormente procesados. Una de sus principales características es que permanecen inactivos la mayoría del tiempo, despertando de manera programada para enviar datos o simplemente para sincronizar sus relojes con los de dispositivos de niveles superiores en la red. El hecho de tener un grado de actividad muy pequeño hace que el consumo de potencia sea mínimo, permitiendo a los RFD poder estar alimentados por baterías, lo que facilita la portabilidad de los dispositivos y colocación en lugares poco accesibles donde no hay posibilidad de conexión a otro tipo de alimentación externa.

Por su parte los FFD pueden adoptar tres roles distintos:

- **Dispositivo final:** con las mismas características que los RFD.
- **Router:** encargado conectar los dispositivos finales con los niveles más altos de la red. Este dispositivo, dependiendo de cómo esté configurada la red, puede tener períodos de inactividad al igual que los RFD. Sin embargo, y debido a que puede estar prestando servicio a varios dispositivos de niveles inferiores de la red, el *router* generalmente estará conectado a una fuente de alimentación en vez de a una batería.
- **Coordinador de red:** constituye el nivel más alto en una red 802.15.4, debiendo haber sólo un dispositivo de éste tipo en la misma. Este es el elemento que normalmente posee mayor carga de trabajo, y realiza como mínimo las siguientes funciones: designar una única dirección de red a cada dispositivo, iniciar y terminar las conexiones y encaminar mensajes a través de la red. Al igual que los *routers*, el coordinador debe conectarse a una fuente de alimentación, ya que la carga de trabajo será muy superior a la de los dispositivos finales.

## 2.3. TOPOLOGÍAS DE RED

Las redes 802.15.4 ofrecen la posibilidad de operar en cualquiera de las siguientes topologías: topología en estrella y topología “peer-to-peer”.

### 2.3.1. TOPOLOGÍA EN ESTRELLA

La topología en estrella, véase la figura 2.2.(a), permite el establecimiento de la comunicación entre dispositivos y un único nodo central que ejerce la función de coordinador de red (*PAN Coordinator*). El coordinador de red es el controlador principal de la WPAN, responsable de iniciar y terminar las conexiones. Cualquier comunicación entre dos nodos debe pasar necesariamente por el coordinador, de forma que el área de cobertura de la red estará determinada por el alcance radio del mismo.

Debido a su sencillez, esta topología puede resultar útil para aplicaciones como control y monitorización, redes inalámbricas de sensores, seguridad centralizada o agricultura inteligente.

En general, todas las redes en estrella que operan en un mismo área de influencia pueden hacerlo de manera independiente. Esto se consigue haciendo que cada red tenga un identificador único. Así, a la hora de iniciar una red nueva, el FFD que va a adoptar el rol de coordinador debe elegir un identificador PAN que no esté siendo usado por ninguna otra red de su entorno. Una vez establecido el FFD como coordinador, éste puede permitir a otros dispositivos, tanto FFD como RFD, unirse a la red.

### 2.3.2. TOPOLOGÍA “PEER-TO-PEER”

En las topologías “peer-to-peer”, como se puede ver en la figura 2.2.(b), cada nodo de la red es capaz de comunicarse con los demás siempre y cuando estén dentro de su área de influencia. Uno de los FFD de la red debe asumir el papel de coordinador PAN, por ejemplo el primero que trata de transmitir en un determinado canal.

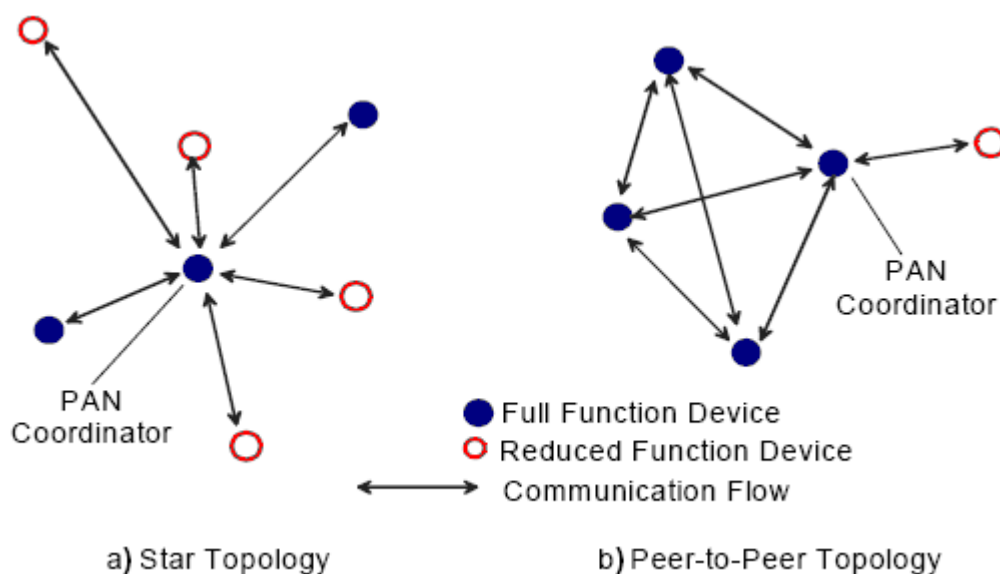
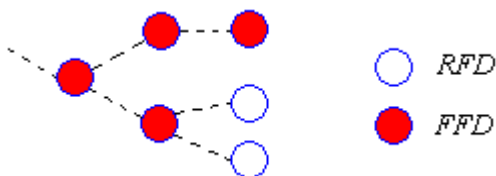


Figura 2.2. Topologías en redes 802.15.4. Fuente: [11].

En este tipo de redes los dispositivos encargados de retransmitir y encaminar mensajes son FFD. Por su parte, los RFD solo pueden ser nodos finales, con capacidad de comunicarse con un único dispositivo del nivel inmediatamente superior, bien sea coordinador o *router*, el cual se encargará de encaminar los mensajes hacia su destino.

Las redes “peer-to-peer” se pueden configurar siguiendo distintas topologías, dependiendo de las restricciones que se apliquen a los dispositivos que se unen a ellas:

- **Mesh Topology:** en este caso cada FFD puede comunicarse con el resto de FFD de la red que estén dentro de su área de cobertura y con los RFD que estén asociados a él. Una ventaja que aporta esta configuración es que, si a la hora de transmitir un mensaje desde un nodo final hasta otro, deja de funcionar un router intermedio, es posible encontrar un camino alternativo para llevar a cabo el envío de información.
- **Cluster Tree Topology:** en esta configuración (que define una jerarquía que realmente escapa al modelo P2P), una vez establecido un FFD como coordinador PAN, los *routers* que se van uniendo a la red lo hacen siguiendo una determinada estructura jerárquica, de forma que pueden prestar servicio a varios nodos del nivel inmediatamente inferior, pero sólo pueden depender de un único nodo del nivel inmediatamente superior. Como se observa en la figura 2.3, esta restricción a la hora de asociarse los nodos a la red, hace que la representación esquemática de esta topología se asemeje a la ramificación de un árbol (de ahí su nombre), en la que el coordinador PAN sería la raíz, los enlaces entre *routers* formarían las ramas, y los nodos finales (típicamente RFD) serían las hojas.



**Figura 2.3. Topología en Árbol**

Independientemente de la topología, una red IEEE 802.15.4 siempre es creada por un coordinador PAN, el cuál se encarga de gestionarla y de realizar como mínimo las siguientes tareas:

- Seleccionar un único identificador PAN para la red. Esto permite a los dispositivos que la integran usar direcciones de 16 bits y, aún así, poder comunicarse con dispositivos pertenecientes a otras redes.
- Iniciar, terminar y encaminar los mensajes a través de la red.
- Asignar una dirección de 16 ó 64 bits a cada uno de los nodos.



## 2.4. CAPA FÍSICA

La capa física (PHY) es la capa del estándar 802.15.4 más cercana al hardware, encargada de controlar su funcionamiento y de proporcionar una interfaz de comunicación con la capa inmediatamente superior de la pila de protocolos (la capa MAC). Dentro de las funciones más relevantes de esta capa se pueden destacar las siguientes:

- Selección del canal de comunicaciones.
- Activar y desactivar el transceptor radio.
- Detección de energía en el canal de comunicaciones (ED: *Energy Detection*).
- Indicación de la calidad del enlace (LQI: *Link Quality Indicator*).
- Evaluación del canal libre (CCA: *Clear Channel Assessment*).
- Transmisión y recepción de datos.

### 2.4.1. SELECCIÓN DE CANAL

La norma 802.15.4, en su primera versión de 2003, ofrece dos posibles implementaciones para la capa física dependiendo de la banda ISM usada para establecer las comunicaciones.

La primera implementación posibilita la comunicación en las bandas de 868 y 915 MHz, de uso libre en Europa y Estados Unidos, respectivamente. En ambos casos la técnica de modulación utilizada es BPSK (*Binary Phase Shift Keying*). Para la banda de 868 MHz se establece un único canal centrado a la frecuencia de 868.3 MHz. Por su parte, en la banda de 915 MHz se definen 10 canales cuyas frecuencias centrales se pueden calcular mediante la siguiente fórmula:

$$F_c = 906 + 2(k - 1) \text{ en MHz, para } k = 1, 2, 3 \dots 10 \quad (2.1)$$

donde  $k$  indica el número de canal.

La otra posible implementación para la capa física permite trabajar en la banda de 2.4 GHz, de uso extendido en prácticamente todo el mundo, y que utiliza la técnica de modulación O-QPSK (*Orthogonal Quadrature Phase Shift Keying*). En este caso se

definen 16 canales de 2 MHz, numerados del 11 al 26 y con una separación de 5 MHz. La frecuencia central de cada canal se puede calcular mediante la siguiente ecuación:

$$F_c = 2405 + 5(k - 11) \text{ en MHz, para } k = 11, 12, 13 \dots 26 \quad (2.2)$$

En la revisión de junio de 2006 del estándar se añaden técnicas de modulación adicionales en las bandas de 868 y 915 MHz, obteniendo tasas de transmisión mayores respecto a la definición original, tal y como se muestra en la tabla 2.1. Así, surge el concepto de “página de canal”, con el objetivo de distinguir entre las implementaciones originales de la capa física y las añadidas en esta revisión. En total se definen 32 páginas, y a cada una de ellas se le asignan 27 canales.

Página	Banda (MHz)	Canal	Velocidad (Kbps)	Modulación
0	868	0	20	BPSK
	915	1 a 10	40	BPSK
	2400	11 a 26	250	O-QPSK
1	868	0	250	ASK
	915	1 a 10	250	ASK
	2400	Banda reservada para uso futuro		
2	868	0	100	O-QPSK
	915	1 a 10	250	O-QPSK
	2400	Banda reservada para uso futuro		
3 a 31	Páginas reservadas para uso futuro			

**Tabla 2.1. Asignación de canales.**

La página 0 da soporte a todos los canales definidos en la edición de 2003 del estándar 802.15.4. Las páginas 1 y 2 se usan para las implementaciones opcionales de la capa física introducidas en la revisión de 2006. Finalmente, las páginas 3 a 31 están reservadas para futuras ampliaciones del estándar.

## 2.4.2. DETECCIÓN DE ENERGÍA (ED)

Se trata de un procedimiento por el cuál se estima el nivel de energía de señal en un canal determinado, promediándolo durante 8 períodos de símbolo. El valor obtenido puede ser usado por la capa de red (NWK) como parámetro del algoritmo de selección de canal.

La función ED (*Energy Detection*) debe ser capaz de detectar valores de energía al menos 10 dB por encima de la sensibilidad del receptor, moviéndose en un rango de 40 dB a partir de este límite. La información que proporciona ED es útil para saber si un canal está siendo usado por otro dispositivo, pero en ningún caso para identificar el tipo de señal que se está transmitiendo.

### 2.4.3. INDICACIÓN DE LA CALIDAD DEL ENLACE

El LQI (*Link Quality Indicator*) es un indicador de la calidad de los paquetes de datos recibidos. Para su estimación se usan parámetros como la potencia de la señal recibida (RSS: *Received Signal Strength*), la relación señal a ruido (SNR: *Signal to Noise Ratio*), o una combinación de ambos. En general, valores altos de RSS y de SNR indican una buena calidad del enlace de comunicaciones.

La medida del LQI se realiza para cada paquete recibido, y el resultado se envía a la capa MAC, quedando disponible para capas superiores en caso de requerirlo. Por ejemplo, la capa de red puede utilizar el valor de LQI para decidir la ruta más óptima que debe seguir el siguiente mensaje que se va a transmitir. En general, la ruta con mayor LQI medio ofrecerá una mayor garantía a la hora de entregar los mensajes a su destino.

### 2.4.4. EVALUACIÓN DEL CANAL LIBRE (CCA)

La función CCA es un mecanismo de la capa física usado para determinar si un canal está siendo utilizado por otro dispositivo. En general, es la capa de red la que ordena a la capa física que realice la evaluación del canal como paso previo a la ejecución del algoritmo CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*), que será explicado con detalle en el apartado dedicado a la capa de red.

La evaluación del canal se puede realizar atendiendo a los siguientes criterios alternativos:

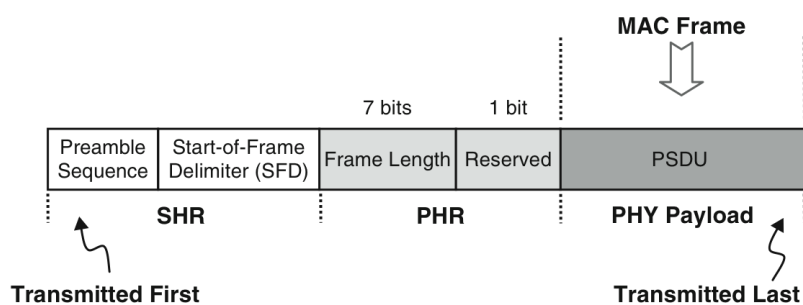
- **Detección de energía:** el canal se considera ocupado si el nivel de energía en el mismo, independientemente del tipo de señal que se esté transmitiendo, supera un umbral determinado por el fabricante.

- **Detección de portadora:** en este caso el canal se considera ocupado si se detecta alguna señal conforme al estándar 802.15.4, coincidiendo el tipo de modulación y técnica de ensanchamiento de espectro.
- **Detección de portadora con detección de energía:** se trata de una combinación de los dos criterios anteriores. El canal se considera ocupado si, tras detectar una señal en el medio del mismo tipo de las que se pretenden transmitir, la energía de la misma supera un determinado umbral.

### 2.4.5. TRANSMISIÓN Y RECEPCIÓN DE DATOS

El servicio de datos de la capa física (*PHY data service*) es el encargado de gestionar la transmisión y recepción de información, ejerciendo de interfaz con la capa MAC y con el medio físico.

La información a transmitir ha de venir en forma de una trama de datos de la capa MAC o MPDU (*MAC Protocol Data Unit*). Una vez recibida la MPDU, ésta se encapsula dentro de una trama de datos de nivel físico, PPDU (*PHY Protocol Data Unit*), cuyo formato se puede ver en la figura 2.4. La información que se añade en esta trama es información útil para la sincronización de datos en la capa física, alcanzando un tamaño máximo de 133 bytes (cuando el tamaño de la trama MAC es de 127 bytes, con 104 bytes de datos).



**Figura 2.4. Formato de trama PPDU. Fuente: [11].**

Una vez creada la PPDU, la secuencia de bits obtenidos se transforma en símbolos. A continuación se realiza la conversión de símbolos a *chips*, y, finalmente, los *chips* resultantes se modulan, obteniendo así la señal analógica que será enviada a través de medio físico.

Por otra parte, cuando un dato es recibido por el transceptor radio, la capa física notifica a la MAC la llegada de información. Entonces se produce un proceso inverso al anterior: la información analógica se demodula obteniendo una secuencia de *chips*, estos *chips* se transforman en símbolos, y los símbolos en los bits que conformarán una PPDU. Finalmente, de la PPDU se extrae la MPDU, que será enviada a la capa MAC junto con información de la calidad del enlace.

## 2.5. CAPA MAC

La capa MAC proporciona una interfaz entre la capas PHY y de red. Es responsable de proporcionar una comunicación fiable entre nodos de una red, ayudando a evitar colisiones entre mensajes y a mejorar la eficiencia en la transmisión.

Dentro de las funciones que realiza la capa MAC, se pueden destacar las siguientes:

- Generación de balizas (*beacons*) en el caso de que el nodo sea coordinador en una red balizada.
- Sincronización con las balizas recibidas.
- Permitir la conexión y desconexión de nodos a la red.
- Empleo de CSMA-CA como algoritmo de acceso al canal.
- Gestión y mantenimiento del mecanismo GTS (*Guaranteed Time Slot*).
- Proporcionar un enlace fiable entre distintas entidades MAC.

### 2.5.1. MODOS DE FUNCIONAMIENTO DE RED

En el protocolo 802.15.4 se contemplan dos modos de funcionamiento de red, dependiendo de la complejidad de la misma y de los requerimientos de consumo de los *routers* intermedios.

#### ***Modo no balizado***

También conocido como “*Non Beacon-Enabled mode*”, el modo de funcionamiento no balizado está dirigido a redes en las que el consumo de los *routers* intermedios y del coordinador PAN no es un parámetro crítico de diseño.

Esta configuración ofrece la posibilidad de que los nodos finales mantengan su transceptor radio apagado mientras no tengan que transmitir ningún dato, o que lo enciendan periódicamente para comprobar si hay algún dato pendiente de recepción. A la hora de realizar la transmisión de una trama, es necesario asegurarse de que el canal radio está libre, ejecutando para ello el algoritmo CSMA-CA no ranurado.

En este tipo de redes, el coordinador no transmite balizas a no ser que reciba una petición por parte de un nodo de la red para así poder localizarle e indicar que tiene datos pendientes para transmitir.

**Modo balizado**

El modo “*Beacon-Enabled*” está orientado a aplicaciones en las que es necesario garantizar la transmisión de un dispositivo en un momento determinado, y a redes en las que el consumo de los *routers* intermedios ha de ser optimizado ya que deben ser alimentados mediante baterías. Para ello debe permitirse que estos nodos tengan períodos de inactividad al igual que los nodos finales. Esto se consigue mediante la utilización de estructuras de *supertrama*.

Una supertrama es una estructura delimitada por balizas que incluyen información referente a la misma, como por ejemplo su duración o el número de GTS (o slots garantizados) que contiene. Tal y como se muestra en la figura 2.5, en una supertrama se pueden distinguir hasta 3 tipos de períodos:

- CAP (*Contention Access Period*).
- CFP (*Contention-Free Period*).
- *Inactive Period*.

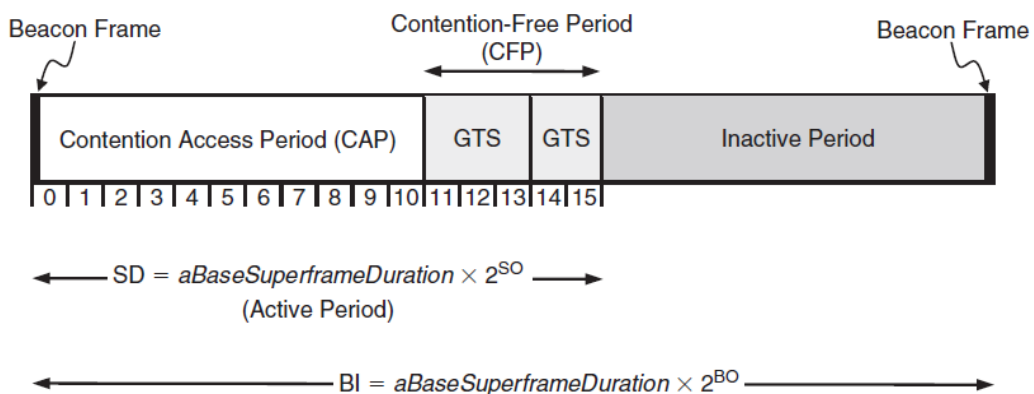


Figura 2.5. Estructura de una Supertrama. Fuente: [10].

Durante el CAP, todos los dispositivos que desean transmitir deben usar el algoritmo CSMA-CA ranurado para conseguir acceso al canal radio. Una vez que un dispositivo accede al canal, lo mantendrá ocupado hasta que finalice su transmisión, siempre y cuando ésta no se solape con el CFP. No existe ninguna garantía de que un nodo pueda transmitir en el momento exacto en que lo necesite, ya que en el caso de encontrar el canal ocupado, debe dejar que pase un tiempo aleatorio para volver a intentar acceder al canal.

En caso de trabajar con aplicaciones de baja latencia, en las que un dispositivo no se puede permitir esperar un período de tiempo aleatorio potencialmente largo hasta que el canal esté disponible, éste puede transmitir durante el CFP. Este período reserva un *slot* temporal específico en el que un determinado dispositivo puede acceder al canal radio sin necesidad de usar CSMA-CA.

La combinación del CAP y el CFP se conoce como “*active period*”, el cuál se divide en 16 *slots* temporales iguales, de los cuales 7 pueden dedicarse a GTS. A cada GTS se le puede asignar más de un *slot* temporal.

Una supertrama puede tener, de manera opcional, un período inactivo, en el que se permite que el dispositivo pueda apagar su transceptor radio y aumentar de manera significativa la duración de su batería.

La estructura de supertrama es definida por el coordinador de red y configurada por la capa de red usando una primitiva. La duración entre balizas, también conocida como BI (*Beacon Interval*) es determinada con el BO (*Beacon Order*) y la constante *aBaseSuperframeDuration*, relacionados mediante la siguiente ecuación:

$$BI = aBaseSuperframeDuration \cdot 2^{BO} \quad (2.3)$$

donde BO toma valores entre 0 y 14 para funcionamiento balizado, y 15 para funcionamiento no balizado.

Análogamente, la longitud del período activo de la supertrama, conocida como SD (*Superframe Duration*), es determinada con el SO (*Superframe Order*), relacionados como sigue:

$$SD = aBaseSuperframeDuration \cdot 2^{SO} \quad (2.4)$$

donde SD se mide en símbolos y su longitud no puede exceder de la del BI, por lo que el valor del SO siempre debe ser menor o igual que el de BO.

En una red balizada, cualquier coordinador, además del coordinador PAN, tiene la opción de transmitir balizas para crear su propia estructura de supertrama. En este caso, el coordinador puede transmitir su baliza durante el período inactivo de la supertrama del coordinador PAN, para evitar que se solapen transmisiones de otros nodos que quieren comunicarse con ambos coordinadores.

Un aspecto importante a tener en cuenta cuando en una red hay varios coordinadores transmitiendo balizas, es que los períodos activos de todas las estructuras de supertrama han de ser de igual duración. Por este motivo, y teniendo en cuenta que los coordinadores solo pueden transmitir sus balizas durante el período inactivo del coordinador PAN, el número máximo de coordinadores que pueden transmitir balizas se verá limitado por los parámetros SD y BI impuestos por el coordinador PAN.

## 2.5.2. ALGORITMO CSMA-CA

El protocolo IEEE 802.15.4 usa el algoritmo CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*) como mecanismo de acceso al canal de comunicaciones. Cualquier dispositivo que desea transmitir ha de asegurarse, mediante CCA, de que el canal no está siendo usado por otro dispositivo.

Hay tres casos en los que no es necesario el uso del algoritmo CSMA-CA para acceder al medio:

- Al transmitir una baliza.
- Acceso al canal durante el período CFP .
- Al transmitir una trama de ACK para confirmar la recepción de un dato.

El algoritmo CSMA-CA utiliza dos variantes, dependiendo de si se está operando en modo balizado o en modo no balizado:

- **CSMA-CA Ranurado:** se usa para conseguir el acceso al canal de transmisión en sistemas con estructura de supertrama habilitada. Una supertrama divide su período activo en 16 *slots* temporales de igual duración, siendo necesario que los períodos de *back-off* del algoritmo CSMA-CA estén alineados con éstos.



- **CSMA-CA no Ranurado:** se usa cuando no existe estructura de supertrama, en redes no balizadas, por lo que no es necesaria la alineación de los períodos de *back-off* del algoritmo CSMA-CA.

En la figura 2.6 se muestra el diagrama de flujo del algoritmo, que contempla las dos variantes, ranurado y no ranurado:

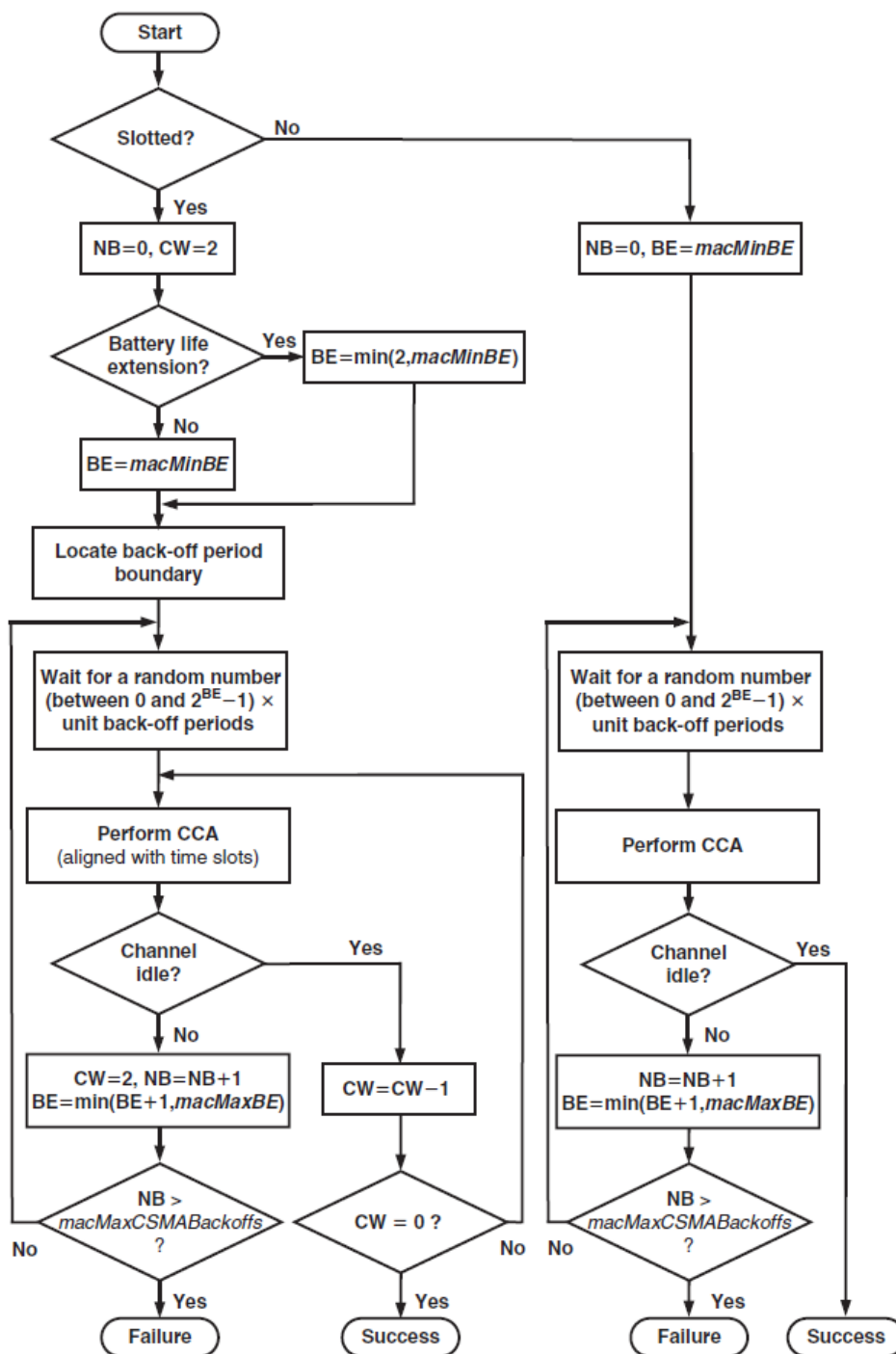


Figura 2.6. Diagrama de Flujo del Algoritmo CSMA-CA. Fuente: [11].

El algoritmo basa su funcionamiento en que cada vez que se quiere acceder al canal de comunicación y éste está ocupado, el dispositivo permanece en espera durante un período aleatorio de tiempo antes de volver a intentar acceder al canal. Los parámetros básicos que intervienen en el algoritmo son:

- **BE (Back-off Exponent):** Determina el período de tiempo que hay que esperar antes de intentar acceder al canal tras un intento fallido. El número de períodos de *Back-off* a esperar entre intentos es un número aleatorio entre 0 y  $2^{BE}-1$ , que al multiplicarlo por la constante de la capa MAC *aUnitBackOffPeriod*, da como resultado el tiempo de espera.

Este parámetro, *BE*, puede tomar valores entre *macMinBE* y *macMaxBE* (constantes de la capa MAC) tanto en el modo ranurado como en el no ranurado, incrementándose de uno en uno cada vez que se encuentra el canal ocupado. Si está activado el denominado modo de ahorro de energía (CSMA-CA ranurado), *BE* solo puede tomar valores dentro del rango  $\min(2, macMinBE)$ .

- **NB (Number of Back-Offs):** Se trata de un contador que almacena el número de veces que el dispositivo encuentra el canal ocupado y vuelve a intentar acceder. Su valor se incrementa en uno cada vez que el dispositivo encuentra el canal ocupado. El valor máximo que puede alcanzar *NB* es *macMaxCSMABackoffs*, en cuyo caso, acaba el algoritmo CSMA-CA y se comunica a la capa de red un fallo de acceso al canal.
- **CW (Contention Window length):** esta variable determina el número de períodos de *Back-Off* que debe estar libre el canal antes de empezar a transmitir (este parámetro sólo es usado en el algoritmo CSMA-CA ranurado).

### 2.5.3. INICIO Y MANTENIMIENTO DE REDES DE ÁREA PERSONAL

La capa MAC toma parte durante las fases de inicio y mantenimiento de redes PAN realizando las siguientes acciones:

- **Detección de Energía (ED):** La capa MAC puede requerir a la capa PHY que realice el procedimiento de Detección de Energía sobre un conjunto de canales determinados, para así decidir el canal óptimo para iniciar la red.

- **Escaneo activo del canal:** tiene como objetivo localizar a cualquier coordinador de red transmitiendo balizas dentro del radio de cobertura del nodo que realiza el escaneo. Este procedimiento puede ser usado por dispositivos antes de unirse a una determinada red, o por otro coordinador PAN, para realizar una prospección de los identificadores de red que están siendo usados y así elegir un identificador libre.
- **Escaneo pasivo del canal:** al igual que en el escaneo activo, el objetivo es localizar los coordinadores de red que estén transmitiendo balizas dentro de su radio de cobertura, pero en este caso simplemente escuchando los canales (sin necesidad de transmitir ninguna baliza).
- **Escaneo del canal por parte de un dispositivo huérfano:** permite que un dispositivo huérfano (que ha perdido la sincronía con su coordinador o router) consiga localizar la red a la que pertenecía y unirse a ella. Durante este escaneo, la capa MAC debe descartar todas las tramas recibidas a través del servicio de datos de la capa física que no sean tramas con comandos de realineación del coordinador.
- **Resolución de conflictos con identificadores PAN:** puede darse el caso de que existan dos o más coordinadores PAN dentro del mismo área de influencia usando el mismo identificador de red, lo que puede provocar problemas a la hora de enviar mensajes a través de las redes implicadas. Cuando un dispositivo detecta la duplicidad de identificadores, debe notificar a su coordinador PAN dicha incidencia, para que éste pueda resolverla. El coordinador deberá hacer un escaneo activo del canal, seleccionar un identificador nuevo y reiniciar la red para que todos los dispositivos cambien el identificador.
- **Inicio de redes:** una PAN puede ser iniciada únicamente por un FFD, que se establecerá como coordinador PAN, después de haber inicializado su capa MAC, haber realizado un escaneo activo del canal y haber elegido un identificador correcto. Tras completar estos pasos, la capa MAC debe entregar un mensaje de confirmación positivo a las capas superiores para poder comenzar a operar como coordinador.
- **Generación de balizas:** tiene como objetivo que dispositivos FFD puedan transmitir balizas para gestionar la sincronización de la red a la que pertenecen, ya sea actuando como coordinadores PAN o coordinadores intermedios. La generación y transmisión de balizas sólo está permitida si el parámetro *macBeaconOrder* toma un valor menor que 15.

## 2.5.4. FORMATO DE TRAMA MAC

El estándar IEEE 802.25.4 define cuatro tipos de trama MAC, en función de la información que contienen:

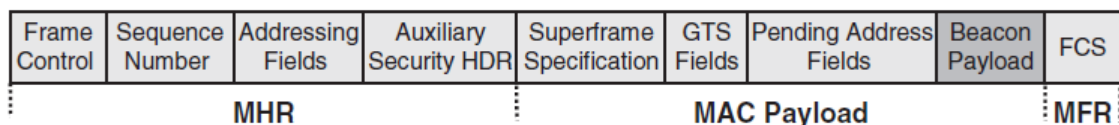
- Trama de baliza
- Trama de datos
- Trama ACK
- Trama de comando

### *Trama de baliza*

Las tramas de baliza o *beacon frames* contienen toda la información necesaria para la sincronización de los dispositivos de una red balizada, aunque también pueden ser usadas por el coordinador en redes no balizadas, para informar a un dispositivo específico de que tiene datos pendientes de recibir.

Tal y como se puede ver en la figura 2.7, la trama MAC se puede dividir en tres secciones:

- **MHR (MAC Header):** La cabecera de la trama la componen cuatro campos, *Frame Control*, que contiene información sobre el tipo de trama y los campos de direccionamiento, *Sequence Number*, que especifica el número de secuencia de baliza (*BSN: Beacon Sequence Number*), *Addressing Fields*, que contienen las direcciones de origen y destino, y *Auxiliary Security Header*, campo opcional que contiene información requerida para seguridad.
- **MAC Payload:** Contiene la información que proviene de la capa de red, como puede ser la especificación de supertrama, información sobre si los GTS son usados para recibir o transmitir, e información complementaria sobre la transmisión de balizas.
- **MFR (MAC Footer):** Contiene un FCS (*Frame Check Sequence*) para comprobar hay algún error en la trama recibida.



**Figura 2.7. Formato de la Trama de baliza de la capa MAC. Fuente: [10].**

### Trama de datos

Son las tramas que se utilizan para enviar información entre nodos de una red. Como se puede ver en la figura 2.8, los campos que la componen son similares a los de las tramas de baliza, a excepción de los campos de supertrama, GTS, y direcciones pendientes, que en este tipo de trama no aparecen.

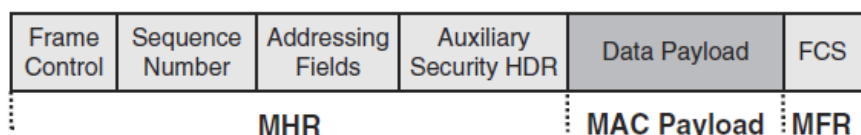


Figura 2.8. Formato de la Trama de Datos de la capa MAC. Fuente: [10].

### Trama ACK

Este tipo de tramas se usa para la confirmación, entre dispositivos, de la recepción correcta de datos o peticiones. La confirmación se hace a nivel MAC, por lo que no es necesario incluir información de capas superiores, reduciendo notablemente la longitud de la trama, tal y como se muestra en la figura 2.9.

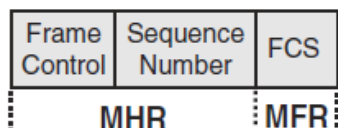


Figura 2.9. Formato de la Trama ACK de la capa MAC. Fuente: [10].

### Trama de comando

El uso de estas tramas está dedicado al envío de comandos de capa MAC, tales como asociación y abandono de dispositivos, petición de baliza, petición de GTS, notificación de conflicto de identidad de la PAN, etc. En la siguiente figura se puede ver la estructura de la trama.

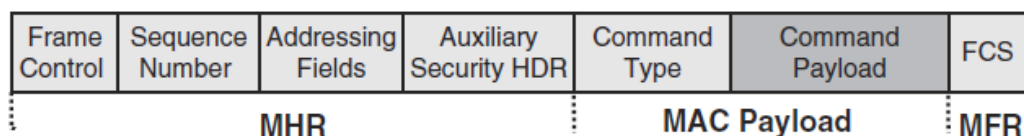


Figura 2.10. Formato de la Trama de Comando de la capa MAC. Fuente: [10].

## 2.6. IMPLEMENTACIÓN DEL ESTÁNDAR ZIGBEE

Como ya se comentó al comienzo de este capítulo, el estándar *ZigBee* define únicamente las capas de Red, Aplicación y Seguridad, apoyándose sobre las capas Física y MAC de la norma IEEE 802.15.4. Respecto a esta, *ZigBee* incorpora funciones de enrutamiento y permite realizar comunicaciones multisalto. Incorpora seguridad a nivel de enlace y de red, aunque quizás, la prestación más interesante sea la posibilidad de creación de *ZigBee Profiles* (perfiles de aplicación que facilitan la interoperabilidad de productos, gracias a la definición de unos servicios y un lenguaje común).

A continuación, y debido a que este proyecto se centra en el uso de TIMAC (implementación de la norma 802.15.4 por parte de Texas Instruments), se va a realizar una descripción de los aspectos más importantes de las capas de *ZigBee*, para así tener una visión global del estándar.

### 2.6.1. CAPA DE RED

La capa de red (NWK) ejerce de interfaz entre las capas de Aplicación y MAC. Entre las funciones de esta capa, pueden destacarse las siguientes:

- Enrutamiento.
- Descubrimiento de ruta óptima.
- Mantenimiento de las rutas y reparación de las mismas.
- Gestión de la formación de la red y de los mecanismos de comunicación

#### *Enrutamiento*

El enrutamiento es, básicamente, la elección del camino que han de seguir los paquetes a la hora de ser dirigidos a través de la red para alcanzar su destino. Los dispositivos que pueden llevar a cabo esta tarea son los *routers* y los coordinadores de red, en ningún caso los dispositivos finales.

Al enlace entre dos nodos consecutivos se le denomina *link*, y a cada uno se le asocia un coste que está directamente relacionado con la probabilidad de entregar correctamente un paquete entre los nodos. Esta probabilidad puede ser estimada usando el parámetro LQI.

El coste de una ruta completa se puede calcular sumando los costes individuales de todos los links que forman el camino. Así, la ruta con menor coste acumulado, será la que tenga más probabilidad de entregar correctamente los paquetes.

Cada nodo conmutador cuenta con una tabla de enrutamiento, para determinar cuál es el siguiente salto a la hora de distribuir mensajes hacia un destino determinado.

### ***Descubrimiento de Ruta Óptima***

El descubrimiento de ruta óptima (en redes malladas) sólo puede ser llevado a cabo por *routers* y coordinadores de red. Este procedimiento se inicia cuando se quiere establecer comunicación entre dos nodos entre los que no había un camino previamente establecido, o cuando se rompe alguno de los enlaces que forman una ruta.

Para establecer una nueva ruta de comunicación, el dispositivo interesado difunde un comando "*route request*" en el que indica la dirección de destino. Todos los dispositivos que estén dentro de su radio de influencia y tengan capacidad de enrutamiento, añadirán en sus tablas de descubrimiento de ruta esta petición y sumarán el coste del enlace al paquete para así volver a difundirlo. Tras varios saltos en la búsqueda del camino óptimo entre dos nodos, puede darse el caso de que un *router* intermedio reciba el mismo mensaje de descubrimiento de ruta procedente de distintos nodos. En este caso, se analizan todas las peticiones y se actualiza la tabla de descubrimiento de ruta óptima con el enlace que proporcione un menor coste acumulado.

Una vez que se alcanza el dispositivo final, éste analiza todos los mensajes que le han llegado desde diferentes nodos, actualizando su tabla de ruta óptima con la de menor coste. Tras establecer la ruta óptima se envía un mensaje de confirmación desde el dispositivo final hacia el dispositivo origen siguiendo dicha ruta.

### ***Mantenimiento de las rutas y reparación de las mismas***

El proceso de mantenimiento y reparación se lleva a cabo cuando se pierde la conexión entre alguno de los enlaces que conforman una ruta.

El procedimiento de reparación es similar al de descubrimiento de ruta óptima, con la salvedad de que el dispositivo que lo inicia es el primero que detecta el enlace roto. En caso de no encontrar una ruta hacia el nodo de destino, se envía un mensaje de error al nodo origen para que inicie una nueva búsqueda de ruta óptima.

### ***Gestión de la formación de la red y de los mecanismos de comunicación***

La capa de red es la responsable de configurar los FFD para que se comporten como coordinadores ZigBee. Para ello ha de comunicarle a la capa MAC que realice un escaneo de energía del conjunto de canales de interés y, dependiendo del resultado del ED en cada uno de los canales, seleccionar el canal óptimo para iniciar la red y asignar un identificador PAN único. La primera tarea del coordinador PAN es configurar los parámetros de supertrama usando el servicio de gestión de la capa MAC.

Otra función destacable de la capa de red en lo que a gestión de la formación de la red se refiere, es configurar a los FFD como *routers* cuando la capa de Aplicación lo requiera, mediante el uso de una primitiva específica. Debido a que los *routers* pueden tener su propia estructura de supertrama, distinta a la del coordinador PAN, y gestionar las peticiones de otros dispositivos que quieran unirse o abandonar la red, la primitiva anterior debe incluir todos los parámetros de configuración de la supertrama.

El orden y las reglas que se sigan a la hora de permitir la unión de dispositivos a la red, determinará la topología de la misma.

### ***Gestión de la unión y el abandono de nodos de una red***

El proceso de unión a una red es iniciado por el nodo interesado. Para ello realiza un escaneo, activo o pasivo, para encontrar los *routers* o el coordinador de la red de la que quiere formar parte. Una vez seleccionado el nodo al que se quiere unir (depende de la topología de la red y de la aplicación que se quiera llevar a cabo) el nodo hijo inicia el proceso de asociación usando otra primitiva.



Una vez que el nodo padre recibe la petición de unión a la red, éste comprueba si el dispositivo se encontraba previamente registrado en su tabla de vecindad y si es posible aceptar dicha petición. En caso de aceptar la unión de un nodo que no estaba registrado, se le asigna una nueva dirección de 16 bits y se actualizan las tablas de vecindad.

Por su parte, el abandono de una red puede ser iniciado tanto por un nodo padre como por un nodo hijo. Cuando un nodo abandona una red, también la abandonan todos los dispositivos que dependían de éste. Tras finalizar el abandono, todos los nodos implicados actualizan sus tablas de vecindad.

## 2.6.2. CAPA DE APLICACIÓN

La capa de Aplicación (APL) es la más alta del protocolo *ZigBee*. En la siguiente figura se muestra un esquema con las partes que la integran:

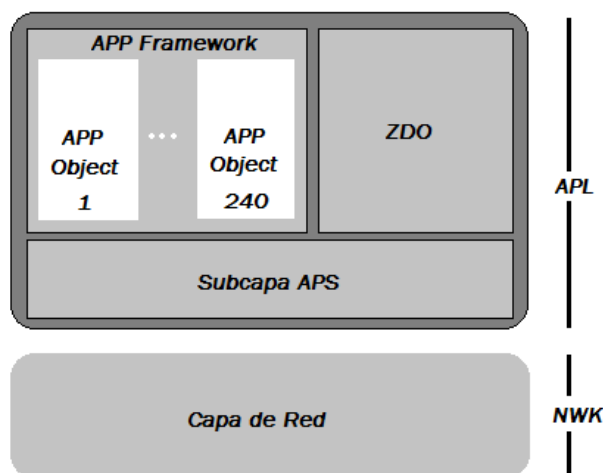


Figura 2.11. Capa de Aplicación de protocolo *ZigBee*.

- **Subcapa APS (*Application Support*):** su misión es la de proporcionar un interfaz entre la capa NWK y la APL. Dentro de las funciones de esta subcapa, se pueden destacar el mantenimiento de las tablas de vecindad, la gestión de las direcciones de grupos, o el “mapeo” entre direcciones de 16 y 64 bits.
- ***Application Framework*:** almacena las aplicaciones que van a ser utilizadas en el dispositivo *ZigBee*. Cada dispositivo puede almacenar hasta un máximo de 240 aplicaciones, que pueden ser direccionadas de manera independiente, posibilitando así compartir el uso del mismo módulo radio.

- **ZDO (*ZigBee Device Object*):** proporciona un interfaz entre la subcapa APS y la capa *Application Framework*. Contiene las funcionalidades que son comunes en todas las aplicaciones que se pueden implementar, como puede ser definir el rol de los dispositivos dentro de la red, descubrir los dispositivos que la integran o realizar las tareas relativas a la seguridad.

### 2.6.3. SEGURIDAD

En lo que se refiere a seguridad, las redes inalámbricas son las más vulnerables ya que el medio de transmisión de información es fácilmente accesible. Por tanto, la seguridad en este tipo de redes juega un papel crucial a la hora de mantener la integridad y privacidad de la información que se transmite.

Los dispositivos *ZigBee*, concebidos para optimizar el consumo energético, están basados en microcontroladores de bajas prestaciones en lo que a capacidad de cómputo se refiere. Por tanto, el sistema de encriptación asociado debe ser sencillo a la hora de la implementación y ejecución.

La norma *ZigBee* define dos capas de seguridad a los niveles de red y aplicación. El algoritmo de encriptación utilizado es AES (*Advanced Encryption Standard*) con claves de 16 bytes. Para garantizar la seguridad en las comunicaciones se definen tres tipos de claves:

- **Claves Maestras:** están preinstaladas en cada nodo. Su función es garantizar la confidencialidad durante el procedimiento de intercambio de “Claves de Enlace” entre nodos.
- **Claves de Enlace:** son únicas entre cada par de nodos que pueden comunicarse de manera directa. Estas claves se gestionan a nivel de aplicación, y se usan para encriptar toda la información intercambiada entre pares de nodos.
- **Clave de Red:** es una única clave de 128 bits, compartida entre todos los dispositivos pertenecientes a una red. Esta clave es la utilizada por los nodos para obtener permiso para unirse a la red.



# CAPÍTULO 3

## Elección de tecnologías

### 3.1. INTRODUCCIÓN

Desde que en 2003 la *ZigBee Alliance* publicase la primera versión del estándar, han sido muchos los fabricantes de semiconductores que se han unido a esta iniciativa, desarrollando sus propias implementaciones tanto a nivel *hardware* como *software*. A continuación se comenta una selección de módulos *ZigBee* de los fabricantes con mayor presencia en este sector.

#### **Texas Instruments [www.ti.com]**

Texas Instruments es uno de los fabricantes de semiconductores que más está evolucionando en el desarrollo de dispositivos diseñados para trabajar en entornos ZigBee/802.15.4. A continuación se detallan algunos de los más importantes:

- **CC2420:** Se trata de un transceptor 802.15.4 de primera generación diseñado para trabajar en la banda ISM de 2.4 GHz y dar respuesta a aplicaciones inalámbricas de bajo consumo y bajo voltaje. El kit de desarrollo asociado es el CC2420DK, y se suministra con 2 módulos de evaluación CC2420EM y 2 placas de evaluación CC2420EB.
- **CC2480:** Procesador ZigBee Z-Accel, de bajo coste y baja potencia que proporciona funcionalidad *ZigBee* Completa con un esfuerzo de desarrollo mínimo. Z-Accel es una solución donde la pila de protocolos *ZigBee* se ejecuta en un procesador dedicado, mientras que las aplicaciones lo hacen en un microcontrolador externo.

- **CC2430/CC2431 SoC (*System on Chip*):** combina transceptor y microcontrolador 8051 en un único chip. Además permite trabajar con Z-Stack, versión propia del estándar *ZigBee* de Texas Instruments, y con TIMAC, versión propia del protocolo IEEE 802.15.4.
- **CC2520:** transceptor de segunda generación, pensado para trabajar junto a un microprocesador MSP430 de Texas Instruments, mejorando características de consumo, voltaje y rangos de temperatura. El kit de desarrollo CC2520DK contiene 3 módulos de evaluación CC2520EM, 2 placas CCMSP-EM430F2618 con microprocesador MSP430, y 3 placas de evaluación SmartRF05EB.
- **CC2530 SoC:** solución para aplicaciones IEEE 802.15.4, *ZigBee* y RF4CE. Combina transceptor de RF y un microcontrolador 8051 mejorado en un único chip. El kit de desarrollo CC2530ZDK incluye 7 módulos de evaluación CC2530EM, 2 placas de evaluación SmartRF05EB, 5 placas de evaluación SmartRF05BB, y un módulo USB CC2531, que puede ser conectado directamente a un PC y usarse como *Packet Sniffer*.

#### **Freescale [www.freescale.com]**

Al igual que Texas Instruments, Freescale cuenta con sus propias implementaciones de los protocolos ZigBee e IEEE 802.15.4, que son las pilas de protocolos *BeeStack* y SMAC respectivamente. A nivel *hardware*, se pueden destacar los siguientes sistemas de desarrollo:

- **MC1320X:** se trata de un transceptor de RF basado en el estándar IEEE 802.15.4.
- **MC1321X *System in package*:** integra el microcontrolador MC9S098GT con el transceptor MC1320X.
- **MC1322X *Platform in package*:** se trata de una combinación de transceptor y microcontrolador integrados en un único chip, para optimizar el ahorro energético del transceptor.

#### **Jennic [www.jennic.com]**

Actualmente la empresa británica de semiconductores destaca por la fabricación y comercialización de sus microcontroladores inalámbricos JN5148, JN5139 y JN5121, para

aplicaciones de bajo consumo y bajo voltaje. Además, comercializa el siguiente kit de desarrollo:

- **JN5148 Evaluation Kit:** proporciona un entorno completo de desarrollo para aplicaciones *ZigBee* PRO, JenNet (versión propia de la capa de red) e IEEE 802.15.4, basadas en el microcontrolador JN5148.

#### **Ember [www.ember.com]**

Hoy en día Ember produce tres soluciones *ZigBee* distintas:

- **EM300 Series:** la serie EM300 comercializa chips tanto SoC como NCP (*Network Co-Processor*), que trabajan en la banda ISM de 2.4 GHz y usan un microprocesador ARM Cortex M3.
- **EM250 SoC:** combina un microprocesador de 16 bits con un módulo radio de tecnología 802.15.4 a 2.4 GHz. El desarrollo de aplicaciones se hace mediante un entorno de desarrollo en lenguaje C que incluye una versión propia de *ZigBee* llamada *EmberZNet ZigBee Stack*.
- **EM260 NCP:** se trata de un co-procesador de red *ZigBee*, que se encarga de alojar la pila de protocolos *EmberZNet PRO ZigBee Stack*, permitiendo al desarrollador programar sus aplicaciones en el microprocesador que elija.

## **3.2. ELECCIÓN DEL SISTEMA DE DESARROLLO**

La elección del sistema de desarrollo para llevar a cabo este Proyecto Fin de Carrera se ha visto condicionada por las limitaciones propias impuestas por los fabricantes de los mismos. Así pues, este proceso ha constado de tres etapas, cada una de ellas centrada en el estudio de un kit de desarrollo.

### **3.2.1 KIT DE DESARROLLO CC2520DK DE TEXAS INSTRUMENTS**

Inicialmente el objetivo del proyecto era el desarrollo de un nodo conmutador *ZigBee* mediante el uso del kit de desarrollo CC2520DK de Texas Instruments basado en el microcontrolador MSP430, como evolución del modelo CC2480DK de la misma casa, y que teóricamente debía permitir la formación de redes balizadas acorde con el estándar *ZigBee*.

El resultado fue negativo, al encontrar que el modo balizado no había sido implementado en la pila de protocolos Z-Stack suministrada junto al HW. La explicación que aportaba Texas Instruments para no implementar dicha funcionalidad fue que, al tratarse de un microprocesador poco potente, era muy complicado gestionar el funcionamiento del modo balizado junto a los distintos modos de enrutamiento propios de las capas superiores del estándar *ZigBee*.

La solución alternativa aportada fue el uso de la pila de protocolos TIMAC, basada en el estándar 802.15.4, de manera que se eliminaban las funcionalidades propias de ZigBee, y así se optimizaba el uso del microprocesador para poder implementar redes balizadas de arquitectura sencilla. El resultado volvió a ser negativo, ya que tras estudiar el funcionamiento de la versión 1.3.0 de TIMAC para microcontroladores MSP430 (que implementa las capas PHY y MAC, y una pequeña capa de aplicación), se vio que la funcionalidad de configuración de redes balizadas tampoco había sido implementada. El problema por el que se excluyó esta funcionalidad era la poca precisión de los relojes con los que cuenta el sistema, perdiendo la sincronización con mucha facilidad cuando los nodos operaban en modo balizado.

### **3.2.2. KIT DE DESARROLLO MC1322X DE FREESCALE**

El siguiente paso, al no conseguir los resultados esperados con el kit CC2520 de Texas Instruments, fue buscar una alternativa de otro fabricante. Después de realizar un estudio de las presentaciones comerciales de los productos que ofrecían los principales fabricantes de semiconductores, se eligió el Kit de Desarrollo MC1322X de Freescale, que era el que, sobre papel, presentaba mejores características.

Tras estudiar en profundidad el funcionamiento de la pila de protocolos *BeeStack*, proporcionada con el Kit MC1322X se observó que la posibilidad de configuración de redes balizadas también había sido deshabilitada sin motivo aparente. En este caso, tras contactar con personal técnico de Freescale se confirmó que el modo balizado no había sido implementado para arquitecturas ARM7, ya que se producían muchos conflictos asociados al consumo de cada uno de los nodos.

### 3.2.3. KIT DE DESARROLLO CC2530DK DE TEXAS INSTRUMENTS

Después de los resultados negativos obtenidos durante el estudio de los kits anteriores, se decidió realizar una última búsqueda con el objetivo de encontrar un sistema que permitiese la configuración de nodos para que formasen parte de redes balizadas. Tras valorar varias opciones de diversos fabricantes se optó por escoger el Kit de Desarrollo CC2530DK de Texas Instruments.

En este caso el modo de funcionamiento balizado tampoco había sido implementado en la pila de protocolos Z-Stack, ya que el problema de sincronización entre nodos con estructuras de red complejas persistía, al seguir contando con un microcontrolador poco potente (al igual que ocurría con el Kit CC2520DK).

Sin embargo, debido a que en este caso la precisión de los relojes era bastante mejor que los del Kit CC2520DK, la versión 1.3.0 de la pila de protocolos TIMAC sí implementaba la funcionalidad necesaria para poder configurar redes balizadas, aunque con restricciones considerables:

- Sólo se pueden configurar redes en topología de estrella.
- Las redes balizadas no permiten la utilización de *slots* garantizados (o GTS).

Por tanto, tras el estudio en profundidad de las soluciones anteriores, se eligió el Kit de Desarrollo CC2530DK de Texas Instruments junto a la pila de protocolos TIMAC 1.3.0 para la realización de este PFC.

## 3.3. KIT DE DESARROLLO CC2530ZDK

El kit de desarrollo CC2530ZDK forma parte de la segunda generación de soluciones ZigBee/IEEE 802.15.4 de Texas Instruments, y está basado en una arquitectura *hardware* SoC, en la que tanto el transceptor como el microprocesador están integrados en un único chip, con un microcontrolador 8051 optimizado, y una radio que funciona en la banda ISM de 2.4 GHz. Contiene todo el *hardware*, *software*, herramientas y documentación necesarios para diseñar aplicaciones *ZigBee* y *ZigBee Pro*.

El CC2530DK se apoya en el compilador C IAR EW8051 (explicado en el apartado 3.5.1.), usado para programar tanto la pila de protocolos Z-Stack como la pila TIMAC.



El kit CC2530ZDK se muestra en la figura 3.1 y se compone de:

- 2 placas de desarrollo SmartRF05EB v1.7.
- 5 placas de desarrollo de alimentación exclusiva mediante baterías SmartRF05BB.
- 7 módulos de evaluación CC2530EM.
- 7 antenas.
- 1 módulo CC2531 USB Dongle.
- Cables, baterías y documentación necesaria para el funcionamiento inicial.



Figura 3.1. Kit de desarrollo CC2530ZDK. Fuente [14].

### 3.3.1. MÓDULO CC2530EM

El módulo de evaluación CC2530EM, mostrado en la figura 3.2, contiene el chip CC2530F256 y los circuitos integrados de RF, los componentes externos y los filtros necesarios para conseguir su correcto funcionamiento. El módulo además cuenta con un reloj de 32MHz, otro de 32KHz y un conector SMA (*SubMiniature versión A*) para antena u otro instrumento RF.

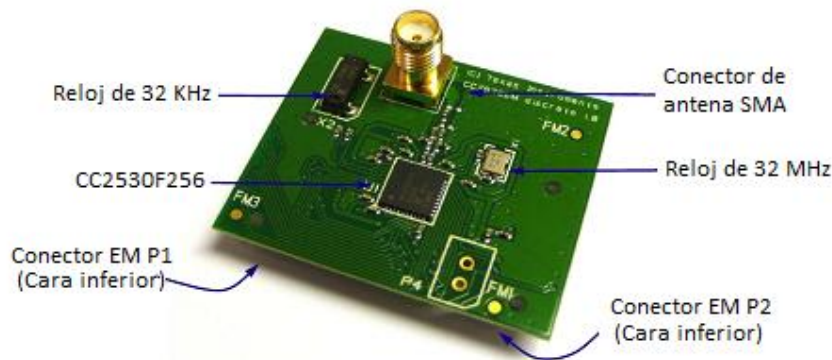


Figura 3.2. Módulo de evaluación CC2530EM. Fuente [14].

### 3.3.2. MÓDULO CC2531 USB DONGLE

El módulo CC2531 USB Dongle es un dispositivo *ZigBee* con interfaz USB completamente operacional que puede ser conectado a un PC. Cuenta con un chip CC2531F256, una antena integrada, 2 diodos LED, 2 pulsadores y 2 pequeños huecos que permiten la conexión de dispositivos y sensores externos. Además, viene pre-programado para ser usado como *packet sniffer*, aunque se le puede cargar cualquier otro *firmware* a través del conector específico que incorpora.

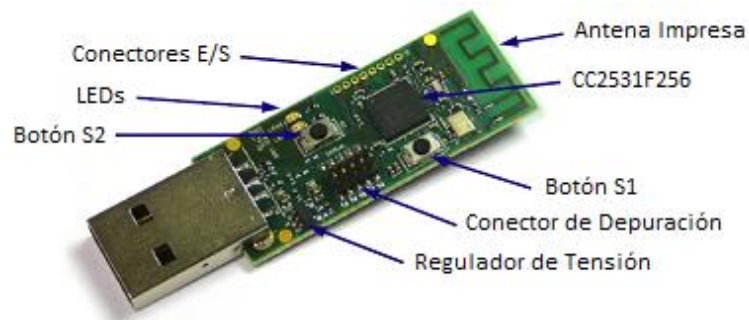


Figura 3.3. Módulo CC2531 USB Dongle. Fuente [18].

### 3.3.3. PLACA DE DESARROLLO SMARTRF05EB

La placa SmartRF05EB hace la función de placa base, proporcionando alimentación y comunicación bidireccional con el módulo CC2530EM a través de sus conectores EM. Cuenta con varias interfaces de usuario y conexiones hacia diversas interfaces externas, lo que proporciona gran rapidez a la hora de diseñar y testear prototipos, tanto *software* como *hardware*. En la figura 3.4 se muestra una vista de la placa, en la que están etiquetados los elementos visibles más importantes.

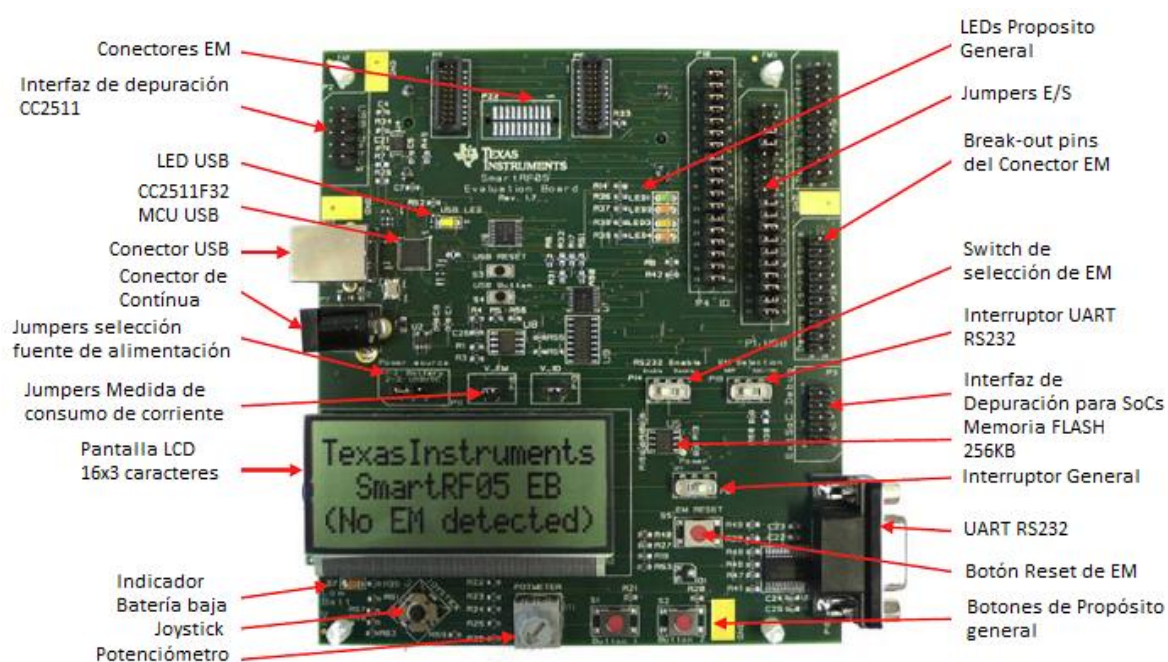


Figura 3.4. Placa de Desarrollo SmartRF05EB de Texas Instruments. Fuente [14].

A continuación se detallan algunos de los elementos más relevantes de la placa de evaluación.

#### Controlador USB CC2511F32

Se puede considerar que el componente principal de la placa es el controlador USB, ya que se encarga de gestionar la comunicación con el PC, a través del conector USB, y con el módulo de evaluación, usando el SPI (*Serial Peripheral Interface*), la UART (*Universal Asynchronous Receive Transmit*) o el interfaz de depuración para SoC. En la siguiente

figura se puede ver un esquema de los elementos más importantes de la placa y los buses que se usan para su interconexión.

Una vez conectado a la placa el módulo de evaluación, en nuestro caso el CC2530EM, se puede tener acceso completo a los periféricos, como la pantalla LCD, la memoria *flash*, los diodos LED, los botones, el *joystick* y la interfaz UART RS232.

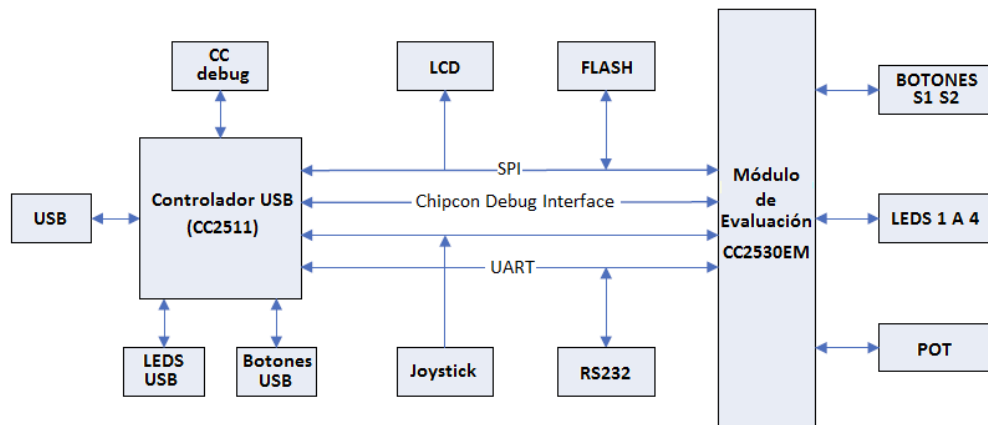


Figura 3.5. Arquitectura de la placa SmartRF05EB. Fuente [16].

### Alimentación de la placa

Existen cuatro métodos para alimentar la placa SmartRF05EB. Estos pueden seleccionarse a través del *jumper* específico etiquetado como P11, que se muestra en la siguiente figura. Una vez alimentada la placa, el encendido y apagado se controla mediante el *switch* etiquetado como P8.



Figura 3.6. *Jumper* de selección de método de alimentación de la SmartRF05EB. Fuente [16].

Los métodos de alimentación son los siguientes:

- **Baterías:** Alimentación mediante dos baterías de 1.5V formato AA. En este caso el *jumper* de selección debe conectar los pines 1 y 2. La placa cuenta un diodo LED que se enciende cuando el voltaje cae por debajo de los 1.56V.

- **Alimentación externa:** Alimentación a través de un conector DC *jack* estándar de 2.5 mm. El jumper de selección debe conectar los pines 2 y 3.
- **Alimentación USB:** Se utiliza cuando la placa está conectada a un PC mediante un cable USB, utilizando parte de la corriente que éste es capaz de suministrar. Al igual que en el caso de la alimentación externa, el jumper de selección debe conectar los pines 2 y 3.
- **Alimentación mediante fuente de laboratorio:** En este caso hay que retirar el jumper de selección de método de alimentación. La toma de tierra de la fuente externa ha de conectarse a cualquiera de los puntos de conexión de la placa etiquetados con GND. La tensión de alimentación, que puede variar entre los 2.7 y 3.6 voltios, se aplica directamente sobre el pin número 2 de la sección P11. Hay que tener cuidado, ya que cuando se utiliza este método de alimentación quedan deshabilitados todos los reguladores de tensión de la placa, pudiéndose causar daños a la misma.

### Interfaz UART RS232

En la siguiente figura se muestra un esquema de interconexión entre las líneas de la UART y los diferentes dispositivos de la placa de evaluación.

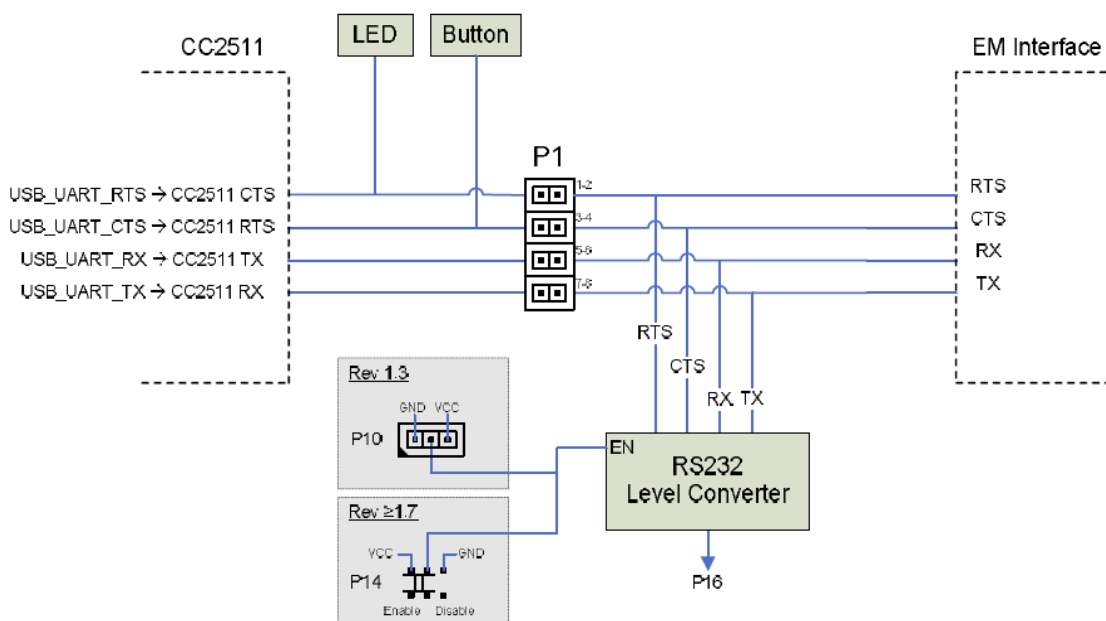


Figura 3.7. Interconexión de dispositivos mediante UART RS232. Fuente [16].

La interfaz UART puede ser utilizada por aplicaciones propias para establecer comunicación con otros dispositivos. Esta interfaz usa un driver de línea de manera que el puerto es compatible con la señalización RS232, permitiendo tres tipos de comunicación:

- Comunicación entre un PC y la placa de evaluación mediante un cable serie.
- Comunicación entre un PC y el controlador USB mediante un cable cruzado.
- Comunicación entre el controlador USB y el microcontrolador del módulo de evaluación, mediante la conexión directa de las líneas de transmisión y recepción.

### ***Joystick***

El *joystick* integrado en la placa puede adoptar cinco posiciones distintas: centro, derecha, izquierda, arriba y abajo, siendo la posición central la posición por defecto y de retorno después de cualquier movimiento (incluida la pulsación del mismo).

Para que todos los módulos de evaluación que se conecten a la placa puedan usar el *joystick*, cuando se produce cualquier movimiento o pulsación se generan dos señales analógicas adicionales. La primera, JOY\_MOVE, indica que se ha producido un movimiento del *joystick* partiendo de su posición central. La segunda, JOY\_LEVEL, dependiendo del voltaje de salida, indica qué movimiento se ha realizado.

El correcto entendimiento del funcionamiento interno de este elemento de la placa ha sido fundamental para el desarrollo de este proyecto, ya que ha permitido el desplazamiento sobre las diferentes opciones que se ofrecen en los menús de configuración de los nodos de red.

### **Pantalla LCD**

La placa SmartRF05EB cuenta con una pantalla LCD (*Liquid Crystal Display*) de 3 líneas y 16 caracteres por línea, cuyo acceso se hace a través del bus SPI.

Al igual que en el caso del *joystick*, ha sido muy importante conocer el funcionamiento de la pantalla para poder mostrar los menús que permiten la configuración de los nodos directamente sobre la placa, así como para mostrar los mensajes de error que facilitan la labor de depuración del código de programación.



## Botones

La placa de desarrollo dispone de cinco botones distintos:

- *S1* y *S2*: se trata de dos botones configurables a través del módulo de evaluación para que interactúen con las aplicaciones instaladas en el mismo.
- *USB*: está conectado al controlador USB.
- *USB Reset*: reinicia el controlador USB y el controlador del módulo de evaluación, ya que durante el inicio del *software* del controlador USB se reinicia por defecto el controlador de dicho módulo.
- *EM Reset*: reinicia el controlador del módulo de evaluación.

## Diodos LED

La placa SmartRF05EB integra 6 diodos LED cuya funcionalidad se describe a continuación:

- Diodos D1 a D4: se trata de 4 diodos LED de propósito general, cuya funcionalidad es totalmente programable a través del módulo de evaluación conectado a la placa, pudiendo actuar, por ejemplo, como indicadores del estado de la aplicación que se esté ejecutando, lo que facilita la depuración del código de la misma.
- Diodo D7: cuando la placa de evaluación está alimentada mediante baterías, este diodo indica el estado de las mismas, encendiéndose cuando su nivel cae por debajo de los 1.56V.
- Diodo D6: está asociado al controlador USB y se usa para indicar el estado de funcionamiento de la placa. Presenta cinco estados diferentes:
  - OFF: cuando el sistema está apagado o el controlador USB está dañado.
  - ON: tras detectar un transceptor si se está usando el *firmware* estándar.
  - Parpadeo Corto: El diodo parpadea una sola vez durante la secuencia de encendido, quedando apagado.
  - Parpadeo a 1Hz: El controlador USB ha entrado en modo de recuperación.

- Parpadeo a 10Hz: El controlador USB no ha podido encontrar una aplicación válida. Los servicios básicos USB permanecen activos de forma que se pueda cargar una aplicación que sí sea correcta.

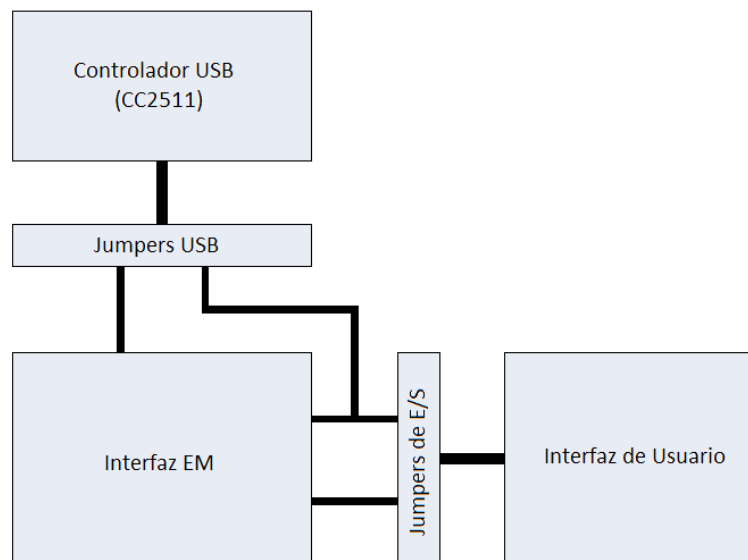
### Conectores EM

Los conectores EM forman la interfaz entre el módulo de evaluación y la placa de desarrollo, permitiendo la comunicación entre ambos. Están formados por dos zócalos de pines etiquetados como P5 y P6 en la placa, y la funcionalidad de cada uno de los pines puede ser consultada en el documento “*SmartRF05 Evaluation Board User’s Guide*”.

### Cabeceras de *Jumpers* USB y E/S

Las señales que provienen de los conectores EM están conectadas a los interfaces de usuario o al controlador USB CC2511 a través de la cabecera de *jumpers* de E/S o la cabecera de *jumpers* USB, respectivamente.

Los *jumpers* de estas cabeceras permiten controlar qué señales van a qué dispositivos, haciendo más fácil la tarea de diseñar y depurar aplicaciones usando dispositivos propios.



**Figura 3.8.** Esquema de conexión entre el EM y las cabeceras de *jumpers*. Fuente [16].



### 3.3.4. CIRCUITO INTEGRADO CC2530

Como ya se ha comentado anteriormente, el chip CC2530 de Texas Instruments es un dispositivo 802.15.4 (y ZigBee) de segunda generación que opera en la banda ISM de 2.4 GHz y está diseñado para operar en redes en las que es esencial obtener muy bajo consumo. En la siguiente figura se muestra el diagrama de bloques genérico de la familia de chips CC253x.

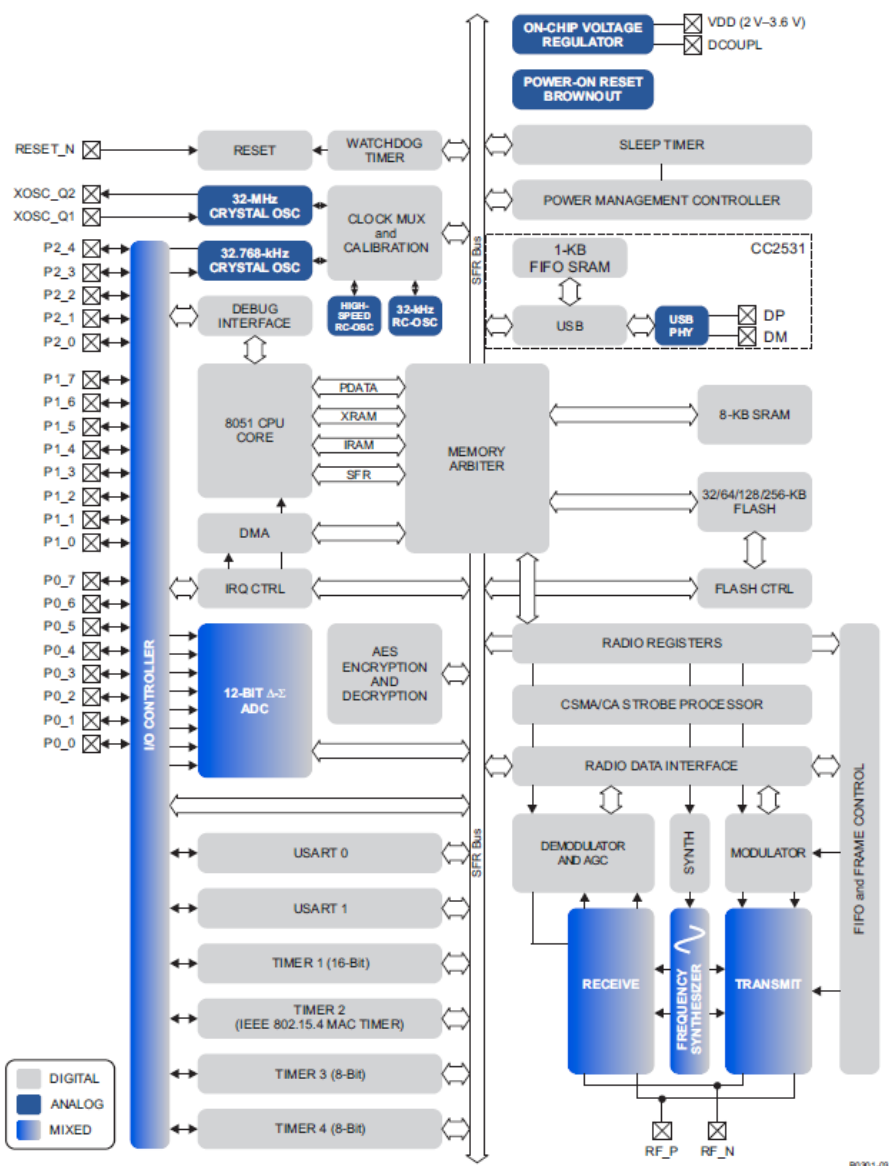


Figura 3.9. Diagrama de bloques de la familia de chips CC253x. Fuente [15].

Estos módulos se pueden agrupar en cuatro categorías, descritas a continuación: CPU y Memoria, Gestión de Alimentación, Periféricos y Radio.

## **CPU y Memoria**

La familia de dispositivos CC2530 usa una CPU 8051 mejorada, que posibilita la ejecución de una instrucción por período de reloj, en contraposición con los 12 ciclos por instrucción usados en la versión estándar de esta CPU. Cuenta con tres buses diferentes para el acceso a memoria, con acceso en un único ciclo al SFR (*Special Function Register*), registro de datos DATA, y la memoria SRAM (*Static Random-Access Memory*) principal.

El controlador de interrupciones es capaz de gestionar un total de 18 fuentes de interrupción, divididas en seis grupos, cada uno asociado a una de las cuatro prioridades disponibles.

El gestor de memoria se encarga de conectar la CPU y el controlador DMA con las memorias físicas y todos los periféricos a través del bus SFR. Además es el responsable de arbitrar y secuenciar los accesos simultáneos a una misma memoria física.

La memoria SRAM de 8-KB es una memoria de muy bajo consumo capaz de conservar el contenido incluso cuando la parte digital del dispositivo está desconectada, característica fundamental para el desarrollo de aplicaciones en las que es necesario optimizar al máximo el consumo.

El bloque de memoria *flash* de 256 KB se utiliza para el almacenamiento de aplicaciones y de datos que deben estar disponibles después de reiniciar el dispositivo.

## **Gestión de Alimentación**

Tanto el núcleo del chip como los periféricos están alimentados a través de un regulador de tensión de 1.8V, que permite el uso de diferentes modos de alimentación para aplicaciones de bajo consumo. Además el dispositivo CC2530 dispone de 5 fuentes de *reset* distintas.

## **Periféricos**

El dispositivo CC2530 cuenta con una serie de periféricos que permiten el desarrollo de aplicaciones avanzadas.

Uno de estos periféricos es el Interfaz de Depuración, que ofrece diferentes funcionalidades. Entre ellas se puede destacar la posibilidad de borrar por completo la memoria *flash*, controlar qué osciladores están habilitados, iniciar y parar la ejecución de programas de usuario y establecer *breakpoints* en código.

Dependiendo del modelo del dispositivo, este incluirá una memoria flash cuya capacidad oscila entre 32 y 256 KB. La elección de este tamaño dependerá de la complejidad del código a almacenar en la misma. Esta memoria puede ser programada desde el *software* externo de usuario y a través del interfaz de depuración. El controlador *flash* gestiona la lectura y escritura en la memoria.

El CC2530 cuenta con un controlador E/S, que es el responsable de gestionar todos los pines de E/S de propósito general. Estos pines pueden ser configurados por la CPU para que puedan ser controlados por *software* o por los periféricos.

Otro periférico es el controlador DMA de 5 canales, con acceso a todas las memorias físicas, que proporciona gran eficiencia en la transferencia de datos para todos los periféricos que lo usan.

El sistema cuenta con 6 temporizadores, cuya funcionalidad se describe a continuación:

- *Temporizador 1*: temporizador de 16 bits con funcionalidad de contador, temporizador y PWM (*Pulse Width Modulator*).
- *Temporizador 2*: también conocido como temporizador MAC, está especialmente diseñado para soportar una capa MAC que cumpla con el protocolo IEEE 802.15.4, o cualquier otro software que implemente un protocolo de tiempo ranurado.
- *Temporizadores 3 y 4*: son temporizadores de 8 bits con funcionalidad de contador, temporizador y PWM.
- *Sleep Timer*: se trata de un temporizador de muy bajo consumo con un reloj de 32 KHz. Se suele usar como contador en tiempo real o en aplicaciones para despertar al dispositivo y sacarlo del modo de alimentación 1 o 2.
- *Watchdog Timer*: permite al dispositivo reiniciarse en caso de que el *firmware* se bloquee.

El CC2530 incluye un coprocesador AES que permite la codificación y decodificación de datos mediante el algoritmo AES con claves de 128 bits.

El convertor analógico-digital proporciona una resolución de entre 7 y 12 bits con anchos de banda que oscilan entre los 30 KHz y los 4 KHz, respectivamente. Además permite automatizar el procesamiento del muestreo periódico o la conversión sobre una secuencia de canales.

La comunicación serie se gestiona con la USART 0 (*Universal Synchronous/Asynchronous Receiver/Transmitter*) y la USART 1, ambas configurables como SPI maestro/esclavo o UART. Además, el CC2531 incluye un controlador USB 2.0 de doble buffer con una memoria RAM de 1KB.

Para finalizar, el CC2530 cuenta con un generador de números aleatorios de 16 bits, que puede ser usado, por ejemplo, para la generación de claves aleatorias para la seguridad en las comunicaciones.

## **Radio**

La familia de dispositivos CC253x incluye un transceptor radio que cumple con la norma IEEE 802.15.4. El núcleo de RF controla los módulos analógicos y proporciona una interfaz entre la MCU y la radio.

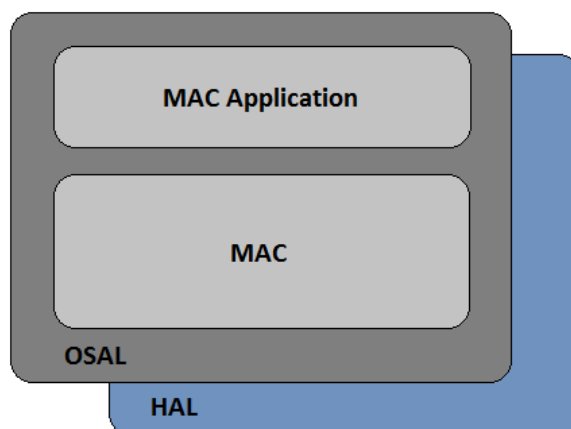
## **3.4. PILA DE PROTOCOLOS TIMAC**

Para el desarrollo de aplicaciones IEEE 802.15.4 y ZigBee, Texas Instruments ha implementado dos pilas de protocolos de acuerdo a estos estándares, TIMAC y Z-Stack, de distribución gratuita.

Por las razones ya comentadas en el apartado 3.2 de este capítulo, para la elaboración del PFC se ha escogido la pila de protocolos TIMAC, en su versión 1.3.0 para el dispositivo CC2530. Esta versión ha sido diseñada y testeada con el compilador IAR EW8051 v.7.60.1, utilizando el lenguaje de programación C. Para el desarrollo de la aplicación de la que es objeto este proyecto, se ha usado la misma versión de compilador para evitar incompatibilidades y errores de compilación, aunque podría haberse usado cualquier versión superior.

En la estructura de la pila de protocolos TIMAC, que puede verse en la figura 3.10, se incluyen las siguientes capas:

- OSAL (*Operating System Abstraction Layer*): Realiza funciones de sistema operativo para controlar el sistema.
- MAC: Capa de control de acceso al medio que se rige por el estándar IEEE 802.15.4.
- MAC *Application*: Capa que aloja y gestiona las aplicaciones diseñadas.
- HAL (*Hardware Abstraction Layer*): Proporciona una plataforma independiente para el manejo de dispositivos, como pueden ser los puertos de entrada y salida o los temporizadores.



**Figura 3.10. Estructura de la pila de protocolos TIMAC.**

En los siguientes apartados se analizará cada una de las capas que conforman la pila de protocolos, con el objetivo de entender el papel que juegan en el desarrollo de la aplicación.

### **3.4.1. CAPA OSAL**

La capa OSAL es una capa de abstracción del sistema operativo, cuyo objetivo es el de proteger la pila del entorno de procesamiento, de modo que la aplicación pueda implementarse con independencia del sistema operativo.

Entre las funciones que realizan se pueden destacar:

- Permite el uso de tareas.
- Provee mecanismos de intercambio de mensajes entre tareas.
- Proporciona funciones de sincronización de tareas, de forma que puedan mantenerse en espera a la ejecución de un evento y ceder el control del procesador.
- Facilita el uso de temporizadores.
- Permite que las tareas sean asociadas a la ejecución de ciertas interrupciones.
- Administra las tareas.
- Facilita la asignación de memoria dinámica.
- Activa el sistema de ahorro energético.

Tanto la elección de la plataforma física como la del protocolo de comunicación están orientadas a la maximización de la autonomía del dispositivo. Aunque en éste se haya configurado el modo de ahorro energético, será la capa OSAL la que se encargue de activarlo o no.

En el bucle principal de esta capa se gestionan las tareas, que están caracterizadas por una serie de parámetros, como pueden ser la prioridad de las mismas, el número de eventos pendientes o la necesidad de activar el ahorro energético. Si todas las tareas que se ejecutan en el bucle tienen activada esta característica, cada vez que finalice una ejecución del bucle sin que queden eventos pendientes, los nodos entrarán en modo de ahorro de energía.

### **3.4.2. CAPA HAL**

La capa de abstracción *hardware* proporciona una interfaz de acceso sencilla desde las capas superiores del protocolo a los elementos *hardware* como pueden ser la UART, los dispositivos de entrada y salida de propósito general, el ADC (*Analog-to-Digital Converter*) o los temporizadores.

La capa HAL dispone de tres tipos de llamadas a funciones para gestionar el *hardware*:

- Funciones de inicialización: este tipo de funciones se utiliza para inicializar servicios o establecer determinados valores sobre parámetros de los elementos *hardware* que intervendrán en la aplicación a desarrollar.
- Funciones de acceso y control: se utilizan para controlar el *hardware* y acceder a ciertos parámetros de los periféricos.
- *Callbacks*: se utilizan para pasar a capas superiores eventos generados por el *hardware*, como pueden ser interrupciones, temporizadores o mecanismos de sondeo.

Para la elaboración de este proyecto ha sido muy importante conocer en profundidad el funcionamiento del hardware del kit de desarrollo CC2530ZDK, y por tanto, los tres tipos de funciones asociadas a la capa HAL, en concreto la inicialización de los servicios de envío y recepción de datos de la UART, ya que las funciones *callback* a este nivel vienen prediseñadas en el código de TIMAC

### 3.4.3. CAPA MAC

La capa MAC provee una interfaz entre la capa de aplicación y la capa de abstracción *hardware*. De manera análoga a la capa HAL, se puede acceder a los servicios que proporciona esta capa mediante los siguientes tipos de funciones:

- Funciones de paso de mensajes: conforman la mayor parte de las funciones asignadas a esta capa, y permiten el paso de mensajes de capas inferiores al procesador de eventos de la MAC o directamente a capas superiores.
- Funciones de ejecución directa: son funciones relativas a eventos de inicialización de servicios.
- *Callbacks*: se emplean para pasar a la capa de aplicación eventos generados en la capa MAC.

### 3.4.4. CAPA DE APLICACIÓN

El código relativo a la programación de la capa de aplicación consta de cuatro elementos principales: la función principal o *main*, la inicialización, el procesado de eventos del sistema y el procesado de *callbacks*.

La función *main* se encarga de comenzar la ejecución del código, desencadenando la inicialización del sistema tanto a nivel *hardware*, como a nivel MAC y OSAL. Además se encarga de iniciar el bucle principal del sistema y añadir tareas a la cola de ejecución.

La ejecución de las tareas vendrá condicionada por la prioridad de las mismas. El protocolo IEEE 802.15.4 define 256 niveles de prioridad, aunque para tareas que tengan el mismo nivel se aplica una política FIFO (*First In First Out*), según el orden de llegada a la cola de ejecución.

En TIMAC se definen tres tipos de tareas, una para cada capa. Las tareas asociadas a la capa MAC tienen los niveles de prioridad más altos, seguida de la capa de aplicación y, finalmente, de la HAL.

Las aplicaciones diseñadas sobre la pila de protocolos TIMAC deben realizar como mínimo las siguientes funciones:

- Configurar los nodos (coordinador o dispositivo final) enviando los parámetros necesarios para ello.
- Gestionar los mensajes recibidos de las capas inferiores mediante una función de procesado de eventos. Los eventos pueden ser propios o del sistema. Los eventos de sistema corresponden a mensajes de la capa MAC, que se ejecutan con prioridad. El resto de eventos se ejecutan de uno en uno, de forma que cada vez que un evento acaba de procesarse se devuelve el control al bucle principal, que se encargará del reparto de recursos.
- Implementar las funciones *callback* para el tratamiento de los eventos generados por la capa MAC o la capa HAL que han sido enviados a la capa de aplicación (pulsación de teclas, recepción de datos a través del puerto serie, excepciones generadas por elementos HW, etc.).
- Diseñar el código necesario para que el funcionamiento del protocolo de comunicaciones cumpla con los objetivos de la aplicación que se quiere desarrollar.



## 3.5 HERRAMIENTAS SOFTWARE

En los siguientes apartados se describe la funcionalidad de las herramientas *software* que se han empleado durante el desarrollo del proyecto.

### 3.5.1. IAR EMBEDDED WORKBENCH EW8051

*IAR Embedded Workbench* es un *software* desarrollado por la empresa *IAR Systems*, que actualmente ofrece una amplia gama de compiladores y depuradores para desarrollar aplicaciones basadas en los microcontroladores más conocidos del mercado.

En concreto, el IAR EW8051 es un compilador C/C++ optimizado, capaz de generar un código muy compacto y eficiente, que da soporte para la creación de aplicaciones sobre una gran variedad de dispositivos basados en microcontroladores de la familia 8051. Ofrece una gran cantidad de funcionalidades, como la posibilidad de insertar puntos de ruptura en el código o realizar el seguimiento de variables para la mejora del rendimiento.

La versión utilizada para la elaboración de este proyecto ha sido la 7.60.1, ya que es la misma con la que se compiló la pila de protocolos TIMAC 1.3.0. En la siguiente figura se muestra el aspecto de la interfaz gráfica de usuario.

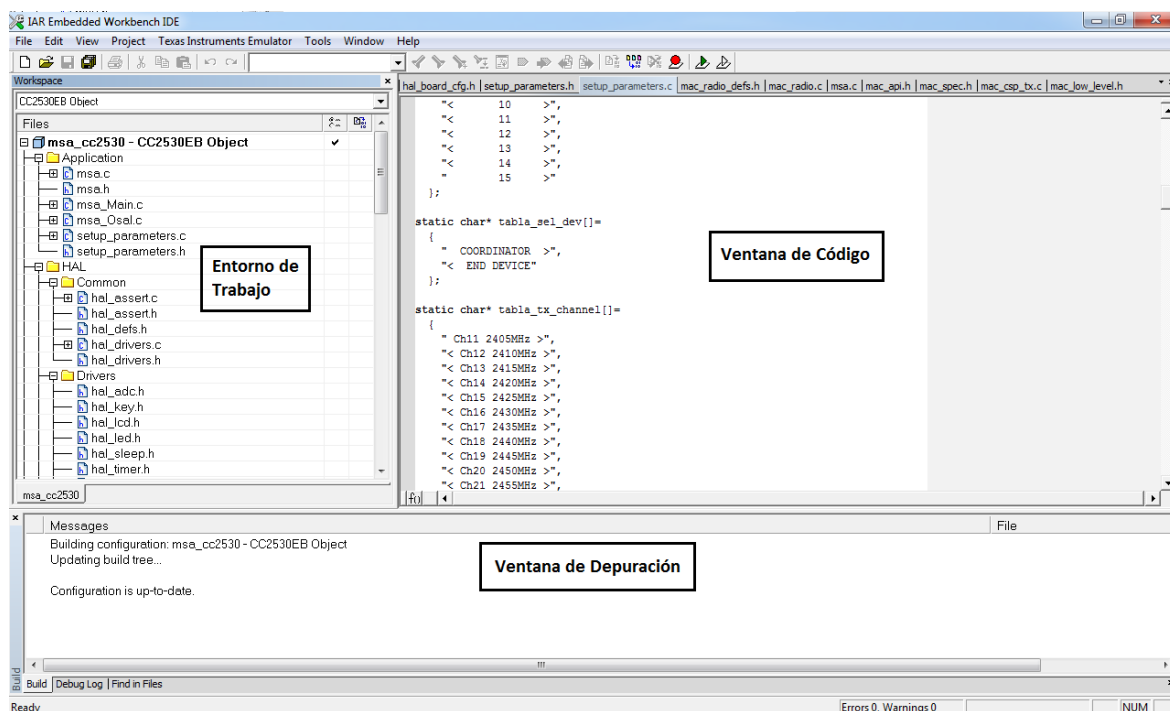


Figura 3.11. Interfaz de Usuario de IAR EW8051 v7.60.1.

La interfaz de usuario se compone de tres ventanas:

- Entorno de trabajo: muestra la estructura de ficheros asociada al proyecto que se esté llevando a cabo. Permite una fácil localización de los archivos ya que estos estarán organizados atendiendo a la estructura de la pila de protocolos.
- Ventana de código: visualiza el código asociado a los archivos que componen el proyecto, permitiendo la edición del mismo.
- Ventana de depuración: muestra el resultado obtenido durante la compilación del código, indicando posibles errores y avisos si los hubiere.

### 3.5.2. LABWINDOWS/CVI

LabWindows/CVI es un *software* creado por la compañía americana *National Instruments*, que proporciona un entorno de desarrollo integrado que cuenta con una gran cantidad de herramientas para la creación, test y control de aplicaciones.

Cabe destacar su facilidad de uso ya que utiliza ANSI C como lenguaje de programación. Está especialmente concebido para el desarrollo de aplicaciones para el control de instrumentación electrónica, adquisición de datos, análisis y creación de interfaces de usuario. Para facilitar esta labor incluye un gran conjunto de librerías con funciones predefinidas.

Para el caso específico de este proyecto, LabWindows/CVI se ha utilizado para el desarrollo de una aplicación que proporcionase una interfaz gráfica sencilla y que permitiese el envío de parámetros de configuración a través del puerto serie a los nodos que conforman el kit CC2530ZDK.

En la figura 3.12 se muestra el aspecto de la interfaz de usuario de la aplicación, en su versión 9.0, en la que se pueden diferenciar tres ventanas:

- Entorno de trabajo: muestra la estructura de ficheros asociada al proyecto que se esté llevando a cabo.
- Ventana de Código: visualiza el código asociado a los archivos que componen el proyecto, permitiendo la edición del mismo.

- Ventana de librerías: muestra la estructura de librerías predefinidas, permitiendo el acceso a las funciones que las componen para poder usarlas en las aplicaciones que se estén desarrollando.

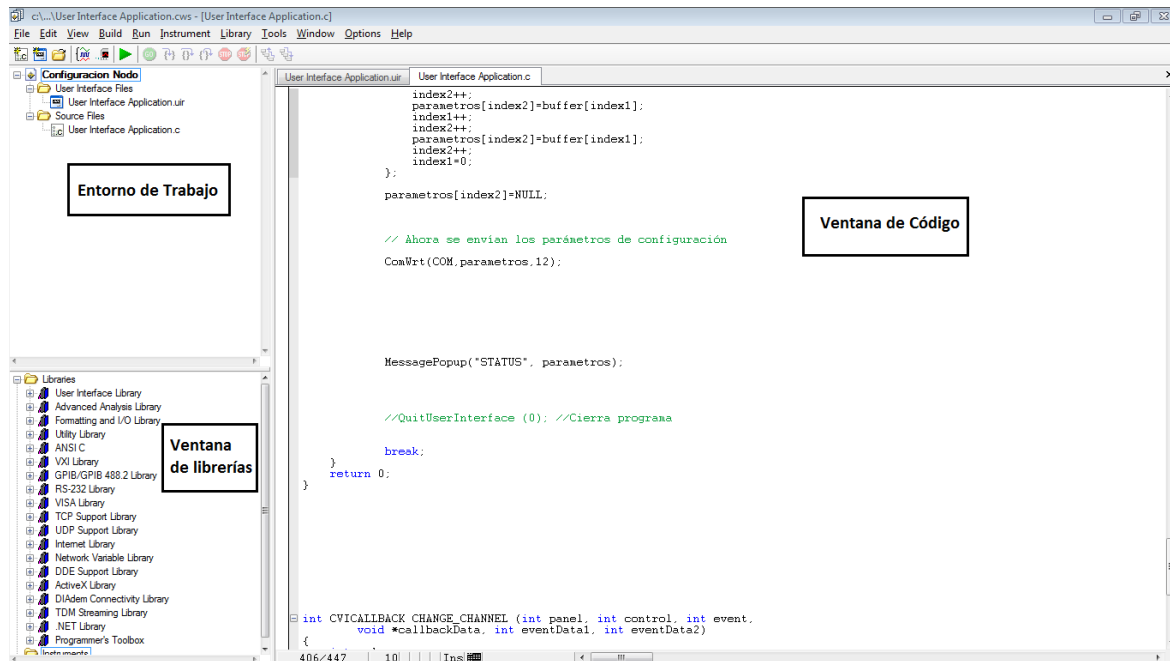


Figura 3.12. Interfaz de usuario de LabWindows/CVI

### 3.5.3. PACKET SNIFFER

El software *Packet Sniffer* es una aplicación gratuita desarrollada por *Texas Instruments* cuya finalidad es la de mostrar y almacenar paquetes de RF capturados por un receptor RF escuchando en un determinado canal. Esta aplicación es capaz de decodificar y filtrar los paquetes para mostrar su información de manera conveniente según las necesidades del usuario. Además soporta varios protocolos de RF, como IEEE 802.15.4, *ZigBee* 2003, *ZigBee* 2006, *ZigBee Pro*, *ZigBee* RF4CE o protocolos genéricos.

El funcionamiento de la aplicación es muy sencillo. Tras seleccionar el protocolo de RF del que se quiere extraer información, se conecta el receptor a un puerto USB del ordenador, se selecciona el canal a escuchar y se inicia la captura de datos. Si hay algún dispositivo de RF que transmita en el canal elegido según el protocolo seleccionado, automáticamente aparecerán los paquetes en pantalla.

Estos paquetes aparecen directamente decodificados, mostrando información relevante como el orden de llegada, el tiempo transcurrido desde la recepción del último paquete, la longitud, el tipo de trama, el número de secuencia, identificación de la red, BO, SO, etc. En la figura 3.13 se muestra un ejemplo de captura de datos.

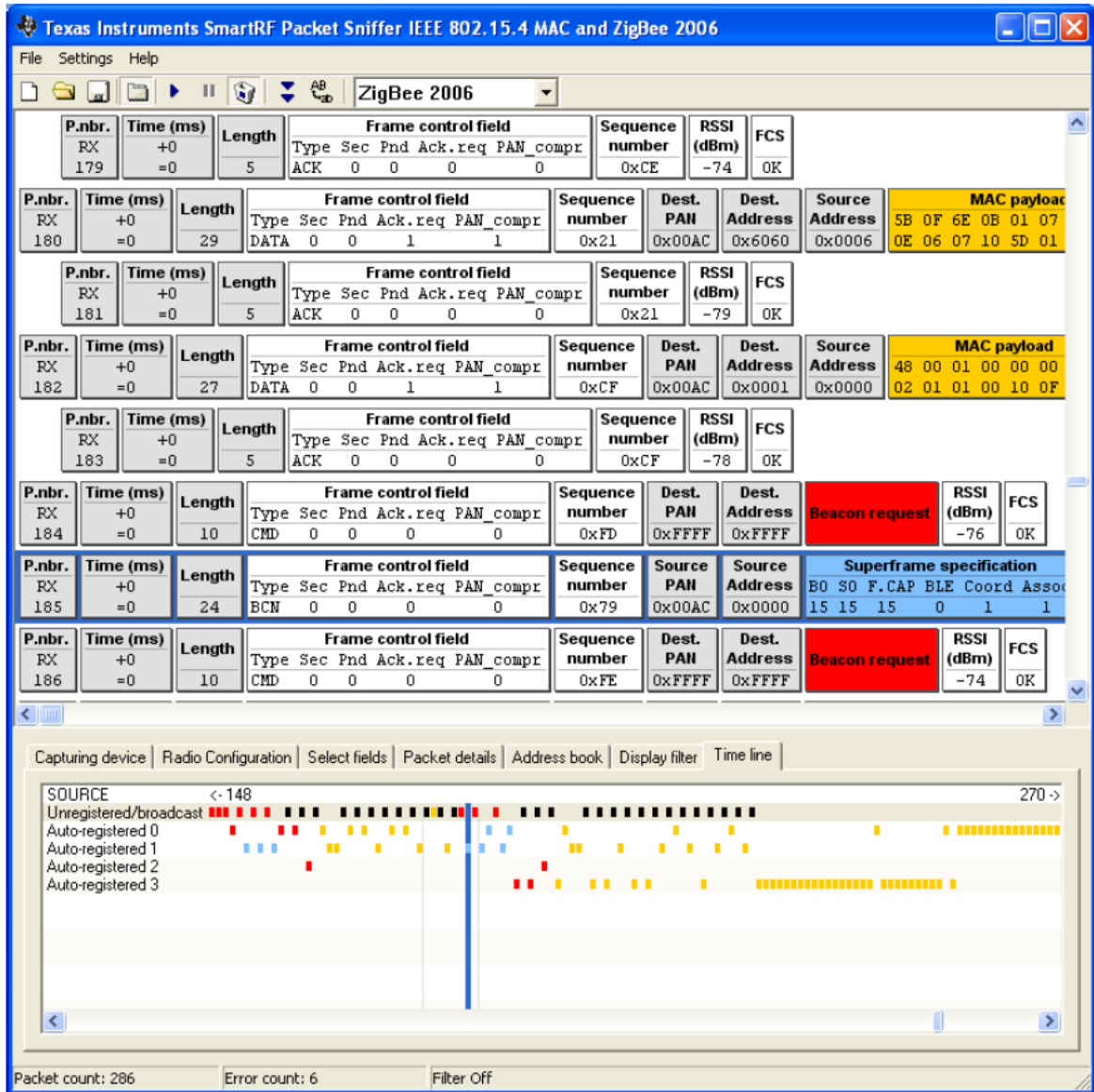


Figura 3.13. Ejemplo de captura de datos usando Packet Sniffer de Texas Instruments

El kit CC2530ZDK incluye el dispositivo CC2531 USB Dongle, con el *firmware* para la captura de datos precargado y listo para ser usado en conjunto con el *software Packet Sniffer*.



# CAPÍTULO 4

## Desarrollo software

### 4.1. INTRODUCCIÓN

Tal y como se explicó en el capítulo introductorio de este PFC, uno de los objetivos más importantes que trata de impulsar la tecnología ZigBee/802.15.4, es la posibilidad de hacer funcionar los nodos que conforman este tipo de redes en modo balizado.

El desarrollo de este proyecto está basado en la aplicación ejemplo *MSA (MAC Sample Application)* proporcionada junto a la pila de protocolos TIMAC de *Texas Instruments*. Esta aplicación permite la creación de redes 802.15.4, balizadas o no balizadas, que siguen una topología en estrella, en las que el coordinador de red será el primer nodo que intente transmitir en un determinado canal. Una vez establecido el coordinador de red, los nodos que intenten transmitir en el mismo canal deberán unirse a dicha red, adaptándose a los parámetros que la rigen.

Antes de crear una red o unirse a una ya existente es necesario establecer los parámetros básicos de configuración de cada nodo, como son el *Beacon Order* (que determinará el carácter balizado o no balizado de la red), el *Superframe Order*, el canal de transmisión y la potencia de transmisión.

Inicialmente, la aplicación ejemplo *msa* solo permite la configuración de los parámetros que determinan el carácter balizado o no balizado de la red mediante la modificación de los mismos directamente en el código de programación de la pila de protocolos. Esto supone que, cada vez que se necesita cambiar alguno de estos valores, hay que compilar de nuevo el código y cargarlo en cada uno de los nodos.

Partiendo de estas premisas, surge la necesidad de crear una aplicación que permita la modificación en tiempo de ejecución de los parámetros que regulan el funcionamiento de los nodos.

El desarrollo de esta aplicación forma parte de los objetivos principales de este proyecto, facilitando así el posterior estudio y evaluación del funcionamiento de redes balizadas.

## 4.2. ESPECIFICACIONES

La aplicación a desarrollar debe permitir la configuración de los parámetros básicos por los que se rige una red balizada, teniendo en cuenta las limitaciones impuestas por la versión utilizada de la pila de protocolos, entre las que destaca el hecho de no poder habilitar la funcionalidad de *slots* temporales garantizados (o GTS), o la imposibilidad de crear redes que sigan una topología distinta a la topología en estrella. Dentro de los parámetros configurables se consideran los siguientes:

- **Beacon Order (BO):** toma valores entre 0 y 14 para redes balizadas, y 15 para redes no balizadas. En el caso de tratarse de una red balizada, el tiempo entre balizas o BI, viene determinado por la ecuación 2.3.
- **Superframe Order (SO):** determina la duración del período activo dentro de una estructura de supertrama. Para ello toma valores entre 0 y 15, debiendo ser siempre menor o igual que el valor de BO. La duración de la supertrama, o SD, viene determinada por la ecuación 2.4.
- **Canal de Comunicación:** el estándar IEEE 802.15.4 establece 16 canales de comunicación, numerados del 11 al 26, para la banda ISM de 2.4 GHz. La frecuencia central de cada uno de estos canales se obtiene aplicando la ecuación [2.2].
- **Potencia de Transmisión:** debe ajustarse para minimizar al máximo el consumo de los nodos y aumentar así su autonomía. Los transceptores de los módulos CC2530EM ofrecen potencias de salida que varían entre los -22 dBm y los 4.5 dBm.

Para aprovechar al máximo la funcionalidad de los nodos, se propuso la programación de dos aplicaciones. La primera debía permitir la introducción manual de los parámetros usando los periféricos disponibles en la placa de desarrollo SmartRF05EB, evitando así la

necesidad de conectar los nodos a un sistema de configuración externo. La segunda debía proporcionar un interfaz gráfico de usuario sencillo que permitiera seleccionar y validar los parámetros antes de ser enviados a través del puerto serie del PC.

Para la programación de la plataforma de configuración a través del puerto serie no se impuso ninguna restricción en lo que a lenguaje de programación se refiere, si bien, tras el estudio de diferentes opciones, se decidió utilizar el *software* LabWindows/CVI, que utiliza el lenguaje de programación C y que permite la generación de un instalador que incluye todas las librerías necesarias para que la aplicación funcione correctamente en entornos *Windows* XP o superiores sin la necesidad de tener instalado LabWindows en los ordenadores destino.

### 4.3. FASE DE DESARROLLO

El desarrollo *software* de este proyecto se ha dividido en dos etapas.

En la primera etapa se ha programado una aplicación que aprovecha la funcionalidad de las placas de desarrollo SmartRF05EB para la configuración de los nodos sin la necesidad de conectarlos a sistemas de configuración externos. Esta aplicación consta de una serie de menús dinámicos que se muestran en la pantalla LCD de la placa, permitiendo cada uno de ellos la selección del valor de uno de los parámetros de configuración del nodo, así como la propia selección del método de configuración. La navegación dentro de los menús y la selección de valores se realiza mediante el uso de los botones de acción de la placa etiquetados como S1 y *Joystick*.

Una ventaja importante a destacar de esta aplicación es la posibilidad de configurar los nodos de manera independiente en emplazamientos de difícil accesibilidad.

En la segunda etapa se ha programado una aplicación que permite la selección de los parámetros de configuración mediante una interfaz gráfica de usuario sencilla y el envío de los mismos a través del puerto serie del ordenador. Para el correcto funcionamiento de esta aplicación es necesaria la modificación de parte del código de la pila de protocolos TIMAC. En concreto es necesario preparar los nodos para la recepción de los datos de configuración a través del puerto serie, así como la decodificación de los mismos. Además, se reutilizan algunas funciones programadas para la configuración de los nodos



directamente en las placas, como pueden ser las funciones relacionadas con la visualización de los menús dinámicos.

Como nexo de unión entre ambas aplicaciones, se ha añadido la posibilidad de seleccionar el método de configuración de manera independiente en cada uno de los nodos. En la figura 4.1 se muestra el diagrama de flujo del funcionamiento básico del sistema.

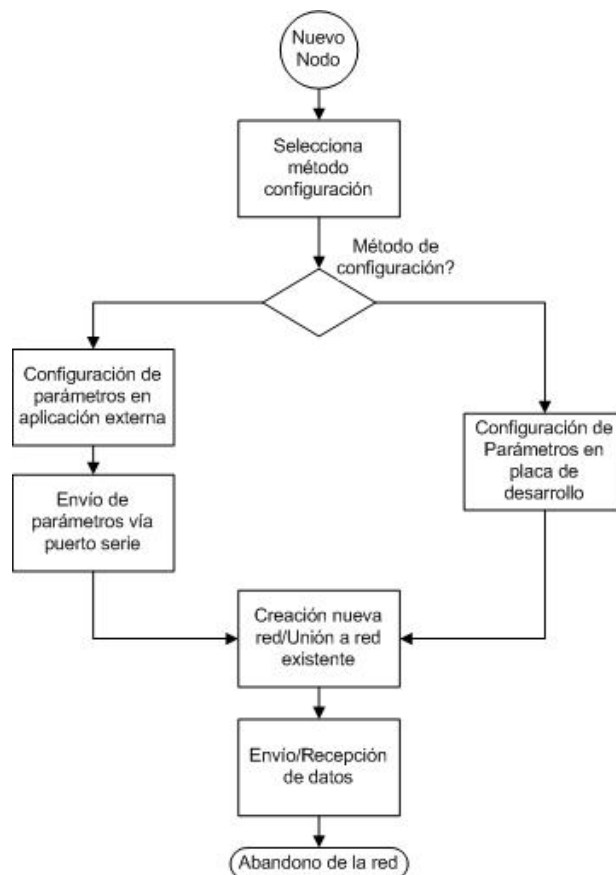


Figura 4.1. Diagrama de flujo de funcionamiento básico del sistema.

### 4.3.1. CONFIGURACIÓN EN PLACAS DE DESARROLLO

Tal y como se ha indicado anteriormente, para añadir la funcionalidad de configuración de los parámetros que determinan una red balizada, ha sido necesario modificar parte del código de la aplicación ejemplo suministrada junto a la pila de protocolos TIMAC. En concreto, todas las modificaciones se han hecho en la capa de aplicación. Para intentar complicar lo mínimo posible el código original, se ha creado la biblioteca

“*setup\_parameters*” que contiene parte de las funciones desarrolladas, las cuales se describen a continuación.

***void print\_menu (uint8 param , uint8 index)***

Función que gestiona la impresión de los diferentes menús de configuración en la pantalla LCD de la placa de desarrollo SmartRF05EB v1.7.

Entradas:

*param*: constante que permite identificar el parámetro que se quiere configurar para así seleccionar el menú adecuado de entre los disponibles. En la siguiente tabla se muestra una lista con los diferentes parámetros disponibles y una breve descripción de cada uno de ellos.

Parámetro	Descripción
COORD	Coordinador de Red
DEV	Dispositivo Final
SEL	Selección de Dispositivo
CHANNEL	Canal de Transmisión
CONF_TYPE	Método de Configuración
TX_POWER	Potencia de Transmisión
BO	Beacon Order
SO	Superframe Order

**Tabla 4.1. Lista de Parámetros de Configuración**

*index*: número entero que indica el desplazamiento dentro de las tablas en las que se almacenan los diferentes menús. Cada menú se almacena dentro de una tabla en la que cada registro es una cadena de caracteres. En el siguiente cuadro se muestra la estructura de datos correspondiente al menú de configuración de la potencia de transmisión.

```
static char* tabla_tx_power [ ] =
{
    " -22 dBm >",
    "< -18 dBm >",
    "< -14 dBm >",
    "< -10 dBm >",
    "< -6 dBm >",
    "< -3 dBm >",
    "< -0.5 dBm >",
    "< 1 dBm >",
    "< 2.5 dBm >",
    "< 4.5 dBm"
};
```

**Tabla 4.2. Estructura de datos para el almacenamiento del menú Potencia de Transmisión.**

Salidas:

Ninguna.

Una vez seleccionado el menú de configuración y el registro a mostrar en la pantalla LCD, la representación se realiza mediante la función de la capa de abstracción *hardware HalLcdWriteString ( char \*str , uint8 option)* a la que se le pasa como primer parámetro la cadena de caracteres que se desea imprimir, y como segundo parámetro la línea de la pantalla LCD en la que se desea visualizar el texto. En el caso de la versión 1.7 de la placa SmartRF05EB, la pantalla LCD cuenta con 3 líneas de 16 caracteres cada una, lo que limita el número de opciones que se le pueden pasar a la función. En la siguiente tabla se recoge la lista de opciones disponibles.

Option
HAL_LCD_LINE_1
HAL_LCD_LINE_2
HAL_LCD_LINE_3

**Tabla 4.3. Opciones de Impresión para la función *HalLcdWriteString*.**

**uint8 set\_tx\_ch (uint8 device)**

Esta función permite seleccionar el canal de transmisión para el nodo que se está configurando.

**Entradas:**

*device*: número entero que representa el tipo de dispositivo que se va a configurar. Toma el valor ‘0’ en caso de tratarse de un coordinador de red, y ‘1’ si se trata de un dispositivo final.

**Salidas:**

La función devuelve un número entero que se corresponde con el canal de transmisión seleccionado. Según el estándar 802.15.4 para comunicaciones en la banda ISM de 2.4 GHz se definen 16 canales de transmisión que van desde los 2405 MHz hasta los 2480 MHz. En la siguiente tabla se muestra la correspondencia entre la frecuencia central de cada canal y el número que lo representa, que en este caso es el valor que devuelve la función.

Frecuencia (MHz)	Número de Canal
2405	11
2410	12
2415	13
2420	14
2425	15
2430	16
2435	17
2440	18
2445	19
2450	20
2455	21
2460	22
2465	23
2470	24
2475	25
2480	26

**Tabla 4.4. Correspondencia entre el número de canal y su frecuencia asociada.**

**uint8 set tx power (uint8 device)**

Función que permite establecer la potencia de transmisión de cada uno de los nodos que conforma la red.

Entradas:

*device*: número entero que representa el tipo de dispositivo que se va a configurar. Toma el valor ‘0’ en caso de tratarse de un coordinador de red, y ‘1’ si se trata de un dispositivo final.

Salidas:

La función devuelve un valor codificado en hexadecimal correspondiente a la potencia de transmisión asignada para la inicialización del nodo. El subsistema de radiocomunicaciones que integra el módulo CC2530EM permite establecer potencias de transmisión que oscilan entre los -22 dBm y los 4.5 dBm. En la tabla 4.5 se muestra la codificación para cada uno de los valores de potencia de transmisión disponibles.

Potencia TX (dBm)	Codificación Hex.
-22	0x05
-18	0x25
-14	0x45
-10	0x65
-6	0x85
-3	0xA5
-0.5	0xC5
1	0xD5
2.5	0xE5
4.5	0xF5

**Tabla 4.5. Codificación Hexadecimal de la Potencia de Transmisión en módulos CC2530**

**uint8 set\_BO (uint8 device)**

Función que permite seleccionar el valor del parámetro *Beacon Order* (BO) para la configuración del nodo, determinando así el carácter balizado o no de la red.

**Entradas:**

*device*: número entero que representa el tipo de dispositivo que se va a configurar. Toma el valor ‘0’ en caso de tratarse de un coordinador de red, y ‘1’ si se trata de un dispositivo final.

**Salidas:**

La función devuelve un número entero con el valor de BO seleccionado, que varía entre 0 y 14 para redes balizadas, y 15 para redes no balizadas.

**uint8 set\_SO (uint8 device, uint8 beacon\_ord)**

Esta función permite seleccionar el valor del parámetro *Superframe Order* (SO) en el caso de tratarse de una red balizada con estructura de supertrama.

**Entradas:**

*device*: número entero que representa el tipo de dispositivo que se va a configurar. Toma el valor ‘0’ en caso de tratarse de un coordinador de red, y ‘1’ si se trata de un dispositivo final.

*beacon\_ord*: valor del parámetro BO con el que se pretende configurar el nodo. Este parámetro es necesario para asegurarse de que el parámetro SO cumple la restricción de que debe ser menor o igual que BO.

**Salidas:**

La función devuelve un número entero con el valor de SO seleccionado.

**uint8 set\_device\_type (void)**

Función que permite seleccionar el tipo de dispositivo que se desea configurar, pudiendo optarse entre Coordinador PAN y Dispositivo Final.

Entradas:

Ninguna.

Salidas:

La función devuelve un número entero que representa el dispositivo seleccionado. En el caso de tratarse de un Coordinador PAN se devuelve un '0', y un '1' si se trata de un Dispositivo Final.

**uint8 set\_conf\_type (void)**

Función que permite seleccionar la aplicación que se va a utilizar para la configuración de cada nodo.

Entradas:

Ninguna.

Salidas:

La función devuelve un '0' en el caso de seleccionarse la aplicación para la configuración de los nodos de red mediante el uso de la placa de desarrollo SmartRF05EB, y un '1' si se selecciona la aplicación externa.

Como característica común entre las funciones anteriormente descritas, para el desplazamiento dentro de los menús de configuración y la aceptación de valores, se han usado tanto el *Joystick* como el botón S1, presentes en la placa SmartRF05EB. La detección de pulsación de los mismos se ha resuelto mediante la consulta de los *flags* de interrupción *hardware* asociados.

<b>Botón</b>	<b>Flag de Interrupción</b>
<i>Joystick</i>	P2IFG
S1	P0IFG

**Tabla 4.6. Flags de Interrupción asociados a los botones *Joystick* y S1.**

Aparte del desarrollo de las funciones anteriormente descritas, ha sido necesario modificar parte del código correspondiente a la inicialización de los nodos, permitiendo que sea durante esta fase cuando se introduzcan los parámetros de configuración de la red. A continuación se muestra la función *MSA\_Init* modificada y comentada:

```

void MSA_Init (uint8 taskId)
{
    uint8 i, conf_type, ch, txp, dev;

    /* Inicialización de los parámetros de configuración de la red */
    dev = set_device_type (); // Selección del tipo de dispositivo a configurar
    ch = set_tx_ch (dev); // Selección del canal de transmisión
    MAC_RADIO_SET_CHANNEL (ch); // Establece canal de transmisión seleccionado
    txp = set_tx_power (dev); // Selección de la potencia de transmisión
    MAC_RADIO_SET_TX_POWER (txp); // Establece potencia de transmisión seleccionada
    MSA_MAC_BEACON_ORDER = set_BO (dev); // Selección del parámetro Beacon Order
    // Selección del parámetro Superframe Order
    if (MSA_MAC_BEACON_ORDER<15)
    {
        MSA_MAC_SUPERFRAME_ORDER = set_SO (dev,MSA_MAC_BEACON_ORDER);
    }
    else if (MSA_MAC_BEACON_ORDER==15)
    {
        MSA_MAC_SUPERFRAME_ORDER=15;
    };
    //Establece los parámetros Beacon Order y Superframe Order
    msa_BeaconOrder = MSA_MAC_BEACON_ORDER;
    msa_SuperFrameOrder = MSA_MAC_SUPERFRAME_ORDER;

    /* Inicialización de la identificación de tarea */
    MSA_TaskId = taskId;

    /* Inicialización de las características generales de la capa MAC */
    MAC_InitDevice();
    MAC_InitCoord();

    /* Inicialización de las características específicas de la capa MAC que permiten la creación
    de redes balizadas */
    MAC_InitBeaconDevice();
    MAC_InitBeaconCoord();

    /* Reseteo de la capa MAC*/
    MAC_MlmeResetReq(TRUE);

```



```

/* Inicialización del paquete de datos */
for (i=MSA_HEADER_LENGTH; i<MSA_PACKET_LENGTH; i++)
{
    msa_Data1[i] = i-MSA_HEADER_LENGTH;
}

/* Inicialización del paquete de eco*/
for (i=0; i<MSA_ECHO_LENGTH; i++)
{
    msa_Data2[i] = 0xEE;
}
}

```

### 4.3.2. PLATAFORMA DE CONFIGURACIÓN VÍA PUERTO SERIE

El desarrollo de la plataforma para la configuración de los nodos a través del puerto serie ha constado de dos fases. En la primera se ha programado la interfaz gráfica de usuario para la configuración de nodos en sistemas operativos *Windows*. En la segunda fase se ha modificado el código de la pila de protocolos TIMAC, preparándolo para la recepción de los datos a través del puerto serie y su posterior decodificación.

#### 4.3.2.1. Interfaz Gráfica de Usuario

Tal y como se indicó al comienzo de este capítulo, la programación de la interfaz gráfica de usuario se ha llevado a cabo mediante el *software* LabWindows/CVI, permitiendo la visualización y edición de los parámetros a configurar de una manera fácil e intuitiva, y posibilitando su posterior envío a través del puerto serie del ordenador.

LabWindows/CVI incorpora una gran cantidad de librerías de objetos (también llamados controles) de diversa naturaleza, como pueden ser objetos de tipo numérico, texto, gráfico, botón de acción, botón de comando, reloj, dial, barra de desplazamiento, etc. Cada uno de ellos responde a una serie de atributos que permiten configurarlo de acuerdo a los requerimientos de la aplicación que se esté diseñando. Además, cabe la posibilidad de generar una función *callback* asociada a cada control para así poder tratar los eventos generados por el mismo.

En general, para cada control existen cuatro tipos de atributos configurables, los cuales se describen a continuación:

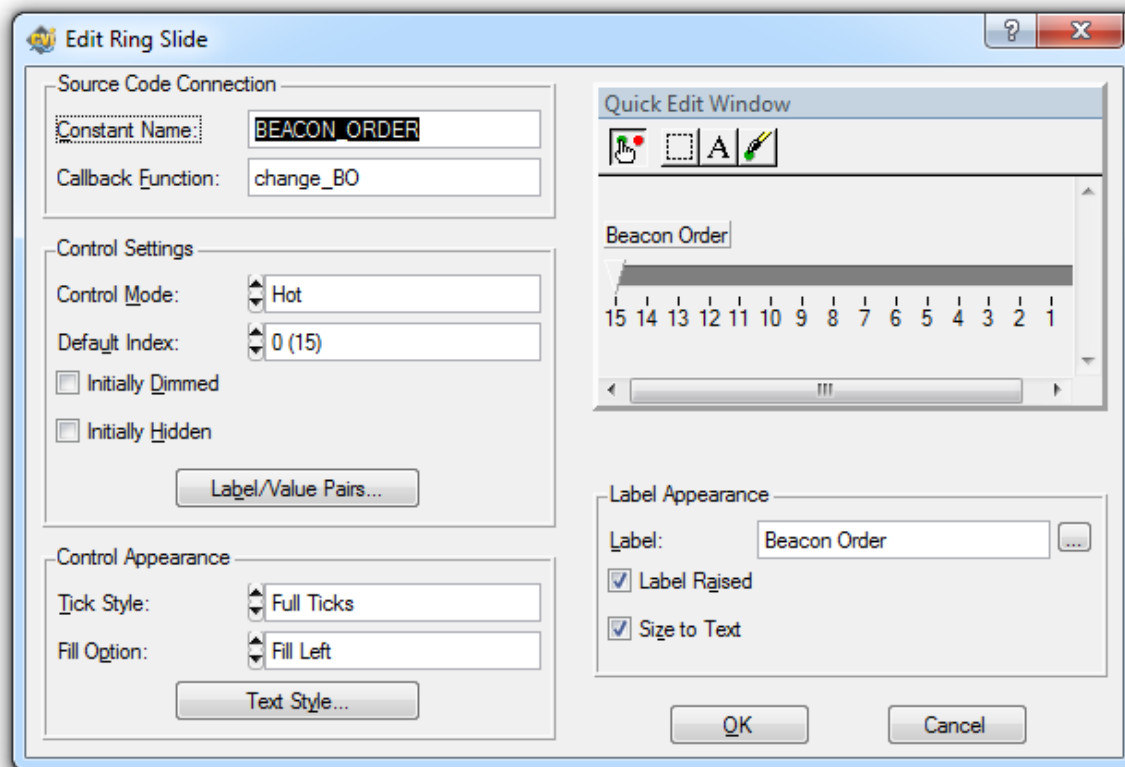
- *Source Code Connection*: en esta sección se define el nombre con el que se puede referenciar el control desde el código fuente y la función *callback* asociada al mismo.
- *Control Settings*: parámetros de configuración directamente relacionados con la funcionalidad del control. El parámetro *Control Mode* hace referencia al comportamiento de un control cuando se actúa sobre él. En la tabla 4.7 se describen los 4 modos disponibles.

<b>Modo</b>	<b>Descripción</b>
<i>Normal</i>	Permite que el usuario pueda actuar directamente sobre el control y que además éste pueda ser modificado en tiempo de ejecución por el propio programa.
<i>Indicator</i>	El control solo puede ser modificado por el propio programa en tiempo de ejecución. Los controles tipo LED, <i>Text Message</i> y <i>Graph</i> son siempre de este tipo.
<i>Hot</i>	Su funcionamiento es igual que el del modo <i>normal</i> excepto que en este modo el control genera un evento cuando el usuario actúa sobre él. Los eventos se devuelven al programa de aplicación y éste determina las acciones a realizar.
<i>Validate</i>	Es idéntico al modo <i>hot</i> , excepto por que todos los controles del panel se comprueban con respecto a un rango predefinido antes de generar el evento. Si se encuentra una condición no válida LabWindows/CVI genera una caja de diálogos indicando el porqué y dando la opción de poder modificar el valor.

**Tabla 4.7. Modos de funcionamiento de los controles en LabWindows/CVI.**

- *Control Appearance*: parámetros relacionados con la apariencia de los controles. Estos parámetros varían dependiendo del tipo de control.
- *Label Appearance*: parámetros de configuración de la etiqueta asociada a cada control.

En la siguiente figura se muestra la ventana de edición de un control del tipo *Ring Slide*.



**Figura 4.2. Ejemplo de ventana de edición de un control del tipo *Ring Slide*.**

Tal y como se ha indicado anteriormente, a cada control se le puede asignar una función *callback*. Este tipo de funciones se encarga de tratar los eventos generados por los controles a los que están asociadas. La estructura de la cabecera de las funciones *callback* es siempre la misma, independientemente del tipo de control al que pertenezcan. A continuación se detallan los parámetros que las componen:

***int CVICALLBACK CallbackFunction (int panel, int control, int event, void \*callbackData, int eventData1, int eventData2)***

Entradas:

*panel*: informa del panel donde se generó el evento.

*control*: informa del control que generó el evento.

*event*: indica el tipo de evento generado (LEFT\_CLICK, EVENT\_COMMIT, etc.).

*\*callbackData*: valor definido por el usuario que se pasa a la función.

*eventData1, eventData2*: informan de distintos aspectos según el tipo de evento generado. Por ejemplo, si se ha pulsado el ratón estos parámetros contienen las

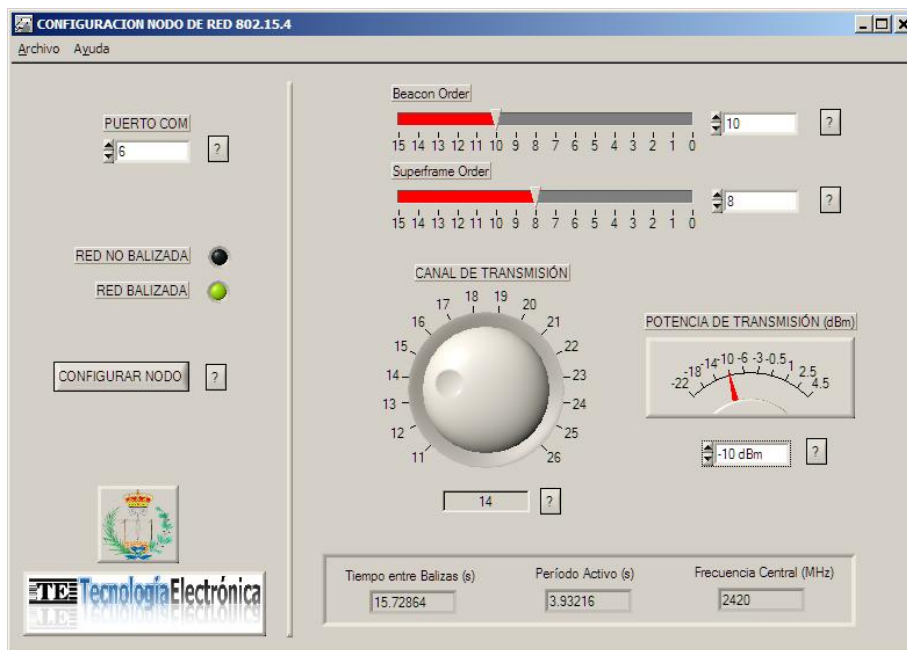
coordenadas X e Y de la posición donde se pulsó el ratón, sin embargo si se pulsa una tecla del teclado estos parámetros contienen el código de la tecla pulsada.

#### Salidas:

La función devuelve un 0 en caso de ejecutarse correctamente.

Para el desarrollo de la aplicación de configuración de nodos 802.15.4 se han seguido los siguientes pasos:

- Crear un nuevo proyecto.
- Asociar un nuevo panel para la generación del interfaz gráfico de usuario.
- Insertar en el panel los controles necesarios para conseguir la funcionalidad deseada (en la siguiente figura se muestra el *layout* de la aplicación).



**Figura 4.3. Layout de la aplicación de configuración de nodos 802.15.4.**

- Configurar cada control de acuerdo a las especificaciones de los parámetros a los que representan, como son el *Beacon Order*, *Superframe Order*, Canal de Transmisión, Potencia de Transmisión, Puerto de Comunicaciones, Diodos LED, Botones de Ayuda, y Botón de Configuración.
- Generar las funciones *callback* para tratar los eventos generados por cada uno de los controles y conseguir la funcionalidad global deseada del sistema.

A continuación se desarrolla el proceso configuración de los controles más representativos y de las funciones *callback* asociadas.

### ***Beacon Order***

El control *Beacon Order* permite la selección del valor del parámetro de configuración con el mismo nombre. Se trata de un control del tipo *Ring Slide* que representa un número entero en un rango de valores entre 0 y 15 (de acuerdo al estándar IEEE 802.15.4). La función *callback* asociada, *Change\_BO*, realiza las siguientes tareas en caso de modificarse el valor del control:

- Si el nuevo valor de BO es igual a 15 se actualiza el parámetro SO con el mismo valor, se activa el diodo LED de indicación de red no balizada, y se desactiva el de indicación de red balizada.
- Si el nuevo valor de BO es menor que 15 se desactiva el diodo LED de indicación de red no balizada y se activa el de indicación de red balizada. Si además el valor de BO es menor que SO, se actualiza el valor de este último con el mismo valor de BO, ya que, según la norma IEEE 802.15.4, en ningún caso el valor del parámetro SO puede ser mayor que BO.

Para consultar el valor que tiene un determinado control desde el código de ejecución se utiliza la función *int GetCtrlVal (int PanelHandle, int ControlID, void \*Value)*, en la que *PanelHandle* indica el panel donde está ubicado el control que se desea consultar, *ControlID* indica la referencia del control a consultar, y *\*Value* es un puntero a la variable en la que se va a almacenar el valor del control.

De manera análoga, la función utilizada para modificar el valor de un determinado control es la función propia de LabWindows/CVI *int SetCtrlVal (int PanelHandle, int ControlID, ...)* en la que el tercer parámetro indica el valor con el que se quiere modificar el control, debiendo coincidir en tipo.

### ***Superframe Order***

El control *Superframe Order* permite la selección del valor del parámetro SO para nodos que operan en redes con estructura de supertrama. Se trata de un control del tipo *Ring Slide* que representa un número entero en un rango de valores entre 0 y 15 (de acuerdo al estándar IEEE 802.15.4), debiendo ser siempre menor o igual que el parámetro

*Beacon Order*. La función *callback* asociada, *Change\_SO*, se encarga de igualar el valor de SO con el de BO si se intenta modificar su valor con un valor mayor que el de BO.

### ***Canal de Transmisión***

El control *Canal de Transmisión*, del tipo *Ring Knob*, permite seleccionar el canal de transmisión del nodo que se pretende configurar. La selección se realiza entre 16 posibles valores enteros, del 11 al 26, según la numeración asignada en la norma IEEE 802.15.4. La correspondencia entre el número de canal y la frecuencia central del mismo se puede consultar en la tabla 4.4.

### ***Potencia de Transmisión***

El control *Potencia de Transmisión* permite seleccionar la potencia de salida de los nodos CC2530 de acuerdo a sus especificaciones. Se trata de un control del tipo *Ring Meter* que representa los 10 posibles valores de potencia de salida, entre los -22 dBm y los 4.5 dBm. A cada valor de potencia se le asigna un número entero entre 0 y 9 que será decodificado por el propio nodo cuando reciba los datos de configuración.

### ***Puerto COM***

El control *Puerto COM*, de tipo *Numeric*, permite seleccionar el puerto serie que se va a utilizar para enviar la información de configuración a los nodos.

### ***Diodos LED***

Los tres diodos son controles del tipo LED. A diferencia del resto de controles, cuyo modo de funcionamiento es *Hot*, el de los diodos es *Indicator*, de forma que su estado solo puede ser modificado por el código de ejecución del programa. La finalidad de estos diodos es indicar si los parámetros de configuración se corresponden con los de una red balizada o con los de una red no balizada. Por ello, el diodo etiquetado como *RED NO BALIZADA* permanecerá encendido mientras el valor del parámetro BO sea igual a 15. En cualquier otro caso se activa el diodo *RED BALIZADA*.

### ***Tiempo entre Balizas***

Se trata de un control de tipo *string* y modo de funcionamiento *Indicator* que muestra el tiempo previsto entre balizas de acuerdo al valor del parámetro *Beacon Order* seleccionado (su valor se actualiza cada vez que se modifica el valor de este parámetro).

### **Período Activo**

Se trata de un control de tipo *string* y modo de funcionamiento *Indicator* que muestra el valor del período activo previsto en el caso de que se esté configurando una red de carácter balizado. Su valor se actualiza cada vez que se modifica el valor del control *Superframe Order*.

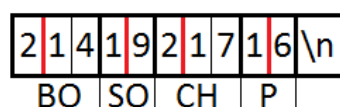
### **Frecuencia Central**

Al igual que los dos controles anteriores, se trata de un control de tipo *string*, cuyo cometido es el de mostrar la frecuencia central correspondiente al canal de transmisión seleccionado. Debido a que su modo de funcionamiento es *Indicator*, su valor se actualiza cada vez que se actúa sobre el control Canal de Transmisión.

### **Configurar Nodo**

El botón Configurar Nodo es un control del tipo *Command Button* cuya pulsación genera un evento que llama a la función *callback* etiquetada como ENVIAR. Las tareas principales que se llevan a cabo en esta función son las siguientes:

- Configurar el puerto COM seleccionado. La configuración se realiza mediante la función de LabWindows *OpenComConfig(int ComPort, char DeviceName[], long BaudRate, int Parity, int DataBits, int StopBits, int InputQueueSize, int OutputQueueSize)*. Los valores de estos parámetros de configuración deben ser los mismos que los que se utilicen en el puerto serie del nodo a configurar para que la comunicación sea efectiva.
- Codificar los datos de configuración de los nodos. Los parámetros de configuración deben ser incluidos dentro de una cadena de caracteres antes de enviarlos a través del puerto serie. Debido a que la longitud de los parámetros puede variar entre 1 y 2 caracteres, delante de cada parámetro ha de añadirse un carácter extra indicando la longitud del mismo para así evitar errores durante la extracción en destino. En la figura 4.4. se muestra un ejemplo de datos de configuración listos para ser enviados.



**Figura 4.4. Ejemplo de datos de configuración codificados.**

- Enviar los datos a través del puerto serie. El envío de datos se realiza mediante la función *ComWrt* (*int PortNumber*, *char Buffer[]*, *int Count*), en la que *PortNumber* indica el puerto que ha sido configurado previamente, *Buffer* es la cadena de caracteres que contiene la información a enviar, y *Count* indica el número máximo de bytes que se van a escribir en el puerto seleccionado.
- Cerrar el puerto de comunicaciones una vez se ha completado el envío. Esta tarea se realiza mediante la función *ComClose* (*int PortNumber*), en la que *PortNumber* indica el puerto que se desea cerrar.

#### 4.3.2.2. Modificación de la pila de protocolos TIMAC

Una vez concluido el desarrollo de la interfaz gráfica de usuario solo falta adaptar el código de la aplicación ejemplo basada en la pila de protocolos TIMAC para la recepción de los datos a través del puerto serie y la posterior configuración de los parámetros. A continuación se detallan las funciones programadas para tal efecto:

##### *void config\_com (void)*

Función destinada a habilitar el puerto de comunicaciones de la placa SmartRF05EB para la recepción de los parámetros de configuración.

##### Entradas:

Ninguna.

##### Salidas:

Ninguna.

En esta función se seleccionan los parámetros por los que se va a regir la comunicación serie y se almacenan en una estructura de datos del tipo *halUARTCfg\_t*, definida en TIMAC. En la siguiente tabla se describen los parámetros para la configuración del puerto de comunicaciones:



Parámetro	Valor	Descripción
<i>baudrate</i>	38400 baudios	Velocidad de comunicación
<i>flowcontrol</i>	FALSE	Control de flujo desactivado
<i>rx.MaxBufSize</i>	512 bytes	Tamaño máximo del <i>buffer</i> RX
<i>tx.MaxBufSize</i>	512 bytes	Tamaño máximo del <i>buffer</i> TX
<i>intEnable</i>	TRUE	Interrupciones de UART habilitadas
<i>callBackFunc</i>	MT_UartProcess_Rx_Data	Nombre de la función <i>callback</i> para el tratamiento de las interrupciones generadas por la recepción de datos a través del puerto serie.

**Tabla 4.8. Parámetros de configuración del puerto serie en la placa SmartRF05EB**

**void MT UartProcessRxData (uint8 port, uint8 event)**

Se trata de una función *callback* que se encarga de procesar los datos recibidos por la UART y la posterior configuración de los parámetros de la aplicación.

Entradas:

*port*: indica el puerto de la UART en el que se ha recibido la información.

*event*: indica el evento generado por la interrupción en la UART. En el caso de que la interrupción no genere ninguno de los eventos que aparecen en la siguiente tabla, este parámetro aparecerá vacío.

Evento	Descripción
HAL_UART_RX_FULL	Buffer de recepción de la UART lleno
HAL_UART_RX_ABOUT_FULL	Buffer de recepción de la UART casi lleno
HAL_UART_RX_TIMEOUT	Tiempo máximo de recepción inactiva agotado
HAL_UART_TX_FULL	Buffer de transmisión de la UART lleno
HAL_UART_TX_EMPTY	Buffer de transmisión de la UART vacío

**Tabla 4.9. Eventos asociados a la UART de la placa SmartRF05EB**

Salidas:

Ninguna.

Esta función se ejecuta tras producirse una interrupción por recepción de datos en la UART, llevando a cabo las siguientes tareas:

- Lectura del *buffer* de recepción de datos: la lectura de este *buffer* se realiza mediante la función *uint16 HalUARTRead (uint8 port, uint8 \*buf, uint16 len)*, en la que *port* indica el puerto de la UART en el que se han recibido los datos, *buf* es la cadena de caracteres en la que se va a almacenar la información recibida, y *len* indica el número máximo de *bytes* a copiar del *buffer* de recepción.
- Decodificar los datos recibidos para obtener los parámetros de configuración del sistema. Para ello, se tiene en cuenta el orden en el que fueron codificados y, además, que cada parámetro va precedido de un número que indica cuantos caracteres ocupa dentro de la cadena en la que está almacenado. Una vez decodificados, los parámetros se guardan en una serie de variables que posteriormente son consultadas para la inicialización del sistema.

El último paso, tras diseñar las funciones que regulan el comportamiento de la UART, consiste en modificar la función de inicialización de la aplicación. Para ello se reutiliza el código que previamente se modificó para la configuración de los nodos mediante el uso de las placas de desarrollo, añadiendo las siguientes funcionalidades:

- Posibilidad de selección del método de configuración del nodo.
- Preparar el sistema para la recepción de datos en caso de seleccionarse la configuración a través de la aplicación externa. Para ello se inicializa el puerto de comunicaciones de la placa de desarrollo y se paraliza el flujo hasta que se registra una interrupción por recepción de datos en la UART.

En el siguiente cuadro se muestra la función *MSA\_Init* modificada y comentada:

```
void MSA_Init (uint8 taskId)
{
    uint8 i, conf_type, ch, txp, dev;

    /* Selección del método de configuración */
    conf_type=set_conf_type();
```

```

/* CONFIGURACIÓN EN PLACAS */

if (conf_type==0)
{
    /* Inicialización de los parámetros de configuración de la red */
    dev = set_device_type (); // Selección del tipo de dispositivo a configurar
    ch = set_tx_ch (dev); // Selección del canal de transmisión
    MAC_RADIO_SET_CHANNEL (ch); // Establece canal de transmisión seleccionado
    txp = set_tx_power (dev); // Selección de la potencia de transmisión
    MAC_RADIO_SET_TX_POWER (txp); // Establece potencia de transmisión seleccionada
    MSA_MAC_BEACON_ORDER = set_BO (dev); // Selección del parámetro Beacon Order
    // Selección del parámetro Superframe Order
    if (MSA_MAC_BEACON_ORDER<15)
    {
        MSA_MAC_SUPERFRAME_ORDER = set_SO (dev,MSA_MAC_BEACON_ORDER);
    }

    else if (MSA_MAC_BEACON_ORDER==15)
    {
        MSA_MAC_SUPERFRAME_ORDER=15;
    };
    //Establece los parámetros Beacon Order y Superframe Order
    msa_BeaconOrder = MSA_MAC_BEACON_ORDER;
    msa_SuperFrameOrder = MSA_MAC_SUPERFRAME_ORDER;
};

/* CONFIGURACIÓN VIA PUERTO COM */

if (conf_type==1)
{
    HalLcdWriteString("CONFIG VIA COM",HAL_LCD_LINE_1);
    HalLcdWriteString("                ",HAL_LCD_LINE_2);
    HalLcdWriteString("Waiting for Data... ",HAL_LCD_LINE_3);

    // Configura y abre el puerto COM que se va a utilizar para la comunicación serie
    config_com();

    // Paraliza la ejecución de la función MSA_Init hasta que se detecta una interrupción
    generada por la UART. Para ello se consulta el estado del flag URX0IF

    while(URX0IF==0){};

    /* Los parámetros que condicionan el funcionamiento de la red se configuran en la función
    callback MT_UartRxProcessData que se ejecuta inmediatamente después de detectar la
    interrupción generada por la UART */
};

```

```
/* Inicialización de la identificación de tarea */
MSA_TaskId = taskId;

/* Inicialización de las características generales de la capa MAC */
MAC_InitDevice();
MAC_InitCoord();

/* Inicialización de las características específicas de la capa MAC que permiten la creación
de redes balizadas */
MAC_InitBeaconDevice();
MAC_InitBeaconCoord();

/* Reseteo de la capa MAC*/
MAC_MlmeResetReq(TRUE);

/* Inicialización del paquete de datos */
for (i=MSA_HEADER_LENGTH; i<MSA_PACKET_LENGTH; i++)
{
    msa_Data1[i] = i-MSA_HEADER_LENGTH;
}

/* Inicialización del paquete de eco*/
for (i=0; i<MSA_ECHO_LENGTH; i++)
{
    msa_Data2[i] = 0xEE;
}
}
```



# CAPÍTULO 5

## Fase de Pruebas

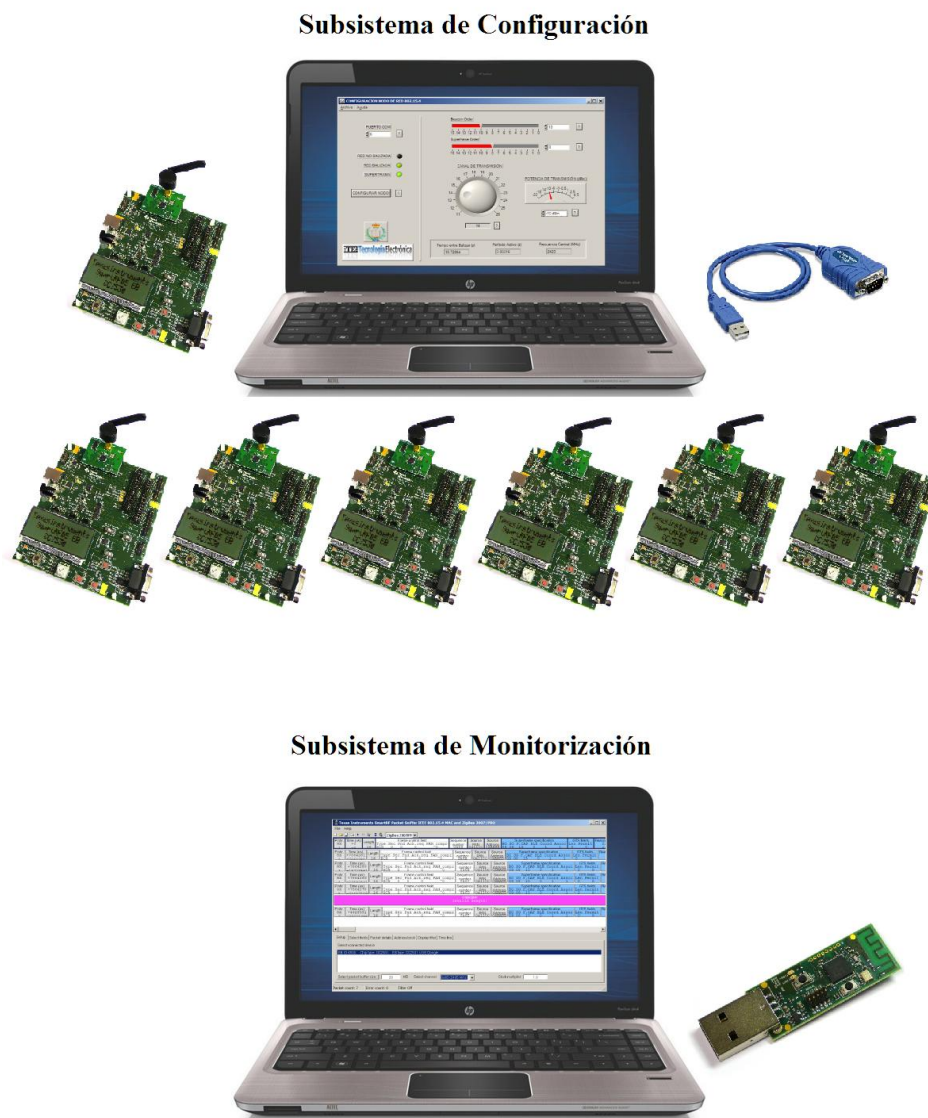
El objetivo de este capítulo es plasmar los resultados obtenidos durante la fase de evaluación del sistema, comprobando su correcto funcionamiento tanto a nivel *hardware* como *software*.

Los resultados positivos obtenidos posibilitan el desarrollo de un banco de pruebas para la evaluación del funcionamiento de redes balizadas sencillas, en las que la configuración de los nodos se realiza de una forma rápida y eficaz mediante las aplicaciones descritas en el capítulo anterior.

Las pruebas realizadas se pueden dividir en tres grandes grupos que atienden a los siguientes criterios:

- Pruebas de funcionamiento de la aplicación para la configuración de los nodos mediante las placas de desarrollo SmartRF05EB. El objetivo de estas pruebas es comprobar que el conjunto *hardware-software* responde adecuadamente durante la fase de configuración de los nodos.
- Pruebas de funcionamiento de la aplicación para la configuración remota de los nodos a través del puerto serie.
- Pruebas para la evaluación de redes balizadas. Basándose en el uso de las aplicaciones diseñadas para la configuración de los nodos, se llevan a cabo una serie de pruebas para evaluar aspectos básicos de funcionamiento de las redes balizadas, como puede ser la pérdida de sincronización de los nodos, los tiempos de formación de redes o los errores en el proceso de asociación a una red.

En la siguiente figura se muestra un esquema del sistema de pruebas empleado, que puede ser dividido en dos subsistemas: el de configuración y el de monitorización.



**Figura 5.1. Sistema de pruebas.**

El subsistema de configuración tiene como objetivo facilitar la labor de configuración de los nodos durante el proceso de formación de redes. Además, permite comprobar el correcto funcionamiento tanto de la aplicación de configuración remota como de la aplicación de configuración mediante las placas SmartRF05EB.

Por su parte, el subsistema de monitorización permite capturar el tráfico de información que se produce en las redes, ayudando a identificar los parámetros que las definen o los diferentes problemas que pueden surgir en los procesos de creación de una nueva red o asociación a una existente.

En las tablas 5.1 y 5.2 se describen las características más relevantes de los equipos que integran cada uno de los subsistemas.

EQUIPO	CARACTERÍSTICAS
<b>PC Portátil</b>	<p>PC basado en una arquitectura <i>Intel Core i5 M430</i> a 2.27 GHz, con 4 GB de memoria RAM y sistema operativo <i>Windows 7 Home Premium</i> de 64 bits.</p> <p>Software instalado:</p> <ul style="list-style-type: none"> <li>○ <i>LabWindows/CVI</i>.</li> <li>○ <i>IAR Embedded Workbench for 8051 v7.60</i>.</li> <li>○ <i>CC2530 Config</i>. (Aplicación para la configuración remota).</li> <li>○ <i>VMware</i>.</li> </ul>
<b>Cable conversor USB-RS232</b>	Cable conversor de protocolos y adaptador de interfaz para suplir la ausencia de puerto RS232 en los equipos portátiles. Permite conectar las placas de desarrollo al equipo portátil para llevar a cabo la configuración de los nodos.
<b>Placas de desarrollo SmartRF05EB v1.7</b>	Realizan la función de placa base, proporcionando alimentación y comunicación bidireccional a los módulos CC2530. Cuenta con las interfaces necesarias para poder llevar a cabo la configuración de los nodos.
<b>Módulos de evaluación CC2530EM</b>	Contienen el chip CC2530F256, los circuitos integrados de RF, los componentes externos y los filtros necesarios para conseguir su correcto funcionamiento. Han sido precargados con la pila de protocolos TIMAC y la aplicación <i>msa</i> modificada tal y como se describe en el capítulo 3 del proyecto.

**Tabla 5.1. Características de los equipos que integran el subsistema de configuración.**

EQUIPO	CARACTERÍSTICAS
<b>PC Portátil</b>	<p>PC basado en una arquitectura <i>Intel Core i3 M380</i> a 2.53 GHz, con 4 GB de memoria RAM y sistema operativo <i>Windows 7 Home Premium</i> de 64 bits.</p> <p>Software instalado:</p> <ul style="list-style-type: none"> <li>○ <i>Packet Sniffer</i> de <i>Texas Instruments</i>.</li> <li>○ <i>VMware</i>.</li> </ul>
<b>Módulo CC2531 USB Dongle</b>	Se trata de un dispositivo <i>ZigBee</i> completamente operacional con interfaz USB que puede ser conectado a un PC. Cuenta con un chip CC2531F256, que viene pre-programado para ser usado como <i>packet sniffer</i> .

**Tabla 5.2. Características de los equipos que integran el subsistema de monitorización.**



## 5.1. APLICACIÓN SMARTRF05EB

La fase de pruebas de la aplicación para la configuración de los nodos mediante la placa de desarrollo SmartRF05EB comienza con el test de los periféricos presentes en la misma que son usados en este proyecto.

### Botones

Los botones presentes en la placa pueden tener distinta funcionalidad dependiendo de la fase en la que se encuentre la ejecución de la aplicación *msa*. En la siguiente tabla se muestra el resultado de los test de funcionamiento de los botones, comprobando que su pulsación tiene el efecto deseado en las situaciones requeridas.

BOTÓN	FUNCIÓN	RESULTADO DEL TEST
<b>EM_RESET</b>	Reseteo del módulo de evaluación.	CORRECTO
<b>S1</b>	Aceptación de los valores de configuración (durante la fase de configuración de los nodos).	CORRECTO
<b>Joystick Derecha</b>	Visualización de la siguiente opción del menú de configuración (durante la fase de configuración de los nodos).	CORRECTO
	Inicio o finalización de envío de mensajes (una vez que la red ha sido creada).	CORRECTO
<b>Joystick Izquierda</b>	Visualización de la opción anterior del menú de configuración (durante la fase de configuración de los nodos).	CORRECTO
<b>Joystick Arriba</b>	Crear nueva red o unirse a una ya existente.	CORRECTO
<b>Joystick, S1 y S2 (cuando se espera pulsar otros interfaces)</b>	Sin efecto.	CORRECTO

Tabla 5.3. Test de funcionamiento de los botones de la placa SmartRF05EB.

### Alimentación de la placa

Mediante el siguiente test (cuyos resultados se recogen en la tabla 5.4) se trata de comprobar que los nodos funcionan correctamente independientemente del modo de alimentación usado.

MODO DE ALIMENTACIÓN	RESULTADO DEL TEST
USB	CORRECTO
Baterías	CORRECTO
Fuente de alimentación externa	CORRECTO
Fuente de alimentación de laboratorio	CORRECTO

Tabla 5.4. Test de los distintos modos de alimentación de la placa SmartRF05EB.

### Diodos LED

En la siguiente tabla se muestran los resultados de los test realizados para comprobar la funcionalidad de los diodos LED en las distintas situaciones de uso. Cada diodo puede presentar un estado distinto dependiendo de la fase de ejecución en la que se encuentre la aplicación.

LED	FUNCIÓN	RESULTADO DEL TEST
LOW BATT	Encendido cuando se detecta batería baja.	CORRECTO
USB LED	Indica que el sistema está encendido.	CORRECTO
LED 1	Parpadeo corto antes de unirse a una red o crear una nueva.	CORRECTO
	Parpadeo largo tras unirse a una red existente (nodo final).	CORRECTO
	Encendido tras crear una nueva red (coordinador de red).	CORRECTO
LED 2	Encendido durante fase de configuración SmartRF05EB.	CORRECTO
LED 3	Encendido durante fase de configuración vía puerto serie.	CORRECTO

Tabla 5.5. Test de los diodos LED.

<b>LED 3</b>	Cambia de estado al recibir una nueva trama de datos.	CORRECTO
<b>LED 4</b>	Encendido durante la pulsación del botón S1.	CORRECTO

**Tabla 5.5. (Continuación) Test de los diodos LED.**

### Pantalla LCD

Las pruebas realizadas con operaciones registradas en la pantalla LCD están orientadas a comprobar que ésta muestra los mensajes requeridos dependiendo de la fase en la que se encuentra la ejecución de la aplicación. En la tabla 5.6 se recogen los resultados obtenidos durante la fase de configuración de los nodos, en la que se muestran los distintos menús para la selección de los parámetros de configuración. La tabla 5.7 hace referencia a los mensajes mostrados durante la fase de ejecución una vez que se han seleccionado los parámetros de configuración.

MENÚ	POSIBLES VALORES	RESULTADO DEL TEST
<b>CONF_METHOD_SEL</b>	SMARTRF05EB > < SERIAL PORT	CORRECTO
<b>DEV_TYPE_SEL</b>	COORDINATOR > < END DEVICE	CORRECTO
<b>SET_TX_CHANNEL</b>	Ch11 2405 MHz > < Ch12 2410 MHz > ... Ch26 2480 MHz	CORRECTO
<b>SET_TX_POWER</b>	-22 dBm > < -18 dBm > ... 4.5 dBm	CORRECTO
<b>SET_BEACON_ORDER</b>	0 >, < 1 >, < 2 >, ... < 15	CORRECTO
<b>SET_SFRAME_ORDER</b>	0 >, < 1 >, < 2 >, ... < 15	CORRECTO

**Tabla 5.6. Test LCD (menús de configuración).**

MENSAJE	DESCRIPCIÓN	RESULTADO DEL TEST
<i>TexasInstruments SmartRF05 EB (No EM detected)</i>	Mensaje de error si no hay módulo de evaluación conectado o éste no funciona correctamente.	CORRECTO
<i>CONFIG_VIA_COM Waiting for Data</i>	Mensaje que se muestra durante la espera de datos de la aplicación externa.	CORRECTO
<i>Create_NWK: Push ^</i>	Instrucciones: Para crear una nueva red pulsar <i>Joystick Up</i> .	CORRECTO
<i>Join NWK: Push ^</i>	Instrucciones: Para unirse a una red existente pulsar <i>Joystick Up</i> .	CORRECTO
<i>COORDINATOR</i>	El nodo se ha establecido como coordinador de red.	CORRECTO
<i>END DEVICE</i>	El nodo se ha establecido como dispositivo final.	CORRECTO
<i>ERROR: SO&gt;BO CONFIG SO=BO Push S1 &gt;&gt;</i>	Error debido a que SO es mayor que BO, se procede a igualar el valor de SO con el de BO. Instrucciones: Pulsar el botón S1 para pasar a la siguiente pantalla.	CORRECTO
<i>Scanning Channel</i>	Se está escaneando el canal seleccionado para crear una red o unirse a una ya existente.	CORRECTO
<i>Sending Beacons</i>	El coordinador está transmitiendo balizas.	CORRECTO
<i>Non Beacons N.</i>	La red creada es No Balizada.	CORRECTO
<i>&gt; Start/Stop Tx</i>	Instrucciones: Para iniciar o finalizar la transmisión de mensajes pulsar <i>Joystick Right</i> .	CORRECTO
<i>Sending Messages</i>	Se están transmitiendo mensajes	CORRECTO

Tabla 5.7. Test LCD (Mensajes en ejecución).

Otro aspecto a tener en cuenta durante la fase de configuración de los nodos es la asignación de valores de cada uno de los parámetros, debiendo éstos mantenerse dentro de los rangos establecidos por la norma IEEE 802.15.4 para garantizar el correcto funcionamiento de las redes. En la tabla 5.8 se recogen los resultados obtenidos durante la comprobación de los valores de configuración.

PARÁMETRO	RANGO	LA APLICACIÓN DE CONFIGURACIÓN PERMITE VALORES FUERA DE RANGO
<b>BEACON ORDER</b>	0-15	NO
<b>SUPERFRAME ORDER</b>	0-15	NO. Además se garantiza que SO $\leq$ BO
<b>TX CHANNEL</b>	11-26	NO
<b>TX POWER</b>	-22 dBm ... 4.5 dBm	NO

**Tabla 5.8. Comprobación de los rangos de valores asignados a los parámetros.**

Para finalizar la fase de pruebas de la aplicación de configuración de nodos mediante la placa de desarrollo SmartRF05EB, se han realizado pruebas de “caja blanca” con el objetivo de detectar errores derivados de la etapa de programación de la aplicación.

Estas pruebas tratan de comprobar que se ejecutan todas y cada una de las líneas de código de la aplicación al menos una vez, generando distintos flujos (o caminos) derivados de los distintos puntos de toma de decisión. Además, se ha tenido en cuenta la ejecución de bucles, prestando especial atención a las condiciones que los resuelven.

Para realizar estas comprobaciones se han dispuesto una serie de puntos de control a lo largo del código de la aplicación, concretamente en el inicio y fin de segmentos de código de ejecución secuencial, en los puntos de toma de decisión y en los bucles. El paso del flujo de ejecución por cada punto de control muestra en la pantalla LCD de la placa de desarrollo un mensaje que indica el estado de ejecución de la aplicación. En el caso de tratarse de un punto de toma de decisión, se muestra el valor del parámetro que ha condicionado dicha decisión. Finalmente, si se trata de un bucle, se muestra por pantalla el valor del parámetro que ha posibilitado su resolución.

En la figura 5.2 se muestra el diagrama de flujo de la aplicación derivado de los resultados obtenidos durante las pruebas de caja blanca.

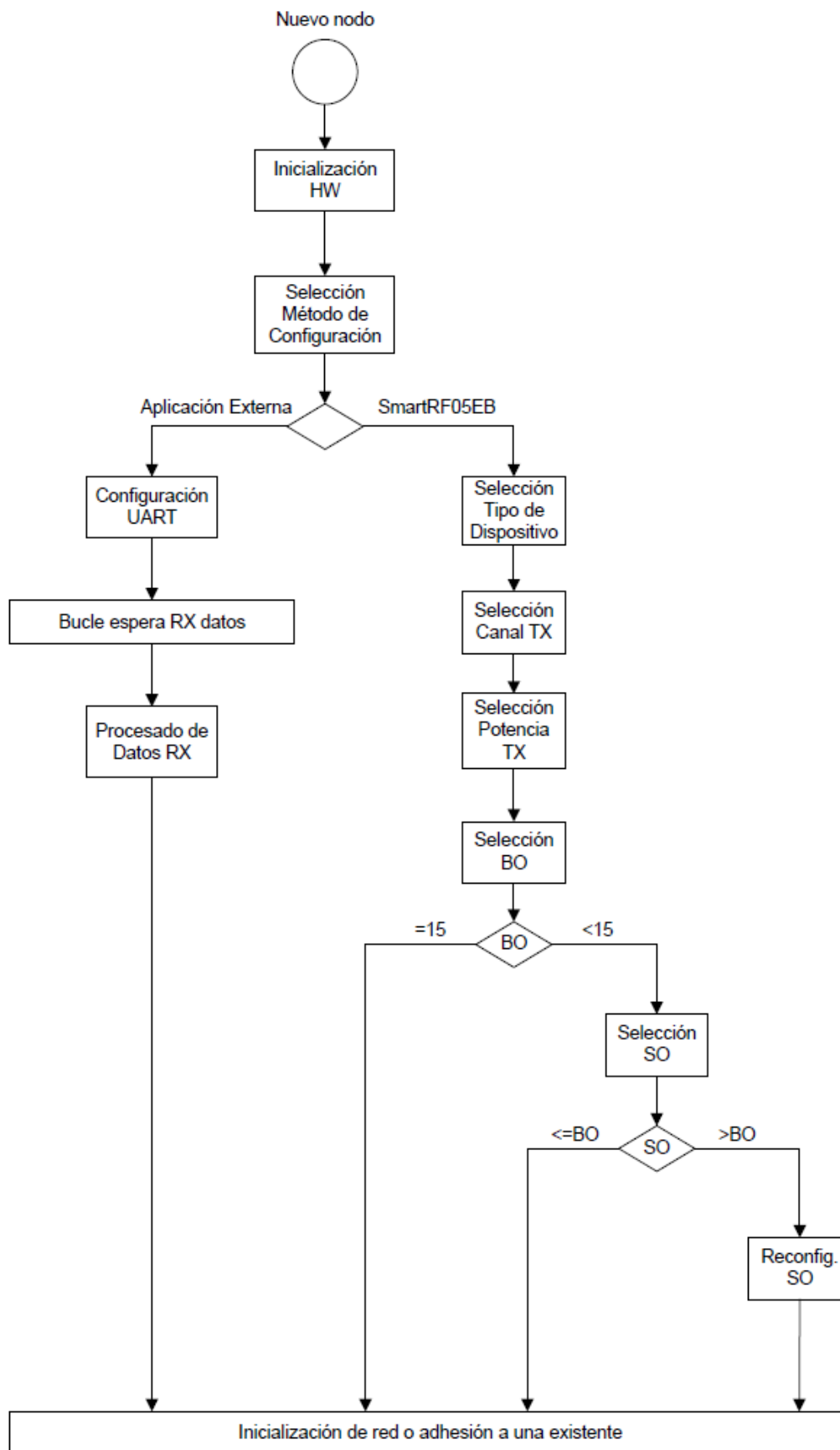


Figura 5.2. Diagrama de flujo para las pruebas de “caja blanca”.

## 5.2. APLICACIÓN DE CONFIGURACIÓN REMOTA

Las pruebas para evaluar el funcionamiento de la aplicación remota de configuración de nodos CC2530 a través del puerto serie se han realizado en un PC portátil basado en una arquitectura *Intel Core i5 M430* a 2.27 GHz, con 4 GB de memoria RAM y sistema operativo *Windows 7 Home Premium* de 64 bits.

La primera prueba tiene como objetivo comprobar la compatibilidad de la aplicación con diversos sistemas operativos. Para ello se ha generado un archivo de instalación de *Windows* que incluye todas las librerías de *LabWindows/CVI* necesarias para garantizar el correcto funcionamiento de la aplicación. El test se ha realizado sobre los siguientes sistemas operativos virtualizados en el PC anteriormente mencionado utilizando el *software VMware Workstation*.

SISTEMA OPERATIVO	INSTALACIÓN DE LA APLICACIÓN	CONECTIVIDAD CON LOS NODOS
<i>Windows XP Professional 32 bits</i>	CORRECTO	CORRECTO
<i>Windows Vista Home Premium 64 bits</i>	CORRECTO	CORRECTO
<i>Windows 2008 Server R2 64 bits</i>	CORRECTO	CORRECTO
<i>Windows 7 Home Premium 32 bits</i>	CORRECTO	CORRECTO

**Tabla 5.9. Test de compatibilidad con diversos sistemas operativos.**

El siguiente conjunto de pruebas, cuyos resultados se recogen en la tabla 5.10, está destinado a comprobar el funcionamiento de los controles ubicados en la interfaz gráfica de la aplicación. Para ello se ha comprobado que todos y cada uno de ellos cumplen con las normas que minimizan la posibilidad de configurar los nodos de manera errónea, garantizando así que los parámetros de configuración se ajustan a las especificaciones impuestas por el estándar IEEE 802.15.4.

CONTROL	PRUEBAS REALIZADAS	RESULTADO
<b>Beacon Order (Ring Slide y Numeric)</b>	La modificación de cualquiera de los controles ( <i>Ring Slide</i> o <i>Numeric</i> ) asociados al parámetro <i>Beacon Order</i> actualiza el control complementario.	CORRECTO
	El parámetro BO no puede ser modificado con valores que estén fuera del rango indicado por la norma IEEE 802.15.4 [0-15].	CORRECTO
	Si el parámetro <i>Beacon Order</i> toma el valor 15 se actualiza el valor de los controles asociados al parámetro <i>Superframe Order</i> con el mismo valor.	CORRECTO
	Si el parámetro BO se modifica con un valor menor que el de SO, éste se iguala al valor del primero.	CORRECTO
	Si el parámetro BO se modifica con un valor mayor que el de SO (y menor que 15), éste permanece inalterado.	CORRECTO
<b>Superframe Order (Ring Slide y Numeric)</b>	La modificación de cualquiera de los controles ( <i>Ring Slide</i> o <i>Numeric</i> ) asociados al parámetro <i>Superframe Order</i> actualiza el control complementario.	CORRECTO
	El parámetro SO no puede ser modificado con valores que estén fuera del rango indicado por la norma IEEE 802.15.4 [0-15].	CORRECTO
	Es posible modificar el valor del parámetro SO con un valor menor que el de BO siempre que éste último sea menor que 15.	CORRECTO
	El parámetro SO no puede ser modificado con un valor mayor que el de BO. En tal caso, se iguala al valor de éste último.	CORRECTO
<b>Canal de Transmisión (Ring Knob y String)</b>	La modificación del control de tipo <i>Ring Knob</i> se refleja en el control de tipo <i>string</i> (no editable)	CORRECTO
	El parámetro Canal de Transmisión no puede ser modificado con valores que estén fuera del rango indicado por la norma IEEE 802.15.4 (canales del 11 al 26).	CORRECTO

Tabla 5.10. Test de funcionamiento de los controles de la aplicación.



<b>Potencia de Transmisión (Ring Meter y Numeric)</b>	El parámetro Potencia de Transmisión no puede ser modificado con un valor fuera del rango establecido en las especificaciones de diseño del módulo CC2530EM (-22 dBm a 4.5 dBm).	CORRECTO
<b>“?” (Command Buttons)</b>	Tras la pulsación sobre cada botón etiquetado como “?” se muestra un mensaje <i>pop-up</i> con la información relativa al control situado a su lado.	CORRECTO
<b>RED NO BALIZADA (LED)</b>	El diodo permanece encendido (color rojo) mientras el valor del control <i>Beacon Order</i> es igual a 15, y apagado cuando es menor que 15.	CORRECTO
<b>RED BALIZADA (LED)</b>	El diodo permanece encendido (color verde) mientras el valor del control <i>Beacon Order</i> es menor que 15, y apagado cuando es igual.	CORRECTO
<b>PUERTO COM (Numeric)</b>	Permite seleccionar el puerto COM del PC para el envío de los datos de configuración.	CORRECTO
<b>CONFIGURAR (Command Button)</b>	Los datos de configuración de los nodos se codifican correctamente antes de ser enviados.	CORRECTO
	El envío de datos de configuración de los nodos se realiza correctamente. Si el puerto de comunicaciones seleccionado no está disponible se muestra un mensaje <i>pop-up</i> indicando el motivo del error (ver figura 5.3).	CORRECTO
<b>Tiempo entre Balizas (String)</b>	Tras modificar el valor del control <i>Beacon Order</i> , muestra el valor del parámetro <i>Beacon Interval</i> (BI) de acuerdo a la ecuación [2.3].	CORRECTO
<b>Período Activo (String)</b>	Tras modificar el valor del control <i>Superframe Order</i> , muestra el valor del parámetro <i>Superframe Duration</i> de acuerdo a la ecuación [2.4].	CORRECTO
<b>Frecuencia Central (String)</b>	Tras modificar el valor del control Canal de Transmisión, muestra correctamente el valor de la frecuencia central del nuevo canal seleccionado de acuerdo a la ecuación [2.2].	CORRECTO

Tabla 5.10. (Continuación) Test de funcionamiento de los controles de la aplicación.

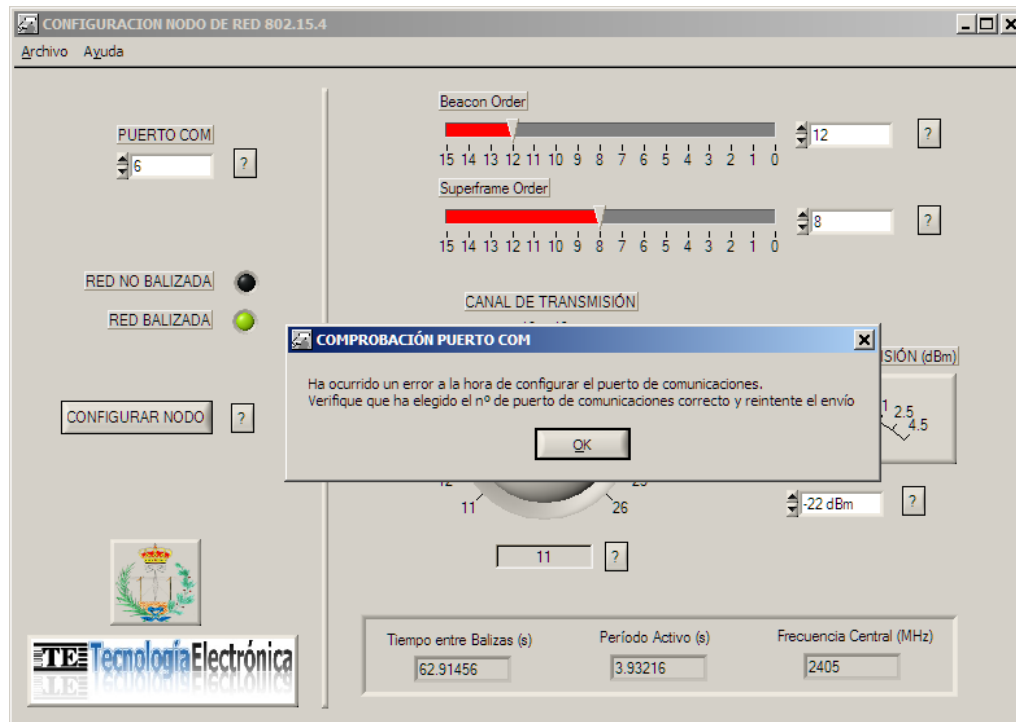


Figura 5.3. Mensaje de error al producirse fallo de envío de datos.

Para finalizar la fase de pruebas de la aplicación de configuración remota de nodos CC2530, se han realizado pruebas de “caja blanca” con el objetivo de detectar errores derivados de la etapa de programación de la aplicación.

A diferencia de la aplicación de configuración mediante el uso de las placas de desarrollo, en la que el código de programación es esencialmente secuencial, esta aplicación está basada en la ejecución de funciones *callback*, que se ejecutan siempre y cuando se actúe sobre el control al que están asociadas. Al igual que en el caso anterior, se ha querido comprobar que todas las líneas de código se ejecutan al menos una vez en un caso en el que se actúa sobre todos los controles de la aplicación.

Para realizar estas comprobaciones se han dispuesto una serie de puntos de control a lo largo del código de cada una de las funciones *callback*, concretamente en el inicio y fin de segmentos de código de ejecución secuencial y en los puntos de toma de decisión (los puntos de control no figuran, obviamente, en la aplicación final de usuario). El paso del flujo de ejecución por cada punto de control muestra un mensaje *pop-up* que indica el estado de ejecución de la aplicación. En el caso de tratarse de un punto de toma de decisión, se muestra el valor del parámetro que ha condicionado dicha decisión.

## 5.3. FUNCIONAMIENTO DE REDES BALIZADAS

### 5.3.1. FORMACIÓN DE REDES

El siguiente paso para comprobar el correcto funcionamiento de las aplicaciones anteriormente descritas es crear redes que se ajusten a los parámetros de configuración seleccionados en las mismas. Para ello se han planteado una serie de escenarios representativos que cubren la formación de distintos tipos de redes balizadas y permiten evaluar su funcionamiento.

Antes de comenzar con estas pruebas conviene señalar las limitaciones más importantes de la versión de la pila de protocolos TIMAC que se ha usado para el desarrollo de este proyecto:

- La formación de redes está limitada a las siguientes topologías:
  - Topología en estrella: tanto para redes balizadas como para redes no balizadas.
  - Topología *peer-to-peer*: para redes no balizadas.
- El número máximo de nodos está limitado a ocho, de los cuales uno actúa como coordinador de red y siete son dispositivos finales.
- No existe la figura de *router* ya que la profundidad máxima de las redes es de un nivel.
- No está habilitada la funcionalidad de *slots* temporales garantizados (GTS) para la transmisión de información sin necesidad de recurrir a mecanismos de contienda para el acceso al canal.
- La aplicación ejemplo *msa*, en la que se basa este proyecto (ya que es la única que se proporciona junto a la pila de protocolos), realiza una asignación fija de los parámetros de identificación tanto de la red de área personal (PAN) como del coordinador de dicha red, pudiendo causar problemas durante el proceso de creación de redes. Las direcciones cortas de identificación son las siguientes:
  - PAN\_ID: 0x11CC
  - COORD\_SHORT\_ADDR: 0xAABB

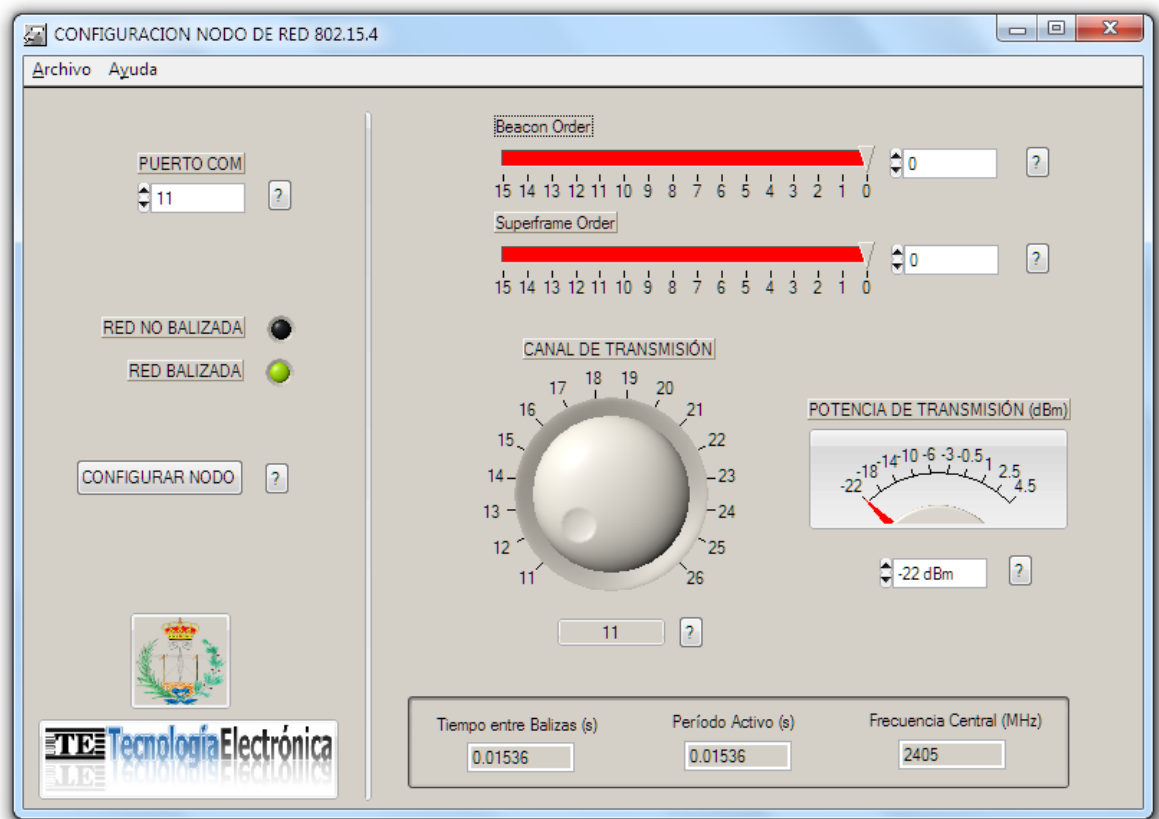
**Escenario 1: Red balizada de dos nodos. BO=0 y SO=0**

El primer escenario trata de recrear la red balizada de topología en estrella más sencilla, formada por dos nodos, un coordinador de red y un dispositivo final, en la que el tiempo entre balizas es el mínimo posible y el período activo es igual al tiempo entre balizas.

Los parámetros de configuración tanto para el coordinador de red como para el dispositivo final son los siguientes:

- *Beacon Order*: BO = 0
- *Superframe Order*: SO = 0
- Canal de transmisión: 11
- Potencia de transmisión: -22 dBm

Para comprobar el funcionamiento de las dos aplicaciones de configuración disponibles, la configuración del coordinador de red se ha realizado con la aplicación de configuración remota (figura 5.4), y la del dispositivo final con la aplicación de configuración SmartRF05EB.



**Figura 5.4. Configuración del coordinador con parámetros del Escenario 1.**

En la figura 5.5 se puede ver una captura de tramas realizada con el dispositivo CC2531 USB Dongle y el software Packet Sniffer sintonizado en el mismo canal que los nodos. Estas tramas muestran el momento en el que el coordinador de red comienza a transmitir balizas, el proceso de asociación del dispositivo final, la transmisión de mensajes de coordinador a dispositivo final, y la transmisión de mensajes de dispositivo final a coordinador.

P.nbr. RX 1	Time (us) +0	Length 16	Frame control field Type Sec Fnd Ack.req PAN_compr BCN 0 0 0 0	Sequence number 0xB5	Source PAN 0x11CC	Source Address 0xAABB	Superframe specification BO SO F.CAP BLE Coord Assoc 00 00 15 0 1 1	GTS fields Len Permit 0 0	Beacon payload 22 33 44	RSSI (dBm) -39	FCS OK	
P.nbr. RX 2	Time (us) +15360	Length 16	Frame control field Type Sec Fnd Ack.req PAN_compr BCN 0 0 0 0	Sequence number 0xB6	Source PAN 0x11CC	Source Address 0xAABB	Superframe specification BO SO F.CAP BLE Coord Assoc 00 00 15 0 1 1	GTS fields Len Permit 0 0	Beacon payload 22 33 44	RSSI (dBm) -39	FCS OK	
P.nbr. RX 3	Time (us) +5441	Length 21	Frame control field Type Sec Fnd Ack.req PAN_compr CMD 0 0 1 0	Sequence number 0x4A	Dest. PAN 0x11CC	Dest. Address 0xAABB	Source Address 0x0000F0E0D0C0B0A0	Association request Alt.coord FFD Power Idle.RX Sec Alloc.addr 0 0 0 0 1 0 1			RSSI (dBm) -3	FCS OK
P.nbr. RX 4	Time (us) +1056	Length 5	Frame control field Type Sec Fnd Ack.req PAN_compr ACK 0 0 0 0	Sequence number 0x4A	RSSI (dBm) -39	FCS OK						
P.nbr. RX 5	Time (us) +8863	Length 24	Frame control field Type Sec Fnd Ack.req PAN_compr BCN 0 0 0 0	Sequence number 0xB7	Source PAN 0x11CC	Source Address 0xAABB	Superframe specification BO SO F.CAP BLE Coord Assoc 00 00 15 0 1 1	GTS fields Len Permit 0 0	Beacon payload 22 33 44	RSSI (dBm) -39	FCS OK	
P.nbr. RX 6	Time (us) +5761	Length 18	Frame control field Type Sec Fnd Ack.req PAN_compr CMD 0 0 1 1	Sequence number 0x4B	Dest. PAN 0x11CC	Dest. Address 0xAABB	Source Address 0x0000F0E0D0C0B0A0	Data request			RSSI (dBm) -3	FCS OK
P.nbr. RX 7	Time (us) +978	Length 5	Frame control field Type Sec Fnd Ack.req PAN_compr ACK 0 0 1 0	Sequence number 0x4B	RSSI (dBm) -39	FCS OK						
P.nbr. RX 8	Time (us) +8621	Length 24	Frame control field Type Sec Fnd Ack.req PAN_compr BCN 0 0 0 0	Sequence number 0xB8	Source PAN 0x11CC	Source Address 0xAABB	Superframe specification BO SO F.CAP BLE Coord Assoc 00 00 15 0 1 1	GTS fields Len Permit 0 0	Beacon payload 22 33 44	RSSI (dBm) -39	FCS OK	
P.nbr. RX 9	Time (us) +4160	Length 27	Frame control field Type Sec Fnd Ack.req PAN_compr CMD 0 0 1 1	Sequence number 0xCC	Dest. PAN 0x11CC	Dest. Address 0x0000F0E0D0C0B0A0	Source Address 0x8877665544332211	Short addr Assoc status Short_addr Assoc.status 0x0001 Successful			RSSI (dBm) -39	FCS OK
P.nbr. RX 10	Time (us) +1248	Length 5	Frame control field Type Sec Fnd Ack.req PAN_compr ACK 0 0 0 0	Sequence number 0xCC	RSSI (dBm) -3	FCS OK						
P.nbr. RX 11	Time (us) +9971	Length 16	Frame control field Type Sec Fnd Ack.req PAN_compr BCN 0 0 0 0	Sequence number 0xB9	Source PAN 0x11CC	Source Address 0xAABB	Superframe specification BO SO F.CAP BLE Coord Assoc 00 00 15 0 1 1	GTS fields Len Permit 0 0	Beacon payload 22 33 44	RSSI (dBm) -39	FCS OK	

a)

P.nbr. RX 549	Time (us) +15360	Length 16	Frame control field Type Sec Fnd Ack.req PAN_compr BCN 0 0 0 0	Sequence number 0xCF	Source PAN 0x11CC	Source Address 0xAABB	Superframe specification BO SO F.CAP BLE Coord Assoc 00 00 15 0 1 1	GTS fields Len Permit 0 0	Beacon payload 22 33 44	RSSI (dBm) -38	FCS OK
P.nbr. RX 550	Time (us) +5121	Length 15	Frame control field Type Sec Fnd Ack.req PAN_compr DATA 0 0 1 1	Sequence number 0x4E	Dest. PAN 0x11CC	Dest. Address 0xAABB	Source Address 0x0001	MAC payload 04 AA BB 02	RSSI (dBm) -1	FCS OK	
P.nbr. RX 551	Time (us) +864	Length 5	Frame control field Type Sec Fnd Ack.req PAN_compr ACK 0 0 0 0	Sequence number 0x4E	RSSI (dBm) -38	FCS OK					

b)

P.nbr. RX 593	Time (us) +15360	Length 16	Frame control field Type Sec Fnd Ack.req PAN_compr BCN 0 0 0 0	Sequence number 0xF9	Source PAN 0x11CC	Source Address 0xAABB	Superframe specification BO SO F.CAP BLE Coord Assoc 00 00 15 0 1 1	GTS fields Len Permit 0 0	Beacon payload 22 33 44	RSSI (dBm) -38	FCS OK
P.nbr. RX 594	Time (us) +4800	Length 15	Frame control field Type Sec Fnd Ack.req PAN_compr DATA 0 0 1 1	Sequence number 0xCD	Dest. PAN 0x11CC	Dest. Address 0x0001	Source Address 0xAABB	MAC payload 04 00 01 00	RSSI (dBm) -38	FCS OK	
P.nbr. RX 595	Time (us) +864	Length 5	Frame control field Type Sec Fnd Ack.req PAN_compr ACK 0 0 0 0	Sequence number 0xCD	RSSI (dBm) -1	FCS OK					
P.nbr. RX 596	Time (us) +2977	Length 19	Frame control field Type Sec Fnd Ack.req PAN_compr DATA 0 0 1 1	Sequence number 0x4F	Dest. PAN 0x11CC	Dest. Address 0xAABB	Source Address 0x0001	MAC payload 04 00 01 00 EE EE EE EE	RSSI (dBm) -1	FCS OK	
P.nbr. RX 597	Time (us) +992	Length 5	Frame control field Type Sec Fnd Ack.req PAN_compr ACK 0 0 0 0	Sequence number 0x4F	RSSI (dBm) -38	FCS OK					

c)

Figura 5.5. Captura de tramas de los procesos de: a) Inicialización de red balizada y asociación de dispositivo final, b) Transmisión de mensaje de dispositivo final a coordinador y c) Transmisión de mensaje de coordinador a dispositivo final. (Escenario 1).

En la siguiente figura se muestra un esquema del intercambio de mensajes entre las capas de red y aplicación de ambos dispositivos para que se lleve a cabo correctamente el proceso de asociación. Se puede observar que el intercambio de mensajes entre las capas MAC de ambos dispositivos se corresponde con la captura de tramas de la figura 5.5.(a).

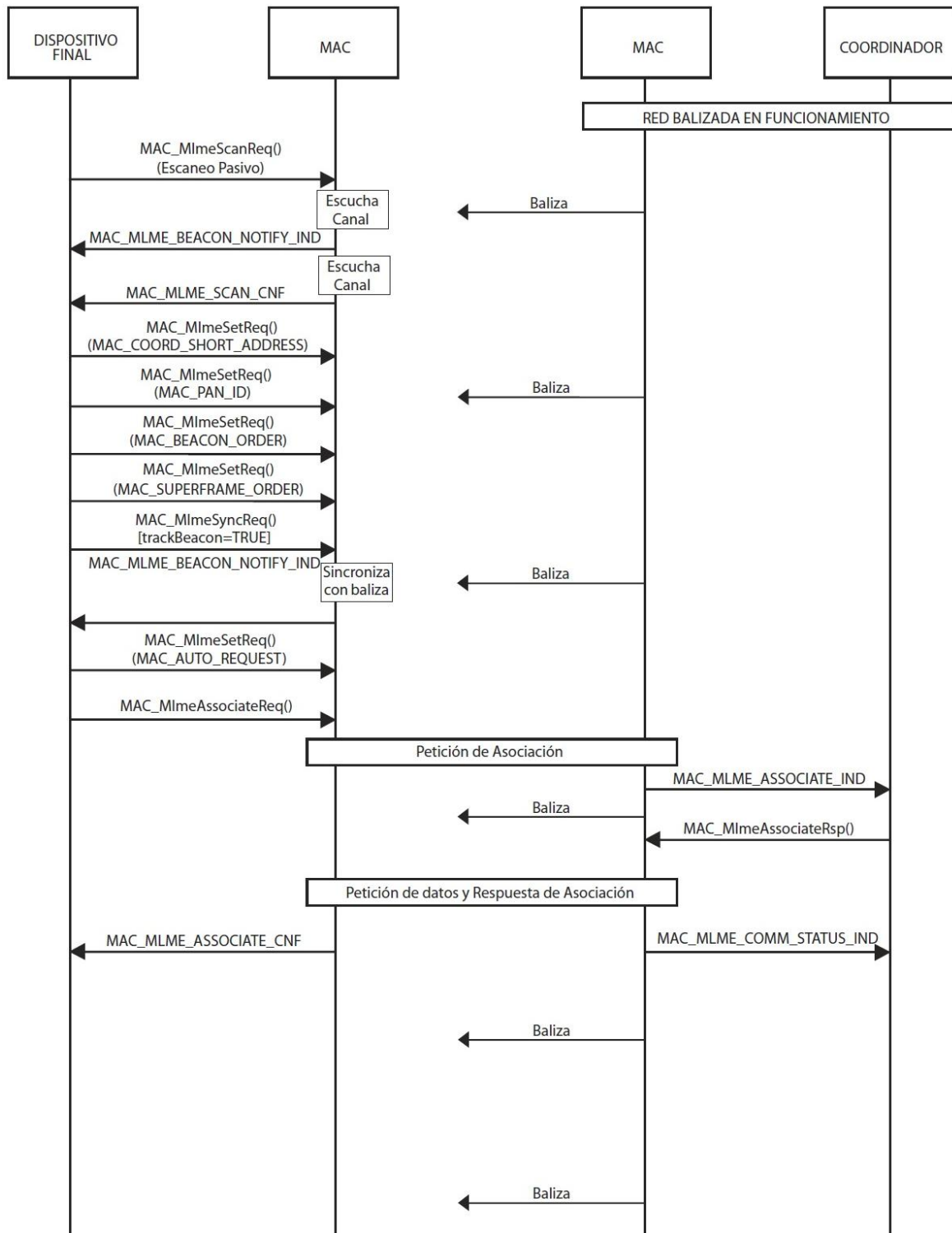
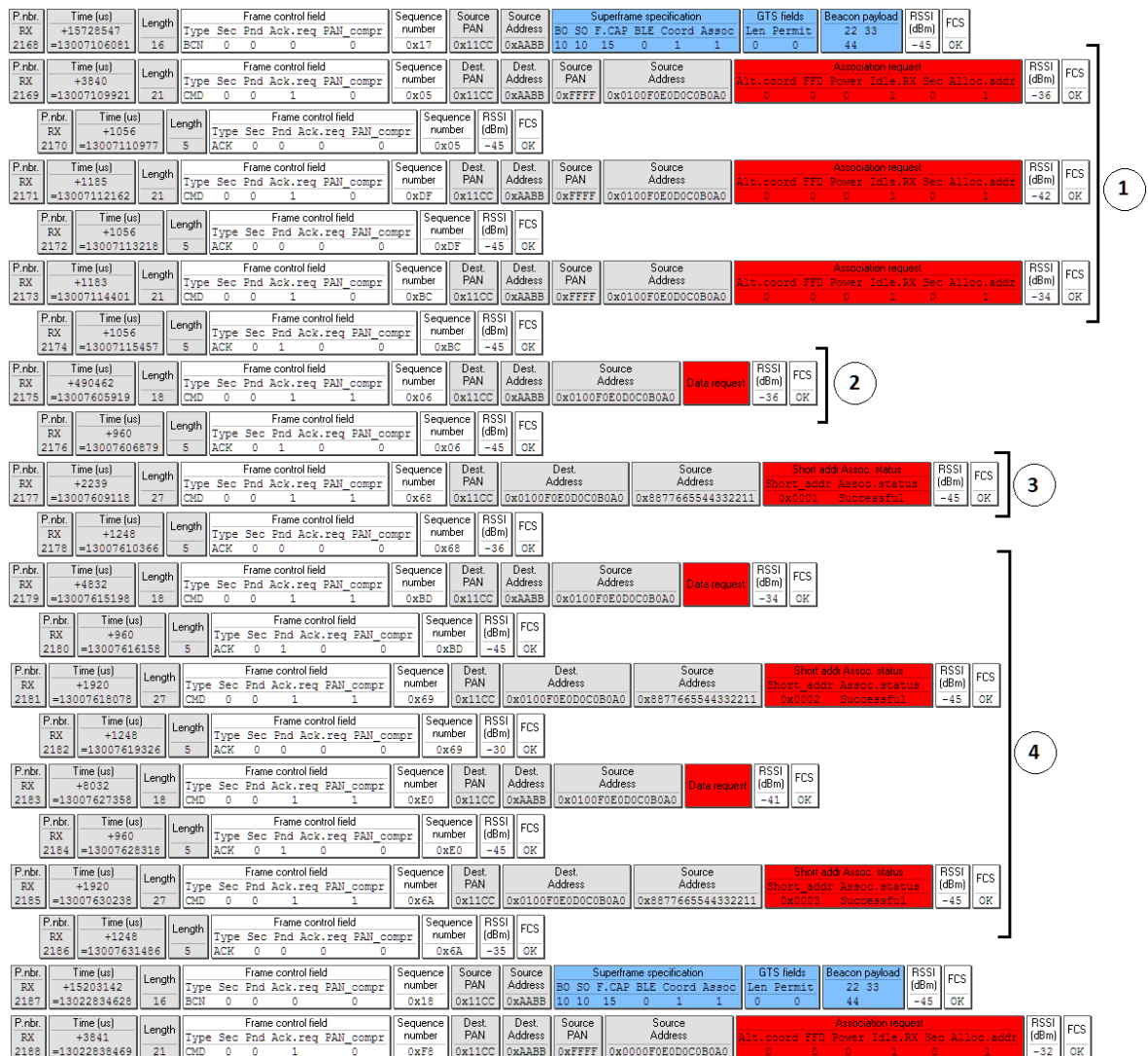


Figura 5.6. Intercambio de mensajes durante el proceso de asociación a una red.

**Escenario 2: Red balizada de 7 nodos. BO=10 y SO=10**

Este escenario trata de ilustrar el problema de la asociación en redes balizadas con más de un dispositivo final. Para ello se ha elegido un tiempo entre balizas y un período activo de aproximadamente 15.72 segundos (de acuerdo con las ecuaciones [2.3] y [2.4]).

Para evaluar el proceso de asociación de los dispositivos finales se ha partido de una red balizada formada únicamente por un coordinador de red que se ajusta a los parámetros anteriores. Tras configurar el resto de nodos con los mismos parámetros (la configuración se ha realizado usando las dos aplicaciones disponibles), éstos inician la petición de asociación a la red prácticamente al mismo tiempo. En la siguiente figura, comentada posteriormente, se muestran las capturas de tramas relativas al proceso de asociación de los dispositivos finales.



**Figura 5.7. Proceso de asociación simultánea de varios dispositivos finales (Escenario 2).**

En la figura anterior se puede ver cómo, tras el proceso de contienda para el acceso al canal, 3 de los 6 dispositivos finales consiguen enviar al coordinador de red las tramas de petición de asociación (etapa (1) en la figura), las cuales son respondidas mediante las correspondientes tramas ACK. Inmediatamente después, y siguiendo la misma filosofía para conseguir el acceso al canal, uno de los nodos que envió la petición de asociación logra enviar una trama de petición de datos, en la que requiere al coordinador que le sea asignada una dirección de red (etapa (2) en la figura). Acto seguido, el coordinador envía una trama indicando la dirección de red del nuevo nodo y una confirmación de asociación (etapa (3) en la figura). Sin embargo, este mensaje es recibido tanto por el nodo que envió la petición de dirección, como por el resto de nodos que consiguieron enviar la petición de asociación. Esto se debe a que durante el proceso de inicialización de los nodos (debido a la concepción de la aplicación ejemplo *msa*) todos tienen asignada la misma dirección de red extendida (*0x0100F0E0D0C0B0A0*) y, como consecuencia de esto, los 3 nodos que consiguieron acceso al canal de transmisión se asocian al coordinador de red con la misma dirección (*0x0001*). A continuación, los 2 dispositivos finales que han recibido la confirmación de asociación de manera errónea, consiguen enviar las tramas de petición de dirección que tenían pendientes al coordinador (etapa (4) en la figura), el cuál responde asignando dos nuevas direcciones de red (*0x0002* y *0x0003*). Estos mensajes enviados por el coordinador no tienen ningún efecto sobre los nodos, ya que, tras recibir la primera trama con la información de asociación, éstos se reconfiguraron con la primera dirección de red asignada.

Después de recibir la siguiente baliza, los 3 nodos restantes que aún no habían conseguido enviar la trama de petición de asociación a la red lo consiguen, repitiéndose el proceso descrito en el párrafo anterior (la figura no incluye los paquetes intercambiados durante estos procesos ya que siguen el mismo patrón que los mostrados).

En las siguientes tablas se muestra el resultado de la asignación de direcciones de red tanto para el coordinador de red como para los dispositivos finales.

<b>COORDINADOR DE RED</b>	
<b>Contenido de la tabla de dispositivos asociados</b>	{0x0001,0x0002,0x0003,0x0004,0x0005,0x0006}
<b>Direcciones reales asignadas</b>	{0x0001,0x0004}

**Tabla 5.11. Direcciones de red asignadas por el coordinador (Escenario 2).**



DISPOSITIVO FINAL	DIRECCIÓN DE RED ASIGNADA
Nodo A	0x0001
Nodo B	0x0001
Nodo C	0x0001
Nodo D	0x0004
Nodo E	0x0004
Nodo F	0x0004

**Tabla 5.12. Asignación real de direcciones de red (Escenario 2).**

Tras finalizar el proceso de asociación por parte de todos los dispositivos finales se comprueba que, debido a que la dirección con la que son inicializados es la misma, pueden surgir problemas a la hora de asignar las direcciones finales de red.

Este problema se registra independientemente del valor de los parámetros *Beacon Order* o *Superframe Order*, si bien se acentúa para valores altos de los mismos, llegando en ocasiones a asociarse todos los nodos a la red usando la misma dirección corta.

Otro problema que se ha observado derivado del intento simultáneo de asociación de varios nodos a una misma red balizada es que, en ocasiones, y de una manera aleatoria, el coordinador pasa a un estado en el que no permite asociarse a ninguno de los nodos que lo están requiriendo (cada vez que un nodo envía una trama de solicitud de dirección corta para la asociación a la red, el coordinador hace caso omiso y no devuelve la información requerida).

### **Escenario 3. Red balizada de 2 nodos. BO=14 SO=2**

El objetivo de este escenario es el de evaluar el tiempo de inicialización de una red y el tiempo de adhesión a la misma. Para ello se han utilizado dos nodos, uno que toma el papel de coordinador de red, y otro que lo hace de dispositivo final.

Al igual que en los escenarios anteriores, la configuración de los nodos se ha realizado utilizando las dos aplicaciones programadas en este proyecto, confirmando así el correcto funcionamiento de las mismas.

Los parámetros de configuración tanto para el coordinador de red como para el dispositivo final son los siguientes:

- *Beacon Order*: BO = 14 (tiempo máximo entre balizas)
- *Superframe Order*: SO = 2
- Canal de transmisión: 18
- Potencia de transmisión: 4.5 dBm

Una de las primeras funciones que debe realizar el dispositivo encargado de iniciar una red balizada es un escaneo pasivo del canal que ha sido previamente seleccionado, para comprobar así si hay alguna otra red balizada con los mismos parámetros de funcionamiento transmitiendo en el mismo canal.

El tiempo de escaneo del canal determinará en gran medida el tiempo de inicialización de la red, debiendo ser, como mínimo, igual al *Beacon Interval* (BI) de la red que se desea iniciar. De esta forma se garantiza que, si hay alguna otra red transmitiendo con los mismos parámetros de balizamiento, ésta va a ser detectada.

Sin embargo, en el caso de la creación de redes balizadas en las que el parámetro *Beacon Order* es igual a 14, se puede observar que el coordinador de red comienza a transmitir balizas de inmediato, sin llegar a efectuar ninguna comprobación de las posibles redes presentes en el mismo canal. Esto plantea un problema, y es que cuando el nodo que estaba destinado a comportarse como dispositivo final es iniciado, éste tampoco realiza un escaneo del canal para captar las balizas del coordinador con el que debe asociarse, de forma que automáticamente se establece como coordinador de red y comienza a transmitir sus propias balizas.

Este problema deriva directamente de la programación de la aplicación *msa*. Tras solicitar la creación de una nueva red balizada o la unión a una ya existente se llama a la función *MSA\_ScanReq(MAC\_SCAN\_PASSIVE, MSA\_MAC\_BEACON\_ORDER + 1)*, en la que el primer parámetro indica el tipo de escaneo del canal, y el segundo indica el exponente que permite determinar la duración del escaneo. En el siguiente cuadro se muestra el código de programación de esta función:

```

void MSA_ScanReq(uint8 scanType, uint8 scanDuration)
{
    macMlmeScanReq_t scanReq;

    /* Fill in information for scan request structure */
    scanReq.scanChannels = (uint32) 1 << MSA_MAC_CHANNEL;
    scanReq.scanType = scanType;
    scanReq.scanDuration = scanDuration;
    scanReq.maxResults = MSA_MAC_MAX_RESULTS;
    scanReq.result.pPanDescriptor = msa_PanDesc;

    /* Call scan request */
    MAC_MlmeScanReq(&scanReq);
}

```

El cometido principal de esta función es preparar la estructura de datos que se le pasa a la función *MAC\_MlmeScanReq()*, que es la encargada de realizar el escaneo del canal. De acuerdo con la documentación de diseño de la capa MAC de la pila de protocolos TIMAC [19], el parámetro *scanDuration* se calcula mediante la siguiente fórmula:

$$scanDuration = aBaseSuperframeDuration \cdot (2^{BO} + 1) \quad (5.1)$$

en la que el parámetro BO solo acepta valores entre 0 y 14 (el valor 15 no se contempla ya que en ningún caso se van a detectar redes balizadas cuyo valor de *Beacon Order* sea igual a 15).

Por tanto para asegurar el escaneo pasivo de canal durante la creación de redes balizadas o la asociación a éstas, simplemente hay que eliminar el “+1” en la llamada a la función *MSA\_ScanReq()*.

Tras corregir los fallos en el código de la aplicación *msa*, se ha vuelto a cargar en los nodos, comprobando que ahora sí es posible crear redes balizadas para cualquier valor de *Beacon Order*, y que además, el tiempo de inicialización de las mismas se reduce aproximadamente a la mitad.

### 5.3.2. SINCRONIZACIÓN EN REDES BALIZADAS

Un aspecto importante a tener en cuenta en el estudio de redes balizadas es la posible pérdida de sincronización de los nodos respecto a las balizas que emite el coordinador de

red. En el caso de la versión 1.3.0 de TIMAC se entiende como pérdida de sincronización la pérdida de 4 balizas consecutivas por parte de un nodo de la red. En esas circunstancias, la capa MAC manda un mensaje del tipo *MAC\_MLME\_SYNC\_LOSS* a la capa de aplicación para que gestione la pérdida de sincronización de manera oportuna.

En la siguiente tabla se resumen los resultados obtenidos en un estudio de pérdida de sincronización realizado sobre una red balizada formada por siete nodos (un coordinador de red y seis nodos finales) que se ha ido configurando con distintos valores del parámetro *Beacon Order*. Los resultados obtenidos se basan en una muestra de 200 balizas emitidas por cada valor de *Beacon Order*.

<i>Beacon Order</i>	Tiempo teórico entre Balizas (ms)	Número total de pérdidas de sincronización (sumando los 6 nodos)
<b>0</b>	15.36	0
<b>1</b>	30.72	0
<b>2</b>	61.44	0
<b>3</b>	122.88	0
<b>4</b>	245.76	0
<b>5</b>	491.52	0
<b>6</b>	983.04	0
<b>7</b>	1966.08	0
<b>8</b>	3932.16	0
<b>9</b>	7864.32	0
<b>10</b>	15728.64	0
<b>11</b>	31457.28	0
<b>12</b>	62914.56	0
<b>13</b>	125829.12	1
<b>14</b>	251658.24	1

**Tabla 5.13. Estudio de la pérdida de sincronización en redes balizadas.**

En la tabla anterior se observa que la pérdida de sincronización ocurre para valores altos del parámetro *Beacon Order*, en los que el tiempo entre balizas es mayor y puede verse alterado por la tolerancia del reloj de 32 MHz del módulo CC2530 (conviene recordar que

la razón por la que la funcionalidad de redes balizadas está deshabilitada en el kit de desarrollo CC2520, es la poca precisión que ofrecen los relojes que integra el sistema). Para contrastar estos datos, en la tabla 5.14 se muestra una comparación entre el tiempo teórico entre balizas y el tiempo medio real para cada valor del parámetro *Beacon Order*. El tiempo teórico se ha obtenido aplicando la ecuación [2.3], y el tiempo medio real midiendo la diferencia temporal entre balizas capturadas con el *software Packet Sniffer*.

<i>Beacon Order</i>	Tiempo teórico entre balizas (ms)	Tiempo medio real entre balizas (ms)	Desviación (ms)
0	15.36	15.36	0
1	30.72	30.72	0
2	61.44	61.44	0
3	122.88	122.88	0
4	245.76	245.758	-0.002
5	491.52	491.516	-0.004
6	983.04	983.031	-0.009
7	1966.08	1966.064	-0.016
8	3932.16	3932.129	-0.031
9	7864.32	7864.231	-0.089
10	15728.64	15728.478	-0.162
11	31457.28	31457.035	-0.245
12	62914.56	62913.998	-0.562
13	125829.12	125828.123	-0.997
14	251658.24	251656.348	-1.892

**Tabla 5.14. Comparación entre el tiempo teórico entre balizas y el tiempo real medio.**

Para finalizar este apartado conviene señalar que, siguiendo las recomendaciones de *Texas Instruments*, las pruebas de sincronización se han realizado estableciendo el valor mínimo permitido para la longitud del segmento de datos de las tramas tipo *beacon* (3 bytes), ya que para valores altos los nodos finales presentan problemas a la hora de decodificar dichas tramas y pierden la sincronización con mucha facilidad.

# CAPÍTULO 6

## Manual de Usuario

### 6.1. CONFIGURACIÓN DE LA PLACA SMARTRF05

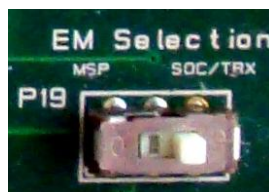
#### Conexión del módulo de evaluación

Antes de iniciar el sistema conviene asegurarse de que el módulo CC2530EM está correctamente conectado a la placa SmartRF05EB. Para ello debe ser insertado en los zócalos de pines etiquetados como P5 y P6.

En el caso de que la conexión no se haya efectuado correctamente aparecerá el siguiente mensaje en la pantalla LCD de la placa:

TexasInstruments  
SmartRF05 EB  
(No EM detected)

Debido a que la placa de desarrollo permite trabajar tanto con sistemas basados en microcontroladores MSP como con sistemas SoC, además de conectar el módulo CC2530 correctamente, es necesario asegurarse de que el *switch* de selección de módulo de evaluación (P19) esté en la posición SoC (tal y como se ilustra en la figura 6.1).



**Figura 6.1.** *Switch* de selección de módulo de evaluación.

## Alimentación

La placa de desarrollo SmartRF05EB cuenta con 4 modos de alimentación que pueden ser seleccionados mediante el *jumper* etiquetado como P11, si bien el modo de alimentación más común para este tipo de dispositivos es mediante 2 baterías del tipo AA que se alojan en la parte posterior de la placa. Una vez alimentada, el encendido y apagado del sistema se controla mediante el *switch* etiquetado como P8.

En la siguiente tabla se muestra el esquema de conexión del *jumper* P11 para la selección del modo de alimentación.

Modo de Alimentación	Conexión del <i>jumper</i> P11
Baterías AA	Pines 1 y 2
Alimentación USB	Pines 2 y 3
Alimentación DC externa	Pines 2 y 3
Fuente de laboratorio	Sin conexión

Tabla 6.1. Selección del modo de alimentación mediante el *jumper* P11.

## Puerto de Comunicaciones

Para poder efectuar la comunicación entre el PC y la placa de desarrollo, es necesario habilitar el puerto de comunicaciones RS232. Para ello hay que posicionar el *switch* etiquetado como P14 en la posición *ENABLE*, tal y como se representa en la figura 6.2.(c).

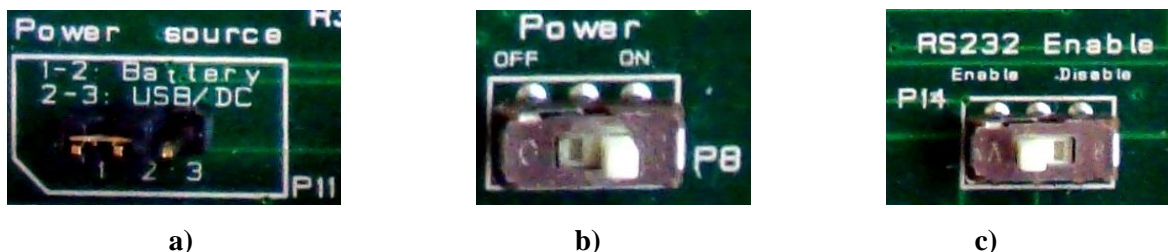


Figura 6.2. Selectores para la configuración de la placa SmartRF05EB a) *Jumper* para la selección del modo de alimentación. b) *Switch* de encendido de la placa SmartRF05EB. c) *Switch* para la habilitación del puerto de comunicaciones RS232.

En la siguiente figura se muestra una vista general de la placa de desarrollo, en la que se destaca la ubicación de los dispositivos más importantes que intervienen en las aplicaciones desarrolladas.

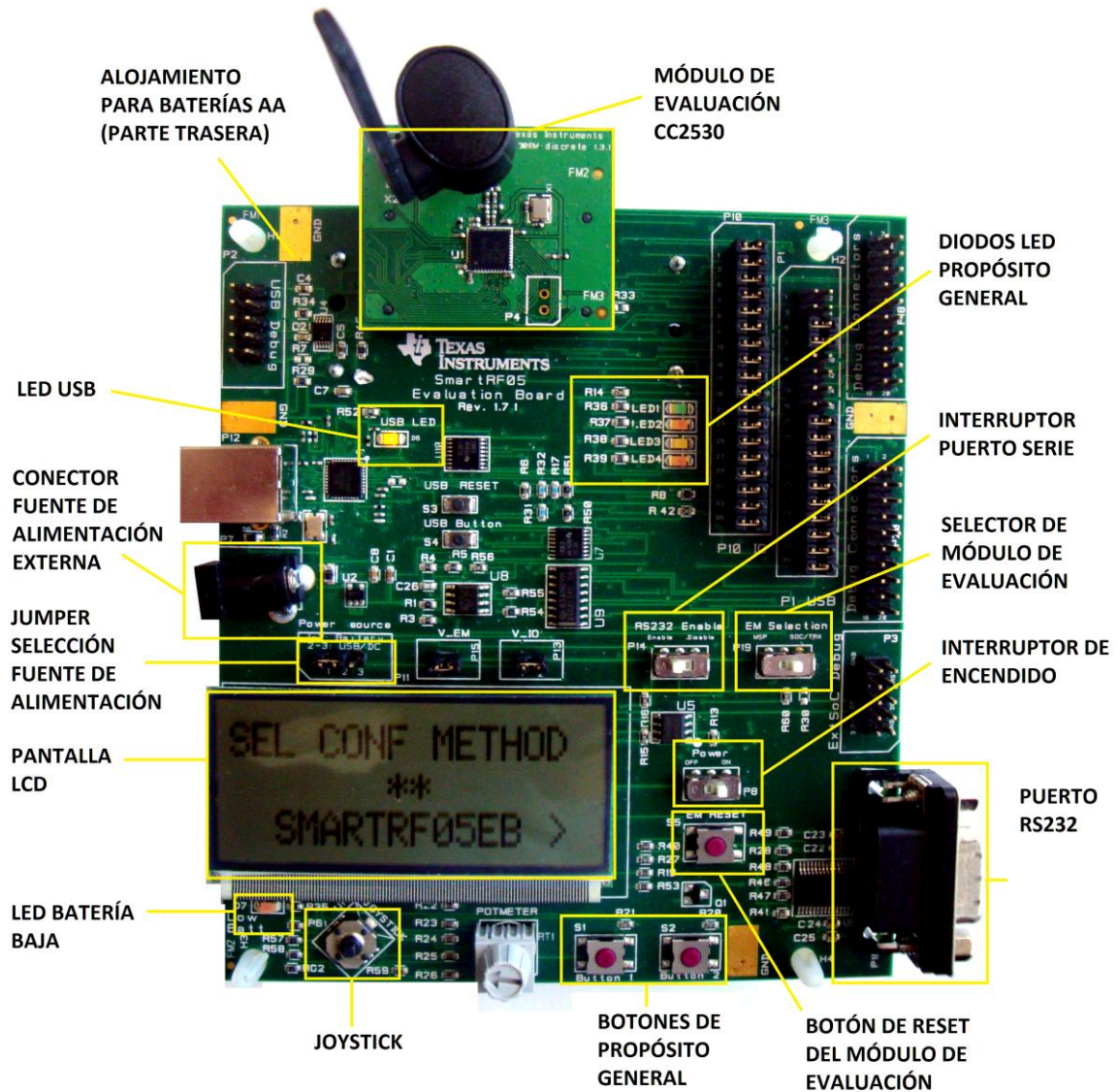


Figura 6.3. Ubicación de los periféricos en la placa SmartRF05EB.



## 6.2. APLICACIÓN SMARTRF05EB

### 6.2.1. FUNCIONALIDAD DE LOS PERIFÉRICOS DE LA PLACA

#### *Pantalla LCD*

La pantalla LCD tiene como cometido guiar al usuario a través del proceso de configuración de los nodos de red, mostrando los menús de configuración asociados a cada parámetro, el estado de los nodos una vez finalizado el proceso de configuración, y los mensajes de error en caso de haberlos.

En las tablas 6.2 y 6.3 se muestran los mensajes que pueden aparecer en la pantalla, así como una descripción de los mismos.

MENÚ	POSIBLES VALORES	DESCRIPCIÓN
<b>CONF_METHOD_SEL</b>	SMARTRF05EB > < SERIAL PORT	Menú que permite seleccionar el método de configuración del nodo.
<b>DEV_TYPE_SEL</b>	COORDINATOR > < END DEVICE	Menú para la selección del tipo de dispositivo.
<b>SET_TX_CHANNEL</b>	Ch11 2405 MHz > < Ch12 2410 MHz > ... < Ch26 2480 MHz	Menú para la selección del canal de Transmisión.
<b>SET_TX_POWER</b>	-22 dBm > < -18 dBm > ... < 4.5 dBm	Menú para la selección de la potencia de transmisión.
<b>SET_BEACON_ORDER</b>	0 >, < 1 >, < 2 > ..., <15	Menú para la selección del parámetro <i>Beacon Order</i> , que determina el carácter balizado de la red.
<b>SET_SFRAME_ORDER</b>	0 >, < 1 >, < 2 > ..., <15	Menú para la selección del parámetro <i>Superframe Order</i> , que determina el período activo en estructuras de supertrama.

**Tabla 6.2. Menús de configuración mostrados en la pantalla LCD.**

<b>MENSAJE</b>	<b>Significado</b>
<i>TexasInstruments SmartRF05 EB (No EM detected)</i>	Mensaje de error si no hay módulo de evaluación conectado o éste no funciona correctamente.
<i>CONFIG_VIA_COM Waiting for Data</i>	Mensaje que se muestra durante la espera de datos de la aplicación externa.
<i>Create_NWK: Push ^</i>	Instrucciones: Para crear una nueva red pulsar <i>Joystick Up</i> .
<i>Join NWK: Push ^</i>	Instrucciones: Para unirse a una red existente pulsar <i>Joystick Up</i> .
<i>COORDINATOR</i>	El nodo se ha establecido como coordinador de red.
<i>END DEVICE</i>	El nodo se ha establecido como dispositivo final.
<i>ERROR: SO&gt;BO CONFIG SO=BO Push S1 &gt;&gt;</i>	Error debido a que SO es mayor que BO, se procede a igualar el valor de SO con el de BO. Instrucciones: Pulsar el botón S1 para pasar a la siguiente pantalla.
<i>Scanning Channel</i>	Se está escaneando el canal seleccionado para crear una red o unirse a una ya existente.
<i>Sending Beacons</i>	El coordinador está transmitiendo balizas.
<i>Non Beaconed N.</i>	La red creada es No Balizada.
<i>&gt; Start/Stop Tx</i>	Instrucciones: Para iniciar o finalizar la transmisión de mensajes pulsar <i>Joystick Right</i> .
<i>Sending Messages</i>	Se están transmitiendo mensajes.

**Tabla 6.3. Mensajes de estado e instrucciones mostrados en la pantalla LCD.**

## Botones

En la siguiente tabla se muestra la funcionalidad de cada uno de los botones presentes en la placa SmartRF05EB, que podrá variar dependiendo del estado de ejecución de la aplicación de configuración:

BOTÓN	FUNCIÓN
<b>EM_RESET</b>	Reseteo del módulo de evaluación.
<b>USB_RESET</b>	Reseteo de la placa SmartRF05EB.
<b>S1</b>	Aceptación de los valores de configuración.
<b>Joystick Derecha</b>	Visualización de la siguiente opción del menú de configuración (durante la fase de configuración de los nodos).
	Inicio o finalización de envío de mensajes (una vez que la red ha sido creada).
<b>Joystick Izquierda</b>	Visualización de la opción anterior del menú de configuración (durante la fase de configuración de los nodos).
<b>Joystick Arriba</b>	Crear nueva red o unirse a una ya existente.

**Tabla 6.4. Funcionalidad de los botones de la placa SmartRF05EB.**

## Diodos LED

En la siguiente tabla se muestra la funcionalidad de cada diodo LED en función del estado en el que se encuentre:

LED	ESTADO	SIGNIFICADO
<b>LOW BATT</b>	Apagado	Nivel de batería adecuado.
	Parpadeo lento	Batería baja.
	Encendido fijo	Batería a punto de agotarse.
<b>USB LED</b>	Encendido	Indica que el sistema está conectado.
<b>LED 1</b>	Parpadeo rápido	Nodo preparado para crear una red o unirse a una existente.
	Parpadeo lento	El nodo se ha establecido como Dispositivo Final.
	Encendido fijo	El nodo se ha establecido como Coordinador de una nueva red.

**Tabla 6.5. Funcionalidad de los diodos LED de la placa SmartRF05EB.**

<b>LED 2</b>	Encendido	El nodo se está configurando con la aplicación SmartRF05EB.
<b>LED 3</b>	Encendido fijo	El nodo se está configurando con la aplicación remota.
	Cambia de estado intermitentemente	Una vez que la red ha sido creada, cada vez que un nodo (coordinador o dispositivo final) recibe un mensaje, cambia el estado de este diodo.
<b>LED 4</b>	Encendido	Se está pulsando el botón S1.

**Tabla 6.5. (Continuación) Funcionalidad de los diodos LED de la placa SmartRF05EB.**

## 6.2.2. CONFIGURACIÓN DE MOTAS CC2530

En las siguientes líneas se describe el proceso de configuración de nodos CC2530 de *Texas Instruments* para la creación de redes 802.15.4 mediante la aplicación de configuración SmartRF05EB.

Antes de comenzar es necesario tener en cuenta los aspectos relativos a la inicialización de la placa de desarrollo señalados en el primer apartado de este capítulo, asegurándose de que el módulo de evaluación está bien conectado y la placa está correctamente alimentada.

Para navegar a través de los distintos menús que irán apareciendo durante el proceso de configuración, utilice el *joystick* situado en la parte inferior izquierda de la placa de desarrollo.

### 1. Conexión del sistema

Para conectar el sistema, desplace el *switch* etiquetado como P8 a la posición ON.

### 2. Selección del método de configuración

Las dos opciones disponibles son “SMARTRF05EB” y “SERIAL COMM”. Para configurar el nodo mediante la placa de desarrollo, seleccione la primera opción y pulse el botón S1. Durante el proceso de configuración, el diodo LED etiquetado como LED 2 permanecerá encendido.

### 3. Selección del tipo de dispositivo

Seleccione el dispositivo que desea configurar y pulse S1. Las opciones disponibles son “COORDINATOR” y “END DEVICE”.

#### 4. Selección del canal de transmisión

En el estándar IEEE 802.15.4 se establecen 16 posibles canales de transmisión (numerados del 11 al 26) para los dispositivos que operan en la banda ISM de 2.4 GHz. Seleccione el canal que desea y pulse S1.

#### 5. Selección de la potencia de transmisión

Seleccione un valor de potencia de transmisión para el nodo y pulse S1. El subsistema de radiocomunicaciones del módulo CC2530 permite potencias de salida que oscilan entre los -22 dBm y los 4.5 dBm.

#### 6. Selección del parámetro *Beacon Order*

El parámetro *Beacon Order* determina el carácter balizado o no balizado de una red.

- Si desea crear una red no balizada o unirse a una ya existente, seleccione el valor 15 y pulse S1.
- En caso de querer crear una red balizada o unirse a una ya existente, seleccione un valor entre 0 y 14 y pulse S1. El tiempo entre balizas (*Beacon Interval*) puede calcularse fácilmente mediante la ecuación [2.3].

#### 7. Selección del parámetro *Superframe Order*

El parámetro *Superframe Order* determina la duración del período activo en redes balizadas. Este tiempo puede ser calculado mediante la ecuación [2.4]. En caso de tratarse de una red no balizada, se omitirá este paso.

Seleccione un valor para el parámetro *Superframe Order* y pulse S1, teniendo en cuenta que éste siempre debe ser menor o igual que el valor del parámetro *Beacon Order* (ya que en ningún caso el período activo puede ser mayor que el tiempo entre balizas). En caso de seleccionar un valor mayor, la aplicación automáticamente reconfigurará este parámetro igualándolo al valor seleccionado para el parámetro *Beacon Order*.

#### 8. Inicialización de una red o asociación a una ya existente

Tras configurar todos los parámetros, el nodo está listo para unirse a una red o crear una nueva. Si la configuración se ha realizado correctamente, el diodo LED etiquetado como LED 1, comenzará a parpadear a una frecuencia aproximada de 4 Hz.

Para crear una red o unirse a una existente pulse el *joystick* hacia arriba.

En caso de tratarse de una red no balizada, tanto el proceso de creación de la red como el de asociación se realizarán de manera prácticamente instantánea, mostrándose en la pantalla LCD de la placa el mensaje “*Non Beacons N.*”.

En caso contrario comenzará el proceso de escaneo del canal seleccionado para comprobar si hay alguna red operando en el mismo. Durante este tiempo se muestra por pantalla el mensaje “*Scanning...*”.

Tras finalizar el escaneo del canal, si éste está libre, el nodo se establecerá como coordinador, creando una nueva red. Si por el contrario se detectan balizas, el nodo se asociará a la red como dispositivo final.

## 9. Envío de mensajes

Para comenzar la transmisión de mensajes periódicos de prueba desde un dispositivo final al coordinador de red o desde el coordinador de red a todos los dispositivos asociados, pulse el *joystick* a la derecha. Para finalizar la transmisión vuelva a pulsar el *joystick* a la derecha.

Durante la transmisión de mensajes, en la pantalla del nodo que está realizando la transmisión aparece el texto “*Sending Messages*”. Además, el diodo LED 3 del dispositivo receptor cambiará su estado cada vez que reciba un nuevo mensaje.

La transmisión de mensajes desde el coordinador al resto de nodos de la red se realiza de forma secuencial, enviando un mensaje individual a cada una de las direcciones que aparece en la tabla de dispositivos asociados.

## 6.3. APLICACIÓN DE CONFIGURACIÓN REMOTA

### 6.3.1. INSTALACIÓN

El proceso de instalación de la aplicación remota para la configuración de nodos CC2530 es similar al de cualquier aplicación estándar que funciona bajo una plataforma *Windows*.

Antes de iniciar la instalación es necesario descomprimir el archivo “PFConfigCC2530.rar”, que contiene todos los archivos necesarios para que la aplicación

se instale y funcione correctamente. A continuación se explican los pasos a seguir durante la instalación de la aplicación:

- Para iniciar la instalación haga doble *click* en el archivo “*Setup.exe*”, ubicado en la carpeta descomprimida. Aparecerá la primera pantalla del proceso de instalación (figura 6.4).

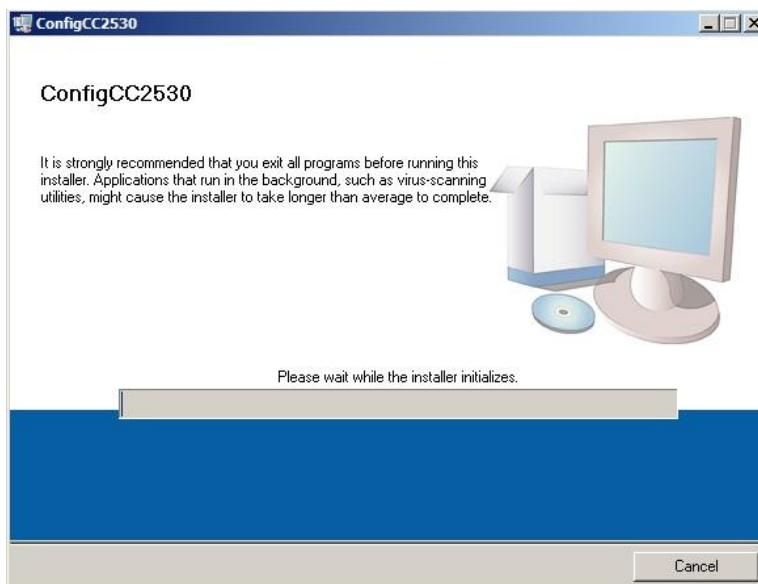


Figura 6.4. Instalación de la aplicación (pantalla inicial).

- Seleccione la ubicación en la que se va a instalar tanto la aplicación de configuración de nodos como el *software* de *National Instruments* necesario para el correcto funcionamiento de la aplicación (figura 6.5). Después pulse el botón “*Next*”.

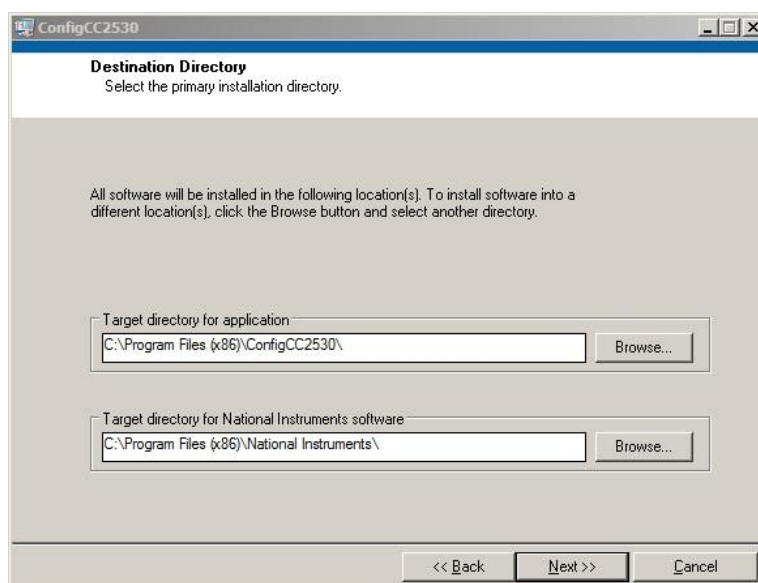
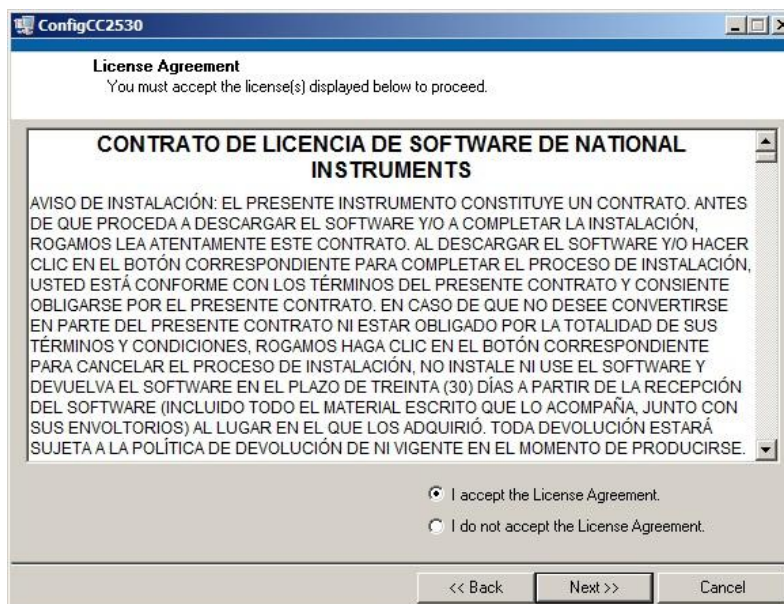


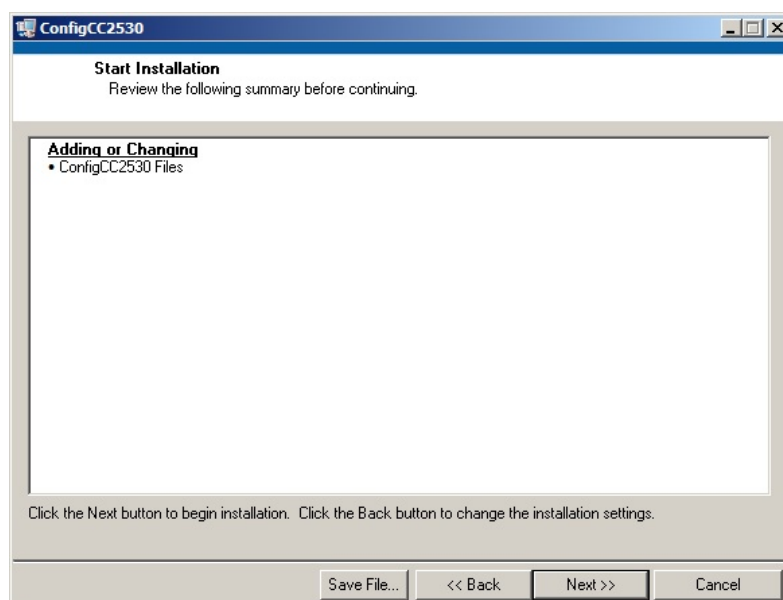
Figura 6.5. Instalación de la aplicación (selección de la ruta de instalación).

- Acepte el contrato de licencia de *National Instruments* (figura 6.6) y pulse el botón “*Next*”.



**Figura 6.6. Instalación de la aplicación (acuerdo de licencia).**

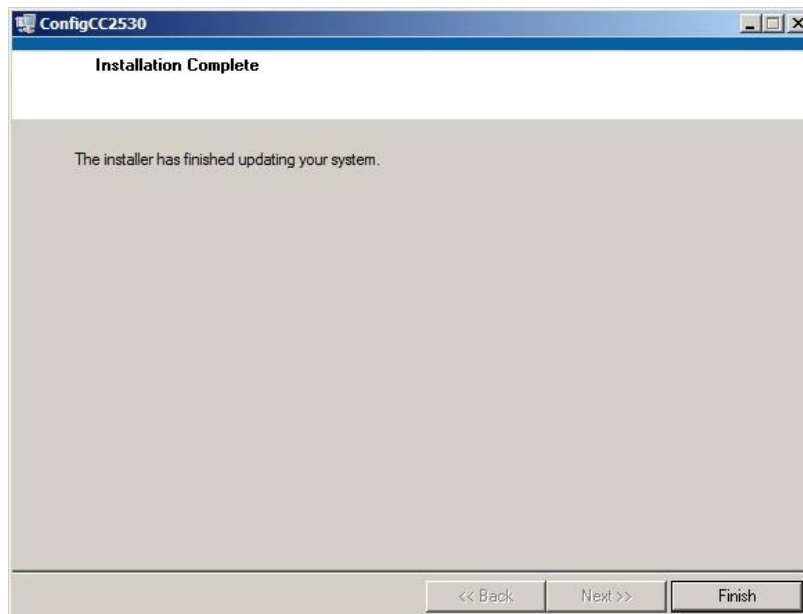
- En la siguiente pantalla (figura 6.7) aparece un resumen del *software* que se va a instalar. Para comenzar la instalación pulse el botón “*Next*”.



**Figura 6.7. Instalación de la aplicación (*software* a instalar).**



- Una vez finalizada la instalación (figura 6.8) pulse el botón “*Finish*” para salir del instalador. A partir de este momento la aplicación es accesible a través del menú Inicio de *Windows*.



**Figura 6.8. Instalación de la aplicación (final de la instalación).**

### 6.3.2. CONFIGURACIÓN DE MOTAS CC2530

Antes de comenzar con la configuración de los nodos es necesario comprobar el número asignado al puerto COM del ordenador, ya que será necesario para poder establecer correctamente la comunicación con el nodo a configurar. Además es necesario tener en cuenta los aspectos relativos a la inicialización de la placa de desarrollo señalados en el primer apartado de este capítulo, asegurándose de que el módulo de evaluación está bien conectado y la placa está correctamente alimentada.

A continuación se detallan los pasos a seguir para llevar a cabo la configuración de los nodos:

#### 1. Inicialización de la placa de desarrollo

Para conectar la placa de desarrollo, desplace el *switch* etiquetado como P8 a la posición ON.

## 2. Selección del método de configuración


Las dos opciones disponibles son “SMARTRF05EB” y “SERIAL COMM”. Para configurar el nodo mediante la aplicación de configuración remota, seleccione la segunda opción y pulse el botón S1. Durante el proceso de configuración, el diodo LED etiquetado como LED 3 permanecerá encendido.

En la pantalla LCD aparecerá el mensaje “*Waiting for data*” que indica que el nodo está listo para recibir los datos de configuración procedentes de la aplicación externa.

## 3. Conexión de la placa de desarrollo al PC

Conecte la placa de desarrollo SmartRF05EB al puerto de comunicaciones del PC usando el cable proporcionado para ello.

## 4. Inicialización de la aplicación de configuración remota

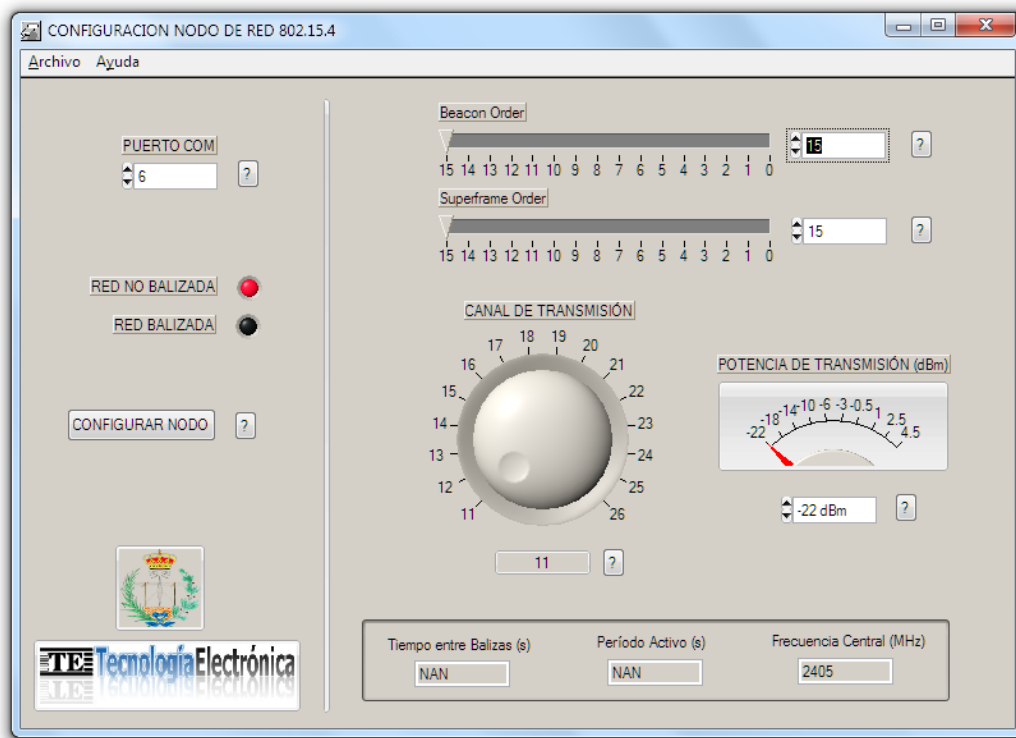
Para iniciar la aplicación de configuración remota haga doble *click* en el acceso directo  situado en los programas del menú inicio de *Windows*.

## 5. Selección de los valores de configuración

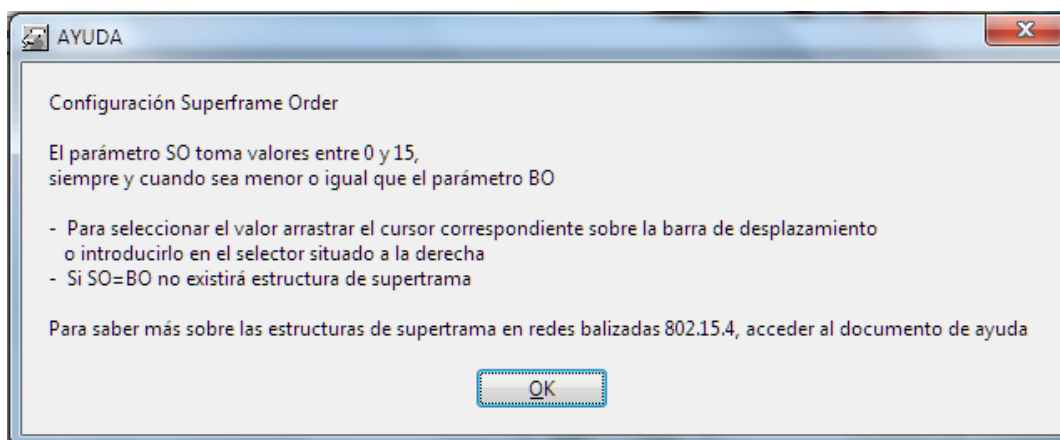
Seleccione los valores de los parámetros de configuración *Beacon Order*, *Superframe Order*, Canal de Transmisión y Potencia de Transmisión situando los controles etiquetados con el mismo nombre en el valor deseado. En la figura 6.9 se muestra una captura de pantalla de la interfaz gráfica de usuario.

Conviene aclarar que la aplicación no permite introducir ningún valor fuera del rango permitido para cada uno de los parámetros ni ninguna combinación de éstos que pueda generar conflicto durante la configuración de los nodos. Así, si, por ejemplo, se intenta introducir un valor del parámetro *Superframe Order* mayor que el del parámetro *Beacon Order*, la aplicación automáticamente igualará el valor del primero al valor del parámetro *Beacon Order*.

Al lado de cada control hay situado un botón de ayuda, que al ser accionado despliega una ventana emergente con la información acerca de ese control necesaria para su configuración. En la figura 6.10 se muestra un ejemplo de ventana de ayuda.



**Figura 6.9. Interfaz gráfica de la aplicación de configuración remota.**



**Figura 6.10. Mensaje de ayuda para la configuración del parámetro *Beacon Order*.**

En la zona inferior de la interfaz de la aplicación aparece un recuadro con la información relativa a los tiempos en redes balizadas y la frecuencia del canal seleccionado. Esta información se actualiza de manera instantánea al modificar los parámetros de configuración.

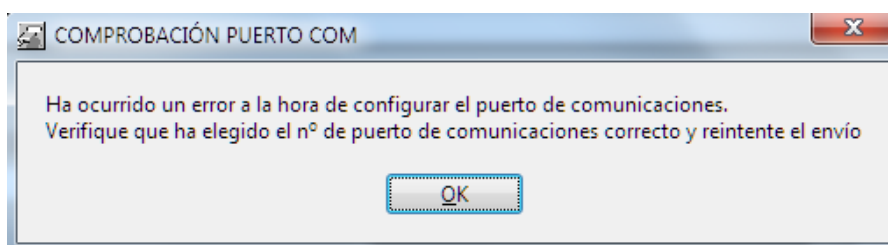
En la parte izquierda de la interfaz están situados los diodos LED de estado, que indican el tipo de red que se está configurando de acuerdo con los parámetros de configuración seleccionados. Estos diodos indican si se trata de una red balizada o no.

## 6. Envío de los datos de configuración al nodo

Antes de enviar la información de configuración al nodo compruebe que éste está correctamente conectado al PC a través del puerto COM.

Introduzca en el control situado en la parte superior izquierda de la interfaz el número del puerto de comunicaciones del PC que va a ser usado para enviar los datos al nodo y pulse el botón “Configurar”.

En el caso de que el puerto seleccionado sea erróneo aparecerá la siguiente ventana emergente indicando que se revise el puerto de comunicaciones o se seleccione uno distinto antes de reintentar el envío.



**Figura 6.11. Mensaje de error del puerto de comunicaciones.**

Si el envío se realiza correctamente, cambiará el estado de la pantalla LCD de la placa de desarrollo mostrando el siguiente mensaje a modo de instrucciones para crear una nueva red o unirse a una existente. Además se apagará el diodo LED 3 y comenzará a parpadear el diodo LED 1.

Push ^ for  
Creating/Joining  
a Network

## 7. Inicialización y asociación a redes. Envío de mensajes.

Siga las instrucciones de los puntos 8 y 9 del apartado 6.2.2 de este capítulo.



# CAPÍTULO 7

## Conclusiones y líneas futuras de trabajo

A lo largo de este capítulo se realiza una pequeña reflexión sobre los resultados obtenidos durante la elaboración del proyecto y sobre la tecnología 802.15.4/ZigBee en general, dejando abiertas posibles líneas de investigación que den continuidad al trabajo realizado.

### 7.1. CONCLUSIONES

Una vez finalizado el desarrollo de las aplicaciones que posibilitan la configuración de los nodos en redes IEEE 802.15.4 y comprobado el funcionamiento de las mismas se puede llegar a las siguientes conclusiones:

- La tecnología 802.15.4/ZigBee destaca por su bajo consumo, aumentando la autonomía de las baterías de los dispositivos de forma considerable respecto a otras tecnologías inalámbricas.
- Otra característica importante de este tipo de dispositivos es su bajo coste de producción, ya que están basados en microcontroladores sencillos (la pila *ZigBee* ocupa muy poco) y transceptores de muy baja potencia.
- Las características anteriores, junto a la sencillez en lo que al desarrollo de aplicaciones se refiere, hacen que esta tecnología sea idónea para su aplicación en sectores que están en plena expansión, como puede ser la domótica, la inmótica industrial y de servicios, o las redes de área personal en aplicaciones médicas.

- Aunque las bases teóricas de esta tecnología están bastante asentadas y existen estudios teóricos que la optimizan, actualmente el desarrollo de dispositivos que se ajustan a las especificaciones de la norma 802.15.4/ZigBee está en una fase muy inmadura.
- A día de hoy los principales fabricantes de semiconductores están produciendo gran cantidad de dispositivos a los que llaman *ZigBee*, pero que sin embargo se rigen por una interpretación propia y simplificada de dicha norma (*Z-Stack* en el caso de *Texas Instruments* o *BeeStack* en el caso de *Freescale*). En general dichos dispositivos se benefician de las características de bajo consumo de los nodos finales y de los algoritmos de formación de redes y enrutamiento definidos en la norma.
- Una de las características más importantes de la tecnología 802.15.4/ZigBee es la posibilidad de crear redes balizadas, en las que se permite optimizar el consumo tanto en coordinadores de red como en *routers*. Sin embargo a día de hoy hay muy pocos fabricantes que produzcan dispositivos con estas características implementadas directamente sobre *ZigBee*. Para poder tener esta funcionalidad hay que prescindir de las funcionalidades de las capas altas de *ZigBee* y utilizar protocolos simplificados basados en la norma IEEE 802.15.4, como es el caso de TIMAC.
- Como se ha visto durante el desarrollo del proyecto, la versión utilizada de TIMAC tiene una funcionalidad muy limitada en lo que a topologías de red se refiere, permitiendo únicamente crear redes en topología estrella. Además no es posible crear redes en las que esté habilitada la funcionalidad de *slots* temporales garantizados (GTS), otra de las características diferenciadoras de la tecnología 802.15.4, que permite reservar una serie de intervalos temporales en los que los dispositivos pueden transmitir su información pendiente sin necesidad de competir por acceder al canal (lo que reduce aún más el consumo de los nodos).
- Sin embargo, pese a las carencias detectadas en el estado actual de evolución de esta tecnología, el desarrollo de este proyecto ha posibilitado realizar una primera aproximación a redes balizadas reales, permitiendo la configuración de los parámetros básicos de las mismas y la evaluación de los procesos de creación y adhesión.

## 7.2. LÍNEAS FUTURAS DE TRABAJO

Partiendo de los resultados obtenidos durante el desarrollo de este proyecto, se abren interesantes líneas de trabajo para profundizar en el estudio de tecnología basada en el estándar IEEE 802.15.4, como pueden ser las siguientes:

- Realizar un estudio real del consumo de dispositivos 802.15.4 que forman parte de redes balizadas.
- Implementación de una aplicación basada en TIMAC que emule el funcionamiento de los *slots* temporales garantizados, permitiendo así realizar un estudio completo del funcionamiento de redes balizadas.
- Mejora y optimización de la aplicación *msa* de TIMAC para resolver los problemas derivados de la duplicidad durante la asignación de direcciones cortas de red a los dispositivos asociados.
- Realizar un estudio de coexistencia de las tecnologías basadas en el estándar IEEE 802.15.4 con las tecnologías inalámbricas que tienen más presencia en el mercado actual (Wi-Fi, *Bluetooth*, BTLE) y que funcionan en el mismo rango de frecuencias.





# BIBLIOGRAFÍA

- [1] Página web oficial del grupo de trabajo 802.11 del IEEE. <http://grouper.ieee.org/groups/802/11/>
- [2] H. Labiod, H. Afifi, C. De Santis Springer. *Wi-Fi, Bluetooth, ZigBee and WiMax*. Dordrecht: Springer 2007.
- [3] J. A. Gutiérrez, E. H. Callaway, R. L. Barrett. *Low-Rate Wireless Personal Area Networks, Enabling Wireless Sensors with IEEE 802.15.4*. New York: IEEE 2004.
- [4] R. Prasad, L. Deneire. *From WPANs to Personal Networks*. Boston: Artech House 2006.
- [5] Página web oficial del SIG de Bluetooth. <http://www.bluetooth.org>
- [6] Página web oficial de la ZigBee Alliance. <http://www.zigbee.org>
- [7] ZigBee Wireless Networking. Drew Gislason.
- [8] R. Morrow. *Wireless network coexistence*. New York: McGraw-Hill, 2004.
- [9] J. Mayné. “Estado actual de las Comunicaciones por Radio Frecuencia”, 2010, Documento en formato pdf accesible por internet en la dirección: [http://www.bairesrobotics.com.ar/data/estado\\_actual\\_de\\_las\\_comunicaciones\\_por\\_radiofrecuencia.pdf](http://www.bairesrobotics.com.ar/data/estado_actual_de_las_comunicaciones_por_radiofrecuencia.pdf)
- [10] S. Farahani. *ZigBee Wireless Networks and Transceivers*. Amsterdam: Newnes/Elsevier 2008.
- [11] IEEE Standard for Information technology. “Telecommunications and information exchange between systems. Local and metropolitan area networks specific requirements. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks

- (WPANs)", 2010. Documento en formato pdf accesible por internet en la dirección: <http://www.ieee.org>.
- [12] Página web oficial de IAR Systems. <http://www.iar.com>
- [13] Página web oficial de National Instruments. <http://www.ni.com>
- [14] Texas Instruments. "CC2530ZDK User's Guide (Rev. B)", 2010. Documento en formato pdf accesible por internet en la dirección: <http://www.ti.com/tool/cc2530ZDK>
- [15] Texas Instruments. "CC253x System-on-Chip Solution for 2.4 GHz IEEE 802.15.4 and ZigBee Application", 2010. Documento en formato pdf accesible por internet en la dirección: <http://www.ti.com/lit/ug/swru191/swru191.pdf>
- [16] Texas Instruments. "SmartRF05 Evaluation Board User's Guide", 2010. Documento en formato pdf accesible por internet en la dirección: <http://www.ti.com/lit/ug/swru210a/swru210a.pdf>
- [17] Texas Instruments. "SmartRF Packet Sniffer User Manual", 2011. Documento en formato pdf accesible por internet en la dirección: <http://www.ti.com/lit/ug/swru187f/swru187f.pdf>
- [18] Texas Instruments. "CC2531 USB Hardware User's Guide", 2011. Documento en formato pdf accesible por internet en la dirección: <http://www.ti.com/tool/cc2531emk>
- [19] Texas Instruments. "802.15.4 MAC API", 2010. Documento en formato pdf ubicado en la carpeta de instalación de la pila de protocolos TIMAC 1.3.0.
- [20] Texas Instruments. "HAL Driver API", 2010. Documento en formato pdf ubicado en la carpeta de instalación de la pila de protocolos TIMAC 1.3.0.
- [21] Texas Instruments. "MAC Sample Application – Software design", 2010. Documento en formato pdf ubicado en la carpeta de instalación de la pila de protocolos TIMAC 1.3.0.
- [22] Texas Instruments. "MAC User's guide cc2530", 2010. Documento en formato pdf ubicado en la carpeta de instalación de la pila de protocolos TIMAC 1.3.0.
- [23] Texas Instruments. "OSAL API", 2010. Documento en formato pdf ubicado en la carpeta de instalación de la pila de protocolos TIMAC 1.3.0.

- [24] Texas Instruments. “CC2531 System-on-Chip Solution for IEEE 802.15.4 and ZigBee Applications (Rev. A)”, 2010. Documento en formato pdf accesible por internet en la dirección: <http://www.ti.com/lit/ds/symlink/cc2531.pdf>.
- [25] Página web oficial de Texas Instruments. <http://www.ti.com>

## *BIBLIOGRAFÍA*