

Trabajo Fin de Grado  
Grado en Ingeniería de las Tecnologías de  
Telecomunicación

Aplicación de una red inalámbrica de sensores para  
detección de presencia y conteo

Autor: Álvaro Villalba Macías

Tutor: María José Madero Ayora

Dep. Teoría de la Señal y Comunicaciones  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2015





Trabajo Fin de Grado  
Grado en Ingeniería de las Tecnologías de Telecomunicación

# **Aplicación de una red inalámbrica de sensores para detección de presencia y conteo**

Autor:

Álvaro Villalba Macías

Tutor:

María José Madero Ayora

Profesor titular

Dep. de Teoría de la Señal y Comunicaciones

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2015



Aplicación de una red inalámbrica de sensores para detección de presencia y conteo

Autor: Álvaro Villalba Macías

Tutor: María José Madero Ayora

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2015

El Secretario del Tribunal



*A mi familia*

*A mis amigos*

*A mi tutora*



# Agradecimientos

---

Tras muchos meses de trabajo, complicaciones y disgustos llega ese día en el que todo va tomando forma y el final se acerca, se le va viendo la punta a todo. Seguramente podría haber hecho las cosas de otra forma, podría haber ahorrado algunos problemas que yo mismo generé con mi propia forma de actuar, pero así ha sido y no hay más. Después de estas palabras con un sentido tan trágico que no parecen sacadas de un trabajo de fin de grado, mis sensaciones no son tales, sino todo lo contrario, puedo decir que la experiencia obtenida al realizar este proyecto ha sido muy positiva y que he aprendido mucho. Todo esfuerzo tiene su recompensa, y en este caso, mi recompensa ha sido darme cuenta de que yo mismo podía llegar a resolver un problema con el que nunca antes había tenido que lidiar. El camino puede ser largo y difícil pero siempre encontraremos ayuda para hacernos más fácil el andar y sortear todas las piedras que vamos encontrando en el, no obstante, las decisiones de qué hacer en cada momento van a depender de nosotros mismos y no debemos perderlo de vista.

Así pues por la ayuda recibida, me gustaría citar algunas personas que me han ayudado mucho a lo largo de esta aventura para darle las gracias.

En primer lugar, y como no podía ser de otra manera mi familia (en especial a mi madre) por aguantarme en todo momento. Ellos son los que conviven bajo el mismo techo conmigo y los que ven tanto mis avances como mis caídas por llamarlo de alguna manera.

En segundo lugar, a Belén por aguantarme y animarme en los momentos buenos y malos e incluso ayudarme en algunas de las pruebas realizadas.

En tercer lugar a mi tutora, Maria Jose Madero por la rapidez de sus respuestas, su atención continua y por guiarme en todo este proceso de elaborar un texto de estas características.

En cuarto lugar, a Francisco Montoya Capote por transmitirme su experiencia y por darme algunas ideas para abordar los problemas que se me iban planteando.

En quinto y último lugar, no puedo olvidarme del foro de Libelium en Internet, por la paciencia de algunos de sus usuarios que me respondieron de buena gana a todas mis preguntas.

A todos ellos, muchísimas gracias de verdad.

*Álvaro Villalba Macías*

*Sevilla, 2015*



# Resumen

---

Las redes inalámbricas de sensores gozan de un gran auge hoy en día, habiendo sido identificadas como una de las tecnologías más prometedoras por diversos analistas tecnológicos y revistas especializadas, debido a que dan respuesta a las exigencias actuales referentes al establecimiento de redes que cubran necesidades de comunicación de forma flexible (en tiempo y espacio) y autónoma (autoconfiguración e independencia de una estructura fija). La posibilidad de implementar dispositivos de bajo coste y elevada duración capaces de obtener información del entorno y reenviarla de forma inalámbrica a un centro de coordinación ofrece posibilidades inimaginables en multitud de aplicaciones.

En este trabajo de fin de grado teórico/práctico se ha intentado dar una visión bastante completa sobre las redes inalámbricas de sensores, de la cantidad de problemas a los que pueden dar solución, sus múltiples ventajas así como sus limitaciones y aspectos a mejorar.

El campo en el que se centra el proyecto es el de la localización en interiores. Así pues, en la parte práctica del mismo se ha desarrollado un sistema IPS (*Indoor Position System*) del que se han obtenido algunas conclusiones que espero sirvan para seguir mejorando este tipo de sistemas.



# Abstract

---

Wireless sensor networks have undergone a remarkable rise in recent times. They have been identified as one of the most promising technologies by many technology analysts and specialized magazines. They give an answer to current demands for ways of communication on a flexible and autonomous basis, i.e., networks that set in time and space independently. It offers the possibility of implementing low-cost and high-life devices able to obtain information from the environment and forward it wirelessly to a coordination center. It will give us unthinkable possibilities at different areas of applicability.

This work aims to survey the wireless sensor networks possibilities, covering the scope of problems they can solve, their advantages and limitations as well as the areas for improvement.

The project focuses on indoor locations. Thus, it has developed a Indoor Position System (IPS) out of which some conclusions have been reached. They will hopefully stand as reference for further improvement of such systems.



# Índice

---

<b>Agradecimientos</b>	<b>i</b>
<b>Resumen</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Índice</b>	<b>vii</b>
<b>1 Introducción y estructura de la memoria</b>	<b>1</b>
1.1 <i>Objetivos</i>	1
1.2 <i>Estructura del TFG</i>	1
<b>2 Redes inalámbricas de sensores</b>	<b>3</b>
2.1 <i>Introducción</i>	3
2.2 <i>Aplicaciones</i>	3
2.3 <i>Factores de diseño</i>	5
2.3.1 Tolerancia a fallos	5
2.3.2 Escalabilidad	5
2.3.3 Costes de producción	6
2.3.4 Limitaciones en el Hardware	6
2.3.5 Topologías de red	6
2.3.6 Entorno	8
2.3.7 Medio de Transmisión	8
2.3.8 Consumo y obtención de energía	9
2.4 <i>Hardware en WSNs</i>	9
2.4.1 Nodos sensores	9
2.4.2 Gateway	11
2.4.3 Estación Base	11
2.5 <i>Software en WSNs: Sistemas operativos</i>	11
2.6 <i>Arquitecturas y tecnologías</i>	12
2.6.1 Arquitectura de capas	12
2.6.2 Capa Física	13
2.6.3 Capa de Enlace	15
2.6.4 Capa de Red	16
2.6.5 Capa de Transporte	17
2.7 <i>Estándares</i>	18
2.7.1 IEEE 802.15.4	18
2.7.2 ZigBee	20
2.7.3 WirelessHART	22
2.7.4 ISA 100.11	23
2.7.5 6LoWPAN	23
2.7.6 Bluetooth Low Energy	23
2.8 <i>Seguridad</i>	23
2.9 <i>Futuro de las WSNs</i>	24
<b>3 Localización en interiores</b>	<b>27</b>
3.1 <i>Problemática de la localización en interiores</i>	27

3.2	<i>Tecnologías de localización</i>	28
3.2.1	GPS	28
3.2.2	RFID (Radio Frequency Identification)	28
3.2.3	Infrarrojos	29
3.2.4	Telefonía Movil	29
3.2.5	Bluetooth	29
3.2.6	WiMAX	29
3.2.7	UWB (Ultra Wide Band)	30
3.2.8	Wi-Fi	30
3.2.9	Redes inalámbricas de sensores	30
3.3	<i>Técnicas de localización en Redes Inalámbricas de Sensores</i>	30
3.4	<i>Técnicas de estimación de distancias</i>	32
3.4.1	Tiempo de llegada (ToA – Time of Arrival)	32
3.4.2	Diferencia de tiempos de llegada (TDoA: Time Difference of Arrival)	32
3.4.3	Ángulo de llegada (AoA - Angle of Arrival)	33
3.4.4	Fuerza de la señal recibida (Received Signal Stength)	34
3.4.5	Combinaciones de parámetros	35
3.5	<i>Técnicas de estimación de posición</i>	35
3.5.1	Técnicas geométricas	35
3.5.2	Técnicas estadísticas	37
3.5.3	Técnica de reconocimiento de patrones	38
3.6	<i>Detección Humana</i>	38
<b>4</b>	<b>Waspnote</b>	<b>43</b>
4.1	<i>Datos Generales</i>	43
4.1.1	Arquitectura modular	43
4.1.2	Especificaciones técnicas	43
4.1.3	Datos eléctricos	44
4.1.4	Sensores incorporados	44
4.2	<i>Entradas/Salidas</i>	45
4.2.1	Analógicas	46
4.2.2	Digitales	46
4.2.3	PWM (Pulse Width Modulation)	46
4.2.4	UART (Universal Asynchronous Receiver Transmitter)	46
4.2.5	I2C (Inter Integrated Circuit)	46
4.2.6	SPI (Serial Peripheral Interface)	47
4.2.7	USB (Universal Serial Bus)	47
4.3	<i>Sensores incorporados en Waspnote</i>	47
4.3.1	Sensor de temperatura	47
4.3.2	Acelerómetro	47
4.4	<i>Fuentes de alimentación</i>	47
4.5	<i>Otros components</i>	48
4.5.1	LEDs	48
4.5.2	Jumpers	48
4.5.3	Memoria EEPROM	48
4.5.4	Tarjeta de memoria Flash micro-SD	48
4.5.5	Modulos externos	49
4.6	<i>Sistema energético</i>	49
4.7	<i>Timers</i>	50
4.8	<i>Watchdog</i>	51
4.9	<i>RTC (Real Time Clock)</i>	51
4.10	<i>Arquitectura y Sistema</i>	51
4.11	<i>Interrupciones</i>	52

4.12	<i>Wasmote IDE</i>	54
4.13	<i>Programación básica de una Wasmote</i>	55
4.14	<i>Errores frecuentes al cargar programas en Wasmote</i>	58
<b>5</b>	<b>Wasmote Gateway</b>	<b>59</b>
5.1	<i>Botones</i>	59
5.2	<i>Leds</i>	60
5.3	<i>Configuración de módulos transeptores a través de la gateway</i>	60
5.4	<i>Programas receptores de datos</i>	60
5.5	<i>XBee Explorer</i>	62
5.6	<i>Over the air programming (OTAP)</i>	62
5.6.1	Conceptos	62
5.6.2	Ventajas	62
5.6.3	Topologías	63
5.6.4	Cifrado y autenticación	63
5.6.5	Modos	63
5.6.6	Sistema de almacenamiento	63
5.6.7	OTA-Shell	63
5.6.8	OTA 802.15.4/Zigbee paso a paso	63
<b>6</b>	<b>XBEE</b>	<b>67</b>
6.1	<i>Introducción</i>	67
6.2	<i>Arquitectura básica de una red XBee</i>	68
6.3	<i>Modo de operacion de un XBee</i>	69
6.3.1	Modo recibir/transmitir.	69
6.3.2	Modo de bajo consumo (sleep mode)	70
6.3.3	Modo de comando	70
6.3.4	IDLE	70
6.4	<i>Configuracion de modulos XBee con X-CTU</i>	70
6.5	<i>Recuperacion y actualizacion de firmware</i>	73
<b>7</b>	<b>Aplicación de localización en interiores con redes inalámbricas de sensores</b>	<b>75</b>
7.1	<i>Entorno de la aplicación</i>	75
7.2	<i>Descripción general</i>	75
7.3	<i>Justificación de la solución empleada</i>	76
7.4	<i>Técnica de reconocimiento de patrones</i>	76
7.4.1	Fase de calibración	76
7.4.2	Fase de localización	77
7.5	<i>Análisis de la solución</i>	78
7.5.1	Código del nodo OVNI	78
7.5.2	Código para los nodos baliza	78
7.5.3	Código de la estación base	78
7.6	<i>Puesta en marcha de la aplicación</i>	80
7.7	<i>Pruebas</i>	81
7.7.1	OVNI estático	82
7.7.2	OVNI en movimiento	84
<b>8</b>	<b>Conclusiones y líneas futuras de trabajo</b>	<b>85</b>
8.1	<i>Conclusiones</i>	85
8.2	<i>Líneas futuras de trabajo</i>	86
8.3	<i>Campos de aplicación</i>	87
<b>Anexo A: XBee Serie 1</b>		<b>89</b>
<b>Anexo B: Códigos de Programación</b>		<b>93</b>
<b>Referencias</b>		<b>101</b>



# 1 INTRODUCCIÓN Y ESTRUCTURA DE LA MEMORIA

---

En este trabajo de fin de grado teórico/práctico se ha intentado dar una visión bastante completa sobre las redes inalámbricas de sensores, de la cantidad de problemas a los que pueden dar solución, sus múltiples ventajas así como sus limitaciones y aspectos a mejorar. Para desarrollar algunos de estos conocimientos adquiridos se va a implementar una aplicación para la detección de presencia en interiores.

En este primer capítulo se definen los objetivos que se pretenden cumplir en el trabajo, así como la estructura utilizada en la memoria.

## 1.1 Objetivos

Los objetivos de este TFG, se indican a continuación:

- Mostrar una panorámica sobre las redes inalámbricas de sensores (WSNs).
- Estudiar las tecnologías de localización en interiores y su problemática asociada.
- Profundizar en la localización en interiores con redes inalámbricas de sensores.
- Desarrollar una aplicación de localización en interiores con una red inalámbrica de Waspnotes.

## 1.2 Estructura del TFG

El TFG esta estructurado en 8 capítulos:

En el Capítulo 1 se realiza la introducción al proyecto, se definen los objetivos y se describe la estructura del TFG.

En el Capítulo 2 se tratan todos los aspectos relacionados con las redes inalámbricas de sensores, sus aplicaciones, factores de diseño, tipos de nodos, arquitecturas y tecnologías, estándares, seguridad, etc.

En el Capítulo 3 se estudia la problemática que presenta la propagación en ambientes de interior, se repasan algunas de las tecnologías de localización actuales, para centrarnos finalmente en las técnicas de localización con redes inalámbricas de sensores.

En los Capítulos 4, 5 y 6 se describen el hardware y software que se utilizarán en la aplicación de localización en interiores. Los sensores elegidos pertenecen a la empresa española Libelium y son las Waspnote v1.1. Los transceptores de RF montados sobre los sensores son los XBee Serie 1 basados en el estándar 801.15.4.

En el Capítulo 7 se desarrolla toda la aplicación. Se habla sobre la lógica de la solución implementada basada en los niveles RSSI y la técnica de reconocimiento de patrones, la topología de la red, la configuración de los componentes, los programas implementados en cada tipo de nodo, la puesta en marcha para terminar con algunas pruebas que muestran el funcionamiento del sistema.

El Capítulo 8 recoge las conclusiones obtenidas del trabajo realizado con el sistema de localización, algunos de los campos de uso para el mismo, así como posibles mejoras al sistema.

Por último se incluyen dos anexos. En el primero de ellos (Anexo A) se describen los módulos de RF XBee utilizados en la aplicación, sus especificaciones y algunos puntos importantes sobre el estándar que implementan. En el segundo (Anexo B) se incluyen los códigos desarrollados para la aplicación de localización por si el lector estuviese interesado en su estudio.

## 2 REDES INALÁMBRICAS DE SENSORES

---

Este capítulo tiene por objetivo acercar al lector a la tecnología en torno a las redes inalámbricas de sensores (WSNs) con el fin de que éste conozca los aspectos más importantes de diseño, el hardware y software disponibles, las topologías y su mantenimiento, estándares y protocolos usados, así como el gran abanico de aplicaciones a las que se puede dar solución con este tipo de redes y las líneas futuras de investigación y desarrollo en este campo.

### 2.1 Introducción

Una WSN (Wireless Sensor Network en inglés) es una red inalámbrica compuesta por nodos sensores (que pueden interactuar entre ellos) que permite monitorizar parámetros físicos de un cierto entorno y ofrecer los datos a uno o más nodos sumideros que se encargan de recolectar la información obtenida. En algunos casos, los mismos elementos de una red de sensores pueden ser empleados para activar actuadores, para llevar a cabo tareas en respuesta a ciertos estímulos (por ejemplo la detección de un determinado evento).

Un nodo sensor es un dispositivo con capacidad de procesamiento y transmisión (en general, inalámbrica) que incluye uno o más sensores, y que puede incluir actuadores.

Los sensores son dispositivos que miden magnitudes físicas, convirtiendo la medida en una señal (que por ejemplo puede ser eléctrica). Algunos ejemplos de magnitudes que puede medir un sensor son: intensidad eléctrica, tensión, presión, humedad, temperatura, intensidad acústica, concentración de oxígeno, etc.

Los actuadores son dispositivos que pueden llevar a cabo una acción (por ejemplo disparar una alarma, apagar un sistema de riego, etc.) en respuesta a un cierto estímulo.

Las WSNs pueden ser redes aisladas, pero es interesante conectarlas a otras redes (por ejemplo Internet) para su acceso remoto y gestión. En tal caso, la conectividad entre la WSN y otra red puede ser lograda por medio de una pasarela o *gateway*, o bien otros dispositivos, dependiendo de los protocolos empleados en la WSN y en la red a la cual ésta se conecta.

La tecnología WSN abre la puerta a un gran abanico de aplicaciones de control y monitorización en multitud de entornos, tales como la automatización residencial y de edificios, el control industrial, la monitorización ambiental, etc.

Las WSNs se caracterizan por el hecho de que los nodos que las componen exhiben limitaciones hardware significativas (incluyendo que en muchos casos pueden estar únicamente alimentados por una batería). Por otra parte, se asume que el número de nodos de una WSN puede ser elevado.

En la mayoría de casos, los nodos sensores emplean arquitecturas de protocolos de comunicaciones basadas en capas.

Las principales topologías que existen en las WSNs son las topologías en estrella, malladas y en árbol. La topología en estrella puede ser adecuada para aplicaciones donde existe un nodo central que controla al resto. Por otra parte, la topología en árbol suele emplearse para la recolección de datos hacia un sumidero que se ubica en la raíz del árbol. En muchas ocasiones, resulta conveniente el empleo de una topología mallada, que ofrece diversos caminos entre dos nodos y, por tanto, es tolerante a caídas de enlaces y/o nodos.

### 2.2 Aplicaciones

El interés por las redes de sensores inalámbricos no ha parado de crecer en la medida que la tecnología ha hecho posible su realización. El éxito de las WSNs radica en las innumerables posibilidades que ofrecen a los usuarios en forma de aplicaciones. A continuación se presentan algunas de ellas:

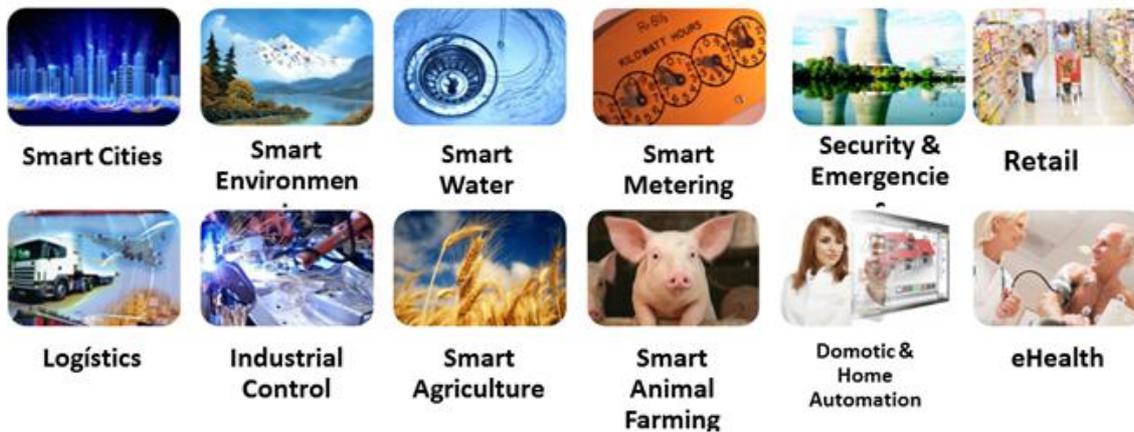


Figura 2-1. Algunas aplicaciones de las WSNs.

- **Teleasistencia médica:** personas con problemas de salud (por ejemplo ancianos, discapacitados, etc.) pueden beneficiarse del uso de sensores que midan sus parámetros biométricos, entre otros, para permitir su monitorización remota permanente. De este modo, se pueden llevar a cabo diagnósticos precisos y se pueden disparar alarmas en caso de que se detecte alguna situación de emergencia.
- **Monitorización medioambiental:** las propiedades de reducido tamaño y gran autonomía hacen de los sensores los elementos ideales para poder monitorizar el entorno. Los sensores se pueden camuflar fácilmente o incluso colocarse encima de animales.
- **Monitorización preventiva:** tanto en infraestructuras como en sistemas mecánicos, las redes de sensores se pueden usar para realizar una monitorización continua de parámetros que permitan predecir el colapso de un puente o la rotura de una pieza. Para este propósito se usan sensores de vibración, desplazamiento o temperatura, principalmente. Se debe disponer de un software que pueda extrapolar la evolución temporal de las medidas en potenciales problemas en los sistemas.
- **Agricultura:** la mejora de la productividad, no sólo en cantidad sino también en calidad, y la escasez de agua en ciertas regiones justifica el uso de sensores que permitan ver el grado de humedad del suelo para determinar el momento y la cantidad de riego. También la humedad de las hojas se puede usar para predecir la posibilidad de que ocurran infecciones de hongos, así como los medidores de radiación solar, para determinar el grado de madurez y con ello el momento ideal para la recolección. El uso de redes de sensores sin hilos se está planteando en cultivos como el de la vid, en los que se requiere un control preciso de la calidad.
- **Automatización de edificios:** un hogar o una oficina son lugares en los que se pueden encontrar diversos sensores, actuadores y dispositivos que deben ser controlados. Hasta la fecha, han existido soluciones específicas para cada área como climatización, control luminoso o seguridad, pero eran costosas, complejas de instalar y propietarias. Con el desarrollo de las redes de sensores sin hilos se pueden unificar todas las soluciones en un único sistema, facilitando mejoras de funcionamiento y de coste.
- **Medida de contadores:** las empresas de suministro de gas, agua y electricidad llevan intentando realizar las lecturas de los contadores situados en casa del cliente de forma remota durante mucho tiempo. Las redes de sensores en un primer paso han de medir el consumo y reportarlo al operador. En un segundo paso, del que ya se está hablando, se debe poder usar estas redes para poder interactuar con los equipos del usuario y poder realizar un control de la demanda y ofrecer servicios adicionales.

- **Ciudades inteligentes:** la gestión urbana engloba aspectos como la gestión de los equipamientos, zonas verdes, recogida de basuras, transporte público, alcantarillado, tráfico rodado e iluminación, entre otros muchos. Las redes de sensores se pueden usar como monitores del correcto funcionamiento y como automatismos inteligentes. Se puede controlar el riego de las zonas verdes, el ahorro en la iluminación en zonas sin presencia de personas, planificar la recogida de basuras en función de la ocupación de los contenedores o la identificación de plazas de aparcamiento disponibles. Esta información puede redundar en una reducción del personal y una mejora de la información ofrecida al ciudadano.
- **Generación de contexto:** los sensores, al poder captar parámetros físicos de entorno, tienen la capacidad de describirlo. Esta virtualización del entorno se puede usar para crear un contexto que facilite el uso de aplicaciones. Por ejemplo, el conocimiento de la localización se puede usar para entregar información adecuada al usuario o las condiciones de nivel de ruido o iluminación para ajustar el reproductor de contenidos. Los teléfonos móviles actuales disponen de una amplia gama de sensores (posición, temperatura, proximidad, aceleración, campo magnético). Además, en un futuro próximo, podrían disponer de interfaces radio Bluetooth con modo de operación *Low Energy*, que les permitirán recoger información de otros sensores situados en las inmediaciones.

## 2.3 Factores de diseño

El diseño de una WSN está bajo la influencia de muchos factores. A continuación se hará un estudio de algunos de ellos que serán de vital importancia para el diseño de protocolos o algoritmos en WSNs. Además, estos factores se pueden usar para comparar diferentes esquemas de diseño.

### 2.3.1 Tolerancia a fallos

Algunos nodos sensores pueden fallar o bloquearse debido a la carencia de energía, tener daños físicos o sufrir interferencias en sus comunicaciones. El fallo de nodos individuales no debería afectar a la tarea global de la WSN. La tolerancia a fallos es la capacidad de mantener las funcionalidades de la red sin interrupciones debido a fallos de nodos individuales. La fiabilidad o la tolerancia a fallos de un nodo, puede ser modelada con la distribución exponencial para hallar la probabilidad de no sufrir un fallo dentro de un intervalo de tiempo  $(0, t)$ :

$$P(t) = e^{-\lambda t} \quad (2-1)$$

donde  $\lambda$  y  $t$  son la tasa de fallo del nodo y el periodo de tiempo, respectivamente. Nótese que los protocolos y algoritmos se pueden diseñar con un nivel de tolerancia a fallos distinto para cada aplicación específica. Si en el entorno donde los nodos se encuentran desplegados no existe interferencia apreciable, entonces los protocolos pueden ser menos estrictos. Por ejemplo, si los nodos son desplegados en una vivienda para monitorizar la humedad y temperatura de la misma, la exigencia de tolerancia a fallos es baja, ya que esta clase de redes no es dañada fácilmente ni interferida por el ruido ambiental. Por otra parte, si los nodos son desplegados en un campo de batalla para la vigilancia y la detección, entonces la tolerancia a fallos tiene que ser alta porque los datos recogidos son críticos y los nodos pueden ser destruidos por acciones hostiles.

Así pues, podemos decir que el nivel de tolerancia a fallos depende de la aplicación de la WSN y el diseño debe tener esta característica siempre presente.

### 2.3.2 Escalabilidad

El número de nodos desplegados en el fenómeno bajo estudio puede ser del orden de cientos, miles e incluso, en aplicaciones muy concretas, millones. Los nuevos esquemas deben ser capaces de trabajar con este número de nodos y utilizar la alta densidad presente. La densidad puede ser calculada como:

---

(2-2)

donde  $N$  es el número de nodos dispersos en la región  $A$  y  $R$  es el rango de transmisión radio de un nodo. Básicamente,  $U(R)$  proporciona el número de nodos dentro del radio de transmisión de cada nodo en la región  $A$ . En definitiva, la densidad de nodos necesaria depende de la aplicación a la que los nodos son destinados.

### 2.3.3 Costes de producción

Dado que las WSNs consisten en un número grande de nodos sensores, el coste de un nodo es muy importante para justificar el coste total de la red. Si el coste de la red es superior al despliegue de sensores tradicionales, entonces la WSN no está justificada en cuanto a su coste. Por consiguiente, el coste de cada nodo tiene que ser muy bajo.

### 2.3.4 Limitaciones en el Hardware

Si bien las WSNs tienen múltiples ventajas que las hacen atractivas para solucionar gran cantidad de problemas y aplicaciones, también es cierto que poseen varias limitaciones, que poco a poco y gracias a la evolución de la tecnología se van resolviendo.

Algunas de estas limitaciones están relacionadas con la energía que alimenta a los nodos, con el corto rango de transmisión, el reducido ancho de banda o las limitadas capacidades de procesamiento y almacenamiento en los mismos.

Por ejemplo, las cuestiones relacionadas con energía pueden ser resueltas obteniendo la energía de la red eléctrica o también de placas fotovoltaicas si consideramos aprovechar la radiación solar. Claro está que estas soluciones conllevan un gasto extra y habrá que estudiar si compensa o no en el contexto concreto, para una aplicación determinada.

### 2.3.5 Topologías de red

Como ya se comentó en la introducción, en WSNs se utilizan principalmente tres tipos de topologías: en estrella, malladas y en árbol. Si bien en algunos casos se implementan combinaciones de las anteriores.

#### 2.3.5.1 Topología en Estrella

El gateway de la red se sitúa en el centro de la misma, es decir, es un sistema donde la información enviada sólo da un salto y donde todos los nodos sensores están en comunicación directa con la puerta de enlace, normalmente a una distancia de 30 a 100 metros. Los nodos finales no intercambian información entre ellos, sino que usan la puerta de enlace para ello, si es necesario. La puerta de enlace también es usada para transmitir datos al exterior y permitir la monitorización de la red.

La topología en estrella como se puede ver en la Figura 2-2 es la que menor gasto de energía desarrolla, pero por el contrario está limitada por la distancia de transmisión vía radio entre cada nodo y la puerta de enlace. Tampoco tiene un camino de comunicación alternativo en caso de que uno de los nodos tenga obstruido el camino de comunicación, lo que lleva a que en este caso la información de ese nodo sea perdida.

#### 2.3.5.2 Topología mallada

La topología en malla es un sistema multisalto, donde todos los nodos son a la vez enrutadores e idénticos. Cada nodo es capaz de enviar y recibir información de otro nodo y de la puerta de enlace. A diferencia de la topología en estrella, donde los nodos solo pueden hablar con la puerta de enlace, en ésta los nodos pueden intercambiar mensajes entre ellos.

Esta topología es altamente tolerante a fallos ya que cada nodo tiene diferentes caminos para comunicarse con la puerta de enlace como se observa en la Figura 2-2. Si un nodo falla, la red se reconfigurará alrededor del nodo fallido automáticamente gracias a los protocolos de encaminamiento que pueden ser implementados.

### 2.3.5.3 Topología en árbol

El coordinador de la red se comunica solo con dispositivos enrutadores, los cuales a su vez comunican de forma directa con los nodos finales. Un ejemplo sería un edificio en el que se establece un clúster por planta, los dispositivos finales que haya instalados en dicha planta comunicarán únicamente con el nodo enrutador de dicha planta, que tendrá que encaminar los datos, poniendo en contacto así a dichos dispositivos con el coordinador, que será la raíz del árbol. Como muestra la Figura 2-2.

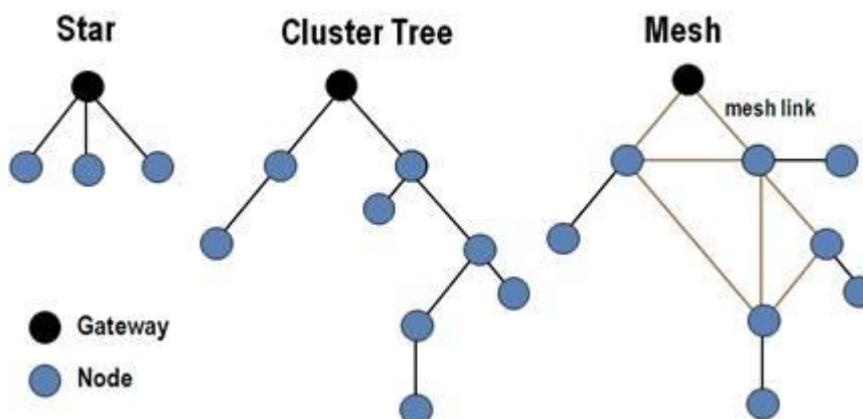


Figura 2-2. Topologías en WSNs.

### 2.3.5.4 Mantenimiento y cambios en la topología

En cuanto al mantenimiento de estas estructuras, dependiendo de la densidad de nodos que formen parte de la red o por ejemplo del lugar donde se efectúe el despliegue de los mismos (nodos inaccesibles y desatendidos) dicha labor puede convertirse en una tarea ardua y difícil.

A continuación, se examinan las cuestiones relacionadas con el mantenimiento y cambio de la topología en tres fases:

- **Fase previa y de despliegue:** los nodos pueden ser desplegados en masa o bien colocarse uno a uno en el área objetivo. De este modo, pueden lanzarse desde un avión o simplemente ser ubicados uno a uno por una persona o robot.

Aunque el número de sensores y su despliegue desatendido evite, por lo general, colocarlos según un plan de despliegue cuidadoso, los esquemas para la ubicación inicial deben reducir el coste de instalación, eliminar la necesidad de cualquier organización y planificación previa, aumentar la flexibilidad de arreglo y promover la tolerancia a fallos y la organización automática.

- **Fase posterior al despliegue:** después del despliegue, los cambios de topología son debidos a cambios de posición en los nodos, accesibilidad (debido a congestión, ruido, movimiento de obstáculos, etc.), energía disponible, mal funcionamiento, etc.
- **Fase de despliegue de nodos adicionales:** en cualquier momento pueden utilizarse nuevos nodos para sustituir a otros que funcionan incorrectamente o debido a cambios en la funcionalidad de la red. La adición de nuevos nodos plantea la necesidad de reorganizar la red.

Enfrentarse a cambios frecuentes de topología en una WSN con gran número de nodos y restricciones de consumo energético muy rigurosas, requiere de protocolos de encaminamiento especiales.

### 2.3.6 Entorno

Los nodos de una WSN son, en general, desplegados de forma densa muy cerca o directamente sobre el fenómeno en estudio, ese del que queremos tomar medidas para obtener información. Las WSNs pueden operar en entornos tan diversos como en las calles, en maquinas, en el fondo del océano, dentro de un tornado, en lugares contaminados, en un campo de batalla, en un edificio o almacén, conectado a animales o personas, ríos, etc. Estos son algunos ejemplos para que nos hagamos una idea bajo qué condiciones se espera que los nodos trabajen. Dada la cantidad de aplicaciones que pueden ser resueltas mediante redes de sensores y las condiciones a las que estos pueden ser sometidos será importante adaptar el diseño de los mismos para resistir al entorno particular donde van a realizar su trabajo.

### 2.3.7 Medio de Transmisión

En una WSN, los nodos se comunican utilizando enlaces inalámbricos. Estos enlaces pueden ser de radiofrecuencia (RF), infrarrojos u ópticos. Los transmisores ópticos consumen menos energía que los de RF y proporcionan seguridad, pero requieren de línea de visión directa entre transmisor y receptor y, además, son sensibles a las condiciones atmosféricas. Los transmisores de infrarrojos, no necesitan de antenas de grandes dimensiones, pero están limitados por su capacidad de difusión. La tecnología RF es la más fácil de usar, pero requiere de antena. En definitiva, las comunicaciones RF son las más comunes en aplicaciones de WSNs y por ello el estudio se centrará en esta tecnología.

Para permitir la operación global de estas redes, el medio de transmisión escogido debe estar disponible mundialmente. Una opción para enlaces radio es el empleo de la bandas de frecuencias para uso no comercial en áreas Industrial, Científica y Médica (ICM), que ofrece comunicación sin licencia en la mayoría de los países. En la Tabla 2-1 se especifican para el caso particular de España, las bandas de frecuencias recogidas en el artículo 5 del Reglamento de Radiocomunicaciones (RR) de la ITU, que pueden ponerse a disposición de las aplicaciones ICM.

Tabla 2-1. Bandas de Frecuencia ICM en España.

Banda de frecuencia	Frecuencia central
6765-6795 kHz	6780 kHz
13553-13567 kHz	13560 kHz
26957-27283 kHz	27120 kHz
40.66-40.70 MHz	40.68 MHz
2400-2500 MHz	2450 MHz
5725-5875 MHz	5800 MHz
24-24.25 GHz	24.125 GHz
61-61.5 GHz	61.25 GHz
122-123 GHz	122.5 GHz
244-246 GHz	245 GHz

Algunas de estas bandas de frecuencias están siendo ya utilizadas en sistemas de telefonía inalámbricos y en redes inalámbricas de área local (WLANs).

Para las WSNs se requieren transeptores de pequeñas dimensiones, bajo coste y de muy baja potencia de transmisión. Las restricciones hardware y el compromiso entre eficiencia de antena y consumo energético hicieron que el estándar IEEE 802.15.4 emplease las bandas de 2,4 GHz (global), 915 MHz (EEUU) y 868 MHz (Europa) en las comunicaciones. Las ventajas principales de usar las bandas ISM son la libertad de transmisión radio en esas bandas, la enorme asignación de espectro y la disponibilidad global. Estas bandas no están destinadas a un estándar en particular, lo que permite la implementación práctica de estrategias de bajo consumo energético en WSNs. Sin embargo, existen varias reglas y restricciones, como son las limitaciones de potencia emitida y la posibilidad de interferir con otros sistemas. Estas bandas de frecuencias también son conocidas en la literatura como frecuencias sin licencia.

De forma adicional, los requerimientos para aplicaciones no usuales complican la elección del medio de

transmisión. Por ejemplo, las aplicaciones marinas pueden requerir el empleo del medio de transmisión acuoso. Aquí, lo ideal sería usar una portadora de longitud de onda larga para que pueda penetrar en la superficie del agua. Por otro lado, en aplicaciones militares se podrían encontrar canales propensos a errores o con grandes interferencias. Además, la antena de un nodo no podría tener la altura y la potencia de emisión requeridas para estas aplicaciones en otros dispositivos inalámbricos. De ahí que la elección del medio de transmisión debe estar acompañada por una codificación robusta y esquemas de modulación adecuados que modelen eficientemente estas características particulares.

### **2.3.8 Consumo y obtención de energía**

Un aspecto fundamental en el diseño y despliegue de WSNs es el consumo de energía de los nodos. Las soluciones tradicionales basadas en el empleo de baterías, o bien conectar los nodos a la red eléctrica (cuando ésta esté disponible) son soluciones viables sólo para un conjunto limitado de usos. Para una aceptación amplia de las WSNs, una vez se ha emplazado un nodo en una WSN, éste debería poder operar de manera no atendida, mientras que el recambio de su batería puede ser una opción inviable o poco práctica. Las limitaciones de energía de las WSNs dan lugar a una serie de principios de diseño distintos a los que habitualmente existen en las redes tradicionales.

El elemento que contribuye en mayor medida al consumo de energía de un nodo sensor es la comunicación del mismo (tanto la transmisión como la recepción de señales). El procesado de datos también consume energía, pero en menor medida. De ello se deduce que es preferible procesar la información de que se dispone si con ello se ahorran bytes a transmitir.

Es destacable el hecho de que, para determinadas potencias de transmisión, un nodo sensor puede consumir más energía en recepción que en transmisión. De todos modos, la potencia empleada en las comunicaciones depende linealmente de la frecuencia con la cual se realizan transmisiones y recepciones. Por otra parte, los periodos de preparación para que un transceptor pase de modo inactivo a activo también consumen energía y no deben menospreciarse.

Se pueden emplear dispositivos de almacenaje de energía para alimentar a los nodos de una WSN. Estos dispositivos pueden ser baterías primarias (como las que se emplean para alimentar electrodomésticos habituales) o secundarias (baterías recargables), o bien los denominados supercondensadores.

Pese a que se está realizando un esfuerzo importante por reducir el consumo energético de los nodos sensores, se están considerando fuentes de energía alternativas para maximizar el tiempo de vida de los nodos de una WSN. En particular, se está considerando la obtención de la energía que puede existir en el ambiente. Algunas posibles fuentes de energía son la radiación solar, la energía mecánica (el movimiento, la presión o la vibración), los gradientes de temperatura, los flujos de viento y/o líquido y la radiación electromagnética ambiental. Este tipo de fuentes ofrecen niveles de potencia bajos, variables y difíciles de predecir, dada su naturaleza.

Además, en muchos casos, requieren el uso de extensiones aparatosas y caras. Sin embargo, existen dispositivos comerciales que incorporan una placa solar para su alimentación o productos que constituyen soluciones auto-alimentadas de WSNs.

## **2.4 Hardware en WSNs**

### **2.4.1 Nodos sensores**

Un nodo sensor está compuesto por un conjunto de elementos como un procesador, memoria, transceptor radio, antena, fuente de energía y sensores o actuadores como se muestra en la Figura 2-3. Algunas características comunes de estos elementos son la minimización del tamaño, del coste y del consumo energético. La mayoría de elementos se presentan montados en una placa de circuito impreso y se ofrecen conectores a todos aquellos elementos que pueden tener más variabilidad en la elección, como antena, sensores o actuadores y fuente de alimentación.

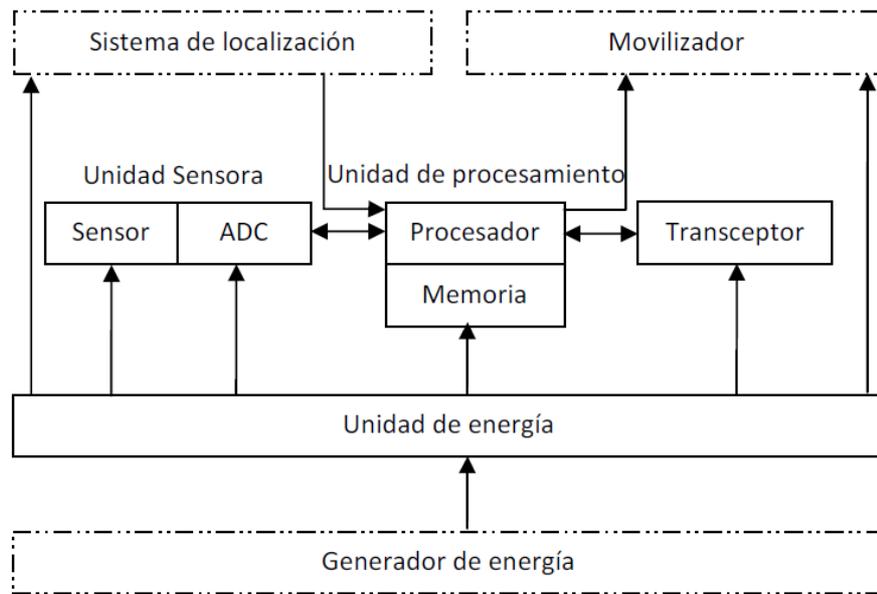


Figura 2-3. Elementos de un Nodo sensor.

El circuito transceptor se encarga de generar una señal de radiofrecuencia a partir de una de banda base digital y viceversa. Las bandas de frecuencias usadas corresponden a las ICM, de mayor aceptación global como la de 2.4 GHz o la de 868 MHz (Europa) de uso más local. En cualquier caso, la placa del nodo debe integrar los transceptores necesarios de acuerdo a las bandas de frecuencia donde se vaya a trabajar. El estándar más adoptado en los transceptores de un nodo sensor es el IEEE 802.15.4, pero también podemos encontrar otros genéricos (que permiten definir un interfaz radio diseñado en función de las necesidades, gustos o preferencias específicas de cada usuario) y otros propietarios. El medio radio es compartido y, por ello, se precisa un protocolo de acceso o MAC. Este protocolo se implementa entre el transceptor radio y el procesador.

El procesador, el elemento inteligente del nodo, se cataloga como MCU (*Microcontroller Unit*) que engloba una unidad de proceso simple, memoria e interfaces de entrada/salida. La gama de MCUs es muy extensa, con diversos fabricantes que ofrecen una amplia gama de modelos. Se pueden observar unos dispositivos que usan procesadores desde 8 bits hasta los de 32 bits con una arquitectura RISC (*Reduced Instruction Set Computer*), que ofrece hasta los 32 MIPS (*Million Instructions Per Second*). La cantidad de memoria disponible, particularmente la RAM, es muy variable. Podemos encontrar MCUs con 1 kB de RAM hasta las que disponen de 128 kB. La memoria no volátil puede ser ROM o flash y varía entre las decenas de kilobytes a las pocas centenas. Un aspecto importante en estos componentes son los modos de funcionamiento y el consumo asociado. La MCU puede pasar de un estado inactivo (*sleep mode*) con alguna de sus componentes desconectadas y con un reloj de 32 kHz a un estado activo con un reloj de decenas de MHz. Entre estos dos estados se pasa de las fracciones de microamperio a los pocos miliamperios. Las MCUs también disponen de una amplia gama de entradas y salidas. Se dispone de varios puertos analógicos y sus correspondientes conversores de analógico a digital y digital a analógico. Estos conversores permiten que ciertos tipos de sensores y actuadores que manejan señales analógicas se puedan conectar fácilmente. También se dispone de puertos serie o UART (*Universal Asynchronous Receiver/Transmitter*) y buses que permiten conectar varios dispositivos como el I2C (*Inter-Integrated Circuit*), el SPI (*Serial Peripheral Interface*) o líneas individuales de entrada/salida. Alguno de estos puertos se usan para conectar sensores/actuadores que manejan información en formato digital, para conectar memoria flash o conectarse con el transceptor radio.

En cuanto a los sensores, estos presentan una gama muy amplia de interfaces, alimentaciones y señales de diferentes formatos. Generalmente se requieren circuitos de adaptación, ya que los niveles de señal y las alimentaciones que se pueden ofrecer no encajan con los de la MCU. Existen iniciativas para la estandarización de las interfaces como la del IEEE 1451 que no han tenido hasta el momento la aceptación necesaria para ser relevante en el área. Existe una iniciativa denominada *phidgets* que ofrece un único interfaz.

Como elementos adicionales, tenemos las antenas. Como el lugar de ubicación de un sensor y los nodos con los que se comunica no es conocido generalmente a priori, las antenas son omnidireccionales. Las bandas de

frecuencia permiten tener antenas monopolo o antena *whip*, o una antena integrada en placa de circuito impreso o cerámica (también llamada antena *chip*). Las primeras generalmente ofrecen mejor ganancia, pero requieren un conector y, en algunos casos, también un cable coaxial. Las segundas son económicas mientras que las antenas *chip* pueden ser de muy reducido tamaño. Estas dos últimas tienen el problema de que la directividad de la antena se ve afectada por los propios componentes y que se deben instalar en compartimentos no metálicos.

Opcionalmente, también podría estar provisto de otros componentes adicionales dependiendo de la aplicación concreta, como un sistema de posicionamiento, un generador de energía o un movilizador.

El movilizador permite desplazar los nodos cuando se requiere llevar a cabo determinadas tareas.

Todas estas unidades tienen que encajar en un módulo de reducidas dimensiones, que puede llegar a ser del orden de un centímetro cúbico. Además del tamaño, existen otras restricciones para los nodos. Estos deben:

- Requerir poca energía.
- Funcionar en despliegues con alta densidad de nodos.
- Tener un coste de producción bajo y ser prescindibles.
- Ser autónomos y funcionar de forma desatendida.
- Ser adaptables al entorno.

Ya que los nodos son a menudo inaccesibles, la vida útil de una WSN depende de la duración de los recursos energéticos de los nodos. Teniendo en cuenta que el tamaño de éstos es limitado, el almacenamiento de energía se ve directamente afectado y, por tanto, en aplicaciones que requieren un tiempo de vida mayor de la WSN es necesario buscar alternativas, tales como incluir componentes que recojan energía del entorno para hacer al nodo autónomo. Esto es posible por ejemplo, incluyendo células solares en el nodo sensor.

### 2.4.2 Gateway

Un Gateway o pasarela de traducción de protocolos es un dispositivo que soporta dos o más interfaces físicas y la semántica y funcionalidad de dos o más pilas de protocolos. Así pues, un dispositivo de este tipo puede conectar una red ZigBee con una red IP por ejemplo, sin embargo, estos dispositivos exhiben también algunas desventajas. Una de ellas es la falta de consistencia extremo a extremo, lo que dificulta realizar tareas como el transporte o la calidad de servicio de forma eficaz. De hecho, los dispositivos de traducción de protocolos pueden emplear el principio del “mínimo común denominador” para permitir la interoperabilidad de las distintas tecnologías que comunican. Por otra parte, dado que la WSN y la red IP a la que se conecta constituyen dominios distintos, existen mayores riesgos a nivel de seguridad, puesto que a cada lado de la pasarela existe una esfera de seguridad distinta.

### 2.4.3 Estación Base

Es el elemento recolector de los datos generados por los nodos, basado en un ordenador común o sistema embebido.

## 2.5 Software en WSNs: Sistemas operativos

Como ya se comentó anteriormente, un nodo sensor dispone de un microprocesador que debe ejecutar una serie de tareas que requieren el acceso al hardware del nodo como son la memoria, entradas/salidas y comunicaciones. El sistema operativo (SO) es el software que permite coordinar el uso de los recursos y que ofrece una interfaz con el hardware para hacerlo independiente de las aplicaciones que en el nodo sensor se

ejecutan. Las funciones básicas que el sistema operativo realiza son: gestión del procesador, gestión de la memoria, gestión de los dispositivos, planificación de los tiempos de ejecución y gestión de la ejecución de tareas concurrentemente.

Adicionalmente, es deseable que un sistema operativo ofrezca funcionalidades que faciliten la carga y descarga de código y que ofrezca un interfaz para que las aplicaciones puedan acceder al hardware sin tener un conocimiento detallado del mismo. Para los sistemas operativos de nodos sensores, otra funcionalidad necesaria será la gestión de la energía con el objetivo de minimizar el consumo. Un sistema operativo es un programa de gran importancia en cualquier elemento procesador, ya que facilita el desarrollo de aplicaciones, pero no debe degradar las prestaciones que este elemento procesador puede ofrecer. En entornos en los que recursos son amplios este último requerimiento no es vital, pero para un nodo sensor en el que la capacidad de proceso y memoria son extremadamente limitados, este requerimiento es una necesidad. En un escenario distribuido como es una red de sensores, el sistema operativo puede ofrecer facilidades adicionales relacionadas con ejecución distribuida de aplicaciones.

Este aspecto se vincula generalmente con una aplicación de intermediación conocida como *middleware* que se puede ubicar en un elemento de la red, como por ejemplo el nodo pasarela, y los nodos sensores. Algunos *middleware* son una tarea más del nodo y en otros casos el sistema operativo incorpora estas funcionalidades.

En la literatura podemos encontrar una gran variedad de sistemas operativos para nodos sensores. Algunos de ellos son adaptaciones de soluciones existentes para plataformas de limitadas prestaciones y otros han sido desarrollados para nodos sensores. La clasificación de los sistemas operativos se puede hacer en función de características básicas como son la arquitectura, la reprogramación, el modelo de ejecución y la planificación (*scheduling*) temporal.

La arquitectura de un SO se refiere a la organización del mismo y de las aplicaciones que se ejecutan. Existe la arquitectura denominada monolítica (con SO y aplicaciones integradas en un programa), la modular y la basada en máquina virtual. En esta última, el SO incorpora componentes de las aplicaciones que permiten hacer las aplicaciones más compactas y permite esconder la red, esto es, que el desarrollador de aplicaciones pueda despreocuparse parcial o totalmente de los detalles de la red sobre la cual se ejecutarán las aplicaciones.

El modelo de ejecución de un SO indica de qué modo se ejecutan las aplicaciones en el nodo. El más común es el basado en eventos, pero también encontramos el basado en hilos o *threads*. Para el primer caso, las aplicaciones generan eventos que son tratados uno detrás de otro. Existen otras opciones, como la basada en máquina de estados.

La reprogramación hace referencia a la facilidad para modificar el software que se ejecuta en el SO. Este aspecto tiene especial interés en las redes de sensores dada la posible inaccesibilidad de los nodos de red, en cuyo caso se haría de forma remota.

En cuanto a la planificación temporal, el procesador se reparte entre las diferentes tareas. Las formas más comunes en redes de sensores son periódica/no periódica y crítica/no crítica. Por ejemplo, una medida de temperatura es periódica, una generación de una alarma es no periódica.

Existe una gran variedad de sistemas operativos en la literatura. Los que podemos considerar que gozan de mayor aceptación son el TinyOS, FreeRTOS y Contiki. Todos ellos son de código abierto.

## 2.6 Arquitecturas y tecnologías

### 2.6.1 Arquitectura de capas

El desarrollo de una pila de protocolos fiable y eficiente en términos de energía es importante para soportar las múltiples aplicaciones en la WSNs. Dependiendo de la aplicación, una WSN puede consistir en decenas o cientos de nodos desplegados en el escenario bajo estudio como se muestra en la Figura 2-4. Cada nodo tiene la capacidad de recoger datos del entorno, de encaminarlos hasta la estación base que a través de otras infraestructuras o redes hará posible la comunicación con el nodo administrador de tareas encargado de recibir las órdenes del usuario final.

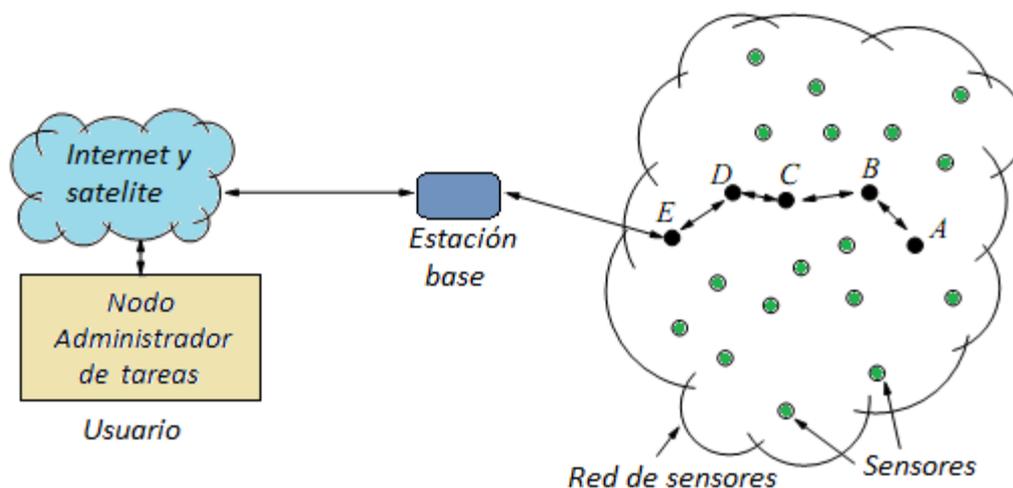


Figura 2-4. Red de Sensores.

Cada nodo usa una pila de protocolos como la mostrada en la Figura 2-5 para comunicarse con cualquier otro y con la estación base. Ésta debe ser energéticamente eficiente en términos de comunicación y ser capaz de trabajar de forma óptima a través de múltiples nodos.



Figura 2-5. Modelo de capas en WSNs.

La implementación de protocolos en las diferentes capas de la pila de protocolos puede afectar significativamente al consumo de energía, al retraso extremo a extremo y a la eficiencia del sistema. Es de vital importancia optimizar la comunicación y minimizar la energía empleada. Para ello y conocidas las restricciones y características concretas en WSNs se hace necesario el diseño de protocolos que soporten la interacción entre capas de protocolo para compartir datos útiles con todas las capas y así poder enfrentarse con garantías a los requerimientos que plantea el diseño de este tipo de redes.

A continuación, se hace una introducción a las diferentes capas presentes en la pila de comunicación de una WSN, destacando los aspectos más influyentes.

### 2.6.2 Capa Física

La capa física proporciona un interfaz para la transmisión de flujos binarios sobre el medio físico. Es responsable de interactuar con la capa MAC, y proporciona capacidades de transmisión, recepción y modulación. La interacción entre la capa física y la capa MAC es un aspecto muy importante en el diseño debido a que la tasa de error en la capa física es alta y variante en el tiempo en un ambiente inalámbrico. Por ello, la capa MAC interactúa con la capa física para detectar y corregir errores. Otras interacciones implican

compartir información de canal y transmisión con la capa MAC para conseguir mayor rendimiento y mejor utilización de recursos.

Para una WSN, los problemas de reducir al mínimo el consumo de energía y maximizar el tiempo de vida de la red comienzan en la capa física. En esta capa, la energía se emplea en la operación del circuito radio y en la transmisión. Mientras que la energía utilizada en la operación del circuito radio es fija, la utilizada en el proceso de transmisión de datos puede variar según las pérdidas de paquetes, interferencias y distancia de transmisión. Existe un compromiso entre potencia de transmisión y la tasa de error producida. Una selección apropiada de la potencia de transmisión reduce al mínimo la pérdida de energía y hace operar a la red de forma más eficiente. Además, los esquemas de modulación son necesarios para transmitir datos sobre un canal inalámbrico. Así, se han desarrollado distintos esquemas de modulación para minimizar la probabilidad de error bajo diferentes condiciones.

Los esquemas de modulación energéticamente eficientes deberían reducir al mínimo tanto la energía de operación del circuito radio como la de transmisión. Recientes estudios de investigación incluyen como requerimientos de capa física un diseño simple de chips radio de baja potencia y esquemas de transmisión y modulación.

La capa física se debe diseñar considerando los requerimientos de una WSN. Según las características de ésta, habrá restricciones en los chips radio en términos de energía, tasas de datos, probabilidad de error, distancia de transmisión y fiabilidad.

Otras restricciones que deben considerarse en la capa física son las interferencias, la sincronización y la habilidad de transmitir en *multicast*. Si los nodos se despliegan de forma densa, la interferencia entre ellos puede ser inevitable. Cada nodo puede reducir su potencia de transmisión para reducir la interferencia; sin embargo, se necesita de sincronización entre los nodos. Debe haber sincronización entre la capa de enlace y la capa física, y entre los nodos. Con sincronización, la interferencia en la comunicación se puede reducir al mínimo.

Finalmente, los chips radio con capacidad de *multicast* son útiles para transmitir datos a múltiples nodos al mismo tiempo. Así, sólo los nodos necesarios recibirán la información.

### 2.6.2.1 Elección del ancho de banda

En WSNs, hay tres clases de tecnologías de capa física según el ancho de banda: banda estrecha, espectro ensanchado y *ultra-wideband*. La tecnología de banda estrecha usa un ancho de banda del orden de la tasa de símbolo. Esta tecnología se centra en la eficiencia de ancho de banda, que es la medida de la tasa de datos sobre el ancho de banda utilizado. Por otra parte, en la tecnología de espectro ensanchado, la señal de banda estrecha es expandida. La función de ensanchado utilizada para determinar el ancho de banda es independiente del mensaje. Esta tecnología tiene la capacidad de reducir la potencia y permitir una comunicación con eficacia, esto es, con robustez ante interferencias y multitrayecto. Comparado con la tecnología de espectro ensanchado, *ultra-wideband* utiliza mayor ancho de banda, del orden del GHz. *Ultra-wideband* expande su señal sobre un ancho de banda tal que la interferencia producida a otros nodos es insignificante. Al igual que la tecnología de espectro ensanchado, *ultra-wideband* permite comunicaciones utilizando baja potencia de transmisión.

Varios estudios muestran que las tecnologías de espectro ensanchado cumplen los requerimientos de las WSNs mejor que las de banda estrecha. La tecnología de banda estrecha se centra en optimizar la eficiencia en ancho de banda y es menos robusta ante interferencias (excepto si se utiliza la técnica de *frequency hopping*), mientras que tanto la tecnología de espectro ensanchado como la *de ultra-wideband* combinan ancho de banda con ahorro energético.

Según el tipo de tecnología de espectro ensanchado, la sincronización puede ser beneficiosa debido a las propiedades de correlación de la secuencia pseudoaleatoria utilizada en el proceso de ensanchado. En cuanto a la capacidad *multicast*, los sistemas de banda estrecha no están diseñados para llevar a cabo esta tarea. En cambio, los sistemas de espectro ensanchado pueden realizar esta tarea con una asignación correcta de códigos pseudo-aleatorios. *Ultra-wideband* tiene muchas características atractivas, pero comparado con tecnologías de espectro ensanchado, tiene sus propias restricciones y limitaciones.

### 2.6.2.2 Esquema de modulación

El esquema de modulación usado por el transceptor radio puede repercutir de forma sensible en el consumo de energía de un nodo. Esquemas de modulación eficientes son necesarios para reducir el consumo de energía.

La modulación multinivel transmite símbolos de un conjunto de  $M$  formas de onda diferentes, mientras que la modulación binaria usa únicamente dos formas de onda distintas. En la modulación  $M$ -aria se transmiten bits por muestra. En general, la modulación multinivel requiere de circuitería más compleja y mayor energía. Sin embargo, está demostrado que la modulación  $M$ -aria es más eficiente energéticamente que la modulación binaria cuando el tiempo de encendido del transceptor es pequeño y la potencia de transmisión RF es baja. En otra comparación, para un valor grande de  $M$ , la modulación M-FSK (*Frequency Shift keying*) es más eficiente que la M-PSK (*Phase Shift keying*) y M-QAM (*Quadrature Amplitude Modulation*), cuando  $M$  es mayor que ocho. Para valores menores de  $M$ , la modulación M-FSK no es tan eficiente porque se requiere de mayor potencia para conseguir la misma tasa de error de bit que en la modulación M-PSK y M-QAM. Sin embargo, para valores grandes de  $M$ , la relación señal a ruido requerida (SNR) en la modulación M-FSK crece lentamente, haciéndola muy eficiente en términos de energía.

En definitiva, la capa física en una WSN debe ser eficiente energéticamente. Dado que el diseño de la capa física comienza en el diseño del transceptor radio, la elección de éste compromete el rendimiento del resto de capas de protocolo. El trabajo futuro se enfoca en nuevas innovaciones en el diseño de transceptores radio de bajo consumo con tecnologías emergentes, explorando las técnicas de *ultra-wideband* como alternativa para la comunicación, creando esquemas de modulación simples para reducir la sincronización y el coste de energía, buscando siempre la potencia de transmisión óptima con el fin de construir protocolos y algoritmos más eficientes.

### 2.6.3 Capa de Enlace

La capa de enlace de datos se ocupa de la transferencia de datos entre dos nodos que comparten el mismo enlace. Dado que la red es inalámbrica, para conseguir una transferencia eficaz de datos, hay una necesidad de controlar el acceso al medio y gestionarlo. El diseño del protocolo o capa MAC debe tener en cuenta los siguientes atributos: eficiencia energética, escalabilidad, sincronización de trama, imparcialidad, utilización de ancho de banda, control de flujo y control de errores.

Los mecanismos propuestos para permitir el acceso al medio están basados en la asignación fija y el acceso al azar. Esquemas MAC basados en demanda pueden resultar inapropiados en WSNs debido a la gran carga de cabeceras que se introducen en los mensajes, así como al retardo de configuración de los enlaces.

La detección y corrección de errores se ofrecen en la capa de enlace así como en la capa de transporte. Una de las técnicas de detección de errores ampliamente utilizadas es CRC (*Cyclic Redundancy Check*). En WSNs CRC funciona de la siguiente manera. El transmisor y el receptor primero deben ponerse de acuerdo sobre el tamaño del bloque de datos antes de la transmisión. El transmisor divide un paquete procedente de la capa de red en bloques de datos del tamaño acordado que serán reconstruidos nuevamente en el receptor. Para detección de errores es usual usar un CRC de 8 bits. Así, se empaquetan en una trama los bloques de datos con el CRC correspondiente y se envía al receptor, que al recibir la trama, identifica si el bloque de datos contiene errores. Si hay bloques erróneos, el receptor iniciará el proceso de recuperación para recuperar aquellos bloques erróneos después de recibir un cierto número de tramas.

Las técnicas de recuperación en WSNs incluyen técnicas como ARQ (*Automatic Repeat Request*), FEC (*Forward Error Correction*), HARQ (*Hybrid ARQ*), SpaC (*Simple Packet Combining*) y MRD (*Multi-radio Diversity*). ARQ emplea asentimientos y temporizadores para notificar al transmisor. La notificación puede ser de la forma de un asentimiento positivo (ACK) o negativo (NACK). El transmisor que reciba un NACK o la expiración de un temporizador retransmitirá la trama de datos. Una restricción de la técnica ARQ es que está limitada a la detección de errores de trama. Una trama entera tiene que ser transmitida de nuevo con que haya un sólo bit erróneo.

Por otro lado, FEC disminuye el número de retransmisiones. El transmisor añade mayor cantidad de datos redundantes en cada trama para que el receptor pueda detectar y corregir errores. La ventaja de FEC es que se reducen las retransmisiones y se evita el tiempo de espera de transmisión del asentimiento y la nueva transmisión de trama.

HARQ es una variación del método ARQ. En HARQ se combinan los métodos ARQ y FEC. Hay dos tipos de esquemas HARQ: tipo I y tipo II. El tipo I incluye tanto bits de detección como de corrección de errores en cada paquete transmitido y utiliza un código de corrección para subsanar errores. El tipo II transmite los bits de detección de error o la información FEC junto con los datos. Si se detecta un error en el primer paquete, se esperará al segundo paquete que contiene las paridades FEC para corregir el error. Si los errores todavía persisten, los paquetes se combinan para corregir los errores.

SPaC almacena el paquete con errores en el receptor y espera a la retransmisión. En vez de retransmitir el paquete original, el transmisor envía los bits de paridad. Al recibir el paquete, el receptor lo combina con el paquete almacenado para recuperar los errores.

MRD usa dos técnicas para reponerse de los errores. La primera técnica es la combinación de múltiples tramas erróneas para poder corregir errores evitando las retransmisiones. La segunda técnica utiliza el esquema RFA (*Request for Acknowledgement*) para recuperar el paquete.

En resumen, el diseño del protocolo MAC en una WSN está sujeto a varias restricciones como la energía, topología y cambios en la red. La reducción al mínimo de la energía para aumentar el tiempo de vida de la red es su objetivo fundamental. El diseño del protocolo MAC debería prevenir el gasto de energía debido a colisiones de paquetes, excesivas retransmisiones, control de cabeceras y escucha ociosa. También debería adaptarse a la topología y cambios de red de manera eficiente. Una gran variedad de protocolos MAC se han propuesto para conseguir optimización en la utilización de canal, eficiencia energética y evitar colisiones.

A pesar de ello, queda mucho trabajo por hacer en lo referente a la optimización del funcionamiento del sistema. La interacción entre capas es un área que necesita ser explorada más profundamente, ya que puede reducir las cabeceras en cada una de las capas, así como minimizar el consumo de energía. Las interacciones con la capa MAC pueden proveer a otras capas de información de control de congestión y mejorar la selección de la ruta. Muchos de los protocolos MAC existentes manejan estudios de rendimiento estáticos, pero hay todavía una carencia en la literatura para comparar estos protocolos en una red móvil. En definitiva, mejorando el protocolo MAC se puede mejorar considerablemente la fiabilidad en la comunicación y el consumo de energía.

#### 2.6.4 Capa de Red

La capa de red se encarga del encaminamiento de los datos a través de otros nodos de la red desde la fuente al destino. Este enfoque evita que la fuente deba emplear una potencia de transmisión elevada para alcanzar al destino.

Los protocolos de encaminamiento son los responsables de elegir el camino más adecuado entre la fuente y su destino. En WSNs, los protocolos de encaminamiento deben encontrar un buen compromiso entre varios parámetros de rendimiento, como son la tasa de entrega, la latencia y el consumo energético, si bien estos protocolos en WSNs se diferencian de los protocolos de encaminamiento tradicionales en varios aspectos.

En primer lugar, los nodos sensores no tienen direcciones IP, debido por ejemplo, a las limitaciones existentes de memoria en los nodos, que impiden el almacenamiento de tablas de encaminamiento (la topología de la red puede variar debido a fallos en los nodos) y los nodos se despliegan generalmente de forma aleatoria y en grandes cantidades. Por lo que no es válido un protocolo de encaminamiento basado en IP en una WSN.

Por otra parte, la escalabilidad es particularmente importante en redes de sensores, dado el número elevado de nodos que pueden existir. Así pues, los protocolos de la capa de red deberían manejar las comunicaciones entre muchos nodos y propagar los datos a la estación base. Los protocolos deben tener en cuenta las limitaciones de recursos de red como son la energía, el ancho de banda, la memoria y capacidades computacionales. Teniendo en cuenta estas restricciones, el tiempo de vida de la WSN se puede prolongar. También, el protocolo debería tener en cuenta cuestiones como tolerancia a fallos, imparcialidad y seguridad.

Un objetivo importante en WSNs es que el encaminamiento pueda estar basado en los datos. El encaminamiento basado en los datos también se conoce como encaminamiento basado en atributos ya que los usuarios están más interesados en un atributo del fenómeno y no tanto en la medida de un nodo determinado. De esta manera, se podría realizar una petición a la red para la ejecución de tareas de sensado en los nodos sensores. Otra posibilidad sería que los propios nodos propagasen sus medidas en difusión anunciando los datos disponibles y que esperasen una petición por parte de los nodos interesados.

El nivel de red también tiene la importante función de proporcionar interconexión con redes externas o con otras redes de sensores, sistemas de control y de comandos, e Internet. Las WSNs requieren conectividad con otras redes por varias razones que incluyen el control remoto, el acceso remoto a los datos recogidos, actualizaciones de software de los nodos sensores, etc.

Si bien la principal red con la cual resulta interesante poder conectar una WSN es Internet. Existen varias propuestas para lograr tal objetivo. Sin embargo, muchas WSNs emplean protocolos especiales, no estandarizados. En tales condiciones, no es posible conectar una WSN con una red IP. Se han propuesto dos opciones para resolver este problema: el uso de la arquitectura DTN (que no está teniendo una aceptación mayoritaria en la comunidad interesada en las WSNs) y el empleo de un *proxy* o pasarela de traducción de protocolos entre la WSN y la red IP. Un enfoque distinto, que está adquiriendo relevancia a medida que la tecnología evoluciona, es conectar directamente la WSN a una red IP mediante el uso de una pila IP en los propios nodos sensores. Este esquema facilita además la inclusión de WSNs en el entorno IMS (*IP Multimedia Subsystem*).

Muchos protocolos de encaminamiento han sido propuestos en WSNs. Las características en las que se centran son la eficiencia energética y flujos de tráfico. Estudios futuros de investigación deberían estar dirigidos a la seguridad, QoS y movilidad del nodo. Además, se deberían analizar los estudios experimentales en cuanto a la seguridad aplicada a protocolos de encaminamiento en WSNs. Por otro lado, hay poca investigación en QoS en WSNs. QoS garantiza el retardo extremo a extremo y un encaminamiento energéticamente eficiente. En aplicaciones donde los nodos son móviles, son necesarios nuevos protocolos de encaminamiento para manejar cambios de topología frecuentes y llevar a cabo entregas de datos de forma fiable.

### 2.6.5 Capa de Transporte

La capa de transporte es la encargada de ofrecer fiabilidad extremo a extremo. En algunos casos, también ofrece control de congestión. En WSNs, el concepto de fiabilidad puede diferir del concepto tradicional, puesto que algunas aplicaciones requieren fiabilidad en la detección de un evento, de modo que no es necesario garantizar fiabilidad para la transmisión de cada paquete.

Por otra parte, el control de congestión puede ser realizado a cada salto, lo cual permite ahorrar energía y ancho de banda. Una solución a nivel de transporte en WSNs es plantear el uso de protocolos de transporte tradicionales (por ejemplo TCP). Pese a que TCP ofrece ventajas en cuanto a interoperabilidad, exhibe un rendimiento subóptimo en WSNs y su implementación es compleja. Una posibilidad para WSNs conectadas a Internet es emplear un elemento de traducción de protocolos, de modo que se pueda usar un protocolo adecuado de transporte para el tramo WSN y TCP en el tramo del resto de Internet.

Dadas las desventajas de TCP en WSNs, se han diseñado numerosos protocolos de transporte para estos entornos. Se pueden clasificar en tres grupos:

- Protocolos que ofrecen control de congestión
- Protocolos que ofrecen fiabilidad
- Protocolos que ofrecen ambos servicios.

Los protocolos de control de congestión para WSNs como CODA (*Congestion Detection and Avoidance*), Fusion ó CCF (*Congestion Control and Fairness*) emplean técnicas como la monitorización de las colas para detectar situaciones de congestión. Tras detectar la congestión, los protocolos citados notifican tal situación a la fuente mediante mensajes y/o *flags* activados en mensajes que son transmitidos a la fuente. Se pueden emplear distintas técnicas para ajustar la tasa de transmisión de información, como AIMD (*Additive Increase Multiplicative Decrease*), o bien ajustes exactos de tasa.

Los protocolos que ofrecen fiabilidad, por ejemplo RMST (*Reliable Multi-Segment Transport*), RBC (*Reliable Bursty Convergecast*) o PSFQ (*Pump Slowly, Fetch Quickly*) emplean distintas técnicas para la detección de pérdidas de datos, como el uso de NACKs, ofreciendo la posibilidad de recuperar los errores en cada salto, o bien extremo a extremo. PSFQ ofrece fiabilidad en sentido *downstream*, esto es, de nodo sumidero a los nodos sensores. RBC y PSFQ ofrecen fiabilidad en sentido *upstream*.

Algunos protocolos como STCP (*Stream Control Transmission Control*) y ESRT (*Event to Sink Reliable Transport*) ofrecen fiabilidad y control de congestión. STCP ofrece mecanismos de fiabilidad variable según los requisitos de las aplicaciones, mientras que ESRT se emplea para garantizar la fiabilidad a nivel de evento.

## 2.7 Estándares

El valor de las redes inalámbricas de sensores se basa en su bajo coste y su distribución en grandes cantidades. Para lograr las economías de escala necesarias para alcanzar un mercado de bajo coste, algunos elementos de las redes inalámbricas de sensores deben ser estandarizados, de manera que los productos de muchos fabricantes puedan interoperar. Esta sinergia añade utilidad a las redes inalámbricas de sensores y, por tanto, fomentará su uso.

Con este fin, se están realizando esfuerzos para estandarizar las distintas capas de los protocolos de comunicación de redes de sensores inalámbricos, incluyendo la carga útil de los datos enviados por sus sensores. El éxito de las redes inalámbricas de sensores como una tecnología se basa en el éxito de estos esfuerzos de estandarización para unificar el mercado, dando lugar a un gran número de dispositivos de bajo coste, interoperables, y evitando la proliferación de la propiedad y de protocolos incompatibles que, aunque tal vez óptimos en sus nichos de mercado, limitan el tamaño del mercado global de sensores inalámbricos.

Además, hoy en día no existe un estándar de facto, ni en protocolos, ni en hardware, ni en representación de datos. Esto es debido por una parte, a que las aplicaciones de las redes de sensores son tan amplias que es difícil englobarlas todas. Y por otra parte, a que las tecnologías están en constante cambio y desarrollo. De entre las soluciones existentes, sin embargo, sí hay quienes tienen más fuerza y quienes tienen menos. A continuación se presentan las principales iniciativas en curso:

### 2.7.1 IEEE 802.15.4

El ámbito de aplicación del grupo de tareas IEEE 802.15.4, tal como se define en su primera solicitud de autorización del proyecto, es «definir las especificaciones de PHY y MAC para conectividad inalámbrica de baja tasa de datos con dispositivos fijos, portátiles y móviles sin requisitos de consumo de batería que normalmente operan en el Espacio de Operación Personal (POS) de 10 metros». Además, el objetivo del proyecto es «proporcionar una norma de baja complejidad, bajo coste, pequeño consumo de energía y conectividad inalámbrica de baja velocidad de datos entre dispositivos. La tasa de datos en bruto será lo suficientemente alta (un máximo de 200 kbps) para satisfacer un conjunto de necesidades simples, pero adaptable a las necesidades de los sensores y de automatización (10 kbps o menos) para comunicaciones inalámbricas». Las tasas de datos máxima y mínima se plantearon más tarde a 250 y 20 kbps, respectivamente.

Este conjunto de objetivos requiere que el estándar IEEE 802.15.4 sea extremadamente flexible. A diferencia de los protocolos que han sido diseñados para una sola aplicación, tales como IEEE 802.11, el estándar IEEE 802.15.4 soporta una variedad casi infinita de posibles aplicaciones en el POS. Estas aplicaciones varían desde requerimientos de alta tasa de transferencia de datos y relativamente baja latencia de mensaje, tales como teclados inalámbricos, ratones, y joysticks, a las que requieren muy baja tasa de transferencia y son capaces de tolerar una latencia mayor, tales como la agricultura inteligente y aplicaciones de detección del entorno. El estándar IEEE 802.15.4 soporta conexiones tanto en estrella como punto a punto, por lo que es capaz de soportar una amplia variedad de topologías de red y de algoritmos de enrutamiento. Cuando se utiliza seguridad, el paquete de seguridad AES-128 es obligatorio.

El estándar emplea balizas, aunque su uso es opcional. El período de baliza es variable, de modo que se puede hacer para cada aplicación un equilibrio óptimo entre la latencia del mensaje y el consumo de energía de cada nodo de la red. Las balizas pueden ser omitidas para aplicaciones que tienen limitaciones de ciclo de servicio, como puede ocurrir en las redes en la banda de 868 MHz o aplicaciones que requieren nodos de red con recepción constante. El acceso al canal está basado en la contención a través de mecanismo de acceso múltiple CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*); la baliza es seguida por un período de acceso de contención (CAP) para los dispositivos que intentan acceder al canal. La duración del CAP es ajustable a una fracción del período entre balizas. Para atender las necesidades de las aplicaciones que requieren mensajes de baja latencia, el estándar admite el uso de ranuras de tiempo garantizadas (GTSS), que reservan el tiempo del canal para dispositivos individuales sin la necesidad de seguir el mecanismo de acceso

## CSMA-CA.

El estándar tiene un campo de direcciones de 16 bits, sin embargo, la norma incluye también la capacidad de enviar mensajes con direcciones extendidas de 64 bits, lo que permite un número casi ilimitado de dispositivos que se pueden colocar en una única red. La transmisión de mensajes puede ser asentida mediante reconocimientos positivos, tras cada trama transmitida (con la excepción de las balizas y los reconocimientos a sí mismos) se puede recibir un reconocimiento explícito dando lugar a un protocolo fiable. La sobrecarga asociada con los reconocimientos explícitos es aceptable dada la baja tasa de datos típica de redes inalámbricas de sensores. El uso de los reconocimientos es opcional con cada trama transmitida, sin embargo, se soporta el uso de técnicas de reconocimiento pasivas.

El estándar IEEE 802.15.4 incorpora muchas características diseñadas para minimizar el consumo de energía de los nodos de la red. Además de la utilización de períodos largos de baliza y un modo de extensión de vida de la batería, el período activo de un nodo con baliza se puede reducir, permitiendo que el nodo se duerma entre balizas.

La coexistencia de otros servicios que usan bandas sin licencia con los dispositivos IEEE 802.15.4 fue también un factor importante en el diseño del protocolo, y es evidente en muchas de sus características. Por ejemplo, es necesaria la selección dinámica de canales; cuando se puede interferir a otros servicios que se encuentran trabajando en un canal que está siendo utilizado por una red IEEE 802.15.4, el nodo de red que controla la misma (coordinador de red de área personal (PAN)) explora otros canales disponibles para encontrar un canal más adecuado. En esta exploración, se obtiene una medida de la energía de cada canal alternativo y, a continuación, utiliza esta información para seleccionar un canal adecuado. Este tipo de escaneo también se puede usar antes del establecimiento de una nueva red. Antes de cada una de las transmisiones (excepto tramas baliza o de reconocimiento), cada nodo de red IEEE 802.15.4 debe realizar dos evaluaciones de canal (CCAs) como parte del mecanismo CSMA-CA para garantizar que el canal está desocupado antes de la transmisión.

Un byte de indicación de la calidad del enlace (LQI) se adjunta a cada trama recibida por la capa física antes de que sea enviada a la capa de control de acceso al medio. El nodo receptor utiliza esta información para diversos fines, dependiendo del diseño de la red.

Para maximizar la utilidad de la norma, el grupo de tarea IEEE 802.15.4 tenía que equilibrar el deseo de permitir nodos de red pequeños, de bajo coste y baja potencia con el deseo de producir una norma que se pudiera utilizar por una amplia variedad de aplicaciones en el mercado. El estándar resultante incluye tres tipos de funcionalidades de nodo de red:

**Coordinador PAN:** es el nodo que inicia la red y es el principal controlador de la misma. El coordinador PAN puede transmitir balizas y se puede comunicar directamente con cualquier dispositivo en su rango. Dependiendo del diseño de red, es posible que tenga suficiente memoria para almacenar información sobre todos los dispositivos en la red, y debe tener memoria suficiente para almacenar la información de enrutamiento según lo exija el algoritmo empleado por la red.

**Coordinador:** puede transmitir balizas y se puede comunicar directamente con cualquier dispositivo a su alcance. Un coordinador puede convertirse en un coordinador PAN en el caso de empezar una nueva red.

**Dispositivo:** no genera balizas y sólo puede comunicarse directamente con un coordinador o coordinador PAN.

Estas tres funciones son incorporadas en dos tipos de dispositivos diferentes:

**Dispositivo de función completa (FFD):** puede operar en cualquiera de las tres funciones de red (coordinador PAN, coordinador, o dispositivo). Debe tener memoria suficiente para almacenar la información de enrutamiento que imponga el algoritmo empleado por la red.

**Dispositivo de función reducida (RFD):** es un dispositivo de muy bajo coste, con requisitos de memoria mínimos. Sólo puede funcionar como un dispositivo de red.

Similar a todos los estándares inalámbricos IEEE 802, el estándar IEEE 802.15.4 normaliza sólo las capas físicas y las MAC. El estándar IEEE 802.15.4 incorpora dos capas físicas:

La banda inferior: 868-868,6 MHz (para Europa), además de los 902-928 MHz (para la mayor parte de América y la Cuenca del Pacífico). La banda superior: 2,400-2,485 GHz (en todo el mundo).

El mayor problema de este estándar es que está definido para WPAN y no es específico para redes de sensores.

Por lo tanto, aunque abarca una gran cantidad de posibilidades y métodos de comunicación, no resulta eficiente en las aplicaciones más restrictivas y exigentes.

Esta dificultad se intenta solventar con el estándar 802.15.4a-2007, que añade nuevas posibilidades a la capa física, como UWB (*Ultra-wideband*), CSS (*Chirp Spread Spectrum*) y DSSS (*Direct Sequence Spread Spectrum*). En la Figura 2-6 se puede observar la pila de protocolos que propone dicho estándar, sobre el que está montado Zigbee.

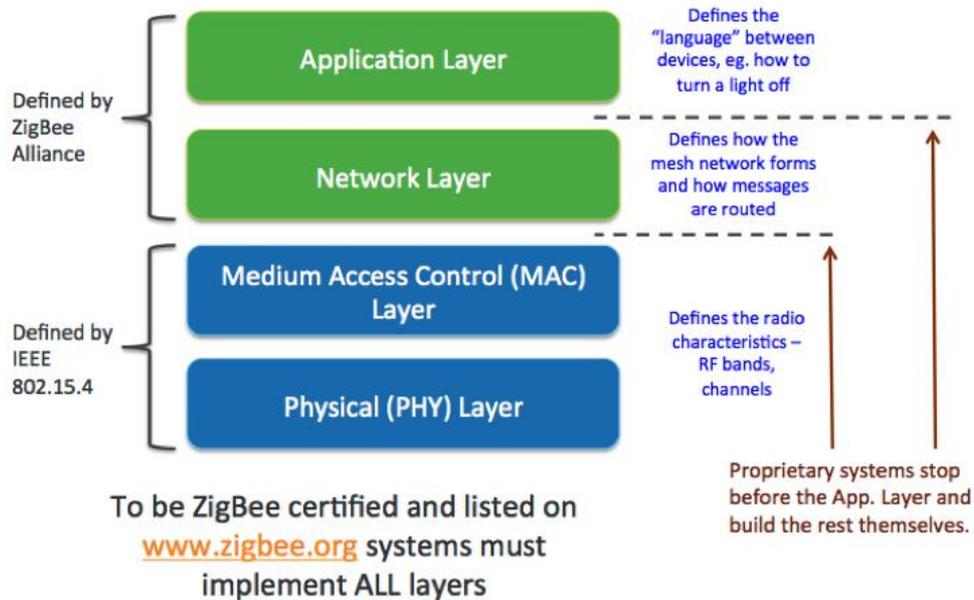


Figura 2-6. Capas de un nodo sensor y estándar asociados.

## 2.7.2 ZigBee

El estándar IEEE 802.15.4 no estandariza las capas más altas del protocolo de comunicación, incluyendo las capas de red y la aplicación. Para asegurar la interoperabilidad entre los dispositivos que funcionen con el estándar IEEE 802.15.4, el comportamiento de estas capas debe estar especificado. La creación de tal especificación se ha incorporado por *ZigBee Alliance* (Alianza ZigBee), un consorcio de la industria de fabricantes de chips, fabricantes OEM (*Original Equipment Manufacturer*), proveedores de servicios y usuarios en el mercado de las redes de sensores inalámbricas, muchos de los cuales trabajaban para desarrollar la norma IEEE 802.15.4. Además de la creación de especificaciones de la capa superior mediante perfiles de aplicación (ver Figura 2-7), ZigBee Alliance es también el apoyo de comercialización y cumplimiento del IEEE 802.15.4, de manera análoga a la relación entre Wi-Fi Alliance y el estándar WLAN IEEE 802.11 (comercializado como "Wi-Fi").

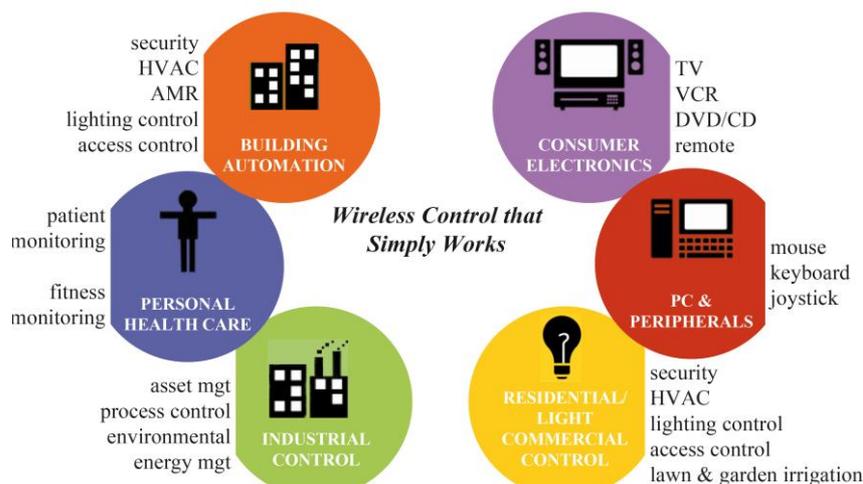


Figura 2-7. Aplicaciones de ZigBee.

Esta especificación inalámbrica es de baja potencia, bajo coste y baja velocidad de transferencia de datos, destinada a electrodomésticos, juguetes, aplicaciones industriales y otras similares. Al igual que ocurre con el estándar 802.15.4-2006, está orientado a WPAN y sufre los mismos defectos.

Las principales ventajas de esta tecnología están en su bajo consumo de energía y su fácil integración a través de nodos. En modo SLEEP su consumo es del orden de los microamperios, mientras que para el modo ON, este alcanza las centenas de miliamperios aproximadamente. Cada sensor solo necesita encenderse cuando tenga que enviar el dato a otro dispositivo (modo SLEEP), mientras que los encaminadores y coordinadores si necesitan estar encendidos todo el tiempo (modo ON). Estas consideraciones son según la topología de red escogida.

Esta tecnología utiliza la banda de frecuencias ISM (Industrial, Scientific, Medical) para uso industrial, científico y médico con frecuencias de 2.4 GHz en todo el mundo, siendo esta la más utilizada por ser de uso libre no licenciada. Además usa 868 MHz en Europa y 915 MHz en EE.UU.

En la interfaz aire utiliza el método DSSS, que consiste en usar la técnica espectro ensanchado, a través de un código de secuencia pseudoaleatoria conocido como “chipping sequence”, el cual se transmite a máxima velocidad (chip rate). El código generado es usado directamente para modular una portadora, de aquí el nombre de secuencia directa, pues codifica el dato que está siendo transmitido. Esta técnica ha demostrado ser útil para evitar interferencias, sobre todo en la banda libre de 2.4 GHz (ISM).

Es muy común en esta banda que dos dispositivos, aunque no tengan exactamente la misma frecuencia, el ancho de banda de transmisión de ambas señales se superpongan. Para sortear estos problemas de interferencias, el estándar IEEE 802.15.4, utiliza otra técnica de acceso conocida como FDMA.

El acceso con el método FDMA (Frequency Division Multiple Access) es usado por otras tecnologías inalámbricas. Este consiste en dividir la banda libre ISM en 16 canales con un ancho de banda de 2 MHz que no se superponen, los cuales tienen una separación de 5 MHz. Su numeración va desde el canal 11 al canal 26. Este método es empleado por otras tecnologías como los sistemas celulares o la IEEE 802.11 b y g. Al menos dos de estos canales, específicamente el 15 y el 20 no se interfieren con los canales 1, 6 y 11 de la 802.11 (Wi-Fi).

Para evitar colisiones utiliza el protocolo CSMA-CA. A diferencia de las redes Ethernet (IEEE 802.3), que pueden escuchar los otros dispositivos durante sus propias transmisiones, cada dispositivo “se anuncia” en el medio antes de realizar su transmisión. De esta manera se evitan colisiones, pues cada equipo en la red va a conocer cuando se producen, de tal forma que un dispositivo va a esperar un tiempo aleatorio adicional antes de intentar de nuevo la transmisión.

En cuanto a la seguridad, ZigBee usa uno de los algoritmos más completos. Para este fin el estándar IEEE 802.15.4 utiliza la encriptación AES de 128 bits (Advanced Encryption Standard), que es un estándar desarrollado para el gobierno de EE.UU y fue adoptado desde el año 2001 como estándar oficial aprobado por la NIST (National Institute of Standards and Technology). Esto hace que las redes de sensores instaladas con

esta tecnología tengan la ventaja de enviar los datos de manera segura, sin que otros sistemas tengan la posibilidad de acceder sin autorización.

ZigBee tiene como gran ventaja su bajo consumo de energía, su bajo coste, su fácil implementación y por ser un protocolo basado en estándares, también su universalidad, es decir, gran cantidad de proveedores en todas las regiones.

La posibilidad de implementar redes con distintas topologías y su capacidad para adaptarse fácilmente a distintas interfaces a través de pasarelas permite a ZigBee integrarse en diferentes ambientes como el control de procesos en la industria, control de tráfico en las ciudades, monitorización de signos vitales en medicina, atención y prevención de desastres, control perimetral de áreas en aplicaciones militares entre muchas otras.

Su gran desventaja está en la baja velocidad de transmisión de datos, comparada con otras tecnologías inalámbricas como el Bluetooth, lo que por el momento no permite utilizarla en aplicaciones de uso masivo de datos como el vídeo.

Las aplicaciones actuales y las que están en desarrollo hacen pensar que el uso de esta tecnología se incrementará de manera exponencial en los próximos años por su alta versatilidad.

### 2.7.3 WirelessHART

WirelessHART surge como una adaptación de la especificación HART ya existente, que se creó en EE.UU. para la monitorización de sensores en entornos industriales. Esta iniciativa especifica perfiles y casos prácticos en los que se puede aplicar directamente el control inalámbrico. La pila de protocolos de HART se puede observar en la Figura 2-8.

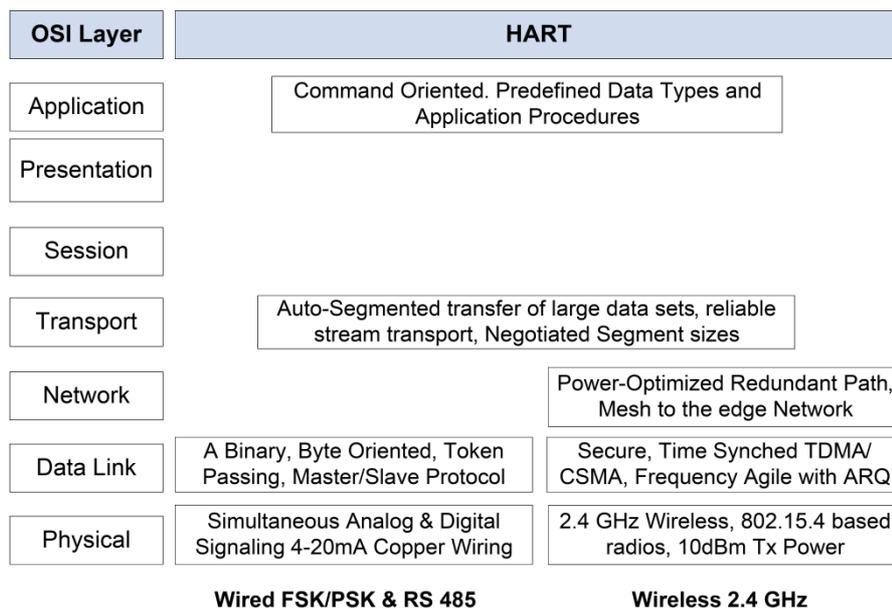


Figura 2-8. Pila de protocolos de HART.

La mayor diferencia con respecto a ZigBee es el acceso al medio, realizado mediante TDMA, ahorrando tiempo y energía en los nodos sensores. La topología también es diferente, ya que en esta especificación la red es mallada. De esta manera consigue una red fiable y eficiente energéticamente. Además, todos los nodos pueden encaminar y son iguales, haciendo mucho más sencilla su instalación y consiguiendo mayor flexibilidad.

Para la generación de mensajes de información, debido a la naturaleza de la aplicación, tiene su propio protocolo (Smart Data) que permite enviar información sólo cuando se registra un cambio en la medición del sensor.

No sólo eso, esta pila de protocolos es compatible con IEEE 802.15.4 y forma parte de un nuevo estándar, el

SP100.11, creado por ISA (Instrumentation, System and Automation Society) para el entorno industrial. En vez de normalizar todos los elementos del sistema, SP100.11 especifica sólo los niveles superiores de la pila, con varias implementaciones posibles a nivel inferior.

#### 2.7.4 ISA 100.11

El estándar ISA100.11 está diseñado para aplicaciones de monitorización y procesos de automatización de baja tasa de datos. Define las especificaciones para las capas OSI de seguridad y gestión. El estándar se centra en un consumo de energía bajo, escalabilidad, infraestructura, robustez e interoperabilidad con otros dispositivos inalámbricos. Las redes ISA100.11 usan la banda de 2.4 GHz y salto de canal (*channel hopping*) para aumentar la fiabilidad y reducir al mínimo las interferencias. Soporta topología en malla y en estrella.

ISA100.11 también proporciona funcionalidad de seguridad simple, flexible y escalable.

#### 2.7.5 6LoWPAN

IPv6-based Low Power Wireless Personal Area Networks permite la comunicación de paquetes IPv6 sobre una red basada en el estándar IEEE 802.15.4.

Este protocolo trata de adaptar a las redes de sensores el protocolo IPv6 para conseguir compatibilidad entre Internet y estas redes, comunicando entre sí miles de sistemas, aprovechando las ventajas que trae consigo IPv6.

Los dispositivos de bajo consumo se pueden comunicar directamente con dispositivos IP utilizando el protocolo IP. Usando 6LoWPAN, los dispositivos de bajo coste tienen todas las ventajas de comunicación y gestión IP.

El estándar proporciona una capa de adaptación, nuevo formato de paquete y gestión de direccionamiento. Como el tamaño de paquete IPv6 es mucho más grande que el tamaño de la trama IEEE 802.15.4 es necesaria una capa de adaptación. La capa de adaptación lleva a cabo la funcionalidad de compresión de cabeceras. Con esta, se crean paquetes de menor tamaño que encajan en una trama IEEE 802.15.4. Por otro lado el mecanismo de gestión de direcciones se encarga de la formación de direcciones de dispositivo para la comunicación.

6LoWPAN está diseñado para aplicaciones en las que los dispositivos requieren conexión a Internet.

#### 2.7.6 Bluetooth Low Energy

Bluetooth low energy o smart bluetooth es una tecnología de comunicación inalámbrica diseñada para dispositivos de bajo coste, bajo consumo energético y comunicación de corto rango. Bluetooth LE permite la comunicación entre dispositivos alimentados con baterías de pequeñas dimensiones tales como relojes, teclados inalámbricos y sensores que se conectan a dispositivos como ordenadores personales o teléfonos celulares.

Bluetooth LE funciona en la banda de 2.4 GHz con una tasa de datos de 1 Mbps. La distancia de enlace entre dispositivos es de 5-10 m. Bluetooth LE está pensado para operar junto a Bluetooth y permite hacer a los dispositivos más pequeños y más eficientes energéticamente, aprovechando la existencia de la comunicación Bluetooth con un consumo de energía más bajo.

## 2.8 Seguridad

La naturaleza de las comunicaciones inalámbricas abre la puerta a multitud de ataques de seguridad. Las WSNs no son la excepción a este hecho. Por ejemplo, la introducción de mensajes malintencionados en una WSN para un entorno industrial, o para un entorno doméstico, puede causar problemas graves.

Los mecanismos de seguridad como el cifrado y autenticación o integridad se basan en métodos criptográficos. Estos mecanismos requieren capacidad de proceso y contribuyen de manera significativa al tamaño del código de las implementaciones. Por tanto, existe un compromiso entre los mecanismos de seguridad y las capacidades hardware de los dispositivos, las cuales son limitadas en WSNs.

El estándar IEEE 802.15.4-2003 por ejemplo define ocho suites de seguridad, con distintos servicios y

prestaciones de seguridad en cada una (ver Tabla 2-2). El algoritmo criptográfico empleado en estas suites es el AES como ya se comentó en apartados anteriores.

ZigBee ofrece servicios de seguridad en todas sus capas (desde la capa física a la de aplicación). Entre otros, ZigBee ofrece servicios de seguridad en cada capa equivalentes a las ocho suites de seguridad de IEEE 802.15.4. Una característica importante de ZigBee es que ofrece una solución para la gestión de claves (servicio que no ofrece IEEE 802.15.4) y la inclusión de forma segura de dispositivos en una red.

La seguridad en una red ZigBee se basa en el uso de claves de enlace (para la comunicación entre dos entidades) y una clave de red (para las comunicaciones *broadcast*). No existe aún la funcionalidad equivalente para 6LoWPAN. Otras soluciones tecnológicas (por ejemplo Wavenis, Z-Wave, INSTEON) soportan el uso de algoritmos de cifrado.

Tabla 2-2. Suites de seguridad de IEEE 802.15.4.

Grupo de seguridad	Nombre de la suite	Servicio de seguridad			
		Control de acceso	Cifrado	Autenticación e integridad de trama	Protección frente a retransmisión
Sin seguridad	Sin seguridad	No	No	No	No
Sólo cifrado	AES-CTR	Sí	Sí	No	Sí
Sólo autenticación /integridad	AES-CBC-MAC-128	Sí	No	Sí	No
	AES-CBC-MAC-64	Sí	No	Sí	No
	AES-CBC-MAC-32	Sí	No	Sí	No
Cifrado más autenticación /integridad	AES-CCM-128	Sí	Sí	Sí	Sí
	AES-CCM-64	Sí	Sí	Sí	Sí
	AES-CCM-32	Sí	Sí	Sí	Sí

## 2.9 Futuro de las WSNs

De cara al futuro podemos diferenciar cinco áreas de trabajo o investigación importantes relacionadas con las redes de sensores.

La primera de ellas consiste en constatar la realidad de que cada vez más y más sensores están y estarán realizando determinadas funciones en nuestro entorno físico próximo (por ejemplo en la ropa u objetos que podamos llevar encima, como en los terminales móviles, como en la infraestructura próxima que nos rodea: edificios, farolas, en las ciudades). Las redes de sensores pueden ser el catalizador que aglutine toda esa información obtenida, facilite que se pueda poner en contexto y a partir de ella se puedan extraer servicios enriquecedores e innovadores para las personas o empresas.

La segunda línea de trabajo estará centrada en los propios nodos sensores, al objeto de conseguir que su rendimiento sea mejor (capacidad de procesado y de transmisión / recepción de datos) al tiempo que se consuma menos energía. Así, un aspecto importante será el de las fuentes de energía alternativas procedentes del propio entorno que palien los problemas de alimentación de los nodos sensores, que deben ser baratos y deben poder estar, en el peor caso, proporcionando servicios durante años en entornos a veces hostiles o de muy difícil acceso.

Pero el mercado y las oportunidades de negocio en torno a las redes de sensores (como habilitadoras de nuevos servicios) no serán una realidad si no se consigue superar el problema de un ecosistema (tecnologías y protocolos) excesivamente fragmentado en el que no aparezcan estándares de referencia. Por tanto, la tercera línea de actuación debería ir en este sentido.

En cuarto lugar, las soluciones de inter-conexión con el mundo exterior para poner en valor esa información

procedente de las redes de sensores precisan de una evolución de las tecnologías existentes hacia aproximaciones estándar, ampliamente aceptadas, típicamente basadas en el protocolo IP. El IETF (*Internet Engineering Task Force*) está trabajando en ello desde hace tiempo (por ejemplo en los grupos de trabajo 6LoWPAN y ROLL (*Routing Over Low power and Lossy networks*)), y ya se puede decir que existen soluciones de redes de sensores sobre IP que ofrecen un buen rendimiento. De hecho algunas organizaciones de referencia en el sector (que desarrollan sus propios estándares inicialmente no IP), como por ejemplo ZigBee o Z-Wave, ya han anunciado que están evolucionando sus soluciones hacia una pila de protocolos que incluirá IP (*IP stack*). Este será un pilar fundamental sobre el que se pueda construir el “Internet de las Cosas”, cuya visión quedó establecida por la UIT hace algunos años, y que básicamente consiste en vislumbrar que a partir de un determinado momento empezará a haber muchas más máquinas que personas interconectadas (por ejemplo a través de Internet), y gran parte de ellas podrían ser nodos de una red de sensores.

En quinto y último lugar, el desarrollo de una gran cantidad de aplicaciones en torno a la información obtenida desde las redes de sensores debería ser habilitado por la existencia de un marco de trabajo adecuado que facilite el desarrollo de servicios innovadores, y que puede pasar por aproximaciones de *middleware* y plataformas “horizontales”, alineadas con el modelo exitoso de la web 2.0 actual, en contraposición a aproximaciones “verticales” que afrontan nichos de mercado concretos y que ofrecen soluciones propietarias.

A un plazo más largo, se puede pensar en las siguientes generaciones de las redes de sensores en las que los nodos sensores estén embebidos en las cosas y personas. En este punto se puede hablar de las redes de sensores en las superficies, en la materia y en las personas. En este contexto se necesitarán nuevos paradigmas de funcionamiento.



# 3 LOCALIZACIÓN EN INTERIORES

---

En este capítulo se tratan algunos de los aspectos más importantes sobre la localización de objetos y personas en entornos cerrados o de interior y de los problemas asociados a tal fin. Se hablará de los sistemas de localización más utilizados actualmente y cuales de ellos son más apropiados para los ambientes de interior. Por último nos centraremos en las redes inalámbricas de sensores (WSNs) y en las técnicas que se emplean para dar solución al problema que estamos tratando sobre la localización en interiores.

## 3.1 Problemática de la localización en interiores

El diseño de sistemas de localización en interiores presenta desafíos significativos. Por un lado, la localización en interiores se ve dificultada por la no disponibilidad de sistemas de posicionamiento como GPS y su baja precisión. Por otro lado, las difíciles condiciones de propagación a las que se enfrenta la onda de radiofrecuencia en interiores dificultan el establecimiento de enlaces inalámbricos robustos. A continuación enumeramos algunos de estos fenómenos que afectan a la onda de RF:

- **Atenuación por distancia:** la potencia de la señal decrece con la distancia de la forma  $P \propto \frac{1}{d^\alpha}$ , donde  $d$  es la distancia y  $\alpha$  es la tasa de atenuación que sufre la señal conforme se propaga. Si bien en condiciones de espacio libre el coeficiente  $\alpha$  toma el valor 2, en interiores debido a fenómenos como la absorción, alcanza valores mayores, provocando que la precisión, en la medida de potencia, disminuya drásticamente con la distancia.
- **Absorción:** si aparecen obstáculos entre emisor y receptor, la atenuación es mucho mayor que en espacio libre debido al fenómeno de la absorción, esto es, cuando una onda se topa con un obstáculo, este absorbe parte de su energía. La cantidad de energía que se pierde, o se transfiere al obstáculo, es una cantidad normalmente modelada por un factor del tipo  $e^{-\gamma d}$ , donde  $\gamma$  es una constante dependiente del material del obstáculo y  $d$  es la distancia.
- **Reflexión:** una señal propagándose por el espacio libre, al toparse con un obstáculo pierde parte de su energía debido a la absorción, además, si el obstáculo es lo suficientemente grande en comparación con la longitud de onda de la señal, parte de la señal se refleja, del mismo modo que hace la luz al llegar a un espejo. Dependiendo de las características del obstáculo, la señal reflejada puede tener distinta fase que la señal original.
- **Difracción:** cuando la señal se topa con el borde de un obstáculo (como por ejemplo una esquina de la habitación), se forman frentes de onda secundarios en todas las direcciones. Este fenómeno recibe el nombre de difracción. El que haya mayor o menor difracción depende principalmente del material del muro u obstáculo y también, de la amplitud y fase de la onda.
- **Dispersión:** al contrario de lo que ocurre con la reflexión, la dispersión está causada por objetos de dimensiones pequeñas en comparación con la longitud de onda de la señal. Estos causan también multitud de frentes de onda secundarios que se unen a los descritos anteriormente.

Los fenómenos de reflexión, difracción y dispersión son los responsables del multitrayecto, esto es, entre una antena transmisora y otra receptora nunca habrá un único camino por el que se propaga la señal. Todos los

frentes de onda que se forman se suman o restan vectorialmente unos con otros, por lo que la señal percibida en el receptor no proviene únicamente del camino más corto, ni siquiera en el caso de que el emisor y el receptor estén en visión directa ó LoS (*Line of Sight*). El multitrayecto es responsable de que por ejemplo en una zona no haya señal en absoluto y a los pocos centímetros exista una potencia recibida muy grande.

Existen dos tipos de multitrayecto: el especular, debido mayoritariamente a reflexiones, y el difuso, debido a difracciones o dispersiones. El multitrayecto difuso se puede modelar como ruido de fondo, ruido aleatorio de magnitud pequeña en comparación con el nivel de señal. Sin embargo, el multitrayecto especular no es tan aleatorio pero puede ser fatal para la comunicación, ya que puede crear áreas sin señal o de sombra en las que la potencia de la señal recibida es muy reducida. Por ello es de vital importancia controlar este último.

A todo lo anterior, hay que añadir la posibilidad de que existan otras transmisiones en el mismo espectro de frecuencias utilizado en la comunicación de nuestra red, con lo cual la predicción del comportamiento de la señal es inviable, fallando incluso los modelos más completos al pasar de la teoría a la práctica. Esto provoca que el problema de la localización en interiores cobre una dimensión diferente, necesitando de soluciones alternativas.

## 3.2 Tecnologías de localización

### 3.2.1 GPS

El sistema GPS (similar al ruso Glonass, al chino Beidou o al europeo Galileo) se apoya en una constelación de 24 satélites. El principio de funcionamiento de estos sistemas está basado en la medida simultánea de la distancia entre el receptor y, al menos, 4 satélites (el sistema obtiene mayor exactitud en el computo de la localización al usar un mayor número de satélites). Estas distancias se obtienen por medio del retardo temporal que existe desde el momento en que el satélite emite la señal (todos los satélites emiten de forma sincronizada) hasta el momento en que el receptor la recibe, por lo que cada satélite debe estar equipado con relojes de alta precisión. Con esto, el receptor obtiene de cada uno de ellos la posición del satélite emisor y el tiempo de envío de cada muestra recibida. Triangulando las distancias obtenidas a partir de estos datos, se puede obtener la posición absoluta del receptor en la superficie de la Tierra (latitud, longitud, altitud).

Este sistema puede llegar a alcanzar una exactitud superior a 10 metros haciendo uso de más de cuatro satélites, lo que lo convierte en un sistema de localización global de gran utilidad.

Sin embargo, para aplicaciones de localización en interiores deja de tener sentido su uso ya que además de ser necesaria una mayor exactitud, este sistema requiere de línea de visión directa (LOS) entre el receptor y los satélites para poder computar la posición de una forma fiable. Esta circunstancia unida a la gran atenuación que sufre la señal en entornos de interior y dado que esta se recibe muy atenuada en la superficie de la Tierra, provoca que el sistema sea inútil en estos entornos.

Otro aspecto muy importante y deseable en un sistema de localización en interiores es la información final que necesitamos conocer. Hemos comentado que es deseable que la referencia sea relativa y no absoluta como la que proporciona el GPS. Es decir, de cara al usuario final sería deseable conocer si se encuentra en una habitación, una zona de una habitación, etc. y no su posición absoluta (X, Y, Z) ya que no sería de gran utilidad.

### 3.2.2 RFID (Radio Frequency Identification)

Una alternativa para la localización en interiores es el sistema de identificación por radiofrecuencia (RFID). Como su propio nombre indica, es un sistema de identificación, aunque también tiene otras aplicaciones.

Este sistema se basa en etiquetas de radiofrecuencia que contienen una antena que al ser excitada por un transmisor emite una señal. Así, un usuario que necesite conocer su posición en un edificio tendría que tener cerca de él un número de etiquetas de radiofrecuencia. El propio usuario tendría un lector de etiquetas RFID y leyendo las etiquetas cercanas a él, se puede llegar a conocer su posición.

Este método puede localizar de manera satisfactoria, pero es aun un método caro debido a la gran cantidad de

lectores RFID que se necesitan para una localización correcta, ya que el alcance de estas señales es muy reducido. Además, es necesaria la utilización de otras tecnologías de soporte para la implantación de estos sistemas. Por estas razones se buscaron otros métodos de localización con un menor coste de infraestructura.

### 3.2.3 Infrarrojos

En la localización por infrarrojos se utilizan etiquetas (*tags*) que emiten radiación infrarroja en modo difusivo. Se trata de un sistema de detección más que de localización ya que la posición del elemento etiquetado se infiere de la posición fija y conocida de los elementos sensores que lo detectan.

Además, al ser un método de corto alcance (unos dos metros) y requerir enlaces LOS, habría que incluir una cantidad importante de emisores de infrarrojos, con el consiguiente encarecimiento de la instalación.

No obstante, este sistema posee la ventaja de evitar interferencias y falsas detecciones de otros sensores, como ocurre en RF.

### 3.2.4 Telefonía Movil

Otra alternativa posible, que además no necesitaría ningún hardware adicional, pasaría por el uso de un teléfono móvil. De hecho, ya hay operadoras de telefonía móvil que ofrecen la opción de localización vía móvil a sus abonados. Sin embargo, el que los sistemas de este tipo sean incapaces de proporcionar exactitudes superiores a 50 metros los convierte en sistemas inútiles para interiores. Esto se debe a que la localización usando el teléfono móvil (localización por GSM) se basa en la detección de la célula a la que está conectado el móvil por lo que la exactitud depende del tamaño de la célula a la que se encuentre conectado el terminal. Mientras que en zonas urbanas la exactitud puede ser de decenas de metros, en las zonas rurales, donde se necesitan menos células para dar servicio, la exactitud es mucho menor.

Por tanto, esta es una clara desventaja de la tecnología GSM, que la hace totalmente inapropiada para su uso en localización en interiores.

### 3.2.5 Bluetooth

En esta tecnología, el receptor escucha las emisiones de los dispositivos Bluetooth cercanos para hallar las distancias a estos. Triangulando estas distancias y conociendo las posiciones de los dispositivos de referencia se podría obtener la posición del móvil.

Las ventajas de esta tecnología se encuentran en su bajo coste y su amplio despliegue en dispositivos comerciales. Además, en determinadas situaciones, el error cometido puede estar en torno a 1,5 metros, lo que la hace una tecnología atractiva para entornos de interior.

El mayor inconveniente que tiene Bluetooth es que el indicador de RSS (Received Signal Strength) no es preciso debido a la heterogeneidad de hardware en los dispositivos disponibles.

En parte, ello resulta de una definición imprecisa del RSS en el estándar Bluetooth. Además, obtener RSS generalmente implica establecer conexiones entre los dispositivos de forma permanente, lo que origina por un lado un consumo elevado de energía y, por otro, una desconfianza en los usuarios ya que es considerado como inseguro por necesitar configurar los dispositivos en modo conectable. Todo esto sumado al hecho de que el alcance es demasiado corto (lo que significa que se necesitarían demasiados dispositivos para dar cobertura a un edificio) hace que Bluetooth no sea una tecnología adecuada para localización en interiores.

### 3.2.6 WiMAX

La tecnología WiMAX sigue el estándar 802.16. Está pensada para la intercomunicación de áreas muy extensas, de hasta 48 kilómetros de radio, y puede llegar a transmitir hasta 100 Mbps con movilidad y hasta 1 Gbps fijo. Dado un alcance tan elevado, parece difícil su empleo como tecnología de localización en entornos de interior.

### 3.2.7 UWB (Ultra Wide Band)

Actualmente, la tecnología UWB está siendo estudiada para la localización en interiores con resultados muy satisfactorios. Permite un gran alcance y altísimas velocidades de transmisión y, además de tener aplicación en un gran número de campos, parece un candidato perfecto como tecnología de localización en determinadas circunstancias.

La robustez que presenta esta tecnología ante cambios de entorno (cambio de posición de objetos estáticos y movimiento de personas) y su capacidad para conseguir exactitudes del orden de unos pocos metros, convierten a esta tecnología en firme candidata para ser utilizada en sistemas de localización de interiores.

### 3.2.8 Wi-Fi

Se trata de una tecnología de gran éxito comercial basada en el estándar 802.11. La comunicación típica en el protocolo 802.11 sigue un modelo centralizado, lo que implica que la red consta de uno o varios puntos de acceso y multitud de clientes conectados a los mismos. Para localizar el móvil, cada punto de acceso puede emitir periódicamente mensajes que son recibidos por los usuarios, los cuales pueden medir la potencia recibida y así enviar esta información a una estación base para el cómputo de la localización.

La ventaja que tiene para ser usada como tecnología de localización es la existencia de infraestructura ya instalada en muchos edificios, contribuyendo así a que el coste hardware sea nulo. Este hecho ha despertado gran interés en la comunidad científica, viéndose reflejado en los grandes esfuerzos que se están realizando para implementar sistemas de localización robustos basados en esta tecnología y en la reciente aparición en el mercado de algunos de estos sistemas de localización.

Como desventaja podemos destacar el alto consumo de energía de los dispositivos inalámbricos necesarios, su gran tamaño, y que generalmente estos sistemas necesitan de una fase previa de calibración que implica la creación de bases de datos grandes y también se deben calibrar de nuevo cuando las condiciones de entorno cambian sensiblemente (por ejemplo, construcciones de nuevos muros en el edificio, cambios grandes de mobiliario, modificar la ubicación de los puntos de acceso, etc).

### 3.2.9 Redes inalámbricas de sensores

Según la comunidad científica, las redes de sensores inalámbricas (WSNs) es la tecnología mejor situada junto con Wi-Fi para la localización en interiores. Los avances en electrónica y en las comunicaciones inalámbricas han hecho viable el desarrollo de redes de sensores multifuncionales de bajo coste y bajo consumo.

Estos pequeños sensores, también conocidos como *motes* o motas, son capaces de medir parámetros físicos, procesar información y comunicarla a otros dispositivos a través de un canal de radiofrecuencia.

Las redes inalámbricas de sensores se asientan sobre el estándar IEEE 802.15.4, que establece la capa física y la capa MAC para redes de área personal inalámbrica de baja tasa (LR-WPANs), y la especificación Zigbee (u otra), que establece la capa de red y la de aplicación sobre las dos primeras. Estos estándares permiten medir y enviar información entre sensores. En particular, los parámetros RSS se pueden medir en la capa física según el estándar 802.15.4 a través del indicador de calidad del enlace (LQI), que reporta a las capas superiores información sobre la fuerza de la señal asociada a los paquetes de datos recibidos.

La capacidad de comunicación de datos a baja tasa es la que posibilita un consumo energético muy bajo. Esto unido a la capacidad de posicionamiento, el bajo coste de infraestructura y el pequeño tamaño de las motas convierte las redes de sensores en una tecnología idónea para los sistemas de localización de interiores.

## 3.3 Técnicas de localización en Redes Inalámbricas de Sensores

La localización usando WSNs es una tarea difícil. Para discutir en detalle este problema, vamos a introducir el contexto en el que operan estas redes. Denominaremos *nodo baliza*, o simplemente *baliza*, a aquellos nodos sensores referencia que tienen una posición conocida y fija. Asimismo, denominaremos *nodo desconocido* a

aquellos nodos de los cuales desconocemos su posición y cuya determinación será nuestro problema.

En redes de pequeña escala, es decir, con pocos nodos y cuando estos son fijos, establecer la localización de los mismos manualmente es viable. Por contra, cuando hablamos de redes de gran escala donde los nodos están distribuidos al azar o éstos son móviles la configuración manual deja de ser una solución.

El GPS (ver [Apartado 3.2.1](#) de este capítulo) es el más conocido sistema de posicionamiento para entornos al aire libre. Sin embargo, es demasiado caro incluir en todos los dispositivos sensores un receptor de este tipo. Además, generalmente los nodos sensores están limitados en energía (proviene de una batería en la mayoría de casos) por lo que no es práctico tampoco el uso del sistema GPS debido al consumo. Finalmente y muy importante, el sistema de GPS no es operativo en ambientes de interior debido a que los edificios pueden bloquear las señales del satélite. Por lo tanto, el GPS puede ser utilizado en algunos casos como apoyo para establecer la localización de los nodos baliza si el entorno lo permite y en función de la aplicación o presupuesto disponible para la misma.

Recientemente, el problema de la localización en redes de sensores ha atraído la atención de muchos miembros de la comunidad científica y se han propuesto un gran número de métodos de localización. Estos métodos se pueden clasificar principalmente en dos categorías: métodos de localización basados en distancia (*Range-based*) y métodos libres de distancia (*Range-free*).

Los métodos basados en distancia utilizan la fuerza de la señal recibida (RSS), Tiempo de Llegada (ToA), diferencia de tiempo de llegada (TDoA), o el ángulo de llegada (AoA) para medir distancias o ángulos entre nodos y luego usar estos datos para calcular las posiciones de los mismos. Los métodos libres de distancia no emplean las técnicas de medida anteriores, en su lugar usan métodos alternativos, como conteo basado en salto o área, para localizar a los nodos.

Los métodos dentro de la categoría *Range-based* o basados en distancia se engloban dentro de un grupo conocido como métodos de localización en dos pasos, que si bien no son óptimos en general, normalmente requieren de menos complejidad computacional que los que se incluyen dentro de otro grupo denominado de localización directa. Además, los métodos de localización en dos pasos son los más comunes y por ello en los que nos centraremos en este TFG. La diferencia entre estos dos grupos mencionados se puede observar en la Figura 3-1.

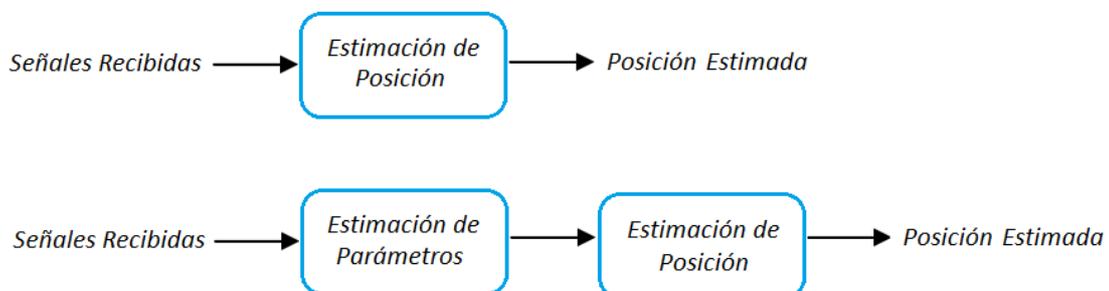


Figura 3-1. Localización Directa (arriba) vs Localización en dos pasos (abajo).

Básicamente, la posición se calcula haciendo uso de la información que se recibe de los nodos baliza en la estación base (puede ser un nodo con capacidades adicionales o un PC). El nodo desconocido transmite información a los nodos baliza y estos una vez recibido el mensaje calculan algún parámetro intermedio (primer paso) usando los métodos *Range-based* o *Range-free*, el cual se transmite al nodo remoto para obtener finalmente la posición del nodo desconocido (segundo paso) con técnicas como el reconocimiento de patrones (basados en mapas de potencia recibida), técnicas geométricas o técnicas estadísticas.

Por último y antes de pasar a describir los métodos basados en distancias, nombraremos algunas características deseables de los mismos para cumplir con los requisitos de localización para redes de sensores inalámbricos. Deben ser precisos (es decir, con pequeñas diferencias entre las posiciones estimadas y las reales), distribuidos (es decir, que no dependan de la infraestructura global), robustos (es decir, tolerantes a fallos en nodos y a errores de medida) y eficientes (esto es, tener pocos gastos de computación y comunicación y por ende energéticos). Sin embargo, es muy difícil lograr todos estos requisitos al mismo tiempo. Los métodos actuales generalmente pueden satisfacer uno o algunos de estos requisitos únicamente a la vez.

### 3.4 Técnicas de estimación de distancias

En función de la precisión requerida y las restricciones del sistema, en el primer paso pueden ser estimados diferentes parámetros. Normalmente, estos parámetros están relacionados con la potencia, dirección y/o tiempo de vuelo de una señal recibida. A continuación, se describen cada una de estas técnicas.

#### 3.4.1 Tiempo de llegada (ToA – Time of Arrival)

La medición del ToA de una señal es un método robusto para estimar distancias que se usa, entre otros sistemas, en GPS.

Esta técnica se basa en el hecho de que una señal (RF, acústica, u otras) tarda un tiempo determinado en llegar de un nodo emisor a uno receptor. Conociendo la velocidad a la que viaja, podremos hallar la distancia, ya que existe una relación lineal entre el tiempo de propagación y la distancia recorrida. Si llamamos al tiempo de propagación entre un nodo  $A$  y otro  $B$ ,  $t_{AB}$ , y a la velocidad de propagación de la señal  $v$ , tenemos que la distancia  $d_{AB}$  entre los nodos será:

$$d_{AB} = v \cdot t_{AB} \quad (3-1)$$

De este proceso se intuye que es necesaria una alta sincronización entre nodos, bien sea compartiendo el mismo reloj o intercambiando información de temporización.

#### 3.4.2 Diferencia de tiempos de llegada (TDoA: Time Difference of Arrival)

Aunque la técnica anterior es muy precisa, se necesita una estricta sincronización entre el emisor y el receptor. La técnica TDoA, en cambio, hace uso de dos señales, cada una con velocidades de propagación diferentes. La primera de ellas es una señal de radiofrecuencia que tiene un tiempo de propagación  $t_{RF}$ , la segunda, una señal acústica (normalmente ultrasonido) cuyo tiempo de propagación se expresa como  $t_{Ac}$ . En la Figura 3-2 se muestra el procedimiento para hallar la distancia con esta técnica, donde se desprecia el tiempo de vuelo de la señal RF ya que ésta es varios órdenes de magnitud inferior al de la señal acústica.

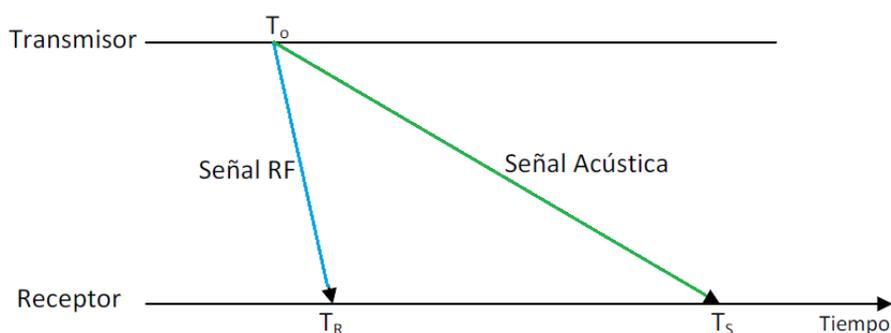


Figura 3-2. Técnica TDoA.

De la Figura 3-2, se intuye que la distancia estimada será:

$$d_{AB} = v_{Ac} \cdot (T_S - T_R) \quad (3-2)$$

Sistemas como Cricket o AHL oS (Ad-Hoc Localization System) utilizan esta técnica.

El mayor problema de las técnicas basadas en el tiempo de vuelo de las señales en entornos de interior es que los receptores deben ser capaces de estimar con precisión el tiempo de llegada de la señal correspondiente a la línea de visión directa entre los nodos. Obtener precisión en estas medidas es una tarea difícil debido al ruido aditivo y las señales multitrayecto que llegan al receptor. Es este último fenómeno el que realmente limita la precisión de la medida, ya que influye en mayor medida que el primero. De forma general, los dos mayores problemas son el escaso retardo que sufren las señales correspondientes al multitrayecto y la gran atenuación que suele sufrir la señal correspondiente al trayecto principal (LOS), pudiendo ser confundida con niveles de ruido o ser confundida en el receptor por alguna de las señales provenientes del multitrayecto.

Si bien es cierto que la atenuación de la señal correspondiente a la línea de visión directa puede ser subsanada desplegando la red de forma que la separación entre los nodos no sea muy grande, el escaso retardo que sufren las señales multitrayecto es muy difícil de paliar. Para evitar esto, se han de implementar estas técnicas en sistemas de mayor resolución temporal y, por tanto, de mayor ancho de banda, como por ejemplo UWB. El problema de estos sistemas es que requieren de un hardware caro y con un consumo de energía elevado.

Otro de los inconvenientes, aunque de menor importancia, son los retardos producidos en transmisor y receptor, debido al hardware y al software, que se añaden a los tiempos medidos. Esto conlleva que cualquier variación respecto a los valores nominales de estos retardos provoquen grandes errores en las medidas.

Las diferentes medidas tienden a producir un error medio en la estimación del 74%. Es decir, si tomamos un intervalo de observación lo suficientemente grande, solo el 26% de las estimaciones serán correctas. Aun así, se pueden obtener medidas más precisas post-procesando los datos con técnicas de cancelación de ruido, filtrado digital y detección y calibrado de picos.

Con el fin de complementar las técnicas de ToA y TDoA, se ha propuesto la técnica denominada ángulo de llegada (AoA), la cual permite que los nodos estimen los ángulos formados entre ellos mismos y sus vecinos.

### 3.4.3 Ángulo de llegada (AoA - Angle of Arrival)

El ángulo de llegada se define como el ángulo que forma la dirección de propagación de una onda incidente y una determinada dirección de referencia, que se conoce con el nombre de orientación, tal y como se observa en la Figura 3-3. Normalmente, se emplean arrays de antenas para hallar el AoA de una señal incidente, aunque también es posible usar medidas de potencia de la señal recibida en al menos dos antenas direccionales situadas en un mismo nodo para medir este parámetro.

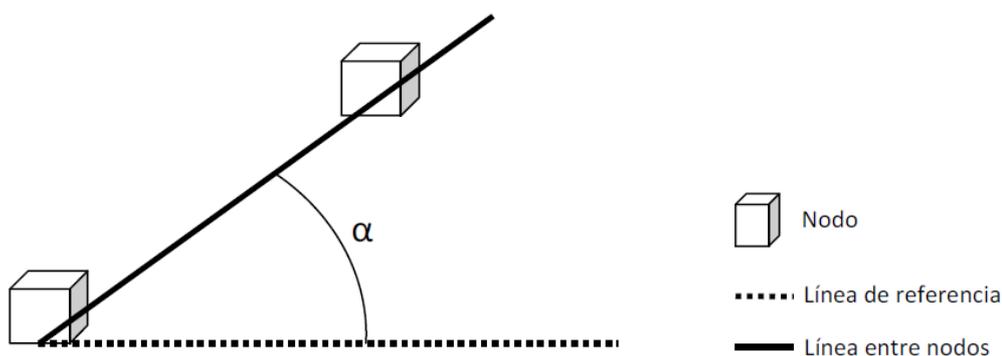


Figura 3-3. Ángulo entre nodos.

El principio en que se basa la técnica de arrays de antenas es en hallar la diferencia entre los tiempos de llegada de una señal recibida a los elementos del array para inferir el AoA conociendo la geometría del array. Para ello se requieren técnicas de procesamiento de señal que son diferentes para el caso de sistemas de banda estrecha y sistemas de banda ancha. En los primeros, la diferencia de los tiempos de llegada de la misma señal a los distintos elementos del array puede ser expresada como cambios en la fase de la señal y, en los segundos, se deben considerar diferentes versiones de la misma señal retardada en el tiempo debido a que un único retardo de la señal no puede ser representado con un solo valor de fase en señales de banda ancha.

Esta técnica, al igual que las anteriores, sufre una gran degradación en las medidas debido al multitrayecto en entornos de interior, lo que conlleva la posibilidad de medir ángulos de señales reflejadas y ruido aditivo, provocando oscilaciones en las medidas. Por ello, esta técnica se emplea a menudo en combinación con las anteriores y no de forma individual.

Al igual que en ToA y TDoA, el cálculo de AoA requiere dotar a la red de un costoso hardware adicional. Para evitar usar este hardware adicional, se ha desarrollado un esquema que utiliza la interfaz radio para inferir las distancias entre nodos. A continuación se describe dicho esquema.

### 3.4.4 Fuerza de la señal recibida (Received Signal Strength)

Esta técnica hace uso de la potencia o energía recibida en un nodo receptor procedente de una señal emitida desde un nodo transmisor. Este parámetro, comúnmente llamado RSSI (*Received Signal Strength Indicator*), puede ser empleado junto con un modelo de pérdidas de trayectorias (*path loss*) y sombras (*shadowing*) para realizar estimaciones de distancias con un coste de energía muy bajo.

En entornos de interior nos encontramos con que una señal viajando de un nodo a otro, experimenta desvanecimiento rápido de múltiples trayectorias, pérdidas en su trayectoria y, además, se ve afectada por el efecto de las sombras que originan los diferentes obstáculos. Estos efectos complican el diseño de un modelo consistente.

Idealmente, promediando el RSSI en un intervalo de tiempo lo suficientemente grande, podríamos evitar los efectos del desvanecimiento de múltiples trayectorias y los de las sombras. Considerando lo anterior, podríamos formular el siguiente modelo de propagación:

$$P_r = P_t - \alpha \cdot d - \eta \quad (3-3)$$

donde  $\alpha$  es el coeficiente de pérdidas de propagación en el trayecto,  $P_r$  la potencia media recibida en dB a una distancia  $d$ , y  $P_t$  es la potencia recibida en dB a una corta distancia de referencia  $d_0$ .

En la práctica, el efecto del multitrayecto puede ser mitigado usando métodos de espectro ensanchado como DSSS (*Direct Sequence Spread Spectrum*) o FHSS (*Frequency Hopping Spread Spectrum*), que promedian la potencia recibida sobre un amplio rango de frecuencias, evitando así el desvanecimiento selectivo en frecuencia causado por el multitrayecto. Además, el empleo de técnicas de espectro ensanchado también reduce las interferencias ocasionadas en la banda libre, que es la usada normalmente en las WSNs. Sin embargo, no es posible disminuir los efectos producidos por las sombras de los obstáculos debido a que estas dependen directamente del entorno en el que se encuentre desplegada la red. Este efecto suele modelarse mediante una variable aleatoria Gaussiana de media  $\mu$  y varianza  $\sigma^2$ . Por tanto, el modelo quedaría como sigue:

$$P_r = P_t - \alpha \cdot d - \eta + \xi \quad (3-4)$$

Nótese que el modelo es válido en situaciones tanto de visión directa (LOS) como sin ella (NLOS). La principal desventaja es que las medidas realizadas son impredecibles por la dependencia directa con el entorno. Además, existen desafíos adicionales como, por ejemplo, la construcción de un mapa de potencia recibida fiable o la variabilidad de la señal RSSI en función del tipo de antena empleado en los receptores, fabricantes e incluso del estado de carga de la batería en cada momento.

Pese a todo ello, la toma de medidas de RSSI se puede realizar a un coste muy bajo tanto de hardware como de energía, ya que estas se pueden realizar en los nodos receptores durante las comunicaciones de datos existentes sin necesidades adicionales de ancho de banda ni requerimientos altos de energía.

Además, la implementación del proceso de toma de estas medidas en hardware es muy simple, lo que

convierte a esta técnica en un método muy atractivo e interesante para la comunidad científica.

### 3.4.5 Combinaciones de parámetros

Como ya se ha comentado, en algunos sistemas de localización se emplean combinaciones de varios parámetros para mejorar la precisión en la localización del nodo desconocido. Así, son usuales los sistemas que emplean combinaciones como ToA/RSS, T(D)oA/AoA y ToA/TDoA.

Además de combinar las distintas técnicas, se puede hacer uso de otros parámetros que normalmente se emplean en un contexto totalmente diferente. Si fuésemos capaces de obtener el perfil potencia-retardo de la señal multirrayecto, podríamos hablar de parámetros PDP (*Power Delay Profile*) y si obtuviésemos la respuesta impulsiva del canal serían parámetros CIR (*Channel Impulse Response*). En algunas situaciones, estos parámetros pueden ayudar significativamente a la localización del nodo desconocido aportando más información que los citados anteriormente. Sin embargo, para poder hacer uso de estos parámetros se requiere de una fase inicial de toma de medidas que a veces es inviable debido a la poca accesibilidad que puede llegar a tener el entorno en cuestión. Asimismo, las bases de datos necesitan de una gran capacidad de almacenamiento por lo que en muchas situaciones el coste computacional se eleva considerablemente.

Aparte de estos, se puede usar otro tipo de parámetros que emplean el perfil potencia-ángulo de la señal multirrayecto. Estos parámetros pueden ser obtenidos únicamente por nodos que posean arrays de antenas, lo que hace necesario la adquisición de hardware adicional.

Aunque todos estos parámetros suponen un coste adicional, mejoran sensiblemente la exactitud y precisión en entornos tan cambiantes como los de interior.

## 3.5 Técnicas de estimación de posición

Como se comentó en la [Sección 3.3](#) de este capítulo, las balizas tras recibir la señal del nodo de posición desconocida generaban unos parámetros intermedios conocidos como parámetros de localización que enviaban a la estación base, y ésta en un segundo paso realizaba la estimación de la posición. Dependiendo de la presencia de bases de datos (existencia de una fase de entrenamiento o calibración), las técnicas de estimación de posición se pueden dividir en dos grupos: técnicas geométricas, estadísticas y técnicas de reconocimiento de patrones (basados en mapas de potencia recibida).

### 3.5.1 Técnicas geométricas

Las técnicas geométricas no necesitan del uso de bases de datos, evitando así la fase previa de calibración. La estimación de la posición del nodo desconocido se infiere directamente de los parámetros de localización hallados en el primer paso, usando relaciones geométricas.

Las técnicas geométricas estiman que la posición de un nodo desconocido es la intersección de las líneas de posición fijadas por los nodos baliza, obtenidas mediante técnicas de estimación de distancias. De estas últimas se dedujo que parámetros como RSSI o ToA definen una circunferencia que representa los posibles puntos donde el nodo baliza determina que puede estar situado el nodo desconocido. Esto se puede observar en la Figura 3-4.

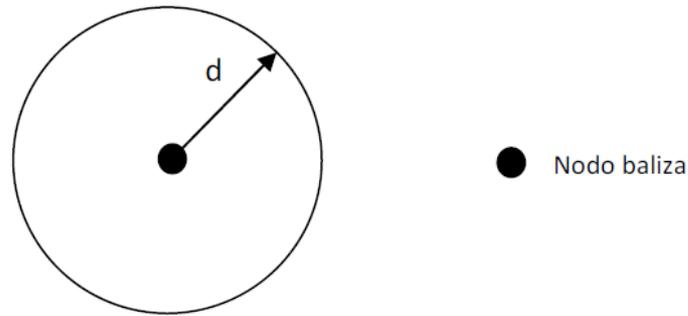


Figura 3-4. Circunferencia de incertidumbre originada por un nodo baliza.

Por consiguiente, si se hace uso de tres nodos baliza (de sus tres estimaciones de distancia), es posible hallar la localización del nodo desconocido a través de la trilateración de las tres distancias, tal y como se puede observar en la Figura 3-5.

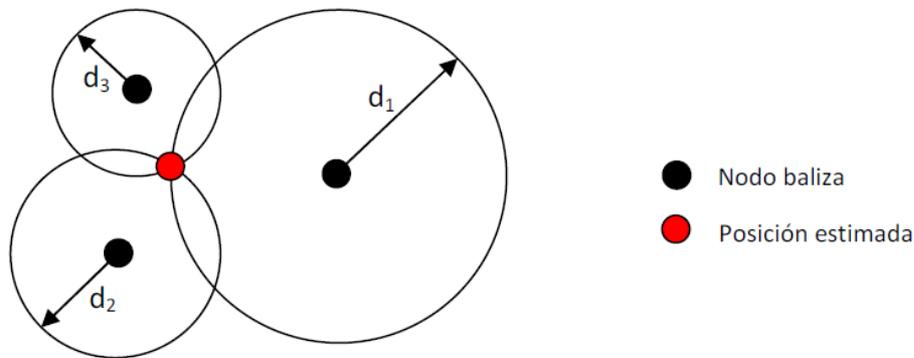


Figura 3-5. Posición de un nodo desconocido determinada mediante trilateración.

Por otro lado, el parámetro AoA define una línea recta que pasa a su vez por el nodo desconocido y por el nodo baliza al que hace referencia, tal y como se muestra en la Figura 3-3. Es por ello por lo que haciendo uso de dos parámetros de este tipo, es posible estimar la posición del nodo desconocido mediante triangulación, tal y como se aprecia en la Figura 3-6.

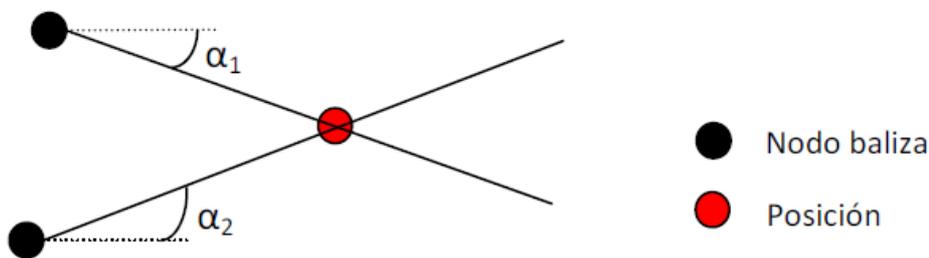


Figura 3-6. Posición del nodo desconocido determinada mediante triangulación.

Las técnicas geométricas también pueden aplicarse en sistemas de localización híbridos. Es decir, si el sistema fuese capaz de medir diferentes tipos de parámetros, las técnicas geométricas podrían combinarlos. Por ejemplo, si un nodo baliza mide los parámetros AoA y RSSI procedentes de las señales del nodo desconocido, sería posible estimar su posición tal y como se muestra en la Figura 3-7. De esta forma la posición del nodo desconocido puede estimarse mediante las siguientes relaciones:

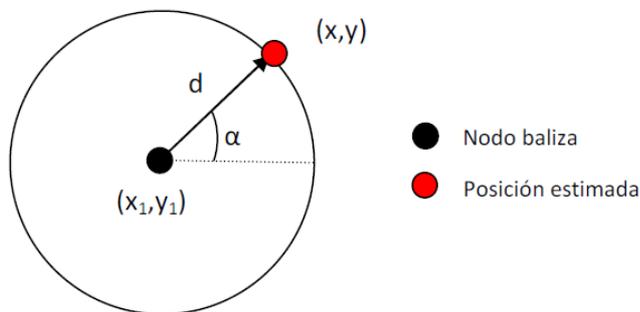


Figura 3-7. Posición del nodo desconocido determinada mediante sistema híbrido.

Podemos concluir por tanto que el número de nodos baliza necesarios para estimar la posición del nodo desconocido depende de las características que tengan los nodos de la red, es decir, de los parámetros que sean capaces de medir.

Uno de los inconvenientes de las técnicas geométricas es que son impredecibles ante errores en las estimaciones de los parámetros. Es muy común la presencia de ruido en la estimación de las distancias, por lo que las técnicas geométricas obtendrán más de un punto de intersección en sus estimaciones, convirtiéndolo en un método ineficaz para la localización. Además, conforme el número de nodos baliza aumenta, el número de puntos de intersección estimados se incrementa. Así, un caso tan simple como el expuesto en la Figura 3-8, en el que solo se utilizan tres nodos baliza y tienen lugar errores en las estimaciones de los parámetros AoA, demuestra cómo las técnicas geométricas son ineficaces.

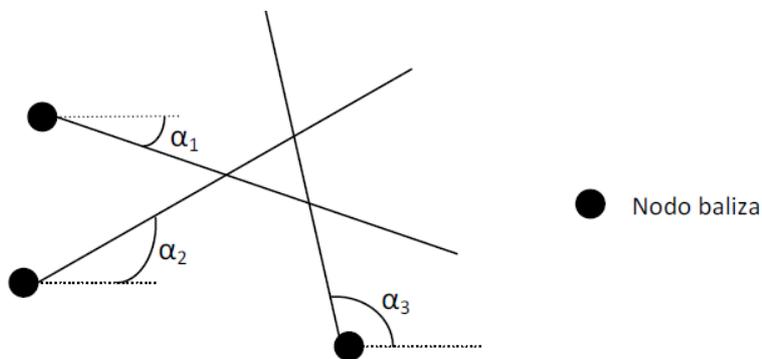


Figura 3-8. Posición del nodo desconocido con tres parámetros AoA y ruido.

Para poder tener en consideración estos efectos tan comunes en entornos de interior sin hacer uso de fases previas de calibración, se suelen usar técnicas estadísticas.

### 3.5.2 Técnicas estadísticas

Al igual que las técnicas geométricas, las técnicas estadísticas no necesitan de una base de datos previa para la estimación de la posición de un nodo desconocido. Además, presentan notables ventajas a la hora de estimar con exactitud la posición de un nodo en presencia o no de ruido y, no se ven afectadas por la ambigüedad producida en la triangulación de las estimaciones de distancias de varios nodos baliza.

Dependiendo de la información disponible acerca del término de ruido añadido a las estimaciones de las distancias, se pueden llevar a cabo técnicas estadísticas paramétricas o no paramétricas. En caso de que la función densidad de probabilidad del ruido sea conocida a excepción de un determinado conjunto de parámetros, pueden utilizarse técnicas paramétricas como la estimación Bayesiana y la de Máxima Verosimilitud (ML, *Maximum Likelihood*). En ausencia de información es necesario usar técnicas no paramétricas como K-NN (*K-Nearest Neighbour*) y otras como SVR (*Support Vector Regression*) o redes neuronales.

Dada la complejidad computacional requerida en este tipo de técnicas y las restricciones de las redes de sensores, no se explicaran en este TFG.

### 3.5.3 Técnica de reconocimiento de patrones

Una alternativa a las técnicas anteriores en la que es necesaria una fase previa de calibración del entorno es la técnica de reconocimiento de patrones. Esta fase consistiría en la toma de medidas de parámetros RSSI en los nodos baliza cuando el nodo desconocido se encuentra en las posiciones que estemos interesados en localizar durante el funcionamiento de la aplicación de localización. De esta manera, obtendremos una pequeña base de datos formada por un conjunto de duplas del tipo: **[localización, vector de potencia]**, donde localización hace referencia a las coordenadas (vector de dos dimensiones) de las posiciones que estemos interesados en localizar y vector de potencia al conjunto de estimaciones de parámetros RSSI en los nodos baliza cuando el nodo desconocido se encuentra en la localización correspondiente (vector cuya dimensión es el número de nodos baliza). Así, estaríamos caracterizando en cierta forma las condiciones de propagación en el entorno específico.

Una vez obtenida esta base de datos, el algoritmo de localización implementado en la estación base debe comparar los vectores de potencia recibidos durante el funcionamiento de la aplicación con los almacenados en la base de datos durante la fase de calibración. De esta manera, realizando un simple cómputo en el que se escoja el vector de potencia de la base de datos de menor distancia euclidiana al vector de potencia obtenido, podríamos estimar la posición del nodo desconocido.

Esta técnica arroja buenos resultados y subsana inconvenientes presentes en las técnicas anteriores siempre que las condiciones de trabajo en la fase de calibración y durante el funcionamiento de la aplicación de localización sean lo suficientemente parecidas, es decir, en el momento en el que se produzcan cambios significativos en el entorno (cambio de posición de mobiliario de grandes dimensiones, construcción de nuevas paredes, etc.) habría que actualizar la base de datos.

## 3.6 Detección Humana

La detección y el seguimiento del movimiento de objetos o personas son campos de interés para gran número de aplicaciones como pueden ser la vigilancia, la robótica, la navegación o los entornos inteligentes (iluminación y accionamiento de puertas de forma automática, monitorización de tráfico, conteo, etc). Numerosos sistemas han sido estudiados utilizando sensores (de movimiento, de campo eléctrico, etc.), cámaras o radares.

Cuando hablamos de detección y seguimiento de personas (*Human Detection and Tracking*) el uso de algunas de estas tecnologías conlleva un gran coste de procesamiento, para conocer, por ejemplo, el número, la posición o la dirección del movimiento de los individuos, debido a la necesidad de diferenciar características concretas del cuerpo humano. Incluso siendo soluciones precisas, estas tienen un alto coste y requieren de un despliegue de infraestructura significativo.

Una opción bien conocida entre la comunidad científica para entornos de interior son los sensores PIR (*Passive Infrared sensors*), catalogados como una solución simple pero potente que puede hacer frente a sistemas más complejos y caros en determinadas situaciones.

Básicamente estos sensores captan la luz infrarroja (IR) radiada de los objetos situados en su campo de visión (FOV, *Field Of View*). Su principio de operación se basa en la idea de que todos los objetos emiten energía en

forma de radiación a causa de tener una temperatura corporal por encima del cero absoluto. Por otro lado, el término pasivo se refiere al hecho de que los dispositivos PIR no generan o irradian energía para fines de detección sino que trabajan en su totalidad para la detección de la energía emitida por otros objetos. A este respecto es necesario aclarar que los sensores PIR no detectan o miden calor, sino que detectan la radiación infrarroja emitida por un objeto, que es diferente aunque relacionado con la temperatura del mismo.

Normalmente estos sensores tienen dos partes sensibles a la radiación IR y lentes de Fresnel que modelan las áreas de detección. Cuando ambas zonas ven la misma cantidad de radiación térmica, el sensor no detecta nada (señal de salida estable), mientras que si una zona capta distinto nivel que la otra, la salida del sensor cambia (ver Figura 3-9). Los sensores PIR pertenecen a la clase de detectores llamados de conversión termoeléctrica pues generan una salida eléctrica a partir de la variación de radiación térmica incidente.

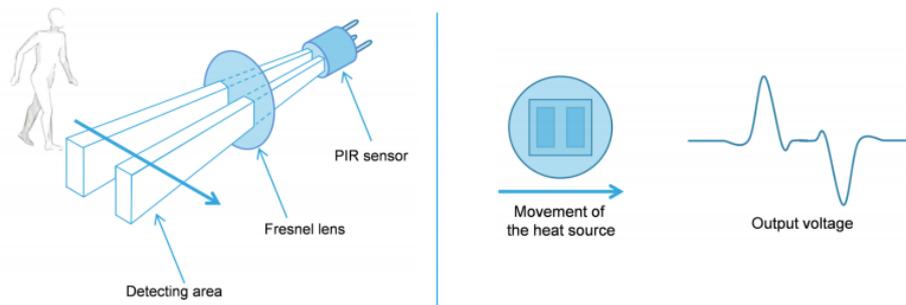


Figura 3-9. Funcionamiento de los sensores PIR.

Aunque comúnmente son usados como detectores de presencia simples con una salida digital, cuando devuelven una señal de salida analógica nos permiten conocer la dirección, la velocidad, la distancia de una persona al sensor, la forma del cuerpo o la presencia de más de un individuo.

En la Figura 3-10a se observa que gracias a la alternancia de las polarizaciones en los elementos sensores, podemos discriminar hacia qué sentido se mueve el individuo que va caminando, es decir, de izquierda a derecha o de derecha a izquierda.

Del mismo modo, en las Figuras 3-10b y c respectivamente se observa que podemos diferenciar a una persona según la distancia a la que se encuentra de los sensores PIR a partir de las señales, que tienen diferentes características (por ejemplo distinta amplitud o tiempos de picos) y la velocidad a la que camina un sujeto, es decir, lo cerca/lejos o lentos/rápidos que se mueven.

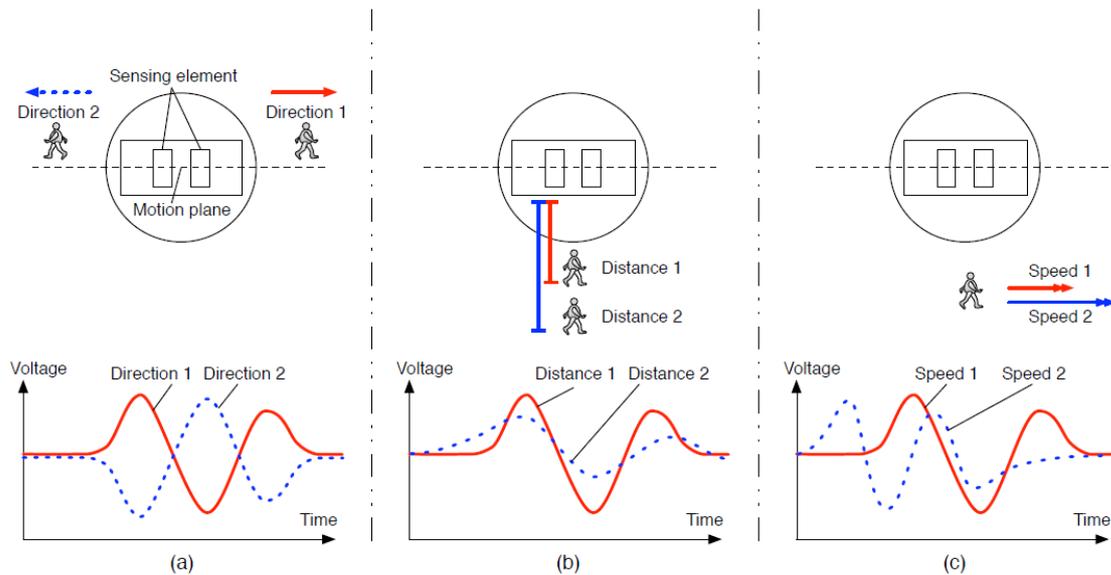


Figura 3-10. Salida del detector PIR ante distintas situaciones.

Otra opción de la que ya se ha hablado en este TFG y con la que no se necesita un hardware adicional se basaba en hallar la localización del nodo desconocido midiendo los parámetros de intensidad de señal (RSSI) desde otros nodos de la red (nodos baliza). Las balizas enviaban esa información a la estación base para que cotejara los niveles de señal recibidos con los una base de datos elaborada previamente y que contenía los niveles de señal medidos en diferentes posiciones del entorno de interior donde podía estar el nodo desconocido. Así pues, la estación base comparaba las medidas recibidas desde los nodos baliza correspondientes al nodo desconocido con las guardadas en la base de datos (técnica de reconocimiento de patrones) y en función de éstas determinaba la localización.

Básicamente las personas a ubicar portarán un nodo que irá emitiendo señal mientras se van moviendo por un área acotada donde pueden ser localizadas. Esta técnica puede implementarse por ejemplo con Wi-Fi o con una WSN.

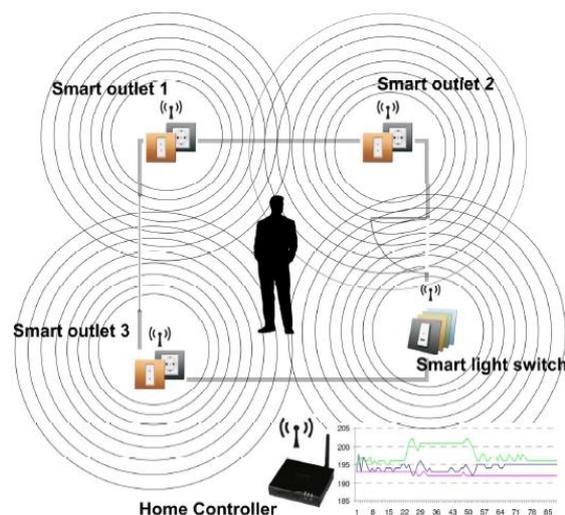


Figura 3-11. Localización basada en RSSI.

Finalmente, tal y como se ha comentado en otras ocasiones, dependiendo de la aplicación específica y de la precisión que sea necesaria usaremos una tecnología u otra. Si bien aquí hemos presentado un poco más en detalle los sensores PIR por ser una de las soluciones más utilizadas, estos no nos permiten, por ejemplo,

detectar personas concretas como los sistemas que emplean cámaras y un gran procesamiento para el reconocimiento facial, u otros rasgos humanos. Los sensores PIR tienen sus limitaciones debido a su principio de funcionamiento y no pueden usarse en entornos de exterior donde las condiciones de temperatura pueden cambiar rápidamente, o en espacios climatizados como también podemos imaginar.

La técnica que combina los parámetros RSSI y el reconocimiento de patrones nos permite aprovechar, por ejemplo, una red Wi-Fi ya implementada para la detección humana, pero no hay que olvidar las limitaciones que tiene el hecho de no ser una técnica creada especialmente para ello.

Otras técnicas que podrían utilizarse para la detección y el seguimiento humano fueron brevemente expuestas en la [Sección 2](#) de este capítulo (RFID, Bluetooth, UWB, etc) aunque existen muchas más. En cualquier caso habrá que valorar el objetivo de la aplicación a la que dar soporte, para elegir la tecnología más adecuada que satisfaga los requerimientos de la misma, tanto en funcionamiento como en coste.



# 4 WASPMOTE

---

**W**aspmote es una plataforma modular desarrollada por la empresa española Libelium y usada en redes de sensores inalámbricos. Como ya se describió en el [Capítulo 2](#) una WSN consta de una serie de sensores espacialmente distribuidos autónomos que toman datos acerca de condiciones físicas o ambientales, como la temperatura, el sonido, la vibración, la presión, el movimiento o agentes contaminantes por citar algunas y que son enviados de forma inalámbrica a un nodo central que almacenará y procesará dicha información del mundo real. En nuestra aplicación los nodos de la WSN serán Waspmote.

## 4.1 Datos Generales

### 4.1.1 Arquitectura modular

Waspmote se basa en una arquitectura modular. Esta idea consiste en integrar sólo los módulos necesarios en cada dispositivo, pudiendo ser cambiados y ampliados dependiendo de las necesidades de la aplicación o el usuario (este es el punto fuerte de la plataforma).

Los módulos disponibles para integrar en Waspmote (v1.1) se clasifican en:

- Módulos ZigBee/802.15.4 (2.4 GHz, 868 MHz, 900 MHz). Baja y alta potencia.
- Módulo GSM/GPRS (Quadband: 850 MHz/900 MHz/1800 MHz/1900 MHz).
- Módulo GPS.
- Módulos Sensoriales (Placas de Sensores).
- Módulo de almacenamiento: SD Memory Card.

### 4.1.2 Especificaciones técnicas

- Microcontrolador: ATmega1281
- SRAM: 8 KB
- EEPROM: 4 KB
- FLASH: 128 KB
- Tarjeta SD: 2 GB
- Peso: 20 gr

- Dimensiones: 73.5 x 51 x 13 mm
- Rango de Temperaturas: [-20°C, +65°C]

#### 4.1.3 Datos eléctricos

- Mínimo voltaje de la batería: 3.3 V
- Máximo voltaje de la batería: 4.2 V
- Voltaje en cualquier pin: [-0.5 V, +3.8 V]
- Máxima corriente para cualquier pin digital I/O: 40 mA
- Voltaje de batería cargada: 4.2 V

#### 4.1.4 Sensores incorporados

- Temperatura: -40°C, +85°C. Accuracy: 0.25°C
- Acelerómetro: +/- 2g (1024 LSb/g) / +/- 6g (340LSb/g) 40 Hz/160 Hz/640 Hz/2560 Hz

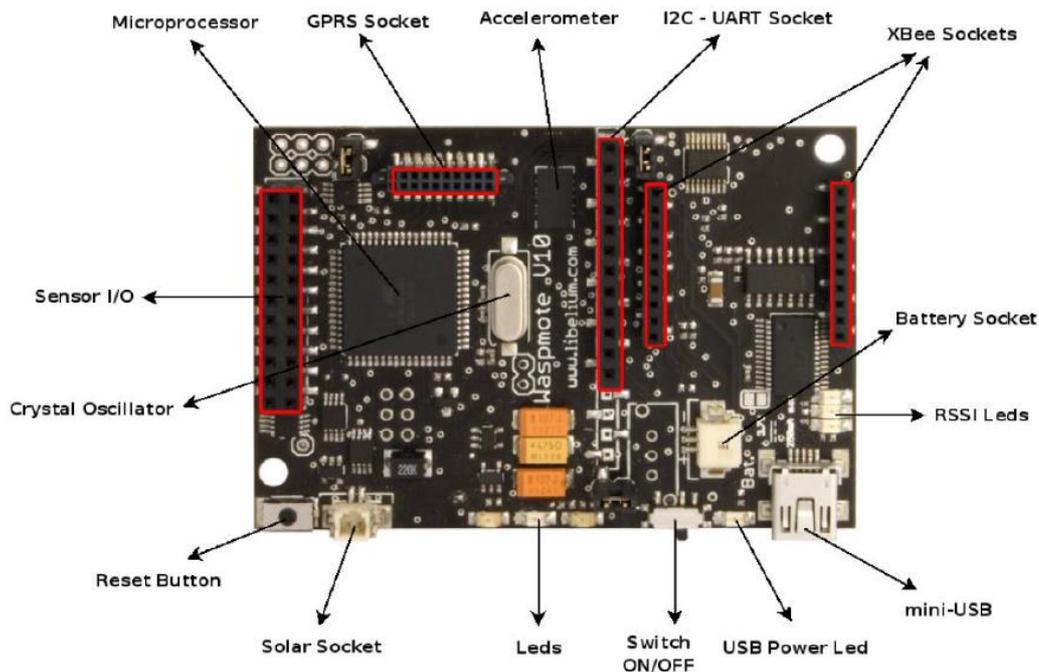


Figura 4-1. Cara superior y zócalos disponibles.

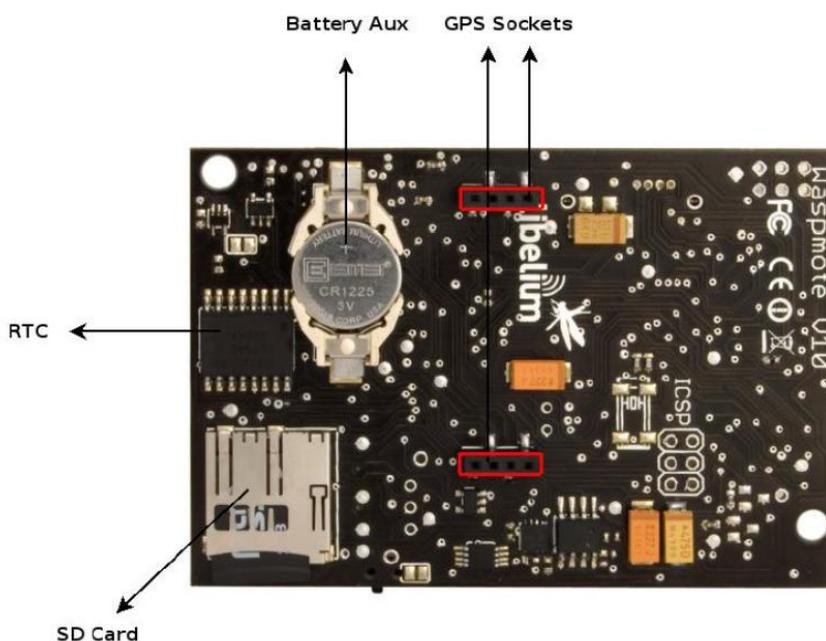


Figura 4-2. Cara inferior y zócalos disponibles.

## 4.2 Entradas/Salidas

Waspote puede comunicarse con otros dispositivos externos mediante los diferentes puertos de entrada/salida que posee.

Los dispositivos con los que Waspote se comunica pueden ser cualquier sensor, componente o módulo electrónico siempre y cuando se respeten las especificaciones requeridas por cada puerto.

Dispone de dos conectores, el *Sensor I/O* y el *I2C UART* como se puede observar en la Figura 4-3 y la Figura 4-4.

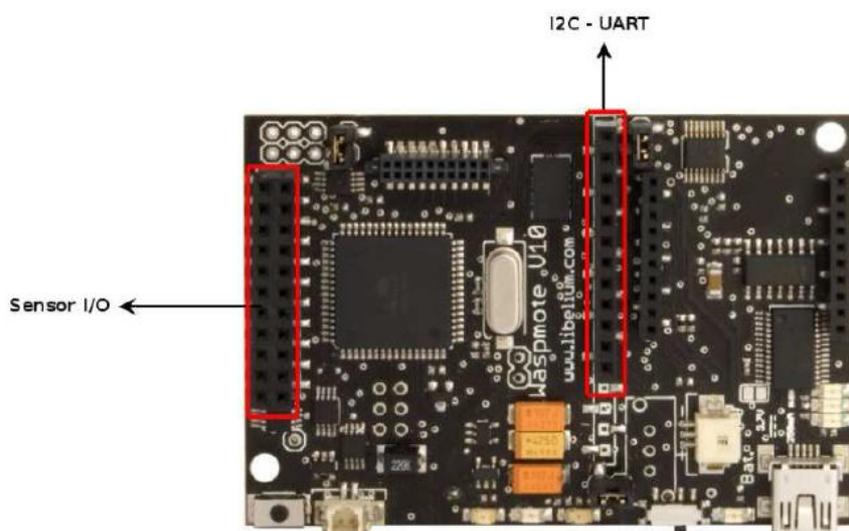


Figura 4-3. Conector Sensor I/O y I2C – UART.

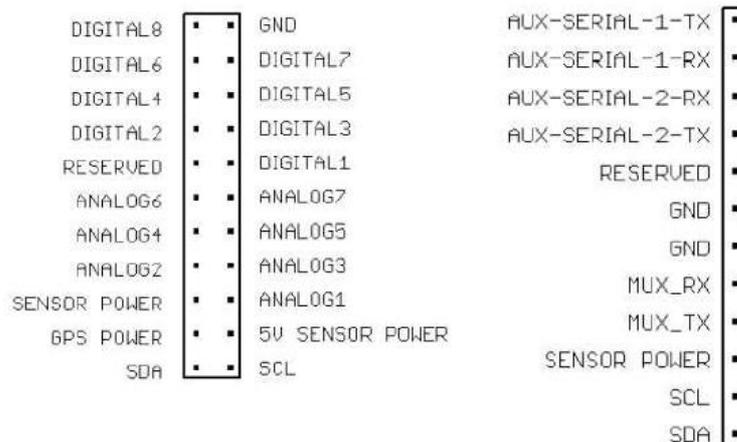


Figura 4-4. Pines de los conectores Sensor I/O e I2C – UART.

A continuación, en las siguientes subsecciones se detallan un poco las entradas y salidas del Waspote:

#### 4.2.1 Analógicas

Siete entradas accesibles en el conector de sensores y conectadas directamente al microcontrolador (éste tiene un ADC de 10 bits cuyo valor máximo de entrada es 3.3 V).

#### 4.2.2 Digitales

Ocho pines digitales configurables como entrada o salida, cuyos valores lógicos 0 y 1 corresponden a 0 V y 3.3 V, respectivamente.

#### 4.2.3 PWM (Pulse Width Modulation)

Existe un pin (DIGITAL1) que permite simular una onda cuadrada entre 0 y 3.3 V mediante modulación por anchura de pulso.

#### 4.2.4 UART (Universal Asynchronous Receiver Transmitter)

Waspote dispone de seis puertos serie. Una de las UART del microcontrolador está conectada simultáneamente al módulo de comunicación XBee y al puerto USB (por tanto, no se pueden usar simultáneamente el XBee y el puerto USB).

La otra UART del microcontrolador está conectada a un multiplexor de cuatro canales, pudiendo seleccionar desde el código cuál de los cuatro queremos conectar a la UART del microcontrolador. Estas cuatro nuevas UART están conectadas de la siguiente manera. Una está conectada a la placa GPRS, otra al GPS y las otras dos quedan accesibles al usuario en el conector I2C – UART auxiliares.

#### 4.2.5 I2C (Inter Integrated Circuit)

En Waspote también se utiliza el bus de comunicación I2C, donde se conectan en paralelo tres dispositivos:

el acelerómetro, el RTC (Real Time Clock) y el potenciómetro que configura el nivel de umbral de alarma por batería baja. En todos los casos el microcontrolador actúa como maestro (*master*) mientras que el resto de los dispositivos conectados al bus actúan como esclavos (*slave*).

#### 4.2.6 SPI (Serial Peripheral Interface)

El puerto SPI del microcontrolador se utiliza para la comunicación de éste con la tarjeta micro SD. Todas las operaciones de uso del bus son realizadas por la librería específica de forma transparente.

#### 4.2.7 USB (Universal Serial Bus)

La comunicación USB se utiliza en Waspote para la comunicación con un ordenador. Esta comunicación permite la carga del programa al microcontrolador y la comunicación de datos durante la ejecución del programa. Para la comunicación USB se utiliza una de las UART del microcontrolador y de la conversión al estándar USB se encarga el FT232RL. De esta forma en el ordenador tendremos un nuevo puerto de comunicación serie (virtual) listo para comunicarse con Waspote.

### 4.3 Sensores incorporados en Waspote

#### 4.3.1 Sensor de temperatura

El RTC dispone de un sensor de temperatura integrado, que utiliza para recalibrarse, y al que se accede mediante el bus I2C. El sensor se presenta en un formato de 10 bits complemento a dos, ocupando 2 registros. Tiene una resolución de  $+0.25$  °C. El rango de temperaturas que pueden medirse está comprendido entre  $-40$  °C y  $+85$  °C.

#### 4.3.2 Acelerómetro

Waspote tiene integrado el sensor de aceleración LIS3LV02DL de STMicroelectronics que informa a la mota de las variaciones de aceleración sufridas en cada uno de los 3 ejes (X,Y,Z).

La integración de este sensor permite la medición de la aceleración en los 3 ejes (X,Y,Z) estableciendo dos tipos de eventos (interrupciones): caída libre (*Free Fall*) y detección de cambio de dirección (*Direction Detection Change*). De esta forma, se pueden realizar acciones como despertar a la mota cuando se produce una caída libre o un cambio de dirección y realizar la acción asignada a tal efecto, en caso de tener a la mota dormida.

### 4.4 Fuentes de alimentación

Para alimentar el Waspote hay tres opciones:

- **Batería:** Se incluye una batería de Ion de Litio (Li-Ion) de tensión nominal 3.7 V. En lo referente a la capacidad de la batería existen dos posibilidades: 1150 mAh y 2300 mAh.
- **Panel solar:** Waspote dispone de un conector y un cable especial para conectar a un panel solar. Se admiten placas solares de hasta 12 V y corriente máxima de carga a través de placa de 240 mA.
- **USB:** Permite conectar mediante un cable USB a un PC una fuente de 220 V o un conector de vehículo.

## 4.5 Otros componentes

Además de los componentes descritos en los apartados anteriores, el Waspote dispone de otros componentes que se nombran a continuación:

### 4.5.1 LEDs

#### 4.5.1.1 LEDs programables

Se dispone de dos leds (LED0 y LED1) que se pueden programar con funciones de la API de Waspote.

#### 4.5.1.2 LED indicador de carga

Un led rojo indica que hay una batería conectada a Waspote que está siendo cargada, la carga se realiza mediante un cable mini USB o por medio de un panel solar unido a Waspote. Una vez que la batería está completamente cargada, el led se apaga automáticamente.

#### 4.5.1.3 LED indicador conexión USB

Un led verde indica cuando Waspote tiene conectado el cable USB correctamente para cargar la batería o programar la mota. Cuando el cable USB esta desconectado el led se apaga automáticamente.

#### 4.5.1.4 LEDs RSSI

Tres leds han sido incluidos para mostrar el nivel potencia con que llegan las señales, también conocido como RSSI (Received Strength Signal Indicator). Los parámetros RSSI indican la calidad de la señal del último paquete recibido de las tramas ZigBee/802.15.4. Los módulos XBee proporcionan esta información en todas las variantes del protocolo y frecuencia.

Tabla 4-1. Relacion entre leds encendidos y RSSI recibido.

LEDs switched ON	RSSI received
LED 1	RSSI > (Sensitivity + 11.1 dB)
LEDs 1+2	RSSI > (Sensitivity + 21.23 dB)
LEDs 1+2+3	RSSI > (Sensitivity + 31.82 dB)

### 4.5.2 Jumpers

La placa posee tres *jumpers* que permiten activar o desactivar ciertas funciones. El primero de ellos, *Programming enabling jumper* debe ser conectado para permitir la programación del microcontrolador, la carga de un programa, etc. El segundo *RSSI indicator enabling jumper* habilita los leds indicadores de nivel de señal de la ultima trama recibida por Waspote. Si no se van a utilizar estos leds conviene desconectar el jumper para ahorrar batería. El tercero es el *Hibernate mode enabling jumper* que es necesario para que la placa pueda ser programada y para habilitar el estado de hibernación del microcontrolador, que será descrito en la [Sección 4.6](#).

### 4.5.3 Memoria EEPROM

Existe una memoria no volátil de 4 KB donde se pueden guardar valores de forma permanente.

### 4.5.4 Tarjeta de memoria Flash micro-SD

Waspote integra un soporte de almacenamiento externo como son las tarjetas micro-SD (*Secure Digital*). Waspote usa el sistema de ficheros FAT16 y puede soportar tarjetas de hasta 2 GB y la comunicación desde el microcontrolador hasta la tarjeta se realiza mediante el bus SPI. Esta es una alternativa para grabar datos de manera masiva frente a la EEPROM.

### 4.5.5 Módulos externos

Libelium pone a disposición una serie de módulos que se pueden instalar en el Waspote, como por ejemplo, el módulo GSM/GPRS, que permite recibir datos, llamadas o SMS de la red móvil, o el GPS, que permite conocer la localización exacta de la mota en exteriores y sincronizar el RTC interno con la hora real. Además, se pueden conectar placas de sensores de gases, eventos, *smart cities*, agricultura, radiación, etc, por lo que el dispositivo es extremadamente versátil.

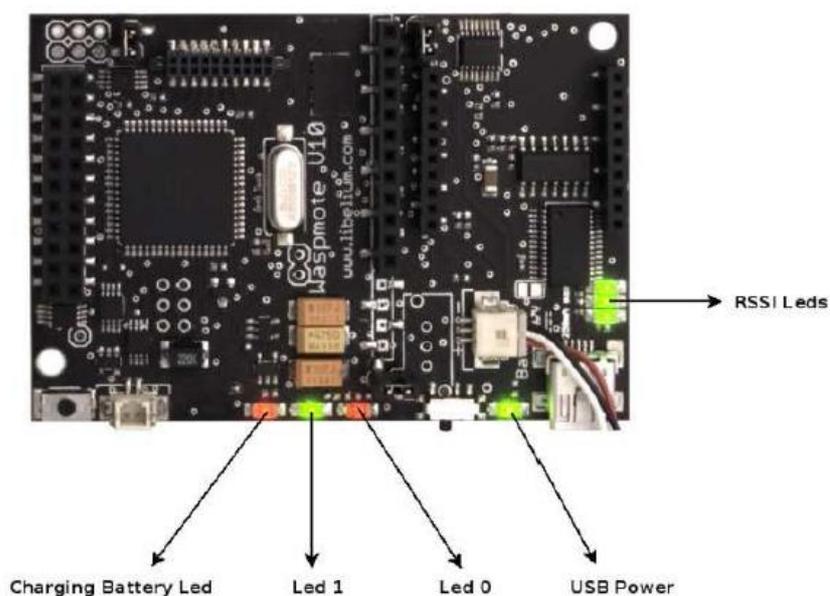


Figura 4-5. Leds de la placa.

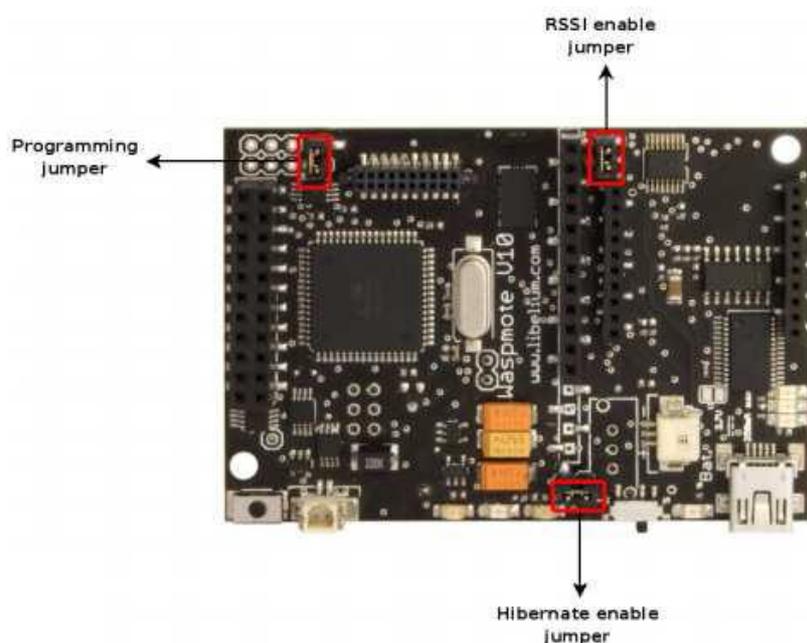


Figura 4-6. Jumpers de la placa.

## 4.6 Sistema energético

Waspote tiene cuatro modos de funcionamiento:

- **ON:** modo normal de funcionamiento. El consumo en este estado es de 9 mA.
- **Sleep:** El programa principal se detiene, el microcontrolador pasa a un estado de latencia, del que puede ser despertado por todas las interrupciones asíncronas y por la interrupción síncrona generada por el Watchdog. El intervalo de duración de este estado va de 32 ms a 8 s. El consumo en este estado es de 62  $\mu$ A.
- **Deep Sleep:** El programa principal se detiene, el microcontrolador pasa a un estado de latencia del que puede ser despertado por todas las interrupciones asíncronas y por la interrupción síncrona lanzada por el RTC. El intervalo de este ciclo puede ir de 8 s a minutos, horas, días. El consumo en este estado es de 62  $\mu$ A.
- **Hibernate:** El programa principal se detiene, el microcontrolador y todos los módulos de Waspote quedan completamente desconectados. La única forma de volver a activar el dispositivo es a través de la alarma previamente programada en el RTC (interrupción síncrona). El intervalo de este ciclo puede ir de 8 s a minutos, horas, días. Al quedar el dispositivo totalmente desconectado de la batería principal el RTC es alimentado a través de una batería auxiliar de la que consume 0,7  $\mu$ A.

Tabla 4-2. Características de los modos de funcionamiento.

Mode	Consumption	Micro	Cycle	Accepted Interruptions
<b>ON</b>	9 mA	ON	-	Synchronous and Asynchronous
<b>Sleep</b>	62 $\mu$ A	ON	32ms-8s	Watchdog and Asynchronous
<b>Deep Sleep</b>	62 $\mu$ A	ON	8s-min/hours/days	Synchronous (RTC) and Asynchronous
<b>Hibernate</b>	0.7 $\mu$ A	OFF	8s-min/hours/days	Synchronous (RTC)

Por otro lado, cada módulo tiene también cuatro modos de funcionamiento:

- **ON:** modo normal de funcionamiento.
- **Sleep:** En este modo se detienen algunas funcionalidades del módulo y se pasa a un uso asíncrono, normalmente dirigido por eventos. En cada módulo funciona de una forma distinta y es específico de cada uno (programado por el fabricante).
- **Hibernate:** En este modo se detienen todas las funcionalidades del módulo y se pasa a un uso asíncrono, normalmente dirigido por eventos. En cada módulo funciona de una forma distinta y es específico de cada uno (programado por el fabricante).
- **OFF:** Mediante el uso de switches digitales controlados por el microcontrolador se apaga por completo el módulo. Este modo ha sido implementado por Libelium como capa independiente de control energético, de forma que se pueda reducir el consumo al mínimo ( $\sim 0$   $\mu$ A).

## 4.7 Timers

Waspote utiliza un oscilador de cuarzo que trabaja a una frecuencia de 8 MHz como reloj del sistema. De esta forma, cada 125 ns el microcontrolador ejecuta una instrucción de bajo nivel (lenguaje máquina).

## 4.8 Watchdog

El microcontrolador Atmega1281 tiene un reloj Watchdog interno mejorado (WDT, *Enhanced Watchdog Timer*). El WDT se encarga de contar de forma precisa ciclos de reloj generados por un oscilador de 128 KHz. El WDT genera una señal de interrupción cuando el contador alcanza el valor establecido.

El WDT permite despertar al microcontrolador del estado Sleep de bajo consumo generando una interrupción. Por ello, se utiliza este reloj a modo de alarma temporal asociado al modo Sleep del microcontrolador. Esto permite controlar intervalos de tiempo muy pequeños y de forma muy precisa, estos intervalos son: 16 ms, 32 ms, 64 ms, 128 ms, 256 ms, 500 ms, 1 s, 2 s, 4 s, 8 s. Para intervalos mayores a 8 s (modo Deep Sleep) se usa el RTC.

## 4.9 RTC (Real Time Clock)

El RTC o reloj de tiempo real integrado de tipo DS3231SN de Maxim, mantiene informado al Waspote del momento temporal en el que se encuentra. Se programa a través del bus I2C y permite programar acciones, como por ejemplo, hacer que la mota hiberne y se despierte en un determinado momento gracias al uso de interrupciones. Su frecuencia es de 32 kHz (32.768 Hz).

El RTC permite despertar al microcontrolador del estado de bajo consumo generando una interrupción. Por ello, se ha asociado al modo DeepSleep e Hibernate del microcontrolador, permitiendo poner a dormir el microcontrolador activando una alarma en el RTC para poder despertarlo. Los intervalos pueden ir desde los 8 s en modo Deep Sleep, hasta minutos, horas o incluso días en Hibernate.

## 4.10 Arquitectura y Sistema

La arquitectura de Waspote está basada en el microcontrolador Atmel ATMEGA1281. Esta unidad de procesamiento comienza ejecutando el bootloader binario, el cual es responsable de cargar los programas compilados y almacenados previamente en la memoria FLASH, así pues, el programa principal que ha sido creado puede empezar a correr.

Cuando encendemos el Waspote y empieza el *bootloader* hay un tiempo de espera de 62.5 ms antes de comenzar a ejecutar la primera instrucción. Este tiempo es usado para empezar a cargar el programa subido a memoria previamente. Si un nuevo programa se recibe a través del USB durante este tiempo, será subido sobre la memoria FLASH (128 kB) sustituyendo a programas ya existentes. Si no se recibe ningún programa nuevo, se cargará el último programa en la memoria y empezará a ejecutarse.

La estructura de los códigos se divide en dos partes: una parte llamada *setup* y otra *loop*. Ambas partes del código tienen un comportamiento secuencial, es decir las instrucciones son ejecutadas respetando el orden en el que se van encontrando.

La función *setup* es la primera parte del código, ésta es ejecutada una sola vez al comienzo de cada programa. Aquí se incluye la inicialización de todos los módulos que van a ser usados, además es la parte del código importante cuando se enciende la Waspote.

Por otra parte, la función *loop* es ejecutada recursivamente como un bucle infinito. El porqué de este comportamiento lo encontramos en que facilita el uso de interrupciones para detectar eventos con Waspote.

Por ejemplo, en la Figura 4-7 se muestra la estructura anterior:

```

char* pointer;

void setup()
{
  // Init USB
  USB.ON();
  USB.println(F("UT_05 example"));

  USB.print(F("Initial free Memory (Bytes):"));
  USB.println(freeMemory());
  USB.print(F("-----"));
}

void loop()
{
  // See how dynamic memory consumes RAM memory during code
  pointer = (char*)calloc(1,100);

  USB.print(F("\nfree Memory:"));
  USB.println(freeMemory());

  // free memory
  free(pointer);
  pointer=NULL;

  delay(3000);
}

```

Figura 4-7. Código de ejemplo para Waspnote

Una técnica común para ahorrar energía estaría basada en bloquear el programa hasta que alguna de las interrupciones disponibles en Waspnote muestre que un evento ha ocurrido. De esta forma, cuando una interrupción es detectada, la función asociada, que fue previamente guardada en un vector de interrupción es ejecutada. Para ser capaz de detectar la captura de interrupciones durante la ejecución del código, una serie de banderas han sido creadas y serán activadas para indicar el evento concreto que ha generado la interrupción.

Cuando la Waspnote es reseteada o encendida, el código empieza otra vez desde el bloque *setup* y luego *loop*. Por defecto, los valores de las variables declaradas en el código y modificados en ejecución serán perdidos cuando se haga un reset o no haya batería. Para guardar valores permanentemente, es necesario usar la memoria no volátil del microcontrolador EEPROM (4 kB). Otra opción es usar la tarjeta SD de 2 GB.

Waspnote usa un oscilador de cuarzo que trabaja a una frecuencia de 8 MHz. De esta forma cada 125 ns el microcontrolador ejecuta una instrucción de bajo nivel (lenguaje máquina). Se debe tener en cuenta que cada línea de un programa compilado en un lenguaje de programación de alto nivel equivale a muchas instrucciones en lenguaje máquina.

Waspnote es un dispositivo preparado para la operación en condiciones adversas con ruido y contaminación electromagnética, por esta razón, para establecer una comunicación estable todo el tiempo con los diferentes módulos conectados a través de las UARTs (XBee, GPRS/3G, USB) se ha fijado una velocidad máxima de transmisión a 38400 bps para XBee, GPRS y USB, y 4800 bps para el GPS.

## 4.11 Interrupciones

A instancias del microcontrolador, todos los tipos de interrupciones las recibe como interrupciones *hardware*. Para ello se usan los pines de la UART (RXD1 y TXD1).

Dichas interrupciones se utilizan normalmente para despertar al Waspnote de un estado de bajo consumo y ejecutar una rutina de interrupción, minimizando el consumo en el caso de funcionamiento sólo con baterías.

Waspnote dispone de seis posibles señales de interrupción: Watchdog, RTC, acelerómetro, GPRS, sensores y batería crítica como se puede ver en la Figura 4-8.

Las señales de interrupción procedentes del RTC, acelerómetro, GPRS y sensores se multiplexan en el pin RXD1. Debido a esta multiplexación de varias señales de interrupción se han creado una serie de vectores de *flags* para permitir conocer el módulo que ha generado la interrupción para poder llamar a la rutina de atención a la interrupción correspondiente.

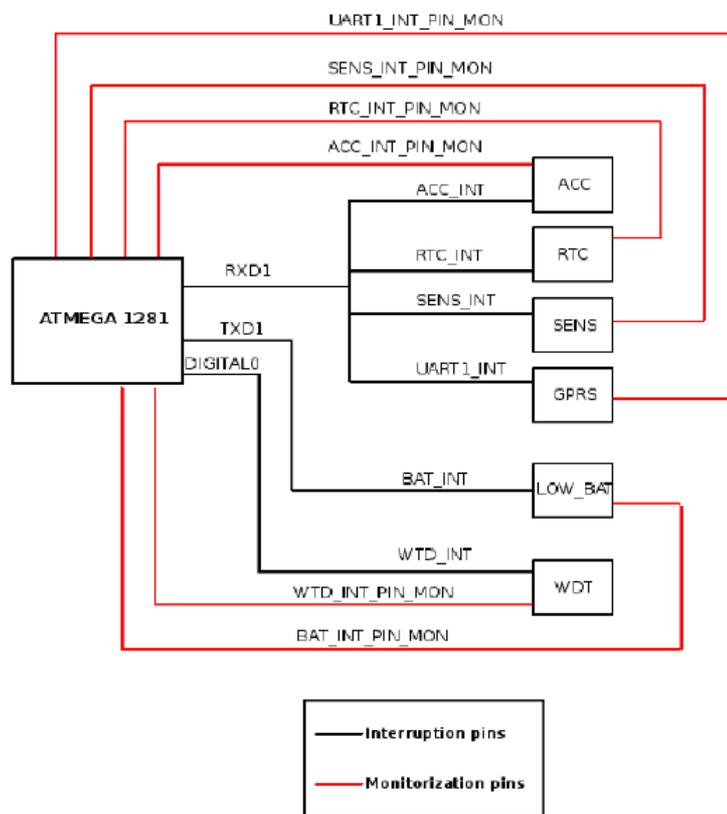


Figura 4-8. Fuentes de interrupción para el Microcontrolador.

## 4.12 Waspnote IDE

Waspnote IDE es un entorno de desarrollo basado en Arduino bastante simple. Su apariencia se muestra en la Figura 4-9.

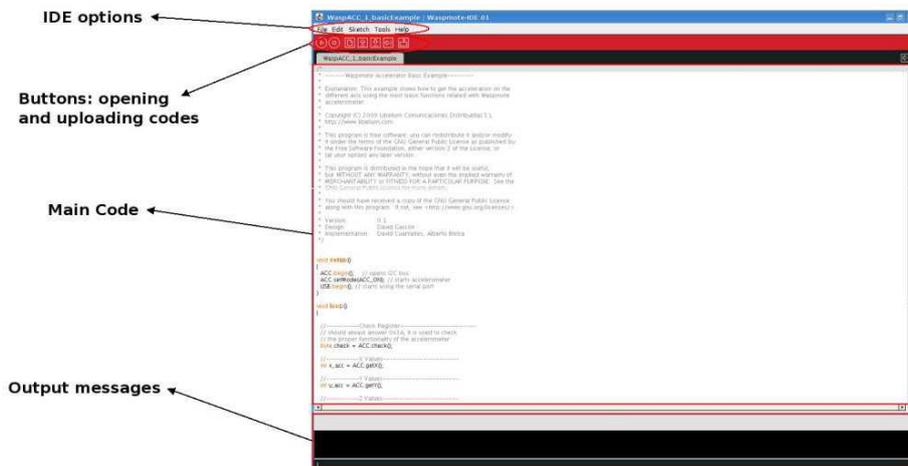


Figura 4-9. Vista del Waspnote IDE.

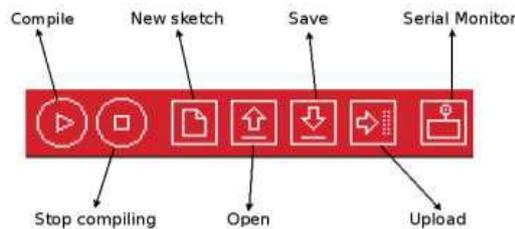


Figura 4-10. Botones del Waspnote IDE.

- **Opciones de la IDE:** Permiten seleccionar la placa utilizada y el puerto concreto entre otras opciones de configuración de Waspnote.
- **Botones:** La IDE incluye botones de compilado del código, de apertura y de guardado del mismo, así como de subida o carga del programa sobre el microcontrolador. También incluye un botón para abrir el monitor serial que permite observar el resultado de los programas implementados.
- **Código principal:** Aquí se encuentra el código del programa que se desea implementar sobre la Waspnote. El programa está dividido en dos partes “setup” y “loop”. La primera se ejecuta una sola vez cuando el programa comienza, mientras que la segunda es ejecutada por el microcontrolador indefinidamente como ya se comentó en la [Sección 4.10](#).
- **Mensajes de salida:** En esta parte aparecen mensajes de éxito o error en el proceso de compilación o carga del programa.

### 4.13 Programación básica de una Wasmote

Los pasos que debemos seguir para programar una Wasmote son los siguientes:

**Paso 1:** Retirar el modulo Xbee de la Wasmote. El USB usado para conectar la Wasmote al PC y el módulo XBee utilizan el mismo bus por lo que solo puede ser utilizado uno de ellos a la vez.

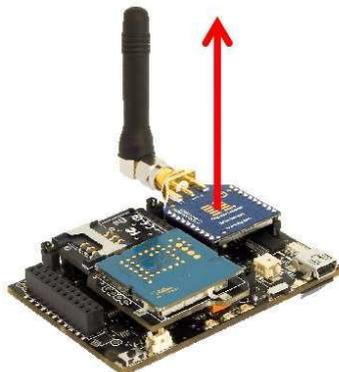


Figura 4-11. Paso 1.

**Paso 2:** Conectar la Wasmote con el cable miniUSB al PC.

**Paso 3:** Encender la Wasmote (ON/OFF -> interruptor izquierda/derecha).

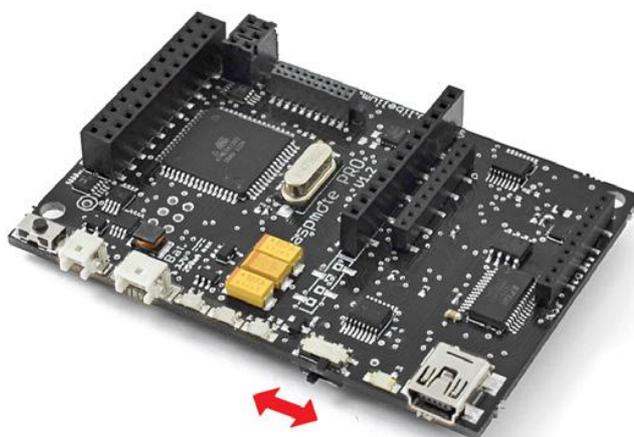


Figura 4-12. Paso 3.

**Paso 4:** Abrir Wasmote IDE y hacer Tools -> Board para seleccionar la placa WASP utilizada. Después hacer Tools -> Serial Port para seleccionar el puerto serie (tty/USBx en Linux, COMx en Windows) asociado al puerto USB donde se ha conectado la mota.

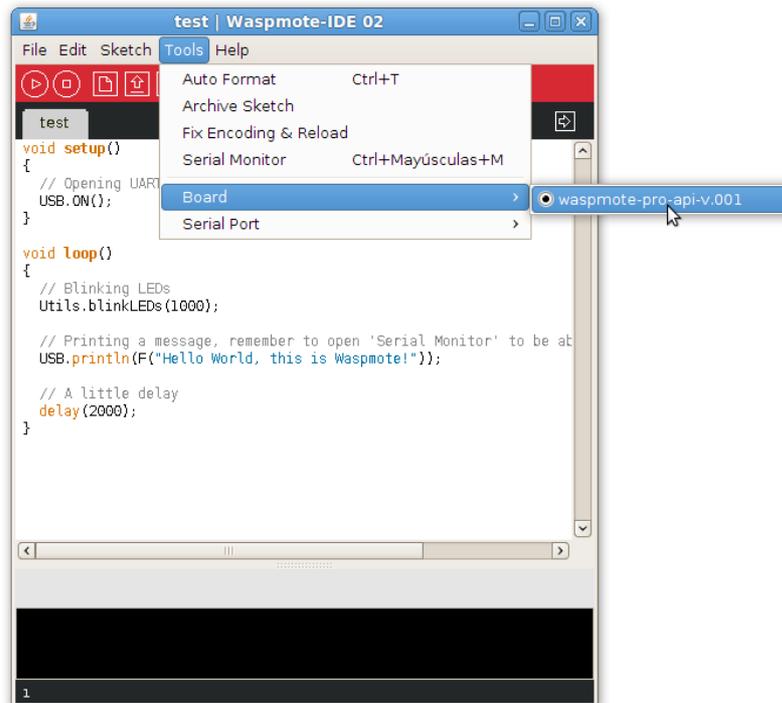


Figura 4-13. Paso 4.

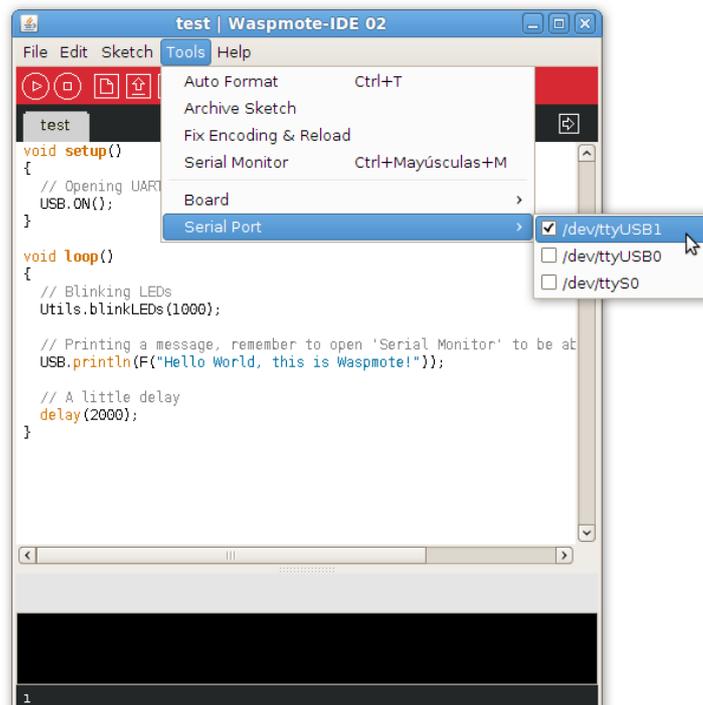


Figura 4-14. Paso 4.

**Paso 5:** Escribir código fuente en el espacio principal del IDE. Guardar el *sketch*.

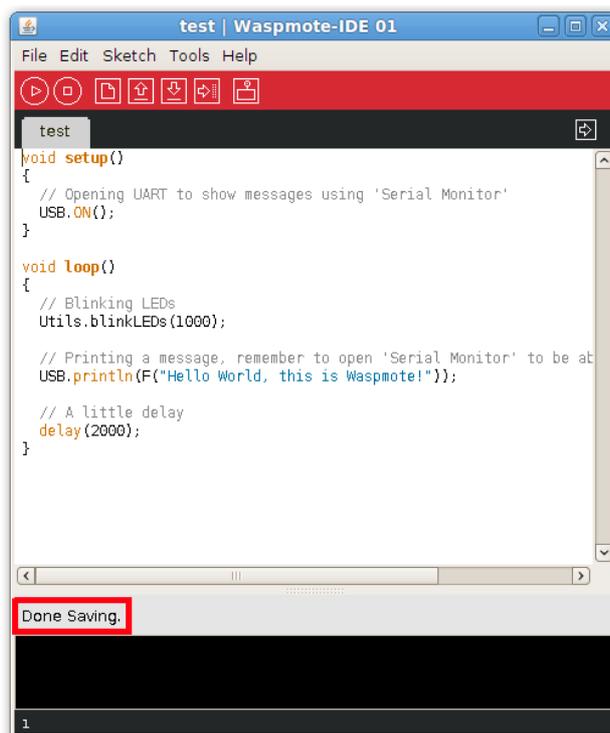


Figura 4-15.Paso 5.

**Paso 6:** Hacer Click sobre en el botón de carga (*upload button*) y esperar hasta que el programa se haya compilado y cargado en la memoria del microcontrolador. Al final si todo ha ido bien veremos el mensaje de carga correcta (*Done Uploading*) de lo contrario en el apartado de mensajes de salida aparecerán los errores en color rojo y deberán ser resueltos.



Figura 4-16. Paso 6.

**Paso 7:** Resetear la placa.

#### 4.14 Errores frecuentes al cargar programas en Wasmote

En la Figura 4-16 se muestra un error muy frecuente cuando se empieza a utilizar Wasmote. Se debe a que la placa no está encendida y por ello no puede cargarse el programa. Para solucionarlo no hay más que encender la Wasmote colocando el interruptor hacia la izquierda.



Figura 4-17. Error programando Wasmote.

Otro tipo de errores son los siguientes:

```
avrdude:stk500-getsync():not in sync:resp=0xXX
```

```
avrdude:stk500-disable():protocol error ,expect=0xXX, resp=0xXX
```

donde XX son distintos números dependiendo de cada caso particular. Para resolverlos responde estas preguntas:

- ¿Está encendida la Wasmote?
- ¿Has retirado el modulo Xbee? Como ya se comento anteriormente el modulo Xbee y el USB comparten la misma UART, por ello si el modulo Xbee no ha sido retirado no podrá cargarse el programa en memoria.
- ¿Están todos los *jumpers* en sus posiciones? Hay tres *jumpers* que deben ser conectados cuando programamos la placa.

Si las preguntas anteriores no ayudan, intenta desinstalar los Drivers FTDI y vuelve a instalarlos otra vez.

# 5 WASPMOTE GATEWAY

Este dispositivo permite recoger datos que fluyen a través de la red de sensores hacia un PC o dispositivo con un puerto USB estándar. La Wasmote *Gateway* o pasarela será un puente de datos o punto de acceso entre la red de sensores y el elemento receptor de datos. Dicho elemento será responsable de almacenar y usar los datos recibidos para las necesidades específicas de una aplicación.



Figura 5-1. Wasmote gateway con transceptor XBee.

## 5.1 Botones

La pasarela tiene los siguientes botones:

- Reset: permite al módulo XBee ser reseteado.
- I/O - 0: botón conectado al pin 0 de I/O del XBee.
- I/O - 1: botón conectado al pin 1 de I/O del XBee.
- RTS - I/O – 6: botón conectado al pin 6 de I/O del XBee.

Todos los botones conectan sus líneas de datos a tierra cuando son presionados. Ninguno de ellos dispone de resistencia de *pull-up*, por ello es necesario activar las resistencias internas *pull-up* del módulo XBee si la aplicación lo requiere.

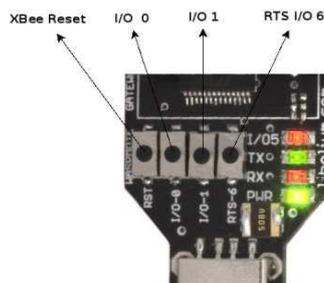


Figura 5-2. Botones gateway.

## 5.2 Leds

La pasarela consta de cuatro LEDs indicadores:

- USB power LED: indica que la placa está alimentada a través del puerto USB.
- RX LED: indica que la placa está recibiendo datos a través del puerto USB.
- TX LED: indica que la placa está transmitiendo datos a través del puerto USB.
- I/O 5 LED: este led es configurable. Está conectado al pin 5 I/O de XBee y puede ser configurado como salida digital de este o como indicador de asociación del modulo XBee a la red sensores.

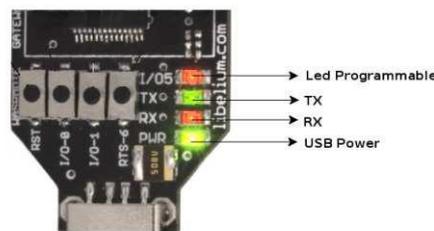


Figura 5-3. Leds de la *gateway*.

## 5.3 Configuración de módulos transceptores a través de la gateway

Para que la comunicación entre dos o más Waspnotes o entre Waspnotes y *gateway* pueda darse, es necesario configurar previamente los módulos transceptores XBee con algunos parámetros de la red como pueden ser canal frecuencial o PANID (*Personal Area Network Identification*).

Para escribir estos parámetros en los módulos XBee se utiliza la *gateway* como intermediario, es decir se conectan los transceptores sobre la *gateway* y esta a su vez al PC, donde por medio de la aplicación X-CTU se establece la configuración del módulo.

Todo este proceso se ha detallado en el [Capítulo 6](#) dedicado a los XBee. En la [Sección 6.4](#) se encontrará cómo configurar los módulos paso a paso.

## 5.4 Programas receptores de datos

Los programas receptores de datos son aplicaciones diseñadas para capturar lo que entra por el puerto serie.

Algunos de estos programas más populares son Hyperterminal para Windows, Cutecom para Linux o Zterm para Mac-OS. Si bien, el propio IDE de Waspnote también permite leer lo que llega a través del puerto serie para comprobar que se reciben los datos correctamente. En el caso de que queramos guardar la información recibida en un fichero con alguna extensión concreta podemos utilizar Realterm desde Windows.

Para abrir el *serial monitor* del Waspnote IDE solo tenemos que pulsar sobre el botón correspondiente. Este se ha destacado en la Figura 5-4.

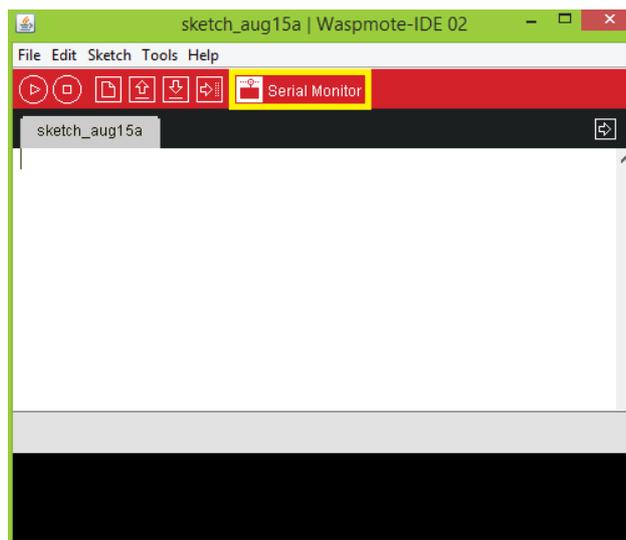


Figura 5-4. *Serial monitor* del Waspote IDE.

Previamente hemos tenido que conectar la *gateway* con el módulo transceptor XBee al puerto USB y haberlo seleccionado desde la pestaña *Tools* del IDE como se indicó en la Sección 4.13.

Si el proceso se ha hecho correctamente empezaremos a recibir los datos enviados y a mostrarse en el monitor. Un punto importante para visualizar correctamente los datos es la selección del *baud rate* cuyo valor debe coincidir con el configurado en los módulos XBee. Tal y como se comentó en el apartado anterior la configuración de los XBee se detalla en la [Sección 6.4](#).

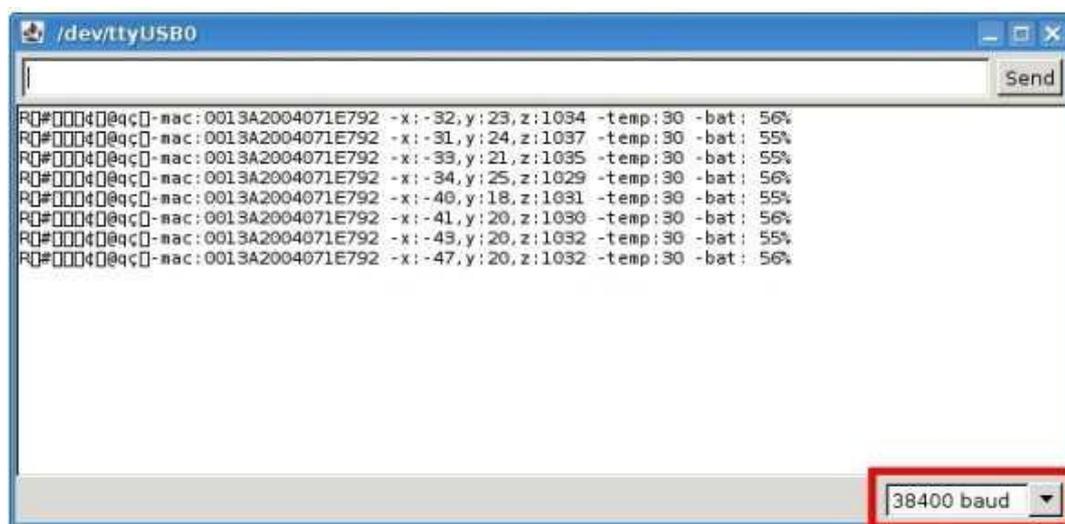


Figura 5-5. Datos capturados en la *gateway* y visualizados en el serial monitor de la Waspote IDE.

En la Figura 5-5 se muestra el resultado de un programa implementado en una Waspote. Esta envía su dirección MAC, la posición en los tres ejes según el acelerómetro, la temperatura del lugar donde se encuentra y el nivel de batería actual hacia la *gateway*. La *gateway* envía todo lo recibido a través del puerto serie del PC y este visualiza esos datos por medio del *serial monitor*.

Cada paquete enviado desde la mota a la *gateway* se muestra en una línea distinta. Al principio de cada línea aparecen caracteres extraños y no son más que las cabeceras del protocolo utilizado en la WSN. Después de la cabecera vienen los datos útiles.

## 5.5 XBee Explorer



Figura 5-6. XBee Explorer.

Este dispositivo permite conectar un modulo XBee directamente a un PC mediante un cable USB. Puede realizar la función de la Wasmote *gateway* pero es mucho más rápido transfiriendo datos, por eso se utiliza en muchas ocasiones sustituyéndola. Como desventaja no dispone de botón de *reset* aunque este puede hacerse de forma manual con un cable, puenteando los pines de RST y GND tal como se muestra en la Figura 5-7.

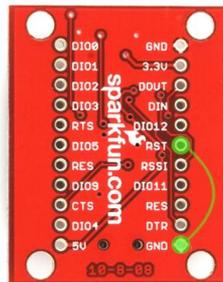


Figura 5-7. *Reset* manual en XBee Explorer.

## 5.6 Over the air programming (OTAP)

El concepto de programación inalámbrica, comúnmente conocida como *Over the Air Programming* (OTAP) se ha utilizado en los últimos años para la reprogramación de los dispositivos móviles. Sin embargo, con los nuevos conceptos de Redes Inalámbricas de Sensores y el Internet de las cosas, donde las redes se componen de cientos o miles de nodos, OTAP toma una nueva dirección, y ahora se aplica tanto a las tecnologías de telefonía móvil tales como 3G y GPRS o protocolos en la banda libre como WiFi, 802.15.4 y ZigBee.

### 5.6.1 Conceptos

La idea es simple. Cuando el programador (normalmente la *gateway*) envía un nuevo programa, este es almacenado en la tarjeta SD de los nodos a programar. Después del comando *start\_new\_program* los nodos copian el programa de la tarjeta SD a la memoria Flash y comienzan a ejecutar el programa.

### 5.6.2 Ventajas

- Permite la actualización y cambio de versiones *firmware* sin acceso físico.
- Descubriremos los nodos de la zona con solo enviar una petición de detección de difusión.
- Permite actualizar una red completa en pocos minutos.

- No hay interferencias: OTAP se realiza en canales distintos entre el programador (*gateway*) y cada el nodo.

### 5.6.3 Topologías

- Acceso directo: cuando se accede a los nodos en un solo salto (no se necesita el reenvío de los paquetes).
- Multisalto: cuando se accede a los nodos en dos o más saltos. En este modo algunos nodos tienen que reenviar los paquetes enviados por la *gateway* con el fin de llegar al destino.

### 5.6.4 Cifrado y autenticación

Todos los datos que se envían en el proceso OTAP se pueden asegurar mediante la activación de los algoritmos de cifrado AES 128b que trabajan en la capa de enlace. Además de esto, se necesita una segunda clave para ser conocido por el programador OTA (la *gateway*) con el fin de ser autenticado y validado cada nodo antes de comenzar con la acción OTAP solicitada.

### 5.6.5 Modos

- *Unicast*: Reprogramar un nodo específico.
- *Multicast*: Reprogramar varios nodos al mismo tiempo enviando el programa una vez.
- *Broadcast*: Reprogramar toda la red enviando el programa una vez.

### 5.6.6 Sistema de almacenamiento

Una vez que hemos enviado el programa a una Waspote este se almacenará en la memoria interna, una tarjeta SD de 2 GB. Si tenemos en cuenta que el tamaño máximo de un programa es de 128 kB, esto significa que podemos almacenar miles de versiones de firmware diferentes dentro de cada nodo.

### 5.6.7 OTA-Shell

La aplicación OTA-Shell permite controlar de una manera rápida y eficiente todas las opciones disponibles sobre OTA. Está disponible para Windows, Linux y MacOS.

### 5.6.8 OTA 802.15.4/Zigbee paso a paso

#### 5.6.8.1 Localizar el nodo o nodos a actualizar

Usando la función *scan\_nodes* podemos buscar un nodo específico o enviar una petición global buscando cualquier nodo que esté preparado para ser reprogramado.

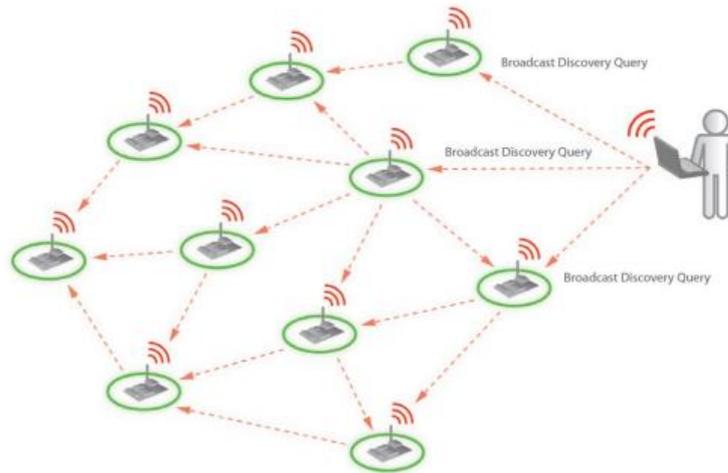


Figura 5-8. Petición global para descubrir nodos.

Los nodos que estén preparados en ese momento responderán con una trama *Ready to OTA*.

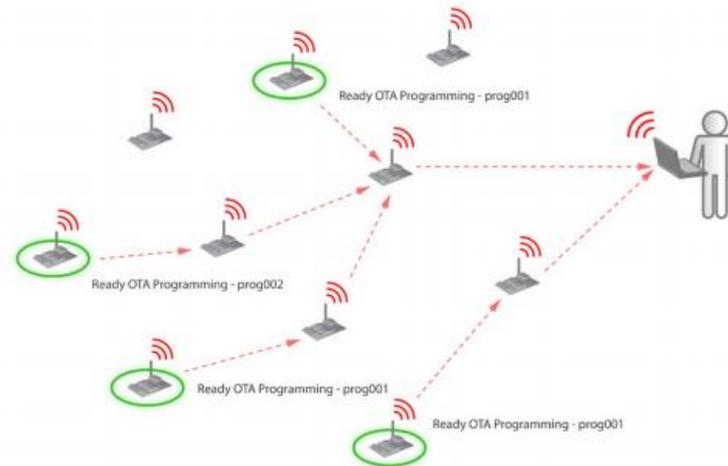


Figura 5-9. Respuesta OTA de los nodos preparados para la programación.

### 5.6.8.2 Enviando un nuevo programa

Podemos utilizar el comando *send* con la opción *unicast*, *multicast* o *broadcast* dependiendo de cuántos nodos queramos reprogramar al mismo tiempo.



Figura 5-10. Enviando el nuevo programa via OTA.

Cada nodo que recibe el programa envía un mensaje a la *gateway* para informar sobre el éxito del proceso.



Figura 5-11. Respuesta de las Wasmotes sobre proceso OTA exitoso.

### 5.6.8.3 Reinicio y comienzo del nuevo programa

Para hacer que los nodos ejecuten el nuevo programa, la *gateway* necesita enviar el comando *start\_new\_program*.

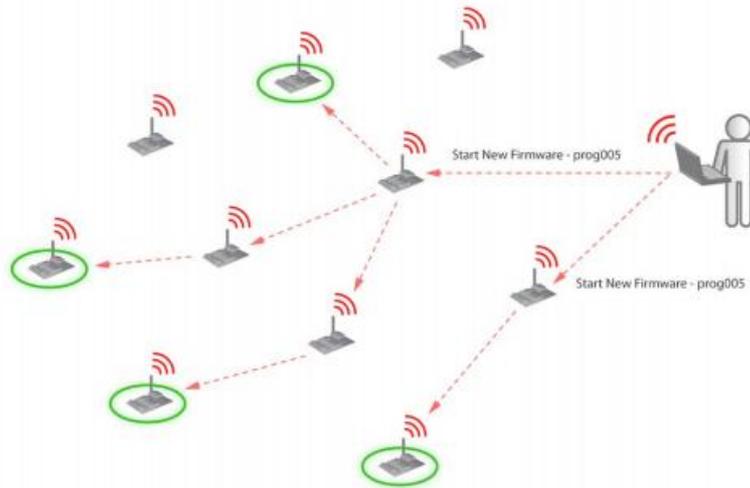


Figura 5-12. Orden de la *gateway* hacia algunas Wasmotes para ejecutar el nuevo programa.

Cada nodo que recibe este paquete copiará el programa desde la SD a la memoria Flash y empezará a ejecutar el nuevo programa.

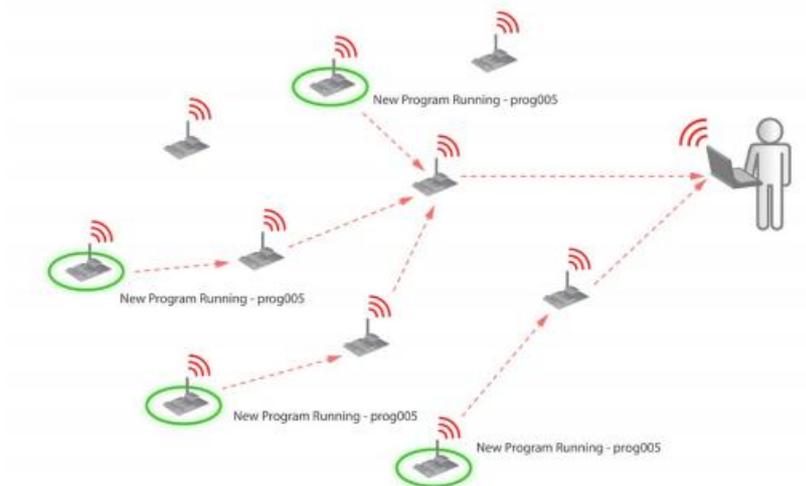


Figura 5-13. Confirmación de que las Wasmotes han comenzado a ejecutar el nuevo programa.

Para obtener más información sobre OTA ver [\[10\]](#).

# 6 XBEE

En este capítulo se describen algunas características generales de los módulos de RF XBee de Digi que se montaran sobre las Wasmote utilizadas en este TFG. Además se describe paso a paso la configuración de los mismos con la herramienta del fabricante llamada X-CTU, así como la restauración o actualización del *firmware* de los módulos.

## 6.1 Introducción

De acuerdo a Digi, los módulos XBee son soluciones integradas que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos. Estos módulos utilizan como base el protocolo llamado IEEE 802.15.4 para crear redes punto a multipunto (*Point to Multipoint*) o para redes punto a punto (*Peer to Peer*). Fueron diseñados para aplicaciones que requieren de un alto tráfico de datos, baja latencia y una sincronización de comunicación predecible.

En términos simples, los XBee son módulos inalámbricos que integran un transmisor - receptor y un procesador en el mismo módulo, lo que le permite a los usuarios desarrollar aplicaciones de manera rápida y sencilla.

Es especialmente útil para redes de sensores en entornos industriales, médicos y, sobre todo domóticos.

Las comunicaciones se realizan principalmente en la banda libre de 2.4 GHz, aunque hay más posibilidades. A diferencia de *bluetooth* no utiliza FHSS (*Frequency hopping Spread Spectrum*), sino que realiza las comunicaciones a través de una única frecuencia, es decir, de un canal. Normalmente puede escogerse un canal de entre 16 posibles. El alcance depende de la potencia de emisión del dispositivo así como del tipo de antenas utilizadas (cerámicas, dipolos,...). El alcance normal con antena dipolo en visión directa suele ser aproximadamente (tomando como ejemplo el caso de MaxStream, en la versión de 1 mW de potencia) de 100 m y en interiores de unos 30 m. La velocidad de transmisión de datos suele ser de hasta unos 250 kbps. La red puede estar formada, teóricamente por hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad de dispositivos. La realidad es menor, aunque sigue siendo, de miles de equipos.

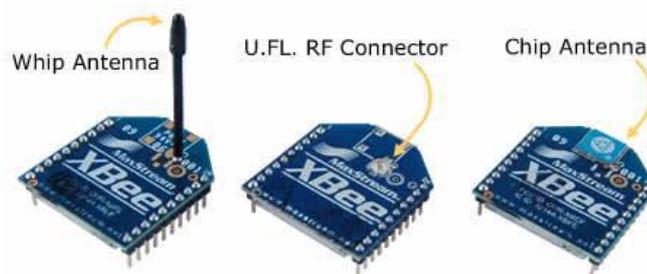


Figura 6-1. Módulos XBee y antenas.

Los módulos Xbee son económicos, potentes y fáciles de utilizar. Algunas de sus principales características son:

- Buen Alcance: hasta 100 m en LoS para los módulos Xbee y hasta 1.6 km para los módulos Xbee Pro.
- 9 pines entradas/salidas con entradas analógicas y digitales.

- Bajo consumo, inferior a 50 mA cuando están en funcionamiento e inferior a 10 uA cuando están en modo *sleep*.
- Interfaz serial.
- 65000 direcciones para cada uno de los 16 canales disponibles. Las redes pueden estar formadas por gran cantidad de dispositivos.
- Fáciles de integrar.

Existen dos series de estos módulos. La serie 1 y la serie 2. Los módulos de la serie 1 y la serie 2 tienen el mismo *pin-out*, sin embargo, no son compatibles entre sí ya que utilizan distintos *chipset* y trabajan con protocolos diferentes.

La serie 1 está basada en el *chipset* Freescale y está pensado para ser utilizado en redes punto a punto y punto a multipunto. Los módulos de la serie 2 están basados en el *chipset* de Ember y están diseñados para ser utilizados en aplicaciones que requieren repetidores o una red *mesh*. Ambos módulos pueden ser utilizados en los modos AT (*Transparent*) y API (*Application Programming Interface*).

Existen módulos con características mejoradas como por ejemplo las versiones *PRO*, entre otros tantos. Algunas de estas mejoras suelen ser las potencias de emisión y por tanto los alcances, la frecuencia de emisión, la sensibilidad del receptor o los protocolos admitidos (802.15.4, Zigbee, DigiMesh).

## 6.2 Arquitectura básica de una red XBee

Como se comentó en el [Punto 2.3.5](#) y en la [Sección 2.4](#) del [Capítulo 2](#), las WSNs están formadas básicamente por tres tipos de elementos. Un único dispositivo coordinador, dispositivos *routers* y dispositivos finales (*end points*).

Los módulos XBee son versátiles a la hora de establecer diversas topologías de red, dependiendo la serie de XBee que escojamos pueden crearse redes:

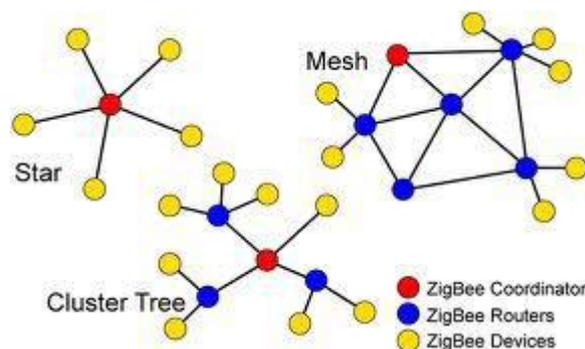


Figura 6-2. Topologías de red.

- Punto a punto
- Estrella
- Malla
- Árbol

- Mixtas

Una vez decidida la red con la que trabajar se deben escoger los módulos XBee. Tendremos varias posibilidades en función de la aplicación que vayamos a realizar. Las dos más simples son usar todo en Serie 1 o todo en Serie 2 pero teniendo en cuenta que podemos encontrar las versiones mejoradas, como ya se comentó anteriormente.

Para la aplicación de localización en interiores que se desarrolla en este TFG (ver [Capítulo 7](#)) se utilizarán los módulos XBee Serie 1 montados sobre Wasmote.

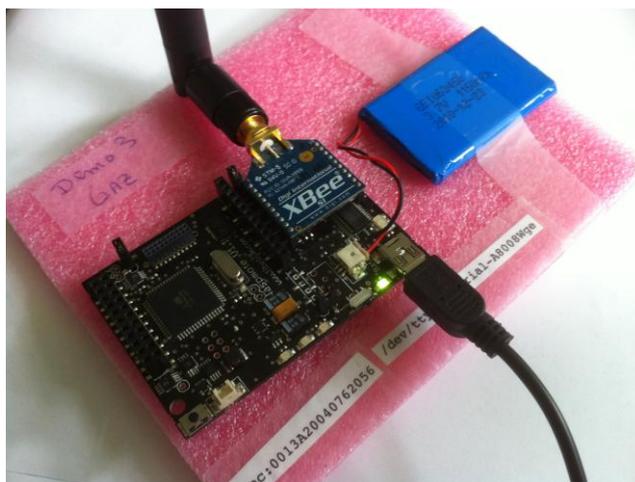


Figura 6-3. XBee y Wasmote.

### 6.3 Modo de operación de un XBee

Los módulos pueden ser configurados para trabajar en distintos modos.

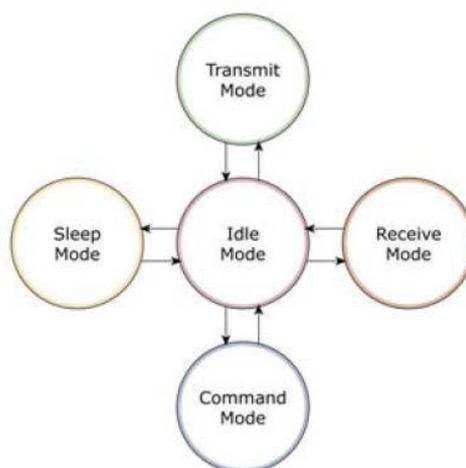


Figura 6-4. Modos de operación de un XBee.

#### 6.3.1 Modo recibir/transmitir.

Se encuentra en este modo cuando al módulo le llega algún paquete RF a través de la antena (modo *Receive*) o cuando se manda información serial al buffer del pin 3 (*UART Data in*) que luego será transmitida (modo

*Transmit*).

La información transmitida puede ser Directa o Indirecta. En el modo directo la información se envía inmediatamente a la dirección de destino. En el modo Indirecto la información es retenida durante un período de tiempo y es enviada sólo cuando la dirección de destino la solicita. Además es posible enviar información de dos formas *Unicast* y *Broadcast*. En la primera la comunicación es punto a punto, y es el único modo que permite respuesta de quien recibe el paquete, es decir, quien recibe puede enviar un ACK (*Acknowledge*) para confirmar esa recepción a la dirección de origen. Quien envió el paquete, espera recibir un ACK, en caso de que no le llegue, reenviará el paquete hasta tres veces o hasta que reciba el ACK. En el modo *Broadcast* la comunicación es entre un nodo y todos los nodos de la red. En este modo, no hay confirmación por ACK.

### 6.3.2 Modo de bajo consumo (sleep mode)

El modo de sueño hace posible que el módulo RF entre en un modo de bajo consumo de energía cuando no se encuentra en uso.

### 6.3.3 Modo de comando

Este modo permite ingresar comandos AT al módulo Xbee, para configurar, ajustar o modificar parámetros. Permite ajustar parámetros como la dirección propia o la de destino, así como su modo de operación entre otras cosas. Para poder ingresar los comandos AT es necesario utilizar el Hyperterminal de Windows, el programa X-CTU o algún microcontrolador que maneje UART y tenga los comandos guardados en memoria o los adquiera de alguna otra forma.

### 6.3.4 IDLE

Cuando el módulo no está en ninguno de los otros modos, se encuentra en éste. Es decir, si no está ni transmitiendo ni recibiendo, ni ahorrando energía ni en el modo de comandos, entonces se dice que se encuentra en un estado al que se le llama IDLE.

## 6.4 Configuración de módulos XBee con X-CTU

X-CTU es la herramienta de DIGI para la configuración de los módulos de RF XBee. Si bien se pueden configurar muchísimos parámetros desde X-CTU, aquí mostraremos los parámetros más básicos y los que hemos tenido que establecer para la aplicación que se ha abordado.

En primer lugar para que los módulos transceptores pertenezcan a la misma red de sensores hay que establecer dos parámetros: PANID y canal frecuencial.

Si dos módulos tienen configurados el mismo PANID pero distinto canal frecuencial no pertenecen a la misma red. Lo mismo ocurre si tienen el mismo canal frecuencial pero distinto PANID. Es decir, para que los módulos que formen la red de sensores estén en la misma red son necesarios que tengan el mismo PANID y canal frecuencial.

En segundo lugar habría que establecer la dirección destino de la comunicación, si la comunicación es punto a punto, sería la del extremo receptor de la comunicación. Para una comunicación en difusión, habría que establecer la dirección de *broadcast* como dirección destino.

En tercer lugar la velocidad de transmisión de la interfaz la fijamos a 38400 bps. Se pueden configurar otras, pero es importante que todos los módulos tengan la misma.

En cuarto y último lugar el API *mode* debe ser fijado con un valor de 2.

A continuación mostramos lo anterior gráficamente:

En la Figura 6-5 se muestra la apariencia inicial al abrir X-CTU.

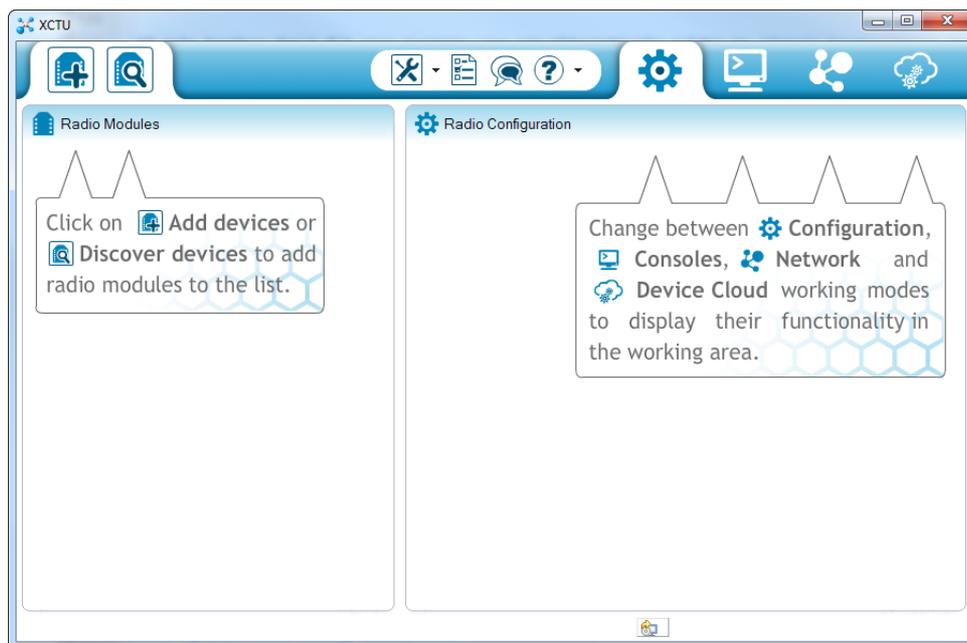


Figura 6-5. Pantalla inicial X-CTU.

Para añadir los módulos XBee primeramente debemos conectarlos al puerto USB mediante la *gateway* o el USB Explorer. Hay que tener en cuenta que en el proceso de configuración se nos puede pedir en más de una ocasión que resetemos los módulos y el USB Explorer no tiene un botón físico para ello como la *gateway*. Recordemos que si usamos este dispositivo debemos hacer el *reset* de forma manual con un cable que una los pines RST y GND.

Una vez que el módulo está conectado al PC, pulsamos el botón  y aparecerá una pantalla como la mostrada en la Figura 6-6. En este punto tenemos que seleccionar el puerto de comunicación donde hemos conectado el dispositivo y luego hacer *Finish*.

Si el módulo ha sido añadido correctamente, aparecerá en la ventana izquierda llamada *Radio Modules*. Seguidamente hacemos Click sobre el módulo y en la ventana derecha *Radio Configuration* se desplegarán los parámetros configurables.

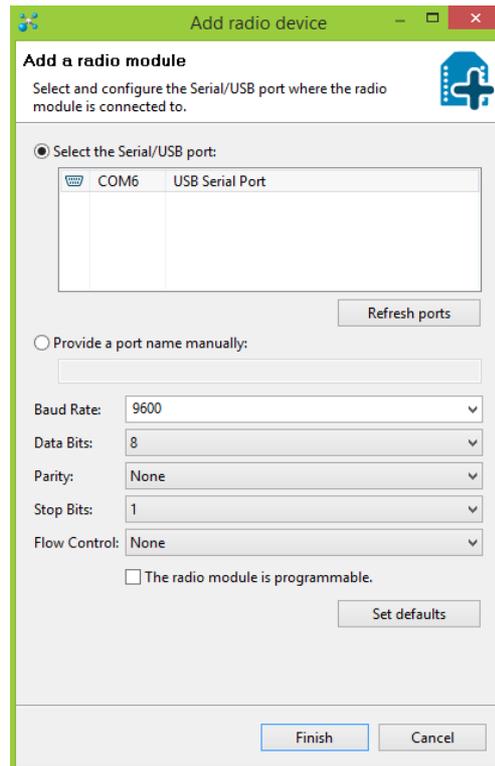


Figura 6-6. Selección puerto de comunicación del módulo XBee.

En la Figura 6-7 y Figura 6-8 se muestran recuadrados en rojo los parámetros básicos a configurar y que se comentaron al principio de esta sección.

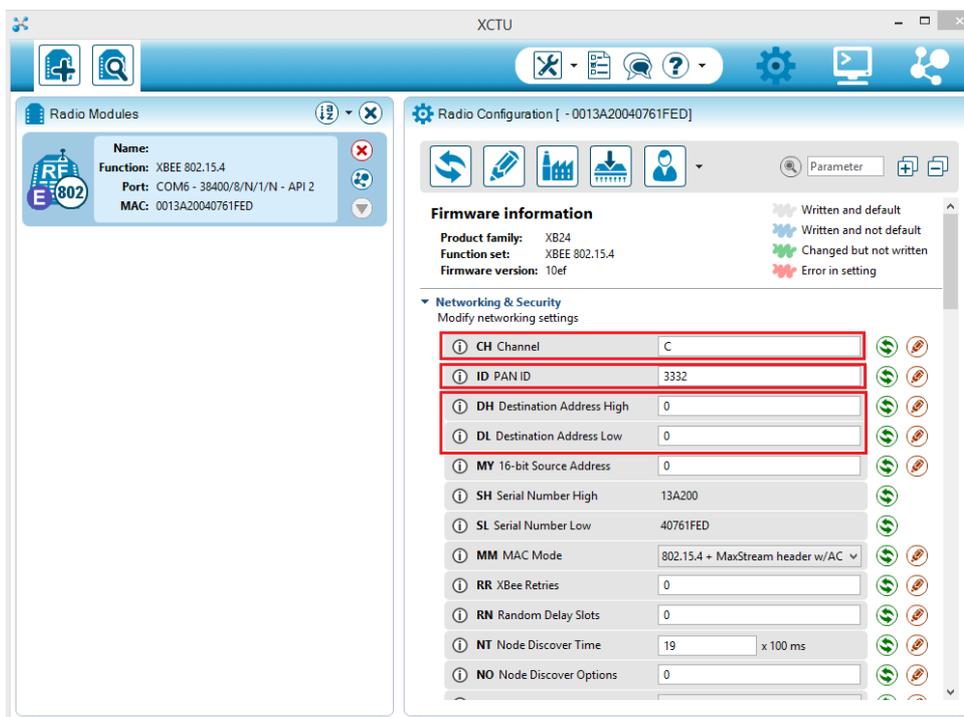


Figura 6-7. Parámetros configurables: CH, PANID, Destination address.

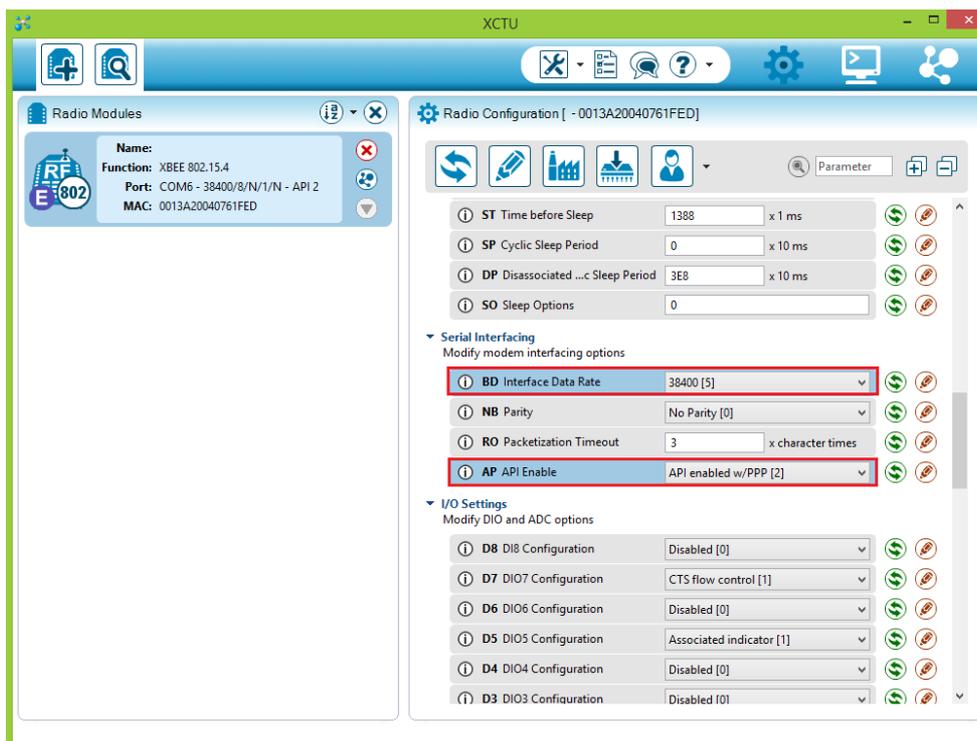


Figura 6-8. Parámetros configurables: BD y AP.

### 6.5 Recuperacion y actualizacion de firmware

X-CTU incluye una herramienta para recuperar módulos XBee que no responden o funcionan correctamente. Cuando esto ocurre podemos utilizar la *XBee Recovery* para cargar de nuevo el *firmware*.

En la Figura 6-9 y Figura 6-10 podemos observar dónde encontrar la herramienta y la facilidad de carga del nuevo *firmware*. Una vez seleccionado el puerto de comunicación, la familia y versión del *firmware* hacemos Click en el botón *Recover* y esperamos que termine el proceso.

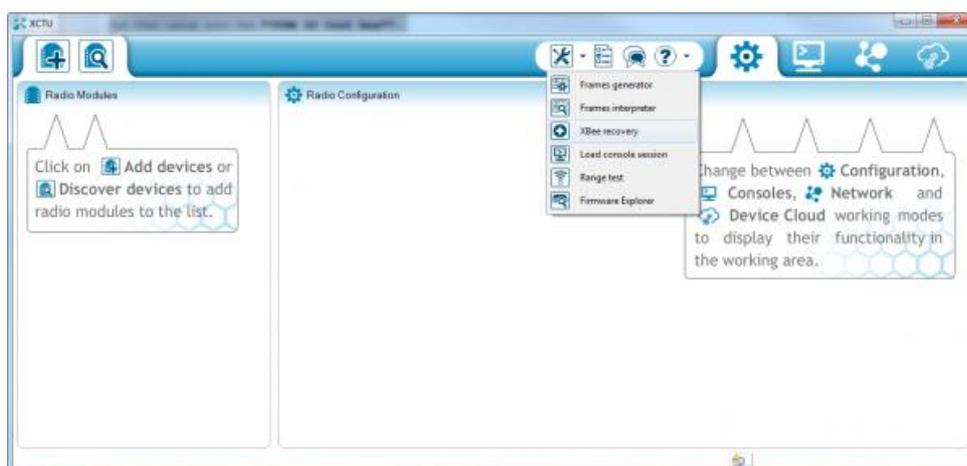


Figura 6-9. Herramienta *XBee Recovery*.

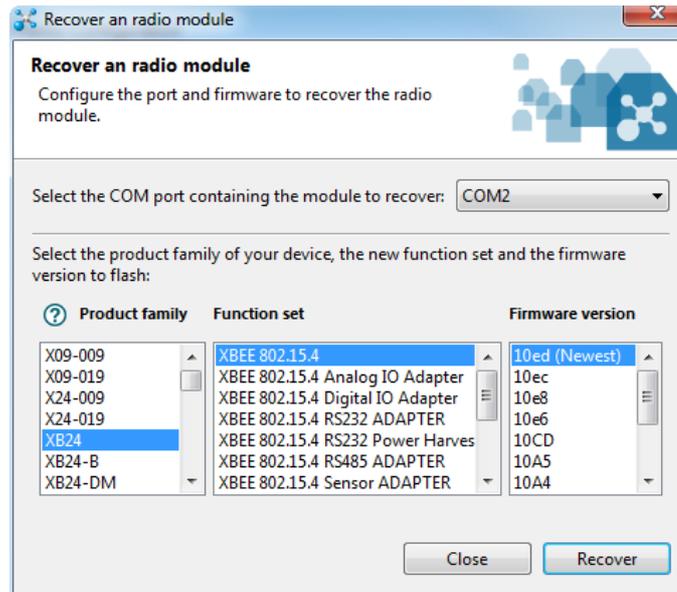


Figura 6-10. Selección puerto XBee bloqueado y selección nuevo *firmware*.

# 7 APLICACIÓN DE LOCALIZACIÓN EN INTERIORES CON REDES INALÁMBRICAS DE SENSORES

---

En este capítulo se va a desarrollar una aplicación para localización en entornos de interior haciendo uso de una red inalámbrica de sensores Wasmote. Los sensores o *motes*, como también se las conoce utilizan la técnica de reconocimiento de patrones junto a los parámetros RSSI para localizar a un nodo móvil de posición desconocida (OVNI) que se desplaza por varias estancias a baja velocidad. El proceso de localización es *OFFLINE*, es decir, no ocurre en tiempo real sino que el sistema toma datos durante un tiempo grande acerca del comportamiento o movimientos del nodo, y a posteriori se analiza dicha información con el fin de sacar conclusiones sobre aquello que estemos monitorizando que, quizá, pueda mejorar la toma de decisiones en una empresa o simplemente mejorar el conocimiento dentro de algún campo de la ingeniería.

## 7.1 Entorno de la aplicación

El entorno considerado para la aplicación serán tres estancias contiguas de una casa unifamiliar. Como es normal los habitáculos están separados por muros unos de otros y cada uno de ellos contienen los elementos típicos del tipo de estancia al que pertenecen, como pueden ser muebles, electrodomésticos, adornos, etc.

Concretamente distinguiremos tres zonas donde el OVNI podría ser localizado:

Zona 1: Salita

Zona 2: Cocina

Zona 3: Dormitorio

Dado que solo se considera esta partición de la vivienda en la aplicación, fuera de estas zonas el OVNI no puede ser localizado.

Ni que decir tiene que la aplicación podría extenderse a más habitáculos y otros entornos distintos aunque habría que plantearse de nuevo aspectos sobre la propagación en interiores, potencia de transmisión, alcance, obstáculos, interferencias, entre otros y hacer las modificaciones oportunas en la configuración del Hardware y Software (programas) que aquí se han creado con el fin de adaptarnos al nuevo ambiente donde se pretende que la aplicación funcione.

## 7.2 Descripción general

La red estará formada por cinco nodos Wasmote, la *gateway* y un PC que hará de estación base. Los módulos de RF o transceptores montados sobre los nodos serán los XBee de la Serie 1 basados en el estándar 802.15.4 y cuyas especificaciones se pueden encontrar en el Anexo A de este TFG.

De las cinco Wasmote, una de ellas hará de nodo OVNI y será el nodo móvil que se desplazará por las diferentes estancias y cuyo objetivo final es localizar y *trackear* a partir de la señal periódica que emite al aire. Los otros cuatro *motes*, que denominaremos en el contexto de la aplicación balizas, serán nodos de posición fija, aunque ubicadas de forma estratégica, ya que el éxito y precisión de la aplicación como veremos más adelante, va a depender en gran medida de la elección de esas localizaciones. La función de las balizas es básicamente medir la potencia (parámetros RSSI) desde sus posiciones fijas con la que se recibe la señal periódica del OVNI y luego enviar esas medidas al *gateway* para que la estación base determine la posición y desplazamientos del mismo en cada instante. La estación base utiliza el algoritmo de error cuadrático medio mínimo para el reconocimiento de patrones y estimación de la localización del OVNI.

Hay que aclarar que las balizas siempre tienen contacto directo con la *gateway*, por lo tanto, no es necesario hacer *routing* para que las medidas de potencia lleguen a ésta. Asimismo la señal del OVNI también es recibida de forma directa en la *gateway*, aunque será ignorada por el programa de la estación base al procesar los datos recibidos, ya que no nos será de utilidad.

### 7.3 Justificación de la solución empleada

La utilización de parámetros RSSI para el cómputo de la localización se ve justificada por ser la solución más simple conceptualmente y sobre todo, la más económica, al no necesitar ningún hardware adicional para llevarla a cabo.

Si bien la medición parámetros RSSI conlleva inconvenientes derivados de los fenómenos que tienen lugar durante la propagación de las señales de RF en interiores, éstos se pueden compensar utilizando como técnica de estimación de posición la técnica de reconocimiento de patrones.

### 7.4 Técnica de reconocimiento de patrones

En el [Punto 3.5.3](#) del [Capítulo 3](#), ya se comentó brevemente dicha técnica de reconocimiento de patrones. Ahora, dado que será importante para nuestra aplicación pasamos a describirla más en profundidad.

Como sabemos, la técnica de reconocimiento de patrones se lleva a cabo en dos fases bien diferenciadas, una fase de calibración y otra de localización.

#### 7.4.1 Fase de calibración

La calibración consiste en una fase de entrenamiento en la que se intenta caracterizar al entorno creando una base de datos de valores RSSI, a partir de las señales emitidas por el nodo OVNI y las recibidas en las balizas. Para comenzar esta fase, primeramente hay que elegir puntos del entorno donde situar al OVNI estático durante un tiempo mientras que transmite de forma periódica. Las balizas, a su vez, irán recibiendo esa señal desde su posición fija y tomando los niveles RSSI o potencia a partir de ella. Las medidas son enviadas a la *gateway* y luego transferidas a la estación base donde se recoge y analiza toda esa información para crear una base de datos, denominada matriz de calibración. La matriz de calibración estará formada entonces, por vectores de potencia o patrones que identifican unívocamente las distintas posiciones espaciales donde ha estado situado el OVNI. Será usada junto con el algoritmo de error cuadrático medio mínimo para detectar a qué patrón de potencia de los registrados se parece más la potencia instantánea del OVNI cuando se encontraba en un punto del entorno y así estimar la posición del nodo.

##### 7.4.1.1 Ubicación de las balizas: un aspecto clave

La ubicación de las balizas en el entorno considerado será un factor decisivo de cara a que la fase de localización sea satisfactoria.

Para lograr precisión y pocos fallos en la estimación de la localización hay que tener en cuenta la correlación que hay entre los patrones registrados en la matriz de calibración, que a su vez dependen del entorno y de la ubicación elegida para las balizas. Si la correlación de estos vectores es grande, la estimación de la localización no será correcta y se confundirán los patrones más parecidos, por el contrario si la correlación entre vectores es pequeña la estimación será mucho mejor.

Así pues, para que la matriz de calibración sea lo más realista posible y represente de la mejor manera al entorno, debemos optar por una matriz cuyos vectores presenten un coeficiente de correlación cruzada muy bajo, y además, la correlación que presente cada vector de calibración con el resto de vectores (patrones) del mismo habitáculo debe ser lo mayor posible para que el proceso de decisión sea coherente.

La forma de conseguir que los vectores tengan una correlación baja, sobre todo los de habitáculos o zonas distintas, es ubicar las balizas de forma irregular, intentando eliminar al máximo las posibles simetrías de potencia en los patrones que se puedan dar y que son causa de fallo en la estimación de la localización con esta

técnica.

En la Figura 7-3 se puede observar la disposición irregular de las balizas para evitar la correlación entre patrones y los puntos donde se han tomado los mismos para rellenar las entradas de la matriz de calibración.

#### 7.4.1.2 Matriz de calibración

La matriz de calibración tendrá tantas filas como patrones queramos registrar del entorno, digamos  $N$ , y tantas columnas como nodos localizadores o balizas usamos en la aplicación, digamos  $M$ .

Así pues, la matriz de calibración tendría la siguiente forma:

Cada vector fila de la matriz  $C$  corresponde a los niveles RSSI medidos por las balizas desde una posición estática y conocida del OVNI en el entorno que pretendemos caracterizar. Así pues, el elemento corresponde a la potencia con la que se recibe la señal del OVNI en la baliza  $j$  cuando el OVNI está en la posición  $i$  del entorno. En cada una de las posiciones referencia elegidas del entorno (ver Figura 7-3), el OVNI debe permanecer un tiempo suficientemente grande para que las balizas tomen varias veces la potencia. Una vez estas mediciones estén en la estación base (PC) se deben seleccionar los valores de cada entrada de la matriz  $C$ , ya sea tomando la media, la moda, o de alguna otra forma para cada una de las posiciones del entorno que hayamos decidido y para cada una de las balizas.

### 7.4.2 Fase de localización

Durante el funcionamiento de la aplicación, el comportamiento del nodo OVNI es *trackeado* si este permanece dentro de las zonas registradas en la matriz de calibración, en caso contrario su localización no tiene sentido y sería errónea.

La estación base para cada instante temporal crea un vector  $P$  que contiene las medidas de potencia realizadas por las balizas.

Este vector se define como sigue:

,

es decir, para cada instante temporal la estación base ejecuta el algoritmo de error cuadrático medio mínimo haciendo uso de la matriz de calibración  $C$  y el vector de potencia  $P$ .

#### 7.4.2.1 Localización mediante minimización del error cuadrático medio e identificación de patrón

Sean  $C$  y  $P$  la matriz de calibración y el vector de potencia recibida

en el instante  $t$ , respectivamente.

El algoritmo que minimiza el error cuadrático medio se implementa de la siguiente manera. Creamos una matriz temporal llamada  $T$ , definida como la diferencia entre la matriz de calibración y una matriz formada por la repetición del vector  $P$  para el instante  $t$ .

A continuación se elevan los elementos de la matriz T al cuadrado, sumamos las columnas entre si y almacenamos el resultado en una matriz R.

Por último, se estima la localización del OVNI como aquella donde el error cuadrático medio es menor, es decir, aquella donde la entrada del vector R sea más pequeña.

Localización =

## 7.5 Análisis de la solución

En este punto se realizará el análisis de la solución dada a la aplicación de localización en interiores con una red inalámbrica de sensores. Básicamente la solución se basa en la creación de tres programas, uno por cada tipo de nodo, OVNI (Wasmote), balizas (Wasmote) y estación base (PC).

A continuación, se comentan los códigos desarrollados en orden creciente de complejidad. Instamos al lector interesado hacia el [Anexo B](#) del TFG para analizar línea a línea los códigos de programación si así lo desea.

### 7.5.1 Código del nodo OVNI

En el código diseñado para el nodo a localizar (OVNI), éste transmite un paquete en difusión (sin carga de datos útiles) con un periodo de 1 segundo. Lo único que nos interesa es que el OVNI transmita periódicamente ese paquete para que las balizas puedan medir la potencia con la que reciben esa señal, por ello no es necesaria ninguna carga de datos útiles, a parte de las cabeceras del protocolo de comunicación, en este caso las de 802.15.4 y las de los módulos XBee empleados en las *motes* (ver [Anexo A](#)).

### 7.5.2 Código para los nodos baliza

En este código las balizas miden la potencia recibida a partir de la señal emitida periódicamente por el OVNI (paquete). Seguidamente, crean un nuevo paquete que incluye ese valor de potencia, la MAC de la baliza que realiza la medida y una marca de tiempo (*timestamp*) del instante en el que éstas envían el paquete hacia la *gateway*.

El *timestamp* se logra gracias al RTC (*Real Time Clock*) de la Wasmote que es inicializado al encender la baliza con una referencia temporal (fecha y hora) fijadas por nosotros en el código. Las cuatro balizas tendrán el mismo programa guardado en memoria y por ello tendrán configuradas la misma marca de tiempo, ahora bien, para que estén sincronizadas deberán ser arrancadas aproximadamente al mismo tiempo, ya que no se dispone de un centro de sincronización que lo haga automáticamente. Este punto será de vital importancia para que la aplicación funcione correctamente.

Por otro lado, en el código también se ha incluido un retraso aleatorio para evitar colisiones en la *gateway*, que es el nodo receptor de todos los paquetes. Este retraso se inserta justo antes de transmitir en la baliza y su rango oscila entre los 1 y 500 ms, un tiempo suficiente para evitar el embudo formado en la *gateway*.

Los únicos valores a configurar en el programa antes de cargarlo en las balizas son las constantes que aparecen en las primeras líneas de código sobre las MACs de la baliza concreta y la de la *gateway*.

### 7.5.3 Código de la estación base

La estación base es el nodo de la red con mayor capacidad de procesamiento, y por ello, donde se trata la mayor carga computacional de la aplicación.

La estación base (PC) recibe a través del puerto serie los paquetes que la *gateway* reenvía tal y como llegan desde las balizas. Para leer el puerto serie en el PC se utiliza el terminal RealTerm que genera un fichero con extensión .txt que contiene todos los paquetes que se transmiten desde la red.

En la Figura 7-1 se muestra un extracto de un fichero obtenido durante una de las pruebas. Ahí se pueden observar los paquetes enviados por las balizas, tanto datos útiles como las cabeceras de los protocolos utilizados. Además, también se encuentra el paquete que la *gateway* recibe directamente del OVNI periódicamente.

El programa de la estación base ha sido desarrollado en Matlab en su totalidad. El primer procesado que se hace en éste tiene como fin eliminar todas las cabeceras de los protocolos y quedarnos únicamente con los datos útiles tal como se puede ver en la Figura 7-2.

```

~ h€ }3¢ @+∅; S # }3¢ @+∅!MacOVNI:0013A2004086A480
MacBal:0013A2004086A4D8 RSSI:-71 Timesp:20091020173854
C~ h€ }3¢ @+ F S # }3¢ @+ !MacOVNI:0013A2004086A480
MacBal:0013A2004086A4A0 RSSI:-71 Timesp:20091020173854
s~ h€ }3¢ @+`L S # }3¢ @+`!MacOVNI:0013A2004086A480
MacBal:0013A2004086A491 RSSI:-50 Timesp:20091020173854
Ô~ h€ }3¢ @+`K S # }3¢ @+`!MacOVNI:0013A2004086A480
MacBal:0013A2004086A491 RSSI:-50 Timesp:20091020173855
Õ~ h€ }3¢ @+ F S # }3¢ @+ !MacOVNI:0013A2004086A480
MacBal:0013A2004086A4A0 RSSI:-71 Timesp:20091020173855
²~ h€ }3¢ @+∅; S # }3¢ @+∅!MacOVNI:0013A2004086A480
MacBal:0013A2004086A4D8 RSSI:-71 Timesp:20091020173855
B~ h€ }3¢ @+ž+ S # }3¢ @+ž!MacOVNI:0013A2004086A480
MacBal:0013A2004086A49E RSSI:-71 Timesp:20091020173855
Ã~ †€ }3¢ @+€X₁R # }3¢ @+€
P~ h€ }3¢ @+ D S # }3¢ @+ !MacOVNI:0013A2004086A480
MacBal:0013A2004086A4A0 RSSI:-71 Timesp:20091020173856
s~ h€ }3¢ @+`K S # }3¢ @+`!MacOVNI:0013A2004086A480
MacBal:0013A2004086A491 RSSI:-50 Timesp:20091020173856
Ô~ h€ }3¢ @+∅; S # }3¢ @+∅!MacOVNI:0013A2004086A480
MacBal:0013A2004086A4D8 RSSI:-71 Timesp:20091020173856
A~ h€ }3¢ @+ž+ S # }3¢ @+ž!MacOVNI:0013A2004086A480
MacBal:0013A2004086A49E RSSI:-71 Timesp:20091020173856
Ã~ h€ }3¢ @+ E S # }3¢ @+ !MacOVNI:0013A2004086A480
MacBal:0013A2004086A4A0 RSSI:-71 Timesp:20091020173857
±~ h€ }3¢ @+`L S # }3¢ @+`!MacOVNI:0013A2004086A480
MacBal:0013A2004086A491 RSSI:-49 Timesp:20091020173857

```

...

Figura 7-1. Datos recibidos en la *gateway* antes de ser procesados para eliminar cabeceras.

```

MacOVNI:0013A2004086A480 MacBal:0013A2004086A4D8 RSSI:-71
Timesp:20091020173854
MacOVNI:0013A2004086A480 MacBal:0013A2004086A4A0 RSSI:-71
Timesp:20091020173854
MacOVNI:0013A2004086A480 MacBal:0013A2004086A491 RSSI:-50
Timesp:20091020173854
MacOVNI:0013A2004086A480 MacBal:0013A2004086A491 RSSI:-50
Timesp:20091020173855
MacOVNI:0013A2004086A480 MacBal:0013A2004086A4A0 RSSI:-71
Timesp:20091020173855
MacOVNI:0013A2004086A480 MacBal:0013A2004086A4D8 RSSI:-71
Timesp:20091020173855
MacOVNI:0013A2004086A480 MacBal:0013A2004086A49E RSSI:-71
Timesp:20091020173855
MacOVNI:0013A2004086A480 MacBal:0013A2004086A4A0 RSSI:-71
Timesp:20091020173856
MacOVNI:0013A2004086A480 MacBal:0013A2004086A491 RSSI:-50
Timesp:20091020173856
MacOVNI:0013A2004086A480 MacBal:0013A2004086A4D8 RSSI:-71
Timesp:20091020173856
MacOVNI:0013A2004086A480 MacBal:0013A2004086A49E RSSI:-71
Timesp:20091020173856
MacOVNI:0013A2004086A480 MacBal:0013A2004086A4A0 RSSI:-71
Timesp:20091020173857
MacOVNI:0013A2004086A480 MacBal:0013A2004086A491 RSSI:-49
Timesp:20091020173857
...

```

Figura 7-2. Datos útiles recibidos en la *gateway* después de eliminar cabeceras.

Una vez que se tiene el fichero de datos limpio de cabeceras, se guardan en matrices y celdas, de forma que podamos utilizar las funciones incluidas en Matlab para procesar todos esos datos.

El programa leerá el fichero en bloques de 2 s, es decir, cada *bin* temporal de 2 s y a partir de la información contenida en ese intervalo de tiempo se creará el vector P con la potencia medida desde cada una de las balizas en la posición donde se encontraba el OVNI en ese *bin*.

Aunque el OVNI emite su señal cada 1 s, en la *gateway* no se llega a tener la cuádrupla de potencia en ese tiempo, por lo que se optó por analizar cada 2 s el fichero de datos y así poder obtener las cuatro medidas de potencia (una por baliza) cada 2 s. Para leer el fichero cada 2 s, el algoritmo se apoya en el *timestamp*.

Así pues, a pesar de que el OVNI transmite cada 1 s su señal, la localización del mismo en el entorno se actualiza cada 2 s.

Cuando se dispone de la cuádrupla de potencia P, el programa realiza el procedimiento descrito en la fase de localización (ver [Sección 7.4.2](#)) apoyándose en la base de datos previamente almacenada de patrones (matriz de calibración) y estima una localización de todas las posibles.

Finalmente para hacer más ilustrativo el proceso de localización y *tracking* del nodo OVNI se ha incluido una figura 2D en la que se representa la ubicación estimada frente al *bin* temporal.

## 7.6 Puesta en marcha de la aplicación

Para poner en funcionamiento la aplicación, en primer lugar se cargan los códigos del [Anexo B](#) en los respectivos nodos.

Para cargar los programas en la Wasmote redireccionamos al lector al [Punto 4.13](#) del [Capítulo 4](#) donde se describieron los pasos a seguir. En cuanto al programa de la estación base, simplemente debemos tener todos los ficheros en un mismo directorio del PC junto a los ficheros de las pruebas realizadas. Los ficheros de prueba se tomaron leyendo el puerto serie asociado a la *gateway* con Realterm.

El siguiente paso y de vital importancia es encender las balizas al mismo tiempo (o aproximadamente al mismo tiempo) de forma que tengan la misma referencia temporal. Esto se puede realizar fácilmente con un poco de práctica entre dos personas que deslicen las pestañas a *ON* al mismo tiempo. Para ser realizado por una única persona se recomienda encender todas las balizas y luego accionar *reset* en todas ellas al mismo

tiempo. Si bien es cierto que no es la mejor manera de realizar un paso tan importante como este, no se disponía para el TFG del hardware necesario para realizar un sincronismo de alta precisión y de forma centralizada, no obstante las pruebas con la solución aportada han sido satisfactorias a pesar de las limitaciones encontradas.

Después de esto se enciende el OVNI mientras que en la estación base empezamos a leer datos del puerto serie con Realterm para generar el fichero .txt de datos.

El último paso consiste en seleccionar desde el programa de la estación base en Matlab el nombre del fichero anterior, que recordemos debe estar en el mismo directorio donde se encuentra el programa. A continuación se ejecuta este para obtener los resultados.

## 7.7 Pruebas

Llegados a la sección final del proyecto, solo queda mostrar algunos de los resultados obtenidos al probar el sistema de localización.

En primer lugar, en la Figura 7-3 se pueden observar las posiciones espaciales donde se tomaron los patrones de potencia para las entradas de la matriz de calibración, así como, la ubicación elegida para cada una de las balizas en el entorno.



Figura 7-3. Puntos toma de patrones y ubicación de balizas en el entorno de localización.

La matriz de calibración que se registró fue la siguiente:

La matriz de calibración para el entorno bajo estudio es una matriz  $18 \times 4$ , es decir, 18 filas correspondientes a las 18 posiciones espaciales donde se han tomado patrones y 4 columnas correspondientes a las 4 balizas utilizadas en la aplicación. Recordemos que el elemento  $C_{ij}$  de la matriz  $C$  es el nivel de potencia con el que se recibe la señal del OVNI en la baliza  $j$  cuando el OVNI está en la posición  $i$  del entorno. Los valores de potencia están en dBm y se han elegido tomando la moda de todos los datos recibidos en la *gateway* desde cada baliza y para cada posición del entorno, siguiendo el procedimiento descrito en el [Punto 7.4.1](#).

### 7.7.1 OVNI estático

En las Figura 7-4, Figura 7-5 y Figura 7-6 se puede observar la estimación de la localización del OVNI cuando este se encuentra parado. A partir de estas figuras se pueden hacer algunos comentarios interesantes sobre la aplicación. Por ejemplo se ha observado que la precisión en la estimación de localización está por debajo de los 2 m, ya que, tras comprobaciones colocando al OVNI en una posición conocida, en ocasiones la estimación da como resultado el patrón contiguo al real.

Por otro lado en las figuras se puede apreciar cómo para algún *bin*, la localización aparece en la posición del eje vertical  $0$ . Esto ha sido desarrollado en el código a propósito, y significa que la localización para ese *bin* temporal es desconocida, es decir, no se ha podido localizar el OVNI para ese *bin*. Esto ocurre cuando al analizar cada *bin* temporal de 2 s a partir del fichero de datos en la estación base, no se dispone de la cuádrupla de potencia necesaria para localizar al nodo.

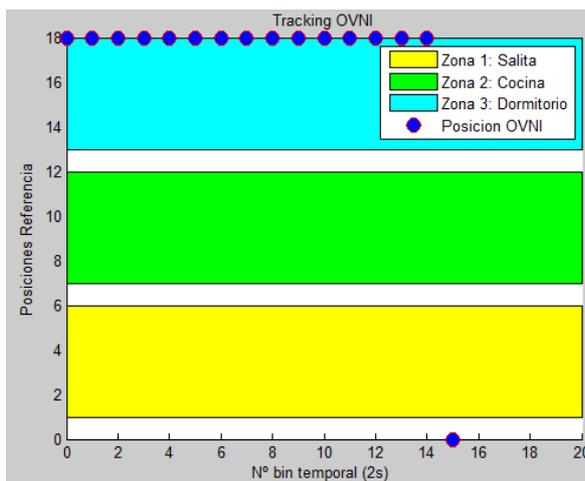


Figura 7-4. Estimación localización del OVNI – Posición 18.

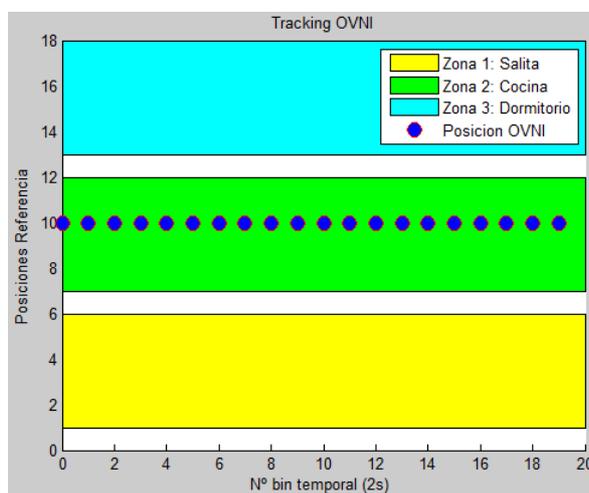


Figura 7-5. Estimación localización del OVNI – Posición 10.

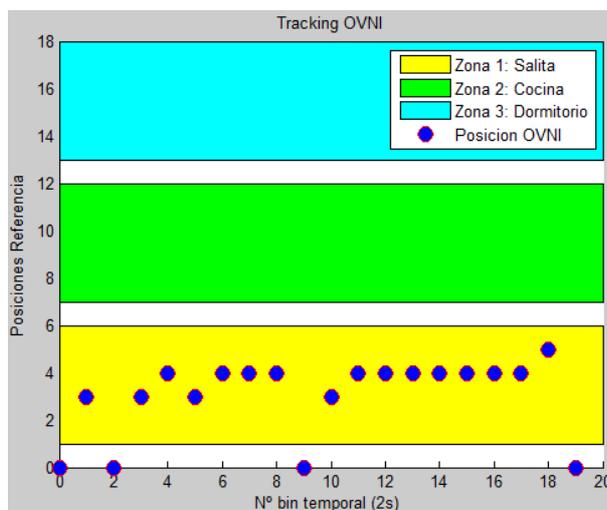


Figura 7-6. Estimación localización del OVNI – Posición 4.

### 7.7.2 OVNI en movimiento

En la Figura 7-7 se puede observar el seguimiento o *tracking* del OVNI desde la zona 1 a la zona 3 y la posterior vuelta.

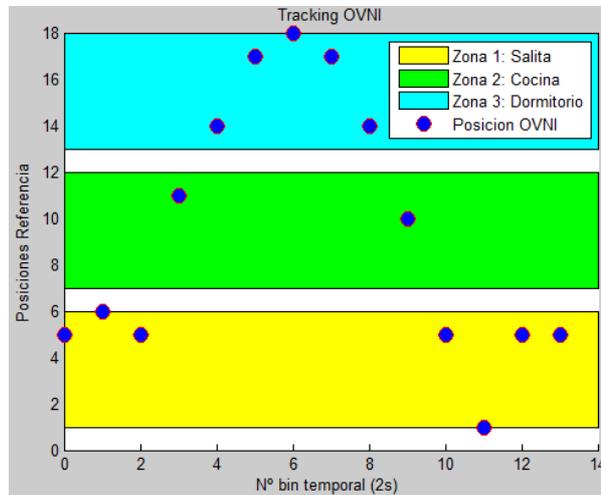


Figura 7-7. Tracking del OVNI. Salita a Dormitorio.

Destacar el buen funcionamiento de la aplicación cuando el OVNI se desplaza a baja velocidad (esto es, cuando el desplazamiento del OVNI se realiza al paso normal de una persona caminando), obteniendo la posición estimada con bastante precisión. Recordar que la tasa de actualización del sistema es de 2 s, por lo que si las zonas son pequeñas en longitud y el OVNI pasa de una zona a otra demasiado rápido puede ocurrir que no se registre el paso por zonas intermedias. Por el contrario, si las zonas son suficientemente amplias, tanto como para no poder ser recorridas en menos de 2 s no se dará esta situación y el registro de *tracking* será continuo, es decir no habrá *saltos*. Así pues, podemos decir que la aplicación está optimizada para baja velocidad y que si se cumple este requisito la precisión en la estimación de localización será buena.

# 8 CONCLUSIONES Y LINEAS FUTURAS DE TRABAJO

---

Para cerrar este trabajo solo queda hacer una reflexión sobre los resultados obtenidos en el estudio, que, a buen seguro, podrían ser de utilidad para seguir mejorando este tipo de sistemas IPS (*Indoor Position System*) basados en niveles RSSI. Si bien, el objetivo se ha conseguido con la solución aportada, el sistema no está optimizado y puede mejorarse en muchos aspectos que analizamos también en este capítulo.

## 8.1 Conclusiones

Al finalizar este TFG se ha demostrado que es posible utilizar con éxito una red inalámbrica de sensores para monitorizar la posición de un nodo (OVNI) en un entorno de interior con suficiente exactitud. Si bien el uso de los niveles RSSI junto a la técnica de estimación de localización de reconocimiento de patrones no es la mejor forma para lograr este objetivo, si es de las soluciones más simples y baratas de conseguirlo, por lo que puede ser un sistema a tener en cuenta frente a otros en función de los requerimientos y presupuesto disponible para el desarrollo de la aplicación.

La localización del nodo realizada en este trabajo puede ser una localización zonal, es decir, el resultado de la localización es la zona, de varias, donde se encuentra dicho nodo. Pero la localización puede ser más precisa aún (en un radio de 2 m), a nivel de a que marca se acerca más la posición real del nodo, de aquellas tomadas como referencia en la matriz de calibración.

En función de lo grande que sea el entorno donde queremos localizar al nodo podemos tomar más o menos patrones de referencia y lograr más o menos precisión, al igual que cuando aumentamos o disminuimos el número de balizas repartidas por el entorno. En cualquier caso para el entorno donde se ha realizado la aplicación de este TFG la medida de seis posiciones o patrones por zona y cuatro balizas ha sido suficiente.

Más allá de que se utilicen más o menos balizas o más o menos patrones de referencia en la base de datos (matriz de calibración) para mejorar o empeorar la precisión respectivamente, un hecho importante para conseguir el máximo grado de la misma que nos permite la combinación de esos parámetros elegida, es la ubicación de las balizas. Dependiendo del entorno donde queramos localizar al OVNI, la ubicación de las balizas en la fase de calibración será determinante, ya que necesitamos que los vectores de la matriz de calibración estén lo menos correlados posibles en zonas distintas y más correlados dentro de una misma zona para que la situación sea realista. Solo así se conseguirá que la precisión en la estimación de la localización sea máxima. Para lograr esto se llegó a que una disposición irregular de las balizas en el entorno de localización conseguía que el parecido (correlación) entre vectores fuera menor y la fase de localización fuera satisfactoria.

En la Figura 8-1 que ya se vio en el capítulo anterior y que repetimos de nuevo, se muestra la ubicación de las balizas y las posiciones donde se tomaron los patrones de la matriz de calibración del entorno tratado en este TFG.

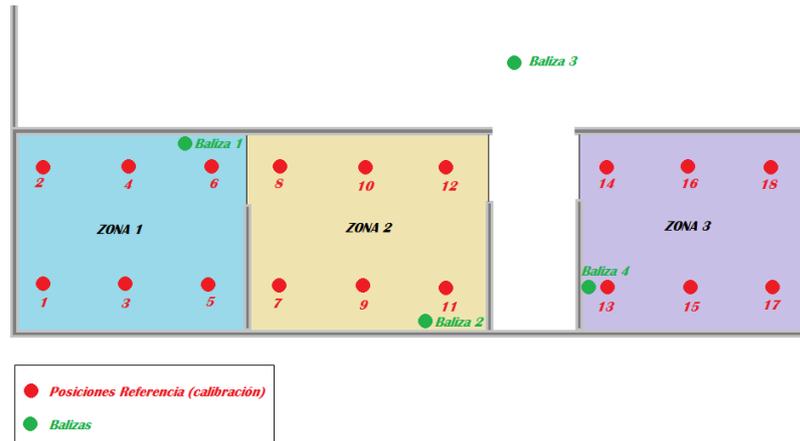


Figura 8-1. Puntos toma de patrones y ubicación de balizas en el entorno de localización.

Aunque el sistema cumple con el objetivo de localización y seguimiento del OVNI en un entorno de interior, cuenta con varios puntos a mejorar que comentamos en la siguiente sección.

## 8.2 Líneas futuras de trabajo

En este punto abordamos las posibles mejoras encontradas al sistema.

En primer lugar sabemos que la frecuencia de transmisión del OVNI es de un segundo, es decir, este transmite un paquete cada segundo. Las balizas, asimismo tal y como reciben ese paquete, a partir de su señal miden la potencia y crean uno nuevo que envían a la *gateway*. La alta frecuencia de transmisión del sistema desemboca en que el nivel de las baterías se ve mermado en poco tiempo, por lo que podría plantearse la alimentación de energía de los sensores fijos (balizas) con mayor carga de procesamiento y por tanto consumo, en la red eléctrica. En cuanto al nodo OVNI podría barajarse utilizar otra tecnología en las baterías, como por ejemplo las de ciclo profundo o gel, eso sí, bastante más caras que las de litio.

Relacionado con este tema y en el caso de que queramos mantener las baterías en todos los nodos se podrían modificar los algoritmos para que cuando se detecte que el OVNI se encuentra estático durante un tiempo los nodos de la red entren en estado de bajo consumo, alargando la vida de las baterías al máximo.

En cuanto a la tasa de actualización de la posición del sistema no se ha conseguido bajar a un valor menor de dos segundos. A pesar de que el OVNI transmite un paquete cada segundo para que las balizas midan los parámetros RSSI, no se logró recibir la cuádrupla de potencia completa en la *gateway* en ese tiempo. Tras numerosas pruebas se consiguió que las balizas no saturaran a la *gateway*, insertando tiempos aleatorios antes de transmitir en cada una de ellas. Finalmente se comprobó que cada dos segundos la *gateway* y la estación base si disponían de la cuádrupla de potencia necesaria para estimar la posición del OVNI quedando la tasa de actualización de posición del sistema a ese valor.

Por otro lado, en el desarrollo de la aplicación se vio que la solución aportada dependía fuertemente del sincronismo de las balizas, que debían tener la misma marca temporal. Para ello y dado las limitaciones de hardware presentadas, había que hacer un sincronismo manual encendiendo al mismo tiempo las cuatro balizas. La solución óptima para este punto sería disponer de un dispositivo que conectado a Internet mantuviera en todo momento la referencia de tiempo actualizada y los nodos de la red, en este caso las balizas, pudieran sincronizarse con este de forma centralizada y en cualquier momento.

En este TFG se ha realizado el seguimiento de un solo nodo, pero podría extenderse a más de uno modificando el programa de la estación base. Habría que incluir las MACs de todos los OVNI que podría haber en la red y repetir el algoritmo buscando paquetes de estos nodos para cada instante/*bin* de tiempo y hacer una representación en Matlab de todos juntos. Bien es cierto que la *gateway* utilizada podría dar problemas de saturación por la gran cantidad de paquetes que recibiría, por lo que con el hardware actual habría que estudiar la nueva situación con detenimiento.

### 8.3 Campos de aplicación

Aquí comentaremos solo algunos de los campos de uso donde utilizar la aplicación desarrollada, pero claro está, el abanico de posibilidades es mucho mayor, tanto como nuestra imaginación nos permita.

- **Entornos inteligentes:** Si conocemos la posición de un nodo que bien podría ser portado por una persona o por una máquina en el interior de un comercio, una nave o una casa podemos activar/desactivar el aire acondicionado, la calefacción, encender luces y un largo etc. de automatismos que se pueden regular o configurar según la zona donde este se encuentre.
- **Control de localización:** En muchos ámbitos como por ejemplo el laboral, el personal debe permanecer en las instalaciones durante cierto tiempo. Mediante un sistema de estas características se podría controlar que los trabajadores estén en su puesto de trabajo durante las horas fijadas en su contrato o que los vigilantes de seguridad hagan su ronda de forma correcta. Se podría también monitorizar remotamente la posición de una persona de la tercera edad dentro de su casa cuando ésta se encuentra sola.
- **Obtención de datos:** La obtención de datos para su posterior análisis es otro campo donde pueden utilizarse este tipo de sistemas. Por ejemplo, en un supermercado, si el carrito de la compra llevara un sensor integrado podría hacerse un estudio sobre los movimientos de los clientes por el establecimiento y obtener datos de qué secciones son más visitadas, por qué zonas pasa más gente y analizar esa información de cara a una decisión para la gerencia. Podría utilizarse para el seguimiento de animales en el interior de una nave con el fin de analizar sus querencias, donde pasan más tiempo en función de la hora de día, etc.
- **Acceso y control de aforo:** La posesión del sensor puede ser la llave para entrar en una zona, habitación o parking privado previo *login* y además esto permitiría controlar el número de personas o vehículos que hay en dicho lugar cerrado para controlar el aforo.



# ANEXO A: XBEE SERIE 1

En este anexo incluimos las especificaciones del XBee 802.15.4 (Serie 1) utilizado en la aplicación de localización en interiores con redes inalámbricas de sensores (ver [Capítulo 7](#)) y tratamos algunos aspectos del estándar 802.15.4 que no se comentaron en el [Punto 2.7.1](#) del [Capítulo 2](#).

## Especificaciones XBee 802.15.4 (Series 1)

Tabla A-1. Tabla de especificaciones del XBee 802.15.4 o S1.

Platform	XBee® 802.15.4 (Series 1)
<b>Performance</b>	
RF Data Rate	250 kbps
Indor/Urban Range	100 ft (30 m)
Outdoor/RF Line-of-Sight Range	300 ft (100 m)
Transmit Power	1 mW (+0 dBm)
Receiver Sensitivity (1% PER)	-92 dBm
<b>Features</b>	
Serial Data Interface	3.3V CMOS UART
Configuration Method	API or AT Commands, local or over-the-air
Frequency Band	2.4 GHz
Interference Immunity	DSSS (Direct Sequence Spread Spectrum)
Serial Data Rate	1200 bps - 250 kbps
ADC Inputs	(6) 10-bit ADC inputs
Digital I/O	8
Antenna Options	Chip, Wire Whip, U.FL, & RPSMA
<b>Networking &amp; Security</b>	
Encryption	128-bit AES
Reliable Packet Delivery	Retries/Acknowledgments
IDs and Channels	PAN ID, 64-bit IEEE MAC, 16 Channels
<b>Power Requirements</b>	
Supply Voltage	2.8 - 3.4VDC
Transmit Current	45 mA @ 3.3VDC
Receive Current	50 mA @ 3.3VDC
Power-Down Current	<10 uA @ 25° C
<b>Regulatory Approvals</b>	
FCC (USA)	OUR-XBEE
IC (Canada)	4214A-XBEE
ETSI (Europe)	Yes
C-TICK Australia	Yes
Telec (Japan)	Yes

## Estándar IEEE 802.15.4

El estándar define 27 canales repartidos en tres bandas, tal y como se recoge a continuación en la Tabla A-2.

Tabla A-2. Bandas frecuencias definidas en IEEE 802.15.4.

Frecuencias	Channels	Zone	Bit Rates
868.0 - 868.6MHz	1	Europe	20/100/250 Kb/s
902.0-928.0MHz	10	EEUU	40/250 Kb/s
2.40-2.48Ghz	16	Worldwide	250 Kb/s

Si bien, el XBee 802.15.4 (Series 1) trabaja en la banda de 2.4 GHz con 16 canales separados 5 MHz unos de otros.

### 2.4GHz Band

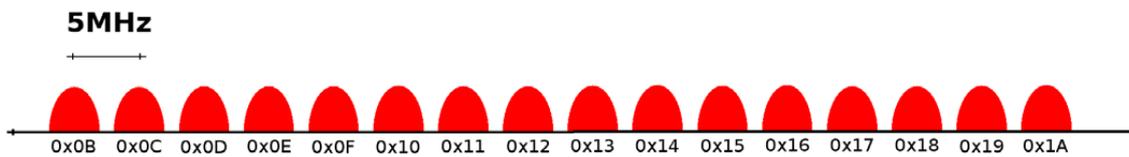


Figura A-1. Canales de frecuencia en la banda 2.4 GHz.

## Modos de operación en 802.15.4

- **Modo Beacon habilitado:** En este modo, un coordinador difunde balizas periódicamente para sincronizar los dispositivos de la red.
- **Modo Beacon no habilitado:** El coordinador no difunde balizas periódicamente aunque puede enviarlas a aquellos dispositivos que la soliciten.

## Topologías de red

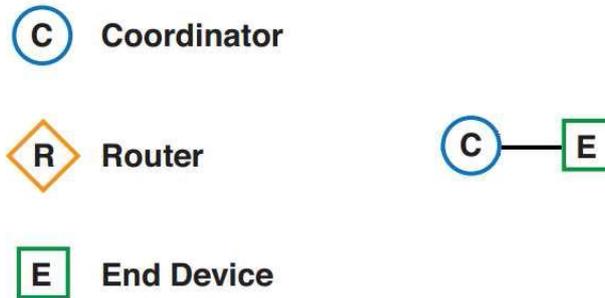


Figura A-2. Topología 802.15.4 Peer to Peer.

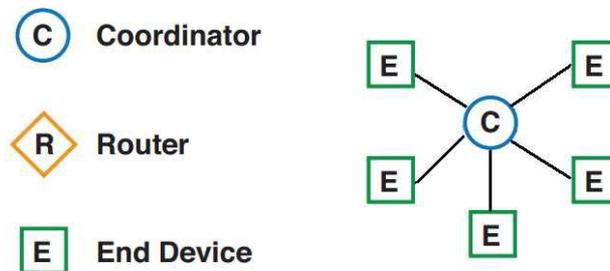


Figura A-3. Topología 802.15.4 en estrella.

### 802.15.4 e interferencias

802.15.4 utiliza dos mecanismos para evitar que todos los nodos empiecen a transmitir a la vez.

- **CSMA-CA (*Carrier Sense Multiple Access-Collision Avoidance*):** Cada nodo escucha el medio antes de transmitir. Si la energía medida está por encima de un umbral, el transceptor espera un tiempo aleatorio para intentar acceder al medio de nuevo.
- **GTS (*Guarantee Time Slots*):** Este sistema usa un nodo centralizado (coordinador PAN) que da *slots* de tiempo a cada nodo para que sepan cuando pueden transmitir. Hay 16 slots de tiempo.

Una de las funcionalidades implementadas en 802.15.4 es el escaneo de energía de los canales (PLME-ED, Perform and Energy Detection Measurement). La idea es ser capaz de saber cuánta energía (actividad, ruido, interferencias) hay en uno o varios canales antes de empezar a usarlos. De esta manera podemos ahorrar energía eligiendo canales libres para transmitir en nuestra red.

### 802.15.4 y bajo consumo energético

El estándar está preparado para trabajar con *duty cycles* pequeños. Esto significa que el transceptor puede estar dormido la mayoría del tiempo (hasta el 99%, dependiendo del tipo de comunicación utilizado) mientras que el resto del tiempo se dedica a tareas de transmisión y recepción.

### 802.15.4 y ruido

El estándar usa espectro expandido por secuencia directa (*Direct Sequence Spread Spectrum, DSSS*) para modular la información antes de ser enviada a la capa física. Esto mejora la relación señal a ruido (SNR) en el receptor debido al hecho de que es más fácil detectar y decodificar el mensaje que está siendo enviado por el transmisor.

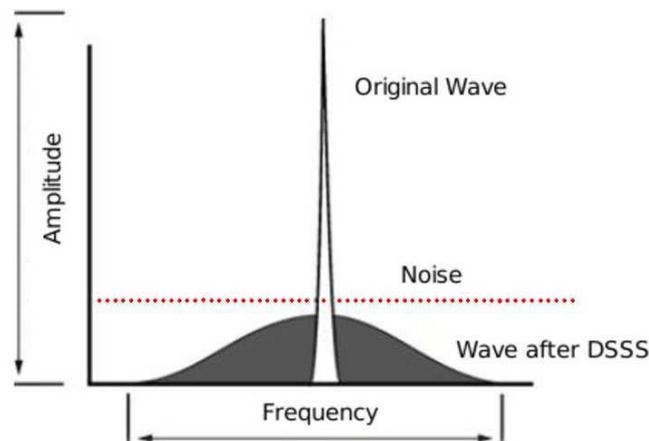


Figura A-4. Espectro de la señal original y después del ensanchado.

### 802.15.4 y capa MAC

802.15.4 es un protocolo que define la capa de acceso al medio según el modelo OSI. Por debajo se encuentra la capa física y por encima la capa de red. El estándar 802.15.4 es la base para distintos protocolos de nivel de red encargados del encaminamiento y de funcionalidades relacionadas con seguridad, entre otras. Algunos de ellos son ZigBee, Wireless HART, ISA -SP100, IETF IPv6 - LoWPAN o DigiMesh. Para saber más acerca de estos protocolos ver la [Sección 2.7](#) donde se dieron algunas nociones de cada uno de ellos.

# ANEXO B: CÓDIGOS DE PROGRAMACIÓN

## Códigos para Wasmote

### Puntos comunes

Destacar antes de mostrar los programas, algunos puntos comunes para los códigos que corren sobre las Wasmotes, es decir OVNI y balizas. En el bloque *setup* se configura la red a la que pertenecen los sensores, esto es, canal frecuencial y PANID, que deben ser los mismos en todos ellos. Además se inician las librerías de los transceptores XBee que tendrían que haber sido configurados con los mismos parámetros de red que los del programa a través de X-CTU.

Por otro lado, un parámetro crítico en una red de sensores es el nivel de batería de sus nodos. Así pues, se han incluido unas líneas de código para controlar dicho nivel mediante un led programable de color rojo y que se enciende cuando el nivel de batería es menor o igual al 10%.

Por último, en la Tabla 7-1 se muestra la equivalencia entre nodos y direcciones MACs, que serán de utilidad al analizar los códigos de la aplicación.

Tabla B-1. Direcciones MACs y nodos.

Nodos	MAC
Baliza 1	0013A2004086A49E
Baliza 2	0013A2004086A4D8
Baliza 3	0013A2004086A4A0
Baliza 4	0013A2004086A491
OVNI	0013A2004086A480
Gateway	0013A2004086A4DF
Difusión o Broadcast	FFFFFFFFFFFFFFFF

## Código del nodo OVNI

```

/*****ALVARO VILLALBA MACIAS*****/
/**Programa incluido en el nodo de posicion desconocida (OVNI) para ser localizado***/
/*****/

//Constante direccion Mac OVNI.
#define MacOVNI "0013A2004086A480"

//Variables
packetXBee* paq_sent;
int8_t state=0;
long previous=0;
char* data=" ";
int g=0;
uint8_t PANID[2]={0x33,0x32};

void setup()
{
  // Iniciamos libreria XBee 802.15.4
  xbee802.init(XBEE_802_15_4,FREQ2_4G,NORMAL);

  //Encendemos modulo XBee
  xbee802.ON();

  // Seleccionando canal frecuencial: canal 0x0C
  xbee802.setChannel(0x0C);
  if( !xbee802.error_AT ) XBee.println("Channel set OK");
  else XBee.println("Error while changing channel");

  // Seleccionando PANID : PANID=0x3332
  xbee802.setPAN(PANID);
  if( !xbee802.error_AT ) XBee.println("PANID set OK");
  else XBee.println("Error while changing PANID");
}

void loop()
{
  //Comprobacion nivel de bateria -> Si nivel bateria menor o igual a 10% se enciende Led rojo
  if (PWR.getBatteryLevel()<=10)
  Utils.setLED(LED0, LED_ON);
  else
  Utils.setLED(LED0, LED_OFF);

  // Configuramos parametros para transmitir.
  paq_sent=(packetXBee*) calloc(1,sizeof(packetXBee));
  paq_sent->mode=BROADCAST;
  paq_sent->MY_known=0;
  paq_sent->packetID=0x52;
  paq_sent->opt=0;
  xbee802.hops=0;
  xbee802.setOriginParams(paq_sent, MacOVNI, MAC_TYPE); // MAC OVNI
  xbee802.setDestinationParams(paq_sent, "FFFFFFFFFFFFFF", data, MAC_TYPE, DATA_ABSOLUTE); // MAC Difusion
  xbee802.sendXBee(paq_sent);

  //Led verde parpadea al Tx
  Utils.setLED(LED1, LED_ON);

  //Periodo Tx Ovni de ls aproximadamente
  delay(1000);

  //Apagamos Led verde
  Utils.setLED(LED1, LED_OFF);

  //Liberamos recursos
  free(paq_sent);
  paq_sent=NULL;
}

```

## Código para los nodos baliza

```

/*****ALVARO VILLALBA MACIAS*****/
/**Programa cargado en las balizas.Se encarga de medir la potencia del OVNI desde su ubicacion fija y envia esas medidas a la Gateway**/
/*****

//Constantes Direcciones Mac de Baliza y Gateway.
#define MacBaliza "0013A2004086A491"
#define MacGw "0013A2004086A4DF"

//Variables.
packetXBee* paq_sent;
int8_t state=0;
long previous=0;
int numAleatorio;
uint8_t PANID[2]={0x33,0x32};
char datos[200];
char* macHighOvni=" ";
char* macLowOvni=" ";
uint8_t rssi;

void setup()
{
  // Inicio libreria XBee 802.15.4.
  xbee802.init(XBEE_802_15_4,FREQ2_4G,NORMAL);

  // Encendiendo modulo XBee
  xbee802.ON();

  // Estableciendo canal frecuencial : canal 0x0C
  xbee802.setChannel(0x0C);
  if( !xbee802.error_AT ) XBee.println("Channel set OK");
  else XBee.println("Error while changing channel");

  // Estableciendo PANID : PANID=0x3332
  xbee802.setPAN(PANID);
  if( !xbee802.error_AT ) XBee.println("PANID set OK");
  else XBee.println("Error while changing PANID");

  // Iniciando RTC, iniciando bus I2C bus.
  RTC.ON();
  // Estableciendo referencia de tiempo en RTC
  RTC.setTime(9,10,20,3,17,35,0);
}

void loop()
{
  //Comprobacion nivel de bateria -> Si nivel bateria menor o igual a 10% se enciende Led rojo
  if (PWR.getBatteryLevel()<=10)
  Uutils.setLED(LED0, LED_ON);
  else
  Uutils.setLED(LED0, LED_OFF);

  // Esperando recibir paquetes.
  previous=millis();
  while( (millis()-previous) < 20000 )
  {
    if( XBee.available() )
    {
      xbee802.treatData();
      if ( !xbee802.error_RX)
      {
        // Obteniendo parametros de los paquetes recibidos.
        while(xbee802.pos>0)
        {

          Uutils.hex2str(xbee802.packet_finished[xbee802.pos-1]->mac0H,macHighOvni,4);
          Uutils.hex2str(xbee802.packet_finished[xbee802.pos-1]->mac0L,macLowOvni,4);

          rssi=xbee802.packet_finished[xbee802.pos-1]->RSSI;

          // Obtenemos marca timestamp del RTC para incluirla en el paquete a enviar.
          RTC.getTime();

        //Formateando Datos
        sprintf(datos,"!MacOVNI:%s%s MacBal:%s RSSI:-%d Timesp:20%02u%02u%02u%02u%02u%02u\n"
,macHighOvni,macLowOvni,MacBaliza,rssi,RTC.year,RTC.month,RTC.date,RTC.hour,RTC.minute,RTC.second);

```



## Código de la estación base

```

%*****Alvaro Villalba Macias*****
%***Codigo implementado en la estación base*****
%*****

clc
clear all
format long

%Matriz de calibracion en archivo txt
C=importdata('matriz.txt');

%Trozo de Programa que se encarga de "limpiar" de cabeceras(XBee+802.15.4)
%el fichero capturado por el terminal en la estacion base.

%Abrimos ficheros
fichTerm = fopen('prueba.txt'); %Fichero de entrada
fichSal = fopen('nuevo.txt','w'); %Fichero de salida
|
tline = fgetl(fichTerm); %Obtiene linea del fichero

while ischar(tline)
|
    [seleccion,resto]=strtok(tline,'!');
    %disp(resto(2:end)); %Muestra lineas obtenidas
    fprintf(fichSal,'%s\n',resto(2:end)); %Escribe lineas en fichero salida
    tline = fgetl(fichTerm);
|
end

%Cerramos ficheros
fclose(fichTerm);
fclose(fichSal);

%*****
%Leemos del nuevo fichero "limpio" de cabeceras los datos utiles y los
%almacenamos en matrices para su analisis y procesado en la aplicacion de
%deteccion en interiores

[MacOvni,MacBaliza,RssiBaliza,Tiempo]=textread('nuevo.txt','MacOVNI:%s MacBal:%s RSSI:%d Timesp:%s');

%Convertimos de celda a matriz Tiempo y luego a string para poder utilizar
%algunas funciones de Matlab
Tiempo=cell2mat(Tiempo);
Tiempo=str2num(Tiempo);
Tiempo=int64(Tiempo);

[B,I] = sort(Tiempo); %B contiene instante de tiempos ordenados de los
                    %paquetes enviados por las balizas
                    %I contiene los indices de los tiempos ordenados

instanteini=B(1); %Instante inicial (el del primer paquete en Gw)
instantesig=NxInstant(instanteini); %Instante siguiente
instanteult=B(length(B)); %Intante final (el del ultimo paquete en Gw)
bin=0;
P=[0 0 0 0]; %Iniciamos vector de potencia del bin
instanteact=instanteini;

%Representacion localizacion OVNI vs tiempo
figure(1)
title('Tracking OVNI')
xlabel('N° bin temporal (2s) ')
ylabel('Posiciones Referencia')
hold on

```

```
%Zona 1: Salita
z1x=[0 14 14 0];
z1y=[1 1 6 6];
fill(z1x,z1y,'y')
```

```
%Zona 2: Cocina
z2x=[0 14 14 0];
z2y=[7 7 12 12];
fill(z2x,z2y,'g')
```

```
%Zona3: Dormitorio Raquel
z3x=[0 14 14 0];
z3y=[13 13 18 18];
fill(z3x,z3y,'c')
```

```
while instanteact<=instanteult
    IndexBalBin=find(Tiempo==instanteact | Tiempo==instantesig);

    Iball=find(ismember(MacBaliza,'0013A2004086A49E'));
    Iballbin=intersect(IndexBalBin,Iball);
    if isempty(Iballbin)
        pktbal1ok=0;
    else
        pktbal1ok=1;
        P(1)=RssiBaliza(max(Iballbin));
    end

    Ibal2=find(ismember(MacBaliza,'0013A2004086A4D8'));
    Ibal2bin=intersect(IndexBalBin,Ibal2);
    if isempty(Ibal2bin)
        pktbal2ok=0;
    else
        pktbal2ok=1;
        P(2)=RssiBaliza(max(Ibal2bin));
    end

    Ibal3=find(ismember(MacBaliza,'0013A2004086A4A0'));
    Ibal3bin=intersect(IndexBalBin,Ibal3);
    if isempty(Ibal3bin)
        pktbal3ok=0;
    else
        pktbal3ok=1;
        P(3)=RssiBaliza(max(Ibal3bin));
    end

    Ibal4=find(ismember(MacBaliza,'0013A2004086A491'));
    Ibal4bin=intersect(IndexBalBin,Ibal4);
    if isempty(Ibal4bin)
        pktbal4ok=0;
    else
        pktbal4ok=1;
        P(4)=RssiBaliza(max(Ibal4bin));
    end

    %Si en el bin actual no se tiene un paquete de cada baliza en la
    %GW:posicion desconocida en ese bin
    if pktbal1ok==0 || pktbal2ok==0 || pktbal3ok==0 || pktbal4ok==0
        disp('Posicion desconocida bin actual')
        instanteact
        instantesig
    end

    I=0;
    plot(bin,I,'o','markersize',9,'markeredgecolor','r','markerfacecolor','b')
    bin=bin+1;
end
```

```

instanteact=NxInstant(instantesig);
instantesig=NxInstant(instanteact);
disp('-----')

% sino es que todo ha ido bien y se tiene medida de las cuatro balizas
% para el bin actual.
else
    disp('Vector Potencia bin actual')
    instanteact
    instantesig
    P

    instanteact=NxInstant(instantesig);
    instantesig=NxInstant(instanteact);

    T=zeros(size(C));

for i=1:size(C,1)
    T(i,:)=P-C(i,:);
end

R=zeros(size(C,1),1);

for i=1:size(C,1)
    R(i,1)=T(i,1)^2+T(i,2)^2+T(i,3)^2+T(i,4)^2;
end

[Y,I]=min(R);
I
R
plot(bin,I,'o','markersize',9,'markeredgecolor','r','markerfacecolor','b')

bin=bin+1;

if (I>=1)&&(I<=6)
    disp('zona 1')
elseif (I>=7)&&(I<=12)
    disp('zona 2')
elseif (I>=13)&&(I<=18)
    disp('zona 3')
else
    disp('fuera zona')
end

disp('-----')
end

end

legend('Zona 1: Salita','Zona 2: Cocina','Zona 3: Dormitorio','Posicion OVNI')

```



# REFERENCIAS

---

[1] Página web Libelium: <http://www.libelium.com/es/>

[2] Página web XBee: <http://xbee.cl/>

[3] Sensors Everywhere Wireless Networks Technologies and Solutions - Gómez 2010

[4] Localización en interiores usando redes inalámbricas de sensores: Estudio teórico - José Carlos

Reyes Guerrero

[5] Análisis de protocolos de enrutamiento para redes de sensores inalámbricas - Antonio Manuel

Palma Gómez

[6] RFID and Sensor Networks - Chen 2009

[7] Reglamento de Radiocomunicaciones (RR) de la ITU, puntos 5.138 y 5.150

[8] Wireless Sensor Networks for Activity Monitoring using Multi-sensor Multi-modal Node

Architecture - Peter Hungt, Muhammad Tahir, Ronan Farrell, Sean McLoone and Tim McCarthy

[9] KERNEL LEAST SQUARES REGRESSION USING WASPMOTE - Francisco Montoya Capote

[10] Waspote technical guide, punto19:

<http://www.libelium.com/es/development/waspote/documentation/waspote-technical-guide/>

