



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación :

INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

Título del proyecto:

GESTION Y MONITORIZACIÓN DE CONTENEDORES
INTERMODALES MEDIANTE TECNOLOGÍA
INALÁMBRICA

Autor: Juan Maria Calvo Astrain

Tutor: José Javier Astrain Escola

Pamplona, 27 de Julio de 2012

1. INTRODUCCIÓN.....	4
1.1 Descripción.....	4
1.2 Objetivos.....	5
1.3 Estado del arte.....	5
1.3.1 Aplicaciones agrícolas.....	6
1.3.2 Aplicaciones civiles y militares.....	7
1.3.3 Aplicaciones en el medio ambiente.....	8
1.3.4 Aplicaciones sociales y sanitarias.....	8
1.3.8 Transeptores de Radio comerciales.....	9
1.3.8.1 Chipcon CC2420.....	9
1.3.8.2 XBee.....	10
1.3.8.3 Chipcon CC2520.....	11
1.3.9 Sistemas operativos para dispositivos embebidos.....	12
1.3.10 Sistemas operativos existentes para los dispositivos utilizados en las WSN.....	14
1.3.10.1 TinyOS.....	14
1.3.10.2 Microsoft .NET Micro Framework.....	15
1.3.10.3 MANTIS.....	17
1.3.10.4 eCos.....	17
1.3.10.5 μ C/OS.....	18
1.3.10.6 Contiki.....	18
1.3.10.7 SOS.....	19
1.3.10.8 Nano-RK.....	20
1.3.10.9 Otras implementaciones.....	20
1.4 Solución propuesta.....	22
2. ANÁLISIS.....	25
2.1 Elementos de una red de sensores inalámbrica.....	25
2.2 Características de una red de sensores inalámbricas.....	25
2.3 Requisitos de una red de sensores inalámbrica.....	26
2.4 Topologías de red y protocolos de comunicación.....	27
2.4.1 El estándar IEEE 802.15.4.....	27
2.4.2 ZigBee.....	30
2.4.3 Otros estándares.....	34
2.4.3.1 WirelessHART.....	34
2.4.3.2 ISA100.11a.....	35
2.4.3.3 6LoWPAN.....	36
2.4.3.4 IEEE 802.15.3.....	36
2.4.3.5 Wibree.....	37
2.5 Protocolos de comunicación más usados en los dispositivos de las WSN's.....	37
2.5.1 Protocolo SPI.....	38
2.5.2 Protocolo I2C.....	40
2.6 Comunicaciones inalámbricas.....	41
2.6.1 WiFi.....	42
2.6.2 Bluetooth.....	43
2.6.2 ZigBee.....	44
3. DISEÑO e IMPLEMENTACIÓN.....	55
3.1 Diseño.....	55
3.2 Implementación.....	59
3.2.1 Motes comerciales.....	59
3.2.1 Micaz.....	59
3.2.2 Mica2 y Mica2dot.....	60
3.2.3 Intel Mote 2.....	60
3.2.4 TelosB.....	62

3.2.5 Zolertia Z1.....	62
3.2.6 Waspote.....	64
3.2.7 Comparativa.....	65
3.3 Pruebas.....	80
4. CONCLUSIONES Y LÍNEAS FUTURAS.....	82
5. REFERENCIAS BIBLIOGRÁFICAS.....	84

1. INTRODUCCIÓN

1.1 Descripción

A lo largo de los años, ha habido situaciones en las que la mercancía se ha estropeado debido a las condiciones en las que está ha sido almacenada.

Por ello, se puede parar uno a pensar, ¿qué nueva medida se podría tomar para intentar reducir el número de mercancía estropeada? , y es así como surge la idea de realizar este Proyecto Fin de Carrera.

Los sensores Waspnote pueden tomar y almacenar muestras ambientales durante todo el transporte del producto y así conocer si la mercancía ha estado expuesta a altas temperaturas, humedades...

De esta forma podemos determinar responsabilidades en el caso de que la mercancía se estropee.

Las motas Waspnote pueden funcionar como una etiqueta inteligente en los contenedores de mercancía. De esta forma, pueden transmitir de manera inalámbrica la información de producto, número de unidades, lotes, etc. Estas motas también pueden almacenar las condiciones a las que debe mantenerse la mercancía y qué temperaturas y condiciones son óptimas para el buen cuidado de la mercancía.

En el caso de este proyecto, los sensores Waspnote miden la temperatura, el monóxido de carbono, el dióxido de carbono, el dióxido de nitrógeno y los contaminantes del aire durante un determinado trayecto.

El trayecto puede ser bien en tren, en barco, en camión o en avión.

Toda esta información que recogen los sensores, se mandan a un sensor, que podríamos denominarlo sensor servidor. Todas estas medidas se almacenan en una base de datos y se muestra en una aplicación Web.

En la aplicación Web, se puede ver las medidas de los diferentes sensores así como ver las medidas tomadas el día de hoy, los últimos 7 días, el último mes , el último año o todas las medidas de un sensor.

También, se pueden ver las medidas en un diagrama de barras y realizar comparaciones entre varios sensores.

Puesto que los sensores recogen la medida en un determinado trayecto, podemos también ver las medidas de un determinado sensor en un trayecto, así como también realizar comparaciones entre diferentes trayectos para poder la posible diferencia de las medidas.

En los trayectos, los sensores pueden estar colocados en diferentes posiciones. Para ello, desde la página Web podemos asignar una determinada posición a los sensores, así como darles un determinado dato del eje x, eje y y eje z para poder saber exactamente la posición en el trayecto.

1.2 Objetivos

Los muelles de los puertos suelen ser puntos de distribución y recogida de mercancías. En ellos se suelen almacenar diversas mercancías, normalmente dentro de contenedores intermodales, de los que a veces se desconoce su localización, su número y las condiciones climatológicas en las que se encuentran. Por eso, es importante controlar y conocer los contenedores intermodales durante su estancia en el puerto y su trayecto.

Un producto puede verse dañado durante su transporte y almacenamiento. La solución que proponemos es utilizar nuestra tecnología sensorial inalámbrica para controlar la mercancía durante esta fase. La tecnología integrada en Waspnote (sensores) permite lo expuesto anteriormente.

El objetivo principal de este proyecto fin de carrera es controlar en tiempo real o a posteriori en qué condiciones se almacena y transporta la mercancía.

Para ello, como hemos explicado antes, distribuiremos diferentes sensores por nuestra mercancía, los cuales se encargaran de mandar la información referente a la temperatura, monóxido de carbono, dióxido de carbono, dióxido de nitrógeno y contaminantes del aire.

Toda esta información la podremos visualizar en una aplicación Web y también ver las medidas en diagramas de barras.

Además podemos crear diversas alarmas, para saber si un determinado sensor situado en una determinada zona, pasa de una cierta temperatura, de una cierta contaminación de monóxido de carbono, ... Dichos casos, aparecerán reflejados en la página Web.

También, al poder ver las diferentes medidas de un determinado sensor en una posición, podemos ver que posición es más óptima para una determinada mercancía.

1.3 Estado del arte

Una red de sensores inalámbrica (WSN) es una red inalámbrica que consta de dispositivos autónomos distribuidos espacialmente utilizando sensores para supervisar conjuntamente elementos físicos o condiciones ambientales. Una definición muy acertada es la siguiente: "una red inalámbrica de sensores es un conjunto de dispositivos inalámbricos o motes que cooperan entre sí para llevar a cabo un objetivo común" tales como la medición de la temperatura, monóxido de carbono, dióxido de carbono, dióxido de nitrógeno, ..., en distintos lugares. El desarrollo de las redes inalámbricas de sensores fue originalmente motivado por las aplicaciones militares, como la vigilancia del campo de batalla. Sin embargo, las redes inalámbricas de sensores en la actualidad se utilizan en muchas áreas de aplicación civil, incluidos el control del tráfico, atención sanitaria y domótica, agricultura, etc. El eficiente diseño e implementación de las redes inalámbricas de sensores se ha convertido en un área emergente de investigación en los últimos años, debido al gran potencial de la red, que permite que la aplicación de sensores conecte el mundo físico al mundo virtual.

Por otro lado, el uso de sensores cableados obligaba al despliegue de grandes cantidades de cable e impedía la toma de medidas en lugares de difícil acceso. Con los avances realizados en los campos de la electrónica y las comunicaciones inalámbricas se pudieron solventar estos problemas, abaratando el uso de dispositivos inalámbricos para

adquisición de datos, lo que ha permitido el auge de las redes de sensores inalámbricas. Esta tecnología, reconocida en un informe del MIT como una de las diez tecnologías emergentes que cambiarán el mundo [1], constituye una elección muy oportuna para los casos en los que se deben recoger datos de múltiples ubicaciones. Su uso permite el despliegue de un gran número de dispositivos sensores de bajo coste que forman una red inalámbrica robusta, escalable y adaptable a los cambios en el entorno o en su topología.

Además de uno o más sensores, cada nodo en una red de sensores, llamado mote, está equipado normalmente con un transmisor de radio u otro dispositivo de comunicaciones inalámbricas, un pequeño microcontrolador y una fuente de alimentación, que suele ser una batería.

El tamaño del sensor de un solo nodo puede variar al igual que el coste de los nodos de sensores, que van desde cientos de dólares a unos pocos centavos, dependiendo del tamaño de la red y de la complejidad necesaria de los distintos nodos de sensores.

Los nodos pueden albergar sensores que midan diferentes variables. Por lo tanto, se pueden tener diferentes datos al mismo tiempo transmitidos a través de la red inalámbrica de sensores al sensor servidor.

Podemos decir que los sensores inalámbricos son un paso de gigante hacia la computación proactiva, un paradigma donde las computadoras se anticipan a las necesidades humanas y, en caso necesario, actuar en su nombre.

1.3.1 Aplicaciones agrícolas

La agricultura constituye una de las áreas donde se prevé que pueda implantarse con mayor rapidez este tipo de tecnología. Por ejemplo, las redes de sensores favorecen una reducción en el consumo de agua y pesticidas, contribuyendo a la preservación del entorno. Adicionalmente, pueden alertar sobre la llegada de heladas, así como ayudar en el trabajo de las cosechadoras. Gracias a los desarrollos que se han producido en las redes de sensores inalámbricos en los últimos años, especialmente la miniaturización de los dispositivos, han surgido nuevas tendencias en el sector agrícola como la llamada agricultura de precisión. Esta disciplina cubre múltiples prácticas relativas a la gestión de cultivos y cosechas, árboles, flores y plantas, etc. Entre las aplicaciones más interesantes se encuentra el control de plagas y enfermedades. Por medio de sensores estratégicamente situados, se pueden monitorizar parámetros tales como el clima, la temperatura o la humedad, con el fin de detectar rápidamente situaciones adversas y desencadenar los tratamientos apropiados. La gran ventaja del uso de esta tecnología es la detección a tiempo y la aplicación óptima de los pesticidas, únicamente en aquellas zonas donde resulta realmente necesario, lo cual unido a la sencillez de despliegue de este tipo de redes, convierte a la agricultura en uno de los campos fundamentales de aplicación de las redes inalámbricas de sensores.



Figura 1.1 – Despliegue agrícola de sensores

La Figura 1.3.1 muestra un despliegue agrícola de sensores [2]. En la primera imagen se puede apreciar un dispositivo final (End- device) que lleva conectado un sensor. La segunda imagen muestra el detalle del hardware del dispositivo final. La tercera ilustración muestra la estación base, se puede apreciar el panel solar necesario para mantener la alimentación de ésta de forma continuada, y la antena en lo alto del mástil. Por último, la cuarta imagen muestra un sensor capaz de detectar la humedad del suelo denominado Watermark ®.

1.3.2 Aplicaciones civiles y militares

IrisNet es una arquitectura software desarrollada por Intel para la gestión de redes mundiales de sensores de diversos tipos, incluyendo vídeo, permitiendo el acceso distribuido a dichos sensores de una forma potente y eficiente. Cuando el número de dispositivos crece de forma significativa, como es el caso de telarañas mundiales de sensores, resulta clave disponer de herramientas eficientes para el acceso a los mismos, en especial por el elevado consumo de ancho de banda. En el proyecto IrisNet se han demostrado distintas aplicaciones prácticas que hacen uso de dicha herramienta. Entre ellas se incluyen las siguientes:

- * Vigilancia de niños y personas mayores mediante videocámaras.
- * Seguridad del hogar.
- * Avisador de riesgo de epidemias (gripe, fiebres, etc.).
- * Monitorización de redes de ordenadores.
- * Observatorios terrestres y marítimos (costas).

Además de las propias aplicaciones civiles, las redes de sensores inalámbricos encuentran un importante campo de aplicación en misiones militares. Por ejemplo, en la identificación y seguimiento de tropas o vehículos militares, así como en la detección de armas químicas, biológicas y radiológicas (sustancias NBQ), como el mostrado en la Figura 1.2.

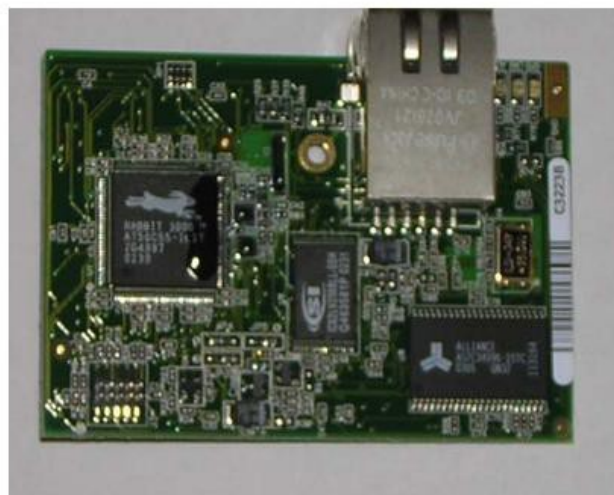


Figura 1.2 – Nodo usado en una red de monitorización de sustancias NBQ [3]

1.3.3 Aplicaciones en el medio ambiente

El mantenimiento y cuidado de espacios y parques naturales resulta complejo en gran medida por las especiales características de los mismos. Se trata de áreas de grandes dimensiones, en algunos casos de difícil acceso, que están repletos de especies vegetales y animales que hay que preservar. Nuevamente las redes inalámbricas de sensores pueden resultar de gran ayuda en este tipo de tareas. Los sensores, de pequeño tamaño, pueden disimularse con el entorno, procesando los datos de diversos parámetros ecológicos y transmitiendo la información de forma inalámbrica hasta un centro de control, situado normalmente en la caseta de los guardas forestales. De este modo, se evita en la medida de lo posible la circulación de personas y vehículos por el parque. Entre los parámetros a monitorizar podemos enumerar: temperatura, humedad, crecimiento de árboles y arbustos, desplazamientos de especies, conteo de animales, caudales de ríos, etc.

1.3.4 Aplicaciones sociales y sanitarias

El cuidado de personas mayores requiere en la mayor parte de los casos de un seguimiento exhaustivo de sus actividades, lo cual limita su privacidad y al mismo tiempo supone una excesiva carga de trabajo para los cuidadores. Mediante el uso de una red de sensores inalámbricos situados en puntos estratégicos del domicilio del anciano, así como en objetos de uso cotidiano, los cuidadores pueden monitorizar en tiempo real el comportamiento de las personas mayores, evitando la realización de tareas tediosas y centrándose en aspectos más importantes como es la mejora de su calidad de vida. Los sensores pueden integrarse en muebles y aparatos. Se comunican entre ellos y con servidores de la habitación, que a su vez se comunican con otros servidores de la habitación. Todos ellos se integran y organizan con los dispositivos integrados existentes para autoorganizarse, autorregularse y autoadaptarse basándose en modelos de control. De esta forma se puede crear un entorno inteligente.

Adicionalmente, el cuidado médico tanto en hospitales como fuera de los mismos, por ejemplo la rehabilitación de pacientes, también se beneficia del uso de esta tecnología. Un ejemplo de ello es el proyecto CodeBlue [4], desarrollado en la Universidad de Harvard. En este caso se han implementado distintos tipos de sensores para la monitorización de parámetros vitales: tasa de latidos del corazón, concentración de oxígeno en sangre, datos EKG de electrocardiograma, etc. Toda esta información se recoge por los sensores y se distribuye de forma inalámbrica a una PDA u ordenador portátil para su procesamiento. De este modo, cualquier señal de alerta puede detectarse a distancia en tiempo real.

Por otro lado, el hardware electrónico actual, ha llegado a un nivel de miniaturización (ver Figura 1.3), tal que los dispositivos electrónicos de hoy en día son capaces de adaptarse al cuerpo humano. Entre la ciencia ficción y la realidad, la idea es crear redes de sensores inalámbricas que permitan un control del estado de salud de las personas y sirvan para prevenir problemas y enfermedades. Con esta ayuda tecnológica, el hombre podría anticiparse a muchas patologías como los ataques al corazón.



Figura 1.3 – Chips de reducido tamaño

1.3.8 Transceptores de Radio comerciales

El término transceptor se aplica a un dispositivo que realiza, dentro de una misma caja o chasis, funciones tanto de transmisión como de recepción, utilizando componentes de circuito comunes para ambas funciones. Dado que determinados elementos se utilizan tanto para la transmisión como para la recepción, la comunicación que provee un transceptor solo puede ser semiduplex, lo que significa que pueden enviarse señales entre dos terminales en ambos sentidos, pero no simultáneamente.

1.3.8.1 Chipcon CC2420

El chip CC2420 compatible con IEEE 802.15.4 es un transmisor de radio a 2,4 GHz de Texas Instruments [21]. Fue diseñado para aplicaciones inalámbricas de baja potencia y baja tensión. CC2420 incluye un espectro digital extendido de secuencia directa y una efectiva transmisión de 250 kbps.

El CC2420 es un dispositivo de bajo coste, está altamente integrado y ofrece una robusta solución para la comunicación inalámbrica en la banda ISM de 2,4 GHz. El CC2420 proporciona un amplio hardware de apoyo a la manipulación de paquetes, los Búfer de datos, ráfagas de las transmisiones, la encriptación y autenticación de los datos y de paquetes de información.

Sus características principales son:

- Chip de 2,4 GHz compatible con IEEE 802.15.4 con base módem y Apoyo MAC
- DSSS base módem con 2 MChips y 250 kbps de Tasa efectivo de datos
- Descarga Bajo consumo (RX: 18,8 mA; TX: 17,4 mA)
- Baja tensión de alimentación (2,1 - 3,6 V) con regulador de voltaje Integrado
- Baja tensión de alimentación (1,6 - 2,0 V) con Regulador de tensión externa
- Potencia de salida programable

1.3.8.2 XBee

Los módulos XBee y XBee - PRO OEM de RF [22] fueron diseñados para cumplir con el estándar IEEE 802.15.4 y cumplir con la necesidad de bajo costo y baja potencia de las redes de sensores inalámbricas. Los módulos requieren un mínimo de energía para la entrega de los datos entre los dispositivos y funcionan en la banda ISM de 2,4 GHz de frecuencia.

Especificaciones		Xbee	Xbee-Pro
Rendimiento	Alcance en ambientes interiores / zonas urbanas	Hasta 30 m	Hasta 100 m
	Alcance de RF en línea de Visión para ambientes exteriores	Hasta 100 m	Hasta 1200 m
	Potencia de salida de Transmisión	1 mW (0 dB)	60mW (18 dB), 100mW EIRP
	Régimen RF de datos	250.000 bps	250.000 bps
	Sensibilidad del receptor	-92 dBm(1% PER)	-100 dBm (1% PER)
Rendimientos de potencia	Suministro de voltaje	2.8-3.4 V	2.8-3.4 V
	Corriente de transmisión	45 mA @3.3 V	270 mA @3.3V
	Corriente de recepción	50 mA @3.3 V	55 mA @ 3.3 V
	Corriente Power-Down	<10 uA	<10 uA
Información general	Frecuencia	ISM 2.4 GHz	ISM 2.4 GHz

Tabla 1.1 – Principales características de los módulos XBee y XBee Pro

1.3.8.3 Chipcon CC2520

El módulo CC2520 es la segunda generación ZigBee / IEEE 802.15.4 transceiver para la banda de 2.4GHz en la banda ISM. Este chip habilita a diferentes tipos de aplicaciones ofreciendo una excelente calidad del enlace de transmisión radio, puede operar hasta a 125°C y operando a bajo voltaje.

Además, el módulo de radio CC2520 proporciona soporte para "frame handling", "data buffering", "burst transmissions", "data encryption", "data authentication", "clear channel assessment", indicación de calidad de enlace e información de los tiempos de la trama. Estas características reducen la carga en el controlador host. En un sistema típico, el CC2520 debe ser usado en conjunto con un microcontrolador y una serie de componentes pasivos (resistencias, bobinas, condensadores, etc.).

El dispositivo proporciona una excelente calidad de enlace (103 dB) y un rango de 400 m en línea de visión directa. El rango de temperaturas de operación se encuentra entre -40°C y +125°C. El rango de voltajes de operación se encuentra entre 1.8 y 3.8 V.

Otra de las características importantes es la capacidad para utilizar encriptación AES-128 y el modo de compatibilidad con el módulo CC2420 de TI.

La principal diferencia respecto al Módulo de radio CC2420 es su menor consumo tanto en recepción como en transmisión:

- (-50 dBm) 18.5 mA en recepción.
- 33.6 mA +5 dBm en transmisión.

1.3.9 Sistemas operativos para dispositivos embebidos

Las necesidades que tiene un nodo de una WSN son totalmente distintas a las que pueda tener cualquier otro dispositivo como puede ser un PC, por lo tanto estos nodos tiene sus propios sistemas operativos.

Los sistemas operativos para WSN son típicamente menos complejos que los de propósito general, tanto debido a los requisitos especiales de las aplicaciones en las que se usan, como a las restricciones de recursos encontradas en las plataformas hardware utilizadas. Por ejemplo, las aplicaciones de WSN no son tan interactivas como son las aplicaciones para PC y debido a esto, estos sistemas no necesitan incluir el soporte de interfaces de usuario. Además, las restricciones de los recursos en términos de memoria hacen imposible de implementar los mecanismos de memoria virtual.

Los nodos sensores incluyen un microcontrolador capaz de ejecutar tareas que requieren el acceso a elementos de hardware (sensores, memoria, radio, etc.). Las funciones principales de un sistema operativo son:

- Gestionar eficientemente los recursos de hardware
- Facilitar la programación de aplicaciones de alto nivel

En la Figura 1.4 se puede apreciar el nivel que ocupa el sistema operativo dentro de las diferentes capas y niveles de abstracción.

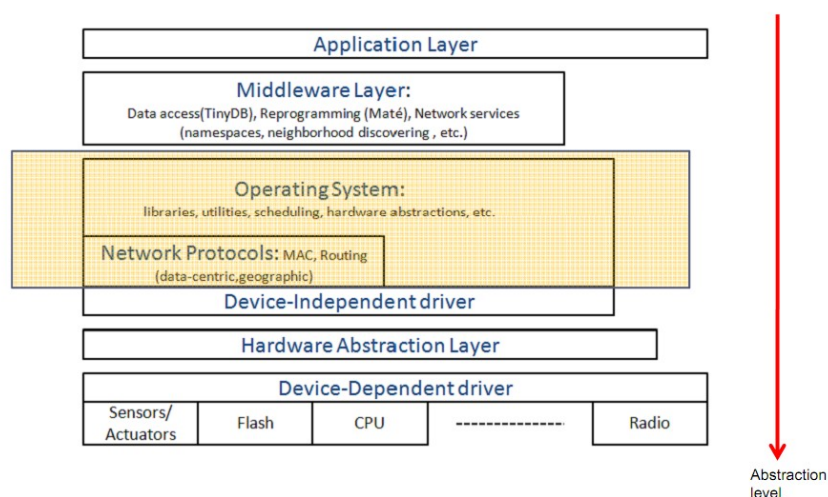


Figura 1.4 – Niveles de abstracción y capas que los componen

Estos sistemas operativos están diseñados específicamente teniendo en cuenta las restricciones de hardware de los nodos sensores. Ya que los sistemas operativos diseñados para otro tipo de sistemas empujados no se adaptan a las fuertes restricciones de los nodos sensores, sobretodo relacionadas con el tamaño de memoria, consumo y requisitos de las aplicaciones.

El sistema operativo es responsable de:

- Gestionar el microcontrolador, tiempo (temporizadores) y concurrencia.
- Gestionar el resto de dispositivos del hardware.
- Ofrecer interfaces de acceso (APIs) a los elementos de hardware (sensores, memoria, radio, etc.).
- Programación de aplicaciones.
- Ofrecer una interfaz para la instalación del código ejecutable en el microcontrolador.
- Ahorro de energía.
- Multiplataforma: Soportan un subconjunto de las plataformas disponibles.
- Arquitectura monolítica: Cuando se compila la aplicación se crea una imagen indivisible compuesta por el S.O. y la propia aplicación.
- Footprint (huella) de pequeño tamaño: Limitado por la capacidad de memoria (e.g. 4KB RAM y 128 KB ROM).
- Ofrecen interfaces de alto nivel (al nivel de aplicación) y usan el interfaz de bajo nivel para acceder a los dispositivos de hardware.
- Proveen implementaciones para la gestión de red (protocolos de enrutamiento, MAC, localización), memoria flash (sistemas de ficheros), temporización (relojes de software), etc.

1.3.10 Sistemas operativos existentes para los dispositivos utilizados en las WSN

En el marco de los sistemas operativos existentes para el trabajo con dispositivos destinados a ser usados en Redes Inalámbricas de Sensores, TinyOS fue el primer sistema operativo diseñado específicamente para ellas. A diferencia de la mayoría sistemas operativos, TinyOS se basa en un modelo de programación controlado por eventos, en lugar de multiprocesos. Los programas de TinyOS están compuestos por eventos y tareas guiadas. Cuando un evento externo ocurre, como puede ser la entrada de un paquete de datos o la lectura de un sensor, TinyOS llama al evento apropiado y lo ejecuta. El lanzador de eventos puede posponer tareas durante cierto tiempo. Tanto TinyOS como los programas escritos para él son escritos en un lenguaje de programación especial llamado NesC, que es una extensión del lenguaje de programación C. NesC está diseñado para determinar las prioridades entre tareas y eventos.

Hay también sistemas operativos que permiten programar en C. Por ejemplo Contiki, MANTIS, BTnut, SOS y Nano-RK. Contiki está diseñado para soportar carga de módulos a través de la red y para soportar cargas de ficheros ELF. El kernel de Contiki está basado en lanzamiento de eventos, como TinyOS, pero puede llegar a soportar multitareas básicas. A diferencia del kernel de Contiki basado en eventos, los kernel MANTIS y Nano-RK están basados en multitareas preventivas. El kernel divide el tiempo entre los procesos activos y decide qué proceso debe ser ejecutado para hacer la programación de la aplicación más fácil. Nano-RK tiene un kernel basado en tiempo real que permite controlar la manera de acceder de las tareas, a la CPU, a la red y a las placas sensoras. Como TinyOS y Contiki, SOS es un sistema operativo basado en eventos. La principal característica de SOS es su soporte para módulos recargables. Un sistema complejo se construye a partir de módulos más pequeños, probablemente en tiempo de ejecución. Para soportar el dinamismo inherente en su módulo de interfaz, SOS soporta una gestión de la memoria dinámica. BTnut se basa en multitareas cooperativas y se propaga en código C plano, además viene empaquetado con un kit de desarrollo y manuales.

1.3.10.1 TinyOS

TinyOS [23] es un sistema operativo orientado a trabajar con redes de sensores, desarrollado en la Universidad de Berkeley. TinyOS puede ser visto como un conjunto de programas avanzados, el cual cuenta con un amplio uso por parte de comunidades de desarrollo, dada sus características de ser un proyecto de código abierto. Este conjunto de programas contiene numerosos algoritmos, que nos permiten generar enrutamientos, así como también aplicaciones pre-construidas para sensores. Además soporta diferentes plataformas de nodos de sensores, arquitecturas bases para el desarrollo de aplicaciones.

El lenguaje en el que se encuentra programado TinyOS es un meta-lenguaje que deriva de C, cuyo nombre es NesC. Además existen varias herramientas que ayudan al estudio y desarrollo de aplicaciones para las redes de sensores, que van desde

aplicaciones para la obtención y manejo de datos, hasta sistemas complejos de simulación.

El diseño de TinyOS está basado en responder a las características y necesidades de las redes de sensores, tales como reducido tamaño de memoria, bajo consumo de energía, operaciones de concurrencia intensiva, diversidad en diseños y usos, y finalmente operaciones robustas para facilitar el desarrollo confiable de aplicaciones. Además se encuentra optimizado en términos de uso de memoria y eficiencia de energía.

El diseño del Kernel de TinyOS está basado en una estructura de dos niveles de planificación:

- Eventos: pensados para realizar un proceso pequeño (por ejemplo cuando el contador del timer se interrumpe, o atender las interrupciones de un conversor analógico-digital). Además pueden interrumpir las tareas que se están ejecutando.
- Tareas: las tareas están pensadas para hacer una cantidad mayor de procesamiento y no son críticas en tiempo (por ejemplo calcular el promedio en un array). Las tareas se ejecutan en su totalidad, pero la solicitud de iniciar una tarea, y el término de ella son funciones separadas.

Con este diseño permitimos que los eventos (que son rápidamente ejecutables), puedan ser realizados inmediatamente, pudiendo interrumpir a las tareas que tienen mayor complejidad en comparación a los eventos.

El enfoque basado en eventos es la solución ideal para alcanzar un alto rendimiento en aplicaciones de concurrencia intensiva. Adicionalmente, este enfoque usa las capacidades de la CPU de manera eficiente y de esta forma gasta el mínimo de energía.

TinyOS se encuentra programado en NesC, un lenguaje diseñado para reflejar las ideas propias del enfoque de componentes, incorporando además un modelo de programación que soporta concurrencia, manejo de comunicaciones y fácil interacción con el medio (manejo de hardware).

1.3.10.2 Microsoft .NET Micro Framework

La .NET Micro Framework [24] fue creada desde el inicio como una solución .NET para dispositivos integrados pequeños de sensores industriales e instrumentación para sistemas empotrados.

Además de estar totalmente integrada con Visual Studio, el kit de desarrollo de software .NET Micro Framework (SDK) viene equipado con un emulador extensible para simular capacidades de hardware. La estructura permite a los desarrolladores de dispositivos conectar diversas soluciones de hardware para prácticamente cualquier dispositivo periférico mediante conexiones de comunicación estándares de la industria y unidades gestionadas personalmente.

La .NET Micro Framework dispone de ofertas integradas de Microsoft en un nuevo mercado de dispositivos que se basan en procesadores de 32 bits de bajo coste y están constreñidos en términos de memoria, potencia de la batería y otros recursos. Ofreciendo paradigmas de programación potentes y modernos a este terreno, .NET Micro Framework pretende acelerar la innovación de dispositivos pequeños y conectados.

- Requiere sólo unos pocos cientos de kilobytes de RAM, y tan poco como 512 K de memoria flash.
- Soporta procesadores con y sin MMU.
- Dispone de una interfaz de control de energía que permite que la aplicación maximice la vida de la batería.

Hasta ahora Microsoft ha publicado el SDK de Microsoft .NET Micro Framework 2.5 el cual permite desarrollar aplicaciones para pequeños dispositivos utilizando para tal motivo, Visual Studio, C#. Esta parte de Visual Studio es gratis y puede descargarse de la página Web de Microsoft, su descarga es de apenas 9Mb y en inglés por ahora para Visual Studio 2005 con SP1. Microsoft .NET Micro Framework es un pequeño subconjunto reducido de clases, en donde podemos desarrollar soluciones en C# para pequeños dispositivos, para trabajar con este pequeño Framework se necesita disponer de Microsoft Visual Studio 2005 Standard o superior, y un sistema operativo Windows XP, Vista o Server 2003. Dispone de un conjunto de librería .NET completamente integradas enfocadas al uso en sistemas empotrados.

Varias empresas están apostando por esta tecnología, Digi International Inc dio a conocer sus planes para un lanzamiento previo del kit de desarrollo Digi Connect ME para Microsoft .NET Micro Framework. El Digi Connect ME incluye soporte para redes Ethernet, un puerto serie y señales de propósito generales entrada/salida (GPIO). Es la primera solución disponible para .NET Micro Framework para dar apoyo a las redes Ethernet.

EmbeddedFusion, que entrega soluciones centrales de hardware y software integrado para desarrolladores de sistemas integrados, anunció el Meridian CPU, que es un módulo central CPU que incorpora procesadores Freescale i.MXS, RAM, Flash y .NET Micro Framework. Para asistir a los desarrolladores en el aprendizaje de cómo se aplica .NET Micro Framework en varios escenarios integrados, EmbeddedFusion también creó la plataforma de desarrollo Tahoe, que permite la experimentación y exploración de .NET Micro Framework.

Freescale también introdujo un kit de desarrollo para .NET Micro Framework que permite a los clientes entregar soluciones diferenciadas en el mercado con rendimiento ARM9 a muy baja potencia.

Además, Rhode Consulting, un especialista en tecnologías Microsoft Windows Embedded, anunció la disponibilidad del kit de evaluación de FlexiDis con .NET Micro Framework instalado. La plataforma FlexiDis utiliza los procesadores centrales Atmel ARM7 y ARM9 con velocidad de hasta 180 Mhz. La combinación de esas velocidades, hasta 16 Mb de memoria flash y SDRAM, y una pantalla QVGA de 2,2 hace de FlexiDis un componente de elección para varios tipos de aplicaciones industriales en las que se requieren HMI integrado o soluciones de visualización.

1.3.10.3 MANTIS

MANTIS [25] (Multimodal NeTworks In-situ Sensors). El sistema operativo MANTIS suministra un nuevo sistema operativo empotrado de plataforma múltiple para redes de sensores inalámbricos. Ante el incremento de complejidad de las tareas realizadas por las redes de sensores como compresión, agregación y procesado de señales, los multiprocesos MANTIS sensor OS (MOS) permiten interpaginar complejas tareas con tareas susceptibles al tiempo para así mitigar los problemas en los saltos de buffers. Para conseguir una eficiencia en el uso de la memoria, MOS es implementado para que utilice una pequeña cantidad de RAM. Usando menos de 500 bytes de memoria, incluyendo el kernel, los controles de tiempo y la pila de comunicación. Para conseguir la eficiencia energética, el controlador de eficiencia energética de MOS hace que el microcontrolador duerma después de ejecutar todas las tareas activas, reduciendo el consumo de energía a un rango de μA .

Una de las características principales de MOS es la flexibilidad en el soporte de múltiples plataformas como PCs, PDAs y diferentes plataformas de microsensores. Otra de las características destacada del diseño de MOS es el soporte de control remoto, permitiendo una reprogramación dinámica y un acceso remoto.

1.3.10.4 eCos

eCos [26] (embedded Configurable operating system) es un sistema operativo de código abierto, gratuito y de operación en tiempo real desarrollado para sistemas empotrados y para aplicaciones que necesiten un procesador con múltiples sesiones. Puede ser personalizado para cumplir los requisitos que la aplicación precise, con cientos de opciones, pudiendo llegar a la mejor combinación entre el rendimiento en tiempo real y el hardware necesario. Este sistema es programable bajo lenguaje C y tiene capas y APIs compatibles para POSIX y μITRON .

eCos fue diseñado para aparatos con un tamaño de memoria sobre cientos de kilobytes o con requerimientos en tiempo real. Puede ser usado en hardware con muy poco RAM soportando Linux empotrado a partir de un mínimo de 2 MB de RAM, sin incluir las necesidades de la aplicación y del servicio.

eCos funciona correctamente en una amplia variedad de plataformas hardware como pueden ser ARM, CalmRISC, FR-V, Hitachi H8, IA-32, Motorola 68000, Matsushita AM3x, MIPS, NEC V8xx, Nios II, PowerPC, SPARC y SuperH. Incluido con la distribución de eCos disponemos de RedBoot, una aplicación de código abierto que usa la capa de abstracción de hardware de eCos que provee soporte de arranque para sistemas empotrados.

eCos fue inicialmente desarrollado por Cygnus Solutions, aunque más tarde fue comprado por Red Hat. A principios de 2002 cesó el desarrollo de eCos y se ubicó al personal que estaba trabajando en el proyecto en otra compañía recién formada, eCosCentric, para continuar con su desarrollo y dar soporte comercial para eCos. En Enero de 2004, a partir de una solicitud de los desarrolladores de eCos, Red Hat aceptó

transferir los derechos de eCos a la Fundación de Software Libre. Esta transferencia fue finalmente ejecutada en octubre de 2005.

ECosPro está formado por una distribución de eCos y RedBoot creada por eCosCentric y está orientada a desarrolladores que quieran integrar eCos y RedBoot dentro de productos comerciales. Está definido como estable, completamente testado, certificado y con soporte, sin embargo, algunas de sus características no han sido liberadas como software libre.

1.3.10.5 μ C/OS

MicroC/OS-III [27], es un sistema operativo multitarea, en tiempo real, basado en prioridad preventiva, de bajo coste donde el kernel está escrito principalmente en el lenguaje de programación C. Es principalmente entendido para uso en sistemas empujados.

La designación II es debido a que es la segunda generación de un kernel que originalmente fue publicado en 1992 en la segunda parte de un artículo en la revista Embedded Systems Programming bajo el título μ C/OS The Real-Time Kernel y escrito por Jean J. Labrosse. El autor intentó describir como funciona un sistema operativo portable por dentro y las razones por las que μ C/OS-II es soportado por Micrium Inc y se obtiene bajo licencia del producto, aunque el uso de este sistema operativo es gratis para uso educacional o no comercial. Adicionalmente Micrium distribuye otros productos software como μ C/OS-View, μ C/CAN, μ C/TCP-IP, μ C/FS, μ C/GUI, μ C/MOD-BUS, μ C/LCD, μ C/USB y un largo grupo de aplicaciones para μ C/TCP-IP como software cliente para DHCP, POP3, SNTP, FTP, TFTP, DNS, SMTP, Y TTCP. El software en su modalidad de servicio incluye http, FTP Y TFTP. PPT es también disponible.

Está disponible para la mayor cantidad de procesadores y placas que existen en el mercado y es adecuado para el uso en sistemas empujados donde la seguridad es crítica como en aviación, sistemas médicos o instalaciones nucleares.

1.3.10.6 Contiki

Contiki [28] es un pequeño sistema operativo de código abierto, altamente portable y multitarea, desarrollado para uso para uso en pequeños sistemas, desde ordenadores de 8-bits a sistemas empujados sobre microcontroladores, incluyendo nodos de redes de sensores. El nombre Contiki viene de la famosa balsa Kon-Tiki de Thor Heyerdahl.

A pesar de incluir multitarea y una pila TCP/IP, Contiki sólo requiere varios kilobytes de código y unos cientos de bytes de RAM. Un sistema totalmente completo con una GUI requiere aproximadamente 30 kilobytes de RAM. El núcleo básico y la mayor parte de las funciones principales fueron desarrollados por Adam Dunkels en el grupo de sistemas de redes empujadas en el instituto sueco de ciencias computacionales.

Contiki fue diseñado para sistemas empotrados con poca cantidad de memoria. Una configuración típica de Contiki es 2 kilobytes de RAM y 40 kilobytes de ROM. Contiki consiste en un núcleo orientado a eventos, el cual hace uso de protothreads, sobre el cual los programas son cargados y descargados dinámicamente. También soporta multihilado apropiativo opcional por proceso, comunicación entre procesos mediante paso de mensajes a través de eventos, al igual que un subsistema GUI opcional, el cual puede usar un soporte directo de gráficos para terminales locales, terminales virtuales en red mediante VNC o sobre Telnet.

Contiki funciona en una variedad de plataformas, desde microcontroladores empotrados, como el MSP430 y el AVR, a viejas computadoras domésticas. El tamaño del código está en el orden de los kilobytes y el uso de la memoria puede configurarse para que sea de sólo unas decenas de bytes.

1.3.10.7 SOS

SOS [29] es un sistema operativo desarrollado en la Universidad de California (UCLA), específicamente en el NESL (Networked & Embedded Systems Laboratory). SOS es un sistema operativo para redes de sensores que procura remediar algunos de las limitaciones propias de la naturaleza estática de muchos de los sistemas precursores a éste (por ejemplo TinyOS).

SOS implementa un sistema de mensajería que permite múltiples hebras entre la base del sistema operativo y las aplicaciones, las cuales pasan a ser módulos que pueden ser o descargadas en tiempo de ejecución sin interrumpir la base del sistema operativo.

El principal objetivo de SOS es la reconfigurabilidad. Ésta se define como la habilidad para modificar el software de nodos individuales en una red de sensores, una vez que éstos han sido desplegados físicamente e iniciado su funcionamiento. En el caso de encontrar un problema, en caso de no contar con esta solución, habría sido necesario recolectar todos los nodos para poder modificar su software.

La capacidad de dinámicamente agregar o remover módulos, permite la construcción de software mucho más tolerante a fallos. Esto presenta dos grandes ventajas: una es el hecho de poder realizar updates fácilmente, y la otra es la capacidad de anular el funcionamiento de algún módulo defectuoso, de algún nodo que pertenece a la red.

Además de las técnicas tradicionales usadas en el diseño de sistemas empotrados, las características del kernel de SOS son:

- Módulos cargados dinámicamente.
- Programación flexible de prioridades.
- Simple subsistema de memoria dinámica.

1.3.10.8 Nano-RK

Nano-RK [30] es un sistema operativo completamente preventivo basado en reserva bajo tiempo real (RTOS) desarrollado en la universidad de Carnegie Mellon con soporte para redes multisalto adecuado para el uso en redes de sensores inalámbricas. Nano-RK funciona adecuadamente con plataformas como redes de sensores FireFly y sobre los motes MicaZ.

Incluye un kernel con recursos empotrados de bajo peso con bastantes funcionalidades y soporte de tiempo usando menos de 2 KB de memoria RAM y 18 KB de ROM. Nano-RK soporta multitareas preventivas con prioridad para asegurar que los plazos de las tareas son conocidos, además de soporte de CPU, red y sensores y actuadores.

Las tareas pueden especificar las demandas de recursos y el sistema operativo provee el acceso controlado y garantizado para los ciclos de CPU y los paquetes de red. Todos estos recursos forman la reserva de energía virtual que permite al sistema operativo controlar el nivel de energía del sistema y de las tareas.

1.3.10.9 Otras implementaciones

Todos los sistema operativos vistos anteriormente son en su mayoría de código abierto, sin embargo, los propios fabricantes de dispositivos electrónicos (Texas Instruments, Jennic, National Instruments, etc.) suelen proporcionar sistemas operativos (con sus correspondientes pilas de comunicaciones) gratuitos para ser usados por sus clientes en sus dispositivos.

La principal diferencia entre los sistemas operativos proporcionados por los fabricantes y los sistemas operativos de código libre analizados anteriormente es que los sistemas operativos de código libre son compatibles con diversos microcontroladores y chips de radio, independientemente del fabricante de los mismos, sin embargo los sistemas operativos proporcionados por los fabricantes sólo podrán ser utilizados en los dispositivos del propio fabricante.

Por ejemplo:

- Jennic[31] proporciona diversas alternativas cada una de ellas con diversas características (Número máximo de nodos soportados, protocolo de comunicación utilizado, facilidad de desarrollo, etc.) dejando a la elección del usuario final de los dispositivos la pila que desee utilizar:

Criteria	IEEE802.15.4	JenNet	JenNet-IP	ZigBee PRO
Recommended Topologies	Star Point-to-point	Tree Star Linear	Tree Star Linear	Mesh
Maximum Network Size	50 nodes	500 nodes	500 nodes	50 nodes
Network Recovery	None	Self-repairing	Self-repairing	Self-repairing
Development Complexity	Medium	Low	Low	High
Standards Compliance	IEEE802.15.4 standards	Proprietary networking layer built on standard IEEE802.15.4 layers	IETF IP, UDP and 6LoWPAN on standard IEEE802.15.4 layers	ZigBee standard networking built on standard IEEE802.15.4 layers
Third-party Interoperability	No provision	No provision	Interoperability possible through public defined MIBs accessed via SNAP	Interoperability through ZigBee public profiles and compliance / certification
Licensing Costs	Free	Free	License-free Compliance-free	ZigBee Alliance membership and product certification fees
Solutions	Cable Replacement Remote Control Logistics Audio	Cable Replacement Active RFID Intelligent Lighting	A/V RF Remote Control Smart Lighting Healthcare Security Smart Energy Asset Management Building Control Environment Monitoring	Smart Energy Home Automation

Tabla 1.2 – Pilas de comunicaciones proporcionadas por el fabricante Jennic

Se puede apreciar en la tabla adjunta, que el fabricante Jennic proporciona 4 implementaciones de sistemas operativos para ser usadas en sus dispositivos:

1. IEEE802.15.4

Implementación del protocolo IEEE802.15.4 punto a punto.

2 .JenNet

Similar a la anterior pero proporciona soporte para topología en árbol, es decir, implementa internamente el autoencaminamiento.

3 .JenNet-IP

Proporciona soporte para IPv6 (6LoWPAN), implementación del protocolo IPv6 desarrollado específicamente para ser usado en dispositivos embebidos.

4 .Zigbee PRO

Proporciona una implementación de la pila de protocolos ZigBee implementada bajo las capas del estándar IEEE802.15.4.

1.4 Solución propuesta

Para cumplir con el objetivo de nuestro proyecto, hemos utilizado las motas Wasmote.

La empresa Libelium [20] lanzó al mercado en el año 2009 este dispositivo destinado a ser usado en redes de sensores inalámbricas denominado Wasmote.



Figura 1.5 – Wasmote con módulo de radio, GPS y GPRS

El módulo principal del Wasmote (ver Figura 1.6) de un microcontrolador ATmega1281 que funciona a 8 MHz y dispone de 8 KB de memoria RAM y 128 KB de memoria FLASH. En conjunto con el microcontrolador, la placa principal cuenta con un Reloj en Tiempo Real con su batería de respaldo y de un slot microSD.

La idea de Libelium es vender el resto de componentes del dispositivo en función de las necesidades del cliente. Es por ello que el dispositivo cuenta con diversos conectores de expansión que le permiten añadir diferentes elementos en función de las necesidades finales.

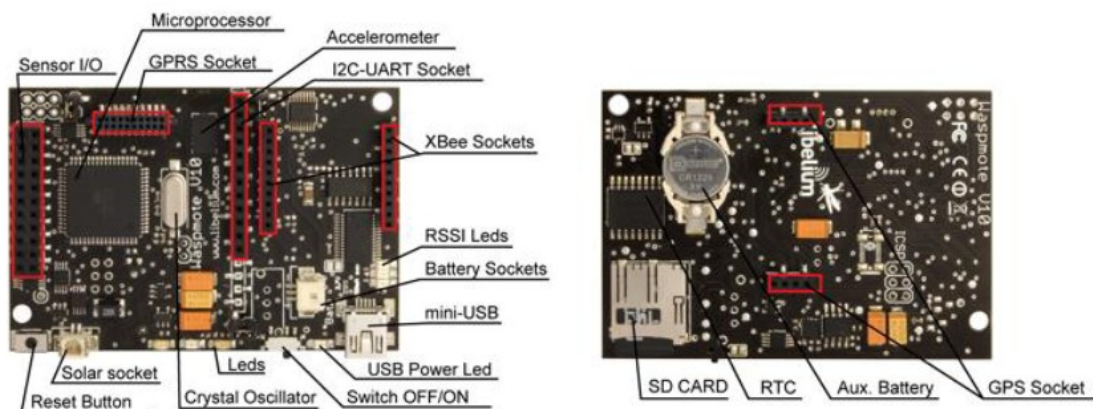


Figura 1.6 – Vista de las dos caras de la placa Wasp mote de Libelium

En lo que respecta a la conectividad vía radio, se le pueden añadir diferentes módulos XBee o bien un módulo Bluetooth. Además también permite añadir un módulo GSM/ GPRS para el envío y recepción de datos a través de la red GSM. También es posible añadir un módulo GPS con su correspondiente antena así como diferentes tipos de sensores destinados en su mayoría a la agricultura.

Libelium ofrece además un dispositivo gateway que permite comunicarse con los Wasp motes por radio, bluetooth o GSM/GPRS y hacer de pasarela hacia internet. De forma que a pesar de que los Wasp motes no soportan el protocolo IP, el gateway hace de pasarela y nos permite la comunicación a través de él, por ejemplo por medio de una red Wi-Fi.

El principal inconveniente del Wasp mote es que no es compatible con Contiki ni con TinyOS. Lo cual limita en gran medida el uso del dispositivo. Ya que se depende totalmente del soporte del fabricante.

La tecnología que he utilizado es ZigBee.

ZigBee define protocolos de comunicación de alto nivel construidos sobre la subcapa MAC del estándar IEEE 802.15.4 para LR-WPANs. ZigBee es simple, de bajo coste, y una tecnología inalámbrica de bajo consumo usada en aplicaciones empotradas.

Referente a la topología, se ha utilizado una en forma de estrella, es decir, la red esta controlada por un único dispositivo, el ZC (ZigBee Coordinator). El ZC es el responsable de iniciar y mantener los dispositivos en la red. El resto de dispositivos, de tipo ZED (ZigBee End Device), se comunican directamente con el ZC.

El coordinador PAN, al estar conectado a un ordenador, transmite mediante el puerto serie los datos al mismo. Estos datos son automáticamente, insertando a una base de datos en donde se almacenan ordenadamente.

Mediante una aplicación Web, desarrollada en html y php, se muestran los datos organizados por temperatura, monóxido de carbono, dióxido de carbono, dióxido de nitrógeno, contaminantes del aire y trayectos.

Se puede también crear sensores, asignarles un trayecto, asignarles una posición, crear alarmas para que nos avise,... en dicha página Web.

Gestión y monitorización de contenedores intermodales mediante tecnología inalámbrica

De esta manera tendríamos controlada, tanto en tiempo real como a posteriori, la mercancía.

2. ANÁLISIS

La empresa Libelium cuenta con un Wasmote capaz de monitorizar de forma remota varios parámetros medioambientales en una WSN como: temperatura del aire, humedad relativa del aire, humedad del suelo, monóxido de carbono, dióxido de carbono, precipitación,... Los datos adquiridos son enviados mediante el protocolo ZigBee inalámbricamente hasta una distancia de 12 kilómetros pudiendo así medir la efectividad del riego, el crecimiento de la cosecha, las condiciones climáticas, prevenir las heladas, detectar la necesidad de riego.

Antes de explicar las distintas topologías de red y protocolos de comunicación explicaremos los elementos, características y requisitos de una red de sensores inalámbrica.

2.1 Elementos de una red de sensores inalámbrica

En relación con los elementos que componen una WSN (ver Figura 2.1), caben destacar (1) los nodos sensores, (2) el nodo sumidero, y (3) la red inalámbrica en su conjunto. Finalmente, hay que mencionar la red inalámbrica propiamente dicha está basada normalmente en un estándar de comunicaciones como el IEEE 802.15.4 [5] o ZigBee [6].

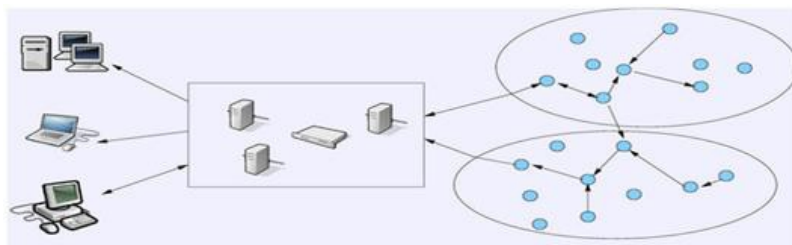


Figura 2.1 – Elementos de una red inalámbrica de sensores

2.2 Características de una red de sensores inalámbricas

Las WSN tienen una serie de características que hay que tener en cuenta en la fase de diseño, a pesar de que la naturaleza de los nodos que forman las redes pueda ser muy distinta y la función que realizarán variará dependiendo del tipo de aplicación. Pero todas ellas tendrán una serie de características comunes:

1. Gran escala: La cantidad de nodos que se despliega en una red puede crecer a lo largo de la vida de ésta, pudiendo llegar a contener miles de nodos. La red estará compuesta de muchos sensores densamente desplegados en el lugar donde se desee llevar a cabo la labor de monitorización.

2. Topología variable: La posición de cada uno de los nodos estará determinada de forma arbitraria o estratégica y normalmente es desconocida por el resto de nodos. La localización no tiene porqué estar diseñada ni preestablecida, lo que va a permitir un despliegue aleatorio en terrenos inaccesibles u operaciones de alivio en desastres.

3. Costes de producción: Una WSN consiste en un elevado número de nodos y por ello, es importante que el coste individual de cada nodo no sea muy elevado. Si el coste de la WSN es mayor que el necesario para desplegar soluciones tradicionales, el uso de una WSN pueda no estar justificado.

2.3 Requisitos de una red de sensores inalámbrica

Para que una red pueda funcionar de acuerdo con las características descritas en el apartado anterior surgen una serie de requisitos no funcionales que una aplicación debe cumplir.

1. Autoorganización: Los nodos desplegados deben formar una topología que permita establecer rutas por las que mandar los datos que han obtenido. Los nodos necesitan conocer su lugar en esta topología, pero resulta inviable asignarlo manualmente a cada uno, debido al gran número de nodos. Por ello, es fundamental que sean capaces de formar la topología deseada sin ayuda del exterior de la red. Este proceso no sólo debe ejecutarse cuando la red comienza su funcionamiento, sino que debe permitir que en cada momento la red se adapte a los cambios que pueda haber en ella.

2. Escalabilidad: Puesto que las aplicaciones van creciendo con el tiempo y el despliegue de la red es progresivo, es necesario que la solución elegida para la red permita su crecimiento sin que las prestaciones de la red caigan drásticamente.

3. Tolerancia a fallos: Los sensores son dispositivos propensos a fallar. Los fallos pueden ser debidos al estado de las baterías, a un error de programación, a condiciones ambientales, al estado de la red, etc. Una de las razones de esta probabilidad de fallo radica precisamente en el bajo coste de los sensores. En cualquier caso, se deben minimizar las consecuencias de ese fallo evitando que un fallo en un nodo individual provoque el mal funcionamiento del conjunto de la red.

4. Tiempo real: Los datos llegan a su destino con cierto retraso. Pero algunos datos deben entregarse dentro de un intervalo de tiempo conocido. Pasado éste dejan de ser válidos, como puede pasar con datos que impliquen una reacción inmediata del sistema, o se pueden originar problemas serios. En caso de que una aplicación tenga estas restricciones debe tomar las medidas que garanticen la llegada a tiempo de los datos.

5. Seguridad: Las comunicaciones inalámbricas viajan por un medio fácilmente accesible a personas ajenas a la red de sensores. Esto implica un riesgo potencial para los datos recolectados y para el funcionamiento de la red. Se deben establecer mecanismos que permitan tanto proteger los datos de estos intrusos, como protegerse de los datos que estos puedan inyectar en la red.

Según la aplicación que se diseñe algunos de los anteriores requisitos cobran mayor importancia. Es necesario encontrar el peso que cada uno de ellos tiene en el diseño de la red, pues normalmente unos requisitos van en detrimento de otros.

A continuación analizaremos los que hay y explicaremos finalmente porque hemos cogido ZigBee.

2.4 Topologías de red y protocolos de comunicación

Una WSN debe funcionar de manera autónoma aunque ocurran situaciones como el fallo de unos o varios nodos sensores o la adición de nuevos nodos en la red. Además, en determinadas aplicaciones los motes se pueden encontrar desplegados en lugares de difícil acceso. Por ello, los nodos tienen que funcionar de manera autónoma y tener la máxima autonomía posible, que se traducirá en un tiempo de vida mayor en la WSN. En esta línea es necesario utilizar topologías de red, así como estándares y protocolos de comunicación que garanticen lo anteriormente expuesto.

2.4.1 El estándar IEEE 802.15.4

El comité de nuevos estándares del IEEE (NesCom, IEEE New Standards Committee) estableció un grupo de trabajo en diciembre del año 2000, para comenzar el desarrollo de un estándar con baja tasa de transmisión de datos en redes inalámbricas de área personal. El grupo de trabajo 4 (TG4, Task Group 4) del IEEE 802.15 se encargaba de investigar una solución con baja velocidad de datos, con una duración de la batería de varios meses o años y de baja complejidad. A su vez la solución debía operar en una banda de frecuencia internacional y sin licencia. Las aplicaciones potenciales de comunicación del estándar serían los sensores, juguetes interactivos, tarjetas inteligentes, mandos a distancia, y en domótica. El TG4 se puso en estado de hibernación en la reunión de marzo 2004 después de formar un nuevo grupo de trabajo (TG4b). El nuevo TG4b completó su trabajo con la publicación de la revisión 2006 del estándar IEEE 802.15.4, que es compatible con la del 2003. Así, nuevos dispositivos 2006 pueden operar en una red 2003.

El estándar IEEE 802.15.4 define la capa física (PHY, Physical) y la especificación de la subcapa del control de acceso al medio (MAC, Medium Access Control) para dispositivos fijos, portátiles o móviles sin batería o con requerimientos de consumo para baterías limitadas que operen en redes inalámbricas personales de área local con tasas bajas de envío de datos (LR-WPAN, Low Rate-Wireless Personal Area Network). Las WPANs son usadas para intercambiar información sobre distancias relativamente cortas. A diferencia de las redes inalámbricas de área local (WLAN, Wireless Local Area Network), las conexiones realizadas a través de una WPAN requieren de infraestructuras muy sencillas e incluso de ninguna infraestructura. Esta característica permite implementar soluciones de pequeño tamaño, eficientes energéticamente y económicas para una amplia gama de dispositivos.

La capa física proporciona una interfaz entre la subcapa MAC y el canal de radio, a través del firmware y el hardware de RF. Conceptualmente la PHY (ver Figura 2.2) incluye una entidad de gestión denominado PLME (Physical Layer Management Entity). Esta entidad proporciona las interfaces de administración de servicios de la capa a través del cual las funciones de gestión de la capa pueden ser invocadas. La PLME también es

responsable de mantener una base de datos de los objetos gestionados relacionados con la PHY. Esta base de datos se denomina la base de información PHY PAN (PIB, PAN Information Base). La PHY proporciona dos servicios: el servicio de datos PHY y el servicio de interfaz de gestión de la PHY, que está ligado con un servicio de acceso puntual (SAP, Service Access Point) de la entidad de gestión de la capa física (PLME, Physical Layer Management Entity), conocido por la siglas PLME-SAP. El servicio de datos de la PHY permite realizar transmisiones y recepciones de unidades de datos del protocolo PHY (MPDU, MAC Protocol Data Unit) a través del canal físico de radio. El PLME-SAP permite el transporte de comandos de gestión entre MLME (MAC Sublayer Management Entity) y la PLME. En definitiva, la PHY define las características físicas y funciones del enlace inalámbrico, entre las que se pueden citar las siguientes: bandas de frecuencia y número de canales de cada banda, la potencia de transmisión, la posibilidad de habilitar y deshabilitar el módulo de radio y la selección del canal. En relación con las bandas de frecuencia, el estándar define 16 canales en la banda disponible a nivel mundial de 2450MHz, 30 canales en la banda de 915MHz disponible en Norte América y 3 en la banda disponible en Europa de 868MHz.

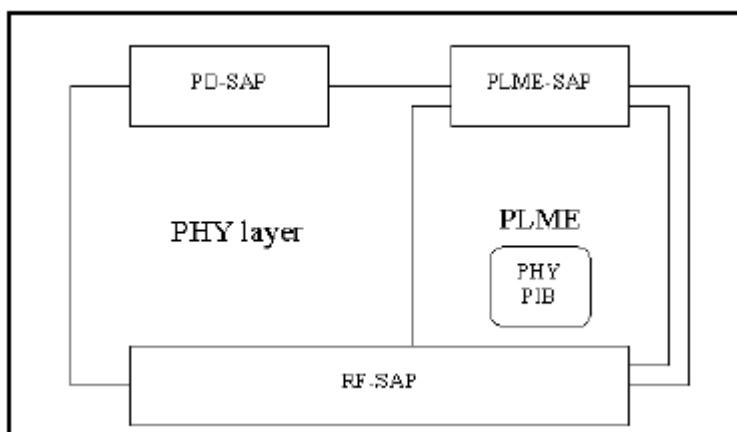


Figura 2.2 - Modelo de referencia de la PHY [5]

La subcapa MAC (ver Figura 2.3) proporciona dos servicios: el servicio de datos MAC y el servicio de gestión de la MAC ligado con el servicio de acceso puntual de la entidad de gestión de la subcapa MAC (MLME-SAP, MAC Sublayer Management Entity- Service Access Point). El servicio de datos MAC permite la transmisión y recepción de MPDUs a través del servicio de datos PHY. Esta subcapa, tiene entre otras, las siguientes características: gestión de las balizas (beacons), acceso al canal, asociación y disociación. Además, la subcapa MAC proporciona servicios para la aplicación de mecanismos de seguridad adecuados en las transmisiones de la aplicación. Los beacons son tramas especiales de sincronización, que si están habilitadas, el coordinador de la red envía periódicamente a los nodos sensores con el fin de tener la red sincronizada y evitar colisiones en las transmisiones.

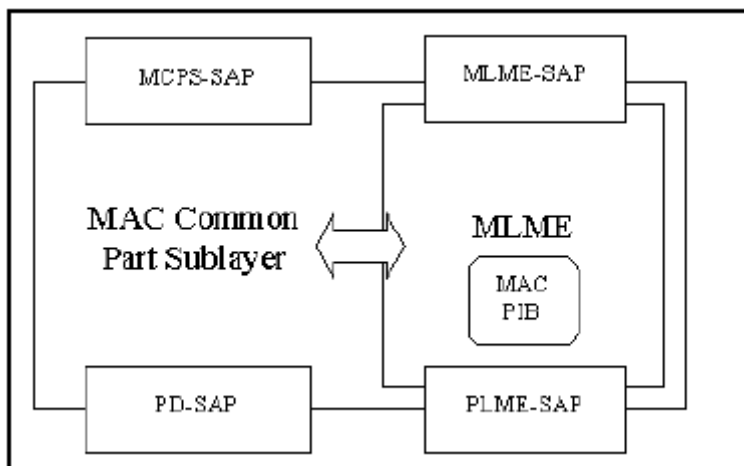


Figura 2.3 – Modelo de referencia de la subcapa MAC [5]

En relación con los dispositivos, hay dos tipos diferentes que pueden participar en una red basada en el estándar IEEE 802.15.4; los dispositivos de funcionalidad completa (FFD, Full Function Device) y los de funcionalidad reducida (RFD, Reduced Function Device). El FFD puede operar en tres modos siendo el coordinador PAN (Personal Area Network), un coordinador o un dispositivo. Además, un FFD puede comunicarse con RFDs y otros FFDs, mientras que un RFD sólo puede establecer una comunicación con un dispositivo con el rol de FFD. Un RFD está pensado para aplicaciones que son extremadamente simples, como el interruptor de una luz o un sensor pasivo de infrarrojos; no necesitan enviar una gran cantidad de paquetes y sólo pueden estar asociados con un único FFD en un instante determinado. En consecuencia, el RFD puede ser implementado usando unos mínimos recursos y una memoria de reducida capacidad.

En función de los requerimientos de la aplicación, una IEEE 802.15.4 LR-WPAN puede estar configurada con dos topologías (ver Figura 1.7): la topología en estrella o star y entre iguales o peer-to-peer. Con la topología en estrella la comunicación se establece entre los dispositivos y un único controlador central, denominado como el coordinador PAN. Típicamente un dispositivo tiene una aplicación asociada y puede ser cualquiera, el punto de inicio o de fin de la red de comunicación. Un coordinador PAN puede tener una aplicación específica, pero tiene la obligación de iniciar, liberar, o intercambiar los paquetes dentro de la red. El coordinador PAN es el coordinador principal de la PAN. Todos los dispositivos que operan en una red con cualquier topología deben tener direcciones únicas de 64-bits. Esta dirección se puede usar para realizar comunicaciones directas dentro de la PAN. También se pueden usar direcciones de 16 bits, que son asignadas a los dispositivos cuando se asocian con el coordinador y que se denotan con el término 'short address'. En relación con los requerimientos energéticos, el coordinador PAN tiene que estar siempre activo. Al contrario, el resto de dispositivos de la PAN no tienen elevados consumos energéticos y generalmente estarán alimentados por baterías. Entre las aplicaciones que se pueden beneficiar de una topología en estrella (ver Figura 2.4), caben destacar las siguientes: automatización del hogar y la domótica, periféricos de un computador, juguetes y juegos, y la monitorización de la salud de pacientes.

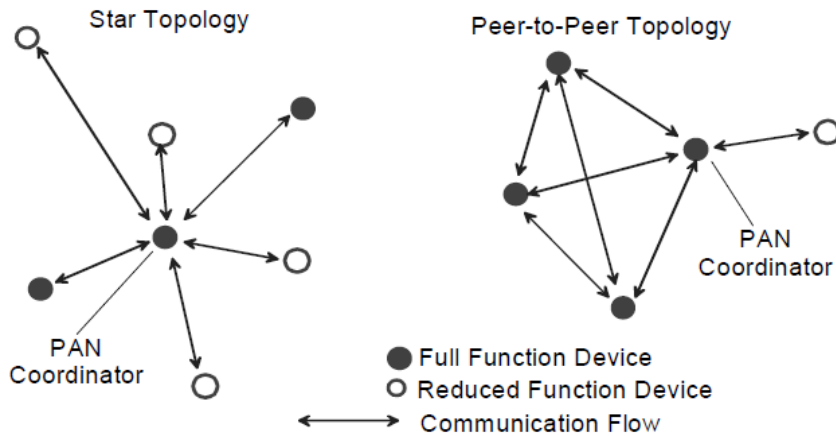


Figura 2.4 – Ejemplos de topologías en estrella y peer-to-peer [5]

La topología peer-to-peer también tiene un coordinador PAN; sin embargo, se diferencia de la topología en estrella en que cualquier dispositivo puede intercambiar información con otro dispositivo, siempre y cuando esté dentro de su rango de alcance. Además, esta topología permite implementar redes más complejas, como una topología de tipo mesh. En este caso las aplicaciones que se pueden beneficiar de esta topología son las siguientes: control industrial y monitorización, WSNs, seguimiento de objetos, agricultura de precisión y la seguridad. Una red peer-to-peer puede ser ad hoc, auto-organizada y con capacidad de auto-reparación. También puede permitir intercambiar mensajes entre dos dispositivos a través de múltiples saltos (hops) intermedios en otros nodos. Estas funciones pueden ser añadidas en capas superiores, pero no forman parte del estándar. Estas funciones han sido añadidas en el protocolo de comunicación de alto nivel ZigBee [7].

2.4.2 ZigBee

La ZigBee Alliance es una asociación de compañías trabajando en conjunto para desarrollar estándares (y productos) para redes inalámbricas de bajo consumo y coste y que sean también seguras. Probablemente, a lo largo de los próximos años la tecnología ZigBee será embebida en una gran cantidad de productos y aplicaciones a nivel mundial, como en el consumo, de tipo comercial, industrial, entre otros [6].

ZigBee define protocolos de comunicación de alto nivel contruidos sobre la subcapa MAC del estándar IEEE 802.15.4 para LR-WPANs. ZigBee es simple, de bajo coste, y una tecnología inalámbrica de bajo consumo usada en aplicaciones empujadas.

Los dispositivos ZigBee pueden formar redes malladas conectando entre ellos desde cientos hasta miles de nodos inalámbricos. Los dispositivos no requieren elevados consumos energéticos y pueden funcionar usando baterías durante varios años. En

general ZigBee define la especificación de la capa de red para topologías en estrella, árbol y peer-to-peer, y proporciona un marco de trabajo para la programación de las aplicaciones.

La versión 2004 del estándar ZigBee fue publicada en junio del año 2005. Esta versión fue revisada con la 2006, que fue publicada en diciembre de 2006. Y durante el último trimestre de 2007 finalizó la publicación de la versión 2007, que se puede encontrar referenciada como ZigBee Pro [7].

ZigBee estandariza las capas superiores (ver Figura 1.8) de la pila del protocolo. La capa de red (NWK, Network) está encargada de organizar y proporcionar rutado sobre una red de tipo multihop, al contrario que la capa de aplicación (APL, Application Layer) proporciona un marco de trabajo para el desarrollo de aplicaciones y comunicaciones distribuidas. La APL engloba el application framework, el ZigBee Device Object y la subcapa de aplicación (APS, Application Sub Layer). El application framework puede tener hasta 240 objetos de aplicación, esto es, módulos de aplicación definidos por el usuario que son parte de la aplicación ZigBee. El ZDO proporciona servicios que permiten descubrir unos APOs (Application Objects) de los otros, así como en organizarse en aplicaciones distribuidas. La APS ofrece una interfaz de datos y servicios de seguridad entre los APOs y el ZDO.

La capa de red (ver Figura 2.5) es necesaria para proporcionar la funcionalidad que asegure una correcta operación de la subcapa MAC del IEEE 802.15.4 y para proporcionar un conjunto de servicios de interfaz con la capa de aplicación.

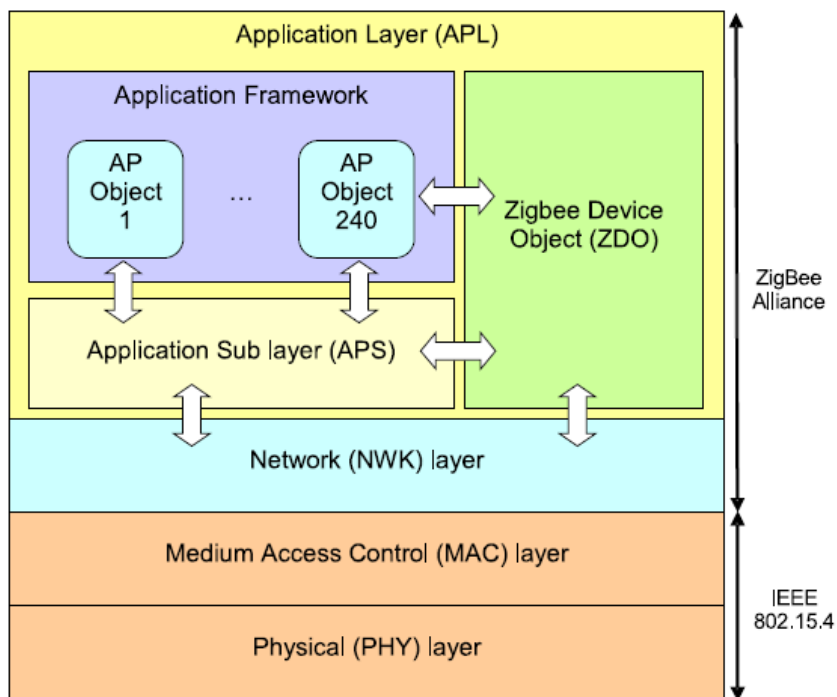


Figura 2.5 – Arquitectura de capas funcional y la pila del protocolo ZigBee [6]

Para intercambiar información con la capa de aplicación, conceptualmente la capa de red incluye dos entidades de servicio que proporcionan la funcionalidad necesaria. Estas entidades de servicio son el servicio de datos y el servicio de gestión. La entidad de datos de la capa de red (NLDE, NWK Layer Data Entity) proporciona los servicios de transmisión de datos a través de su SAP asociado, el NLDE-SAP, y la entidad de gestión de la capa de red (NLME, NWK Layer Management Entity) proporciona el servicio de gestión por medio de su SAP asociado, el NLME-SAP. El NLME utiliza el NLDE para conseguir algunas de sus tareas de gestión y también mantiene una base de datos de los objetos manejados conocida como la network information base o por las siglas NIB [8].

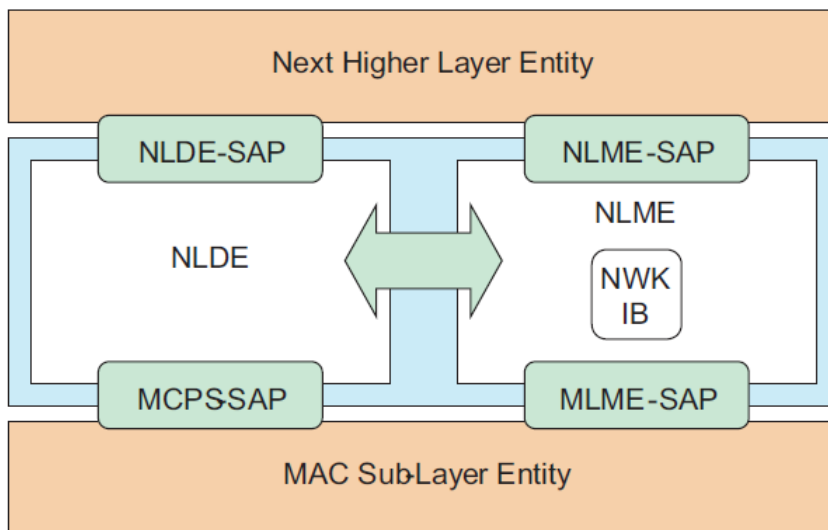


Figura 2.6 – Arquitectura de referencia de la capa de red [6]

La capa de aplicación ZigBee está compuesta por la subcapa APS, el ZDO y los objetos aplicación definidos por el desarrollador. Las responsabilidades de la subcapa APS son mantener las tablas de binding, que es la habilidad de conectar dos dispositivos entre ellos basándose en sus servicios y sus necesidades, e intercambiar mensajes entre los dispositivos que tienen establecido el binding. Otra responsabilidad de la subcapa APS es el descubrimiento (discovery), que es la capacidad de determinar que otros dispositivos están funcionando en el espacio personal de operación del dispositivo. Las obligaciones del ZDO incluyen definir el rol del dispositivo dentro de la red, iniciar y/o responder a peticiones y establecimientos de binding y establecer una relación segura entre los dispositivos de la red.

Los objetos aplicación implementados y definidos por el desarrollador implementan la aplicación en relación con las descripciones de la aplicación ZigBee definida.

Entrando más en detalles, la subcapa APS proporciona una interfaz entre la siguiente capa (NHLE, Next Higher Layer Entity) y la capa de red. Conceptualmente la subcapa APS incluye una entidad de gestión llamada la entidad de gestión de la subcapa APS (APSME, APS Sub-layer Management Entity). Esta entidad proporciona las interfaces de servicio a través de las funciones de gestión que sean invocadas. El APSME es también responsable de mantener una base de datos de los objetos manejados pertenecientes a la subcapa APS. Esta base de datos se denota como la base de información de la subcapa APS (AIB, APS Information Base). La Figura 2.7 muestra las interfaces y los componentes de la subcapa APS.

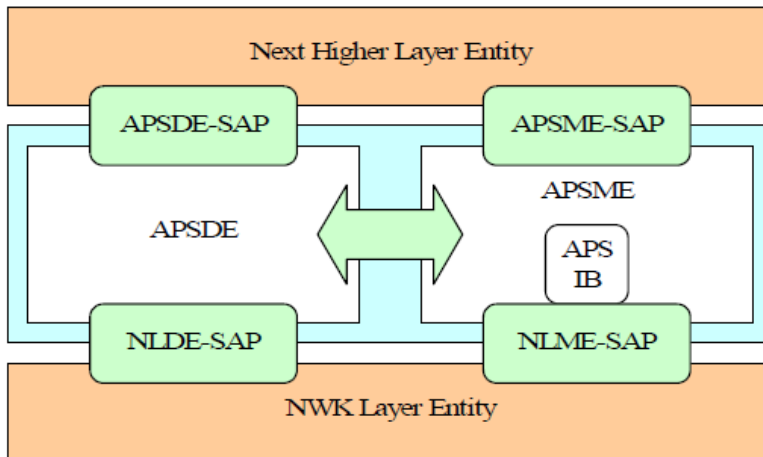


Figura 2.7 – Arquitectura de referencia de la capa de la subcapa APS [8]

La subcapa APS proporciona dos servicios, accedidos por medio de dos SAPs. Estos son el servicio de datos APS, accedido a través del SAP de la entidad de datos de la subcapa APS (APSDE-SAP, APS Sublayer Data Entity SAP), y el servicio de gestión de la APS, accedido por medio del SAP de la entidad de gestión de la subcapa APS (APSME-SAP, APS Sub-layer Management Entity SAP). Estos dos servicios proporcionan la interfaz entre el NHLE y la capa de red, a través del NLDE-SAP y, de forma limitada, con la interfaces NLME-SAP. La interfaz NLME-SAP entre la capa de red y la subcapa APS sólo soporta primitivas NLME-GET y NLME-SET; el resto de las primitivas NLME-SAP están disponibles sólo por medio del ZDO. Además de estas interfaces externas, también hay una interfaz implícita entre el APSME y el APSDE que permite que el primero use el servicio de datos de la APS.

En una red ZigBee hay involucrados tres tipos de dispositivos que define el estándar: el coordinador ZigBee (ZC, ZigBee Coordinator), el router ZigBee (ZR, ZigBee Router) y el dispositivo final ZigBee (ZED, ZigBee End Device). El ZED se corresponde con un RFD o un FFD definido por el estándar IEEE 802.15.4 que actúa como un dispositivo simple. Un ZR es un FFD con capacidad de realizar tareas de enrutado en la red. El ZC es un FFD que maneja la red completa y sólo puede haber uno en cada red.

Algunas de las funcionalidades de la capa de red son: enrutado multihop, descubrimiento y mantenimiento de rutas, seguridad y conexión/desconexión de la red, con la consecuente asignación de una short address (16-bits) a los nuevos dispositivos conectados. Además, la capa de red soporta topologías más complejas que las del estándar IEEE 802.15.4. Estas topologías son las siguientes (ver Figura 2.8): estrella (star), árbol (tree) y malladas (mesh). En una topología en estrella, la red está controlada por un único dispositivo, el ZC. El ZC es el responsable de iniciar y mantener los dispositivos en la red. El resto de dispositivos, de tipo ZED, se comunican directamente con el ZC. En las topologías en árbol y malladas, el ZC tiene la responsabilidad de iniciar y configurar ciertos parámetros importantes de la red, pero la red puede extenderse haciendo uso de los ZRs. En redes en árbol, los ZRs intercambian datos y mensajes de control a lo largo de la red usando una estrategia de rutado jerárquica. Además, este tipo de redes pueden emplear comunicaciones con los beacon habilitados como describe el estándar IEEE. Las redes malladas permiten una comunicación peer-to-peer completa. Para ofrecer este soporte, se utiliza con ligeras modificaciones el protocolo de rutado AODV (Ad-hoc On-demand Distance Vector) [9]. Finalmente, cabe destacar que la revisión actual de ZigBee sólo describe redes intra-PAN, esto es, redes en las que las comunicaciones empiezan y terminan dentro de la misma red.

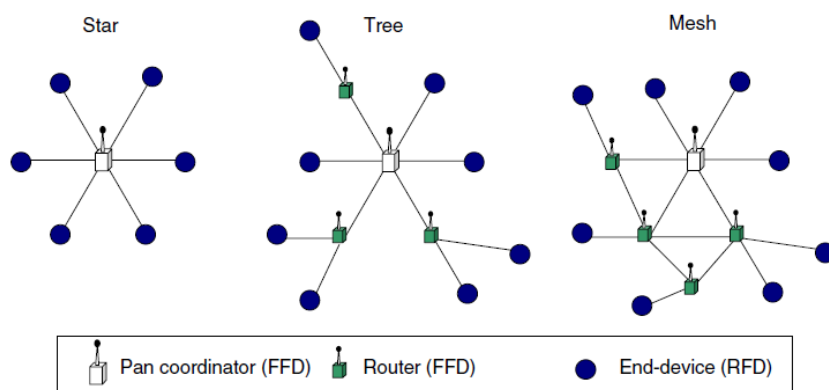


Figura 2.8 – Topologías de las redes ZigBee [6]

2.4.3 Otros estándares

2.4.3.1 WirelessHART

El estándar WirelessHART [9; 10] proporciona un protocolo de comunicación inalámbrica para aplicaciones de control de procesos e instrumentación. El estándar está basado en el IEEE 802.15.4 con una operación de bajo consumo en la banda de 2,4GHz. El WirelessHART es compatible con todos los productos existentes, herramientas y

sistemas. Además, es fiable, seguro y eficiente en términos energéticos. Soporta redes malladas, salto de frecuencia y mensajes sincronizados. La red de comunicación es segura gracias a la encriptación, verificación, autenticación y a la gestión de la clave de la red. Las opciones de gestión energética permiten que los dispositivos inalámbricos sean más eficientes energéticamente. WirelessHART está diseñado para soportar topologías malladas, en estrella, así como combinaciones de las anteriores. Una red WirelessHART está compuesta de los dispositivos inalámbricos situados en campo, gateways, controladores de la automatización del proceso, las aplicaciones host y el gestor de la red (ver Figura 2.9). Los dispositivos inalámbricos de campo están conectados con el proceso o los equipos de la planta. Los gateways garantizan la comunicación entre los dispositivos inalámbricos de campo y las aplicaciones host. El controlador de la automatización del proceso actúa como un sencillo controlador para el proceso. El gestor de la red configura la red y organiza la comunicación entre los dispositivos. Además, se encarga del rutado y del tráfico de la red. El gestor de la red se puede integrar en un Gateway, en una aplicación host, o en un controlador del proceso. El estándar WirelessHART estuvo disponible para la industria en septiembre del 2007.

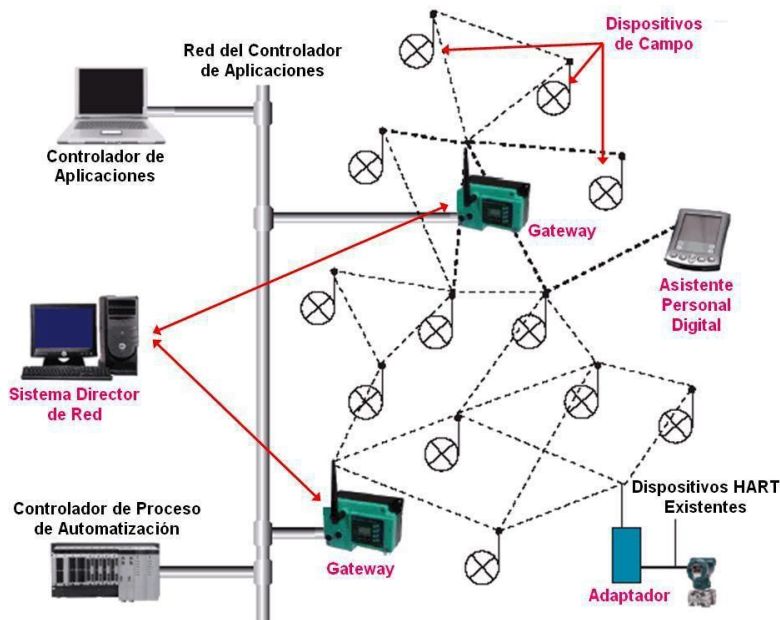


Figura 2.9 – Red WirelessHART y sus elementos

2.4.3.2 ISA100.11a

El estándar ISA100.11a [11] está diseñado para monitorización inalámbrica y control de automatización de procesos con redes inalámbricas con baja velocidad de

datos. Define la especificación para la capa OSI, seguridad y el sistema de gestión. El estándar tiene como objetivos un bajo consumo, la escalabilidad de la red, una infraestructura robusta y la interoperabilidad con otros dispositivos inalámbricos. La red sólo usa la banda de frecuencia de 2,4GHz y saltos de frecuencia para incrementar la fiabilidad de la red y minimizar las interferencias. En cuanto a las topologías, ofrece topologías malladas y en estrella. Además, el ISA100.11a (ver Figura 2.10) proporciona una funcionalidad de seguridad simple, flexible y escalable.

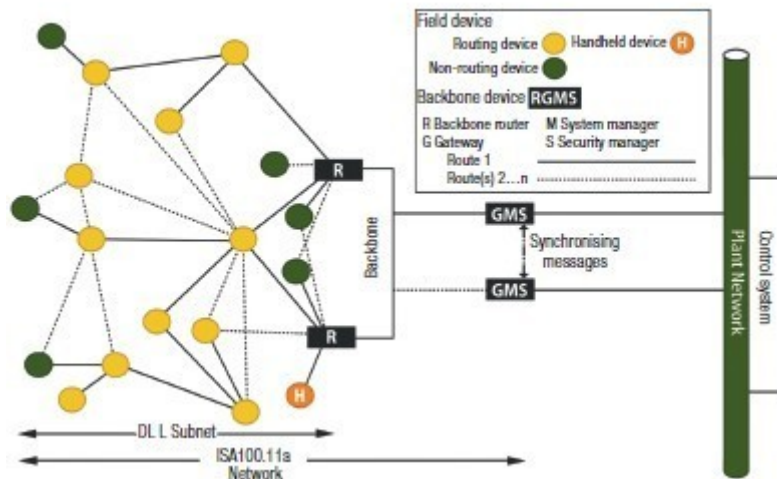


Figura 2.10 – Arquitectura de ISA100.11a

2.4.3.3 6LoWPAN

El estándar 6LoWPAN (IPv6-based Low power Wireless Personal Area Networks) [12; 13; 14] proporciona el envío de paquetes IPv6 sobre una red basada en el IEEE 802.15.4. Los dispositivos de bajo consumo pueden comunicarse directamente con dispositivos IP usando protocolos IP. Usando 6LoWPAN, dispositivos de bajo consumo tienen todos los beneficios de una comunicación y una gestión IP. El estándar 6LoWPAN proporciona una capa de adaptación, un nuevo formato de paquetes, así como gestión del direccionamiento. La capa de adaptación es usada porque el tamaño de los paquetes IPv6 son mucho mayores que el tamaño de trama del IEEE 802.15.4. La capa de adaptación lleva a cabo la funcionalidad para la compresión de la cabecera. Usando la compresión de la cabecera, se crean pequeños paquetes que encajan con el tamaño de trama del IEEE 802.15.4. El mecanismo de gestión de la dirección se encarga de la gestión de las direcciones de los dispositivos para garantizar la comunicación entre ellos. En definitiva, el estándar 6LoWPAN está pensado para aplicaciones en las que usan dispositivos que tienen una baja tasa de envío de datos y que requieren comunicación con Internet.

2.4.3.4 IEEE 802.15.3

El estándar IEEE 802.15.3 [15] define una capa física y una capa de acceso al medio para WPANs con una elevada tasa de datos. Está diseñado para soportar transmisiones de audio y vídeo en tiempo real. El estándar define como la banda de operación la de 2,4GHz y tasas de envíos de datos desde 11 hasta 55Mbps. Para garantizar la calidad del servicio el estándar usa TDMA (Time Division Multiple Access). Soporta transferencias de datos síncronos y asíncronos y lleva a cabo las gestiones del consumo de potencia, la escalabilidad de la tasa de datos enviada y la frecuencia. El estándar se usa en dispositivos, como auriculares inalámbricos, equipos de vídeo portátil y conectividad inalámbrica para juegos, teléfonos inalámbricos, impresoras y televisores.

2.4.3.5 *Wibree*

Wibree [16] es una tecnología de comunicación diseñada para dispositivos de bajo consumo, comunicaciones dentro de pequeños alcances y bajo coste. Wibree permite la comunicación entre dispositivos provistos de pequeñas baterías y dispositivos Bluetooth. Entre los dispositivos provistos de pequeñas baterías caben mencionar los siguientes: relojes, teclados inalámbricos y los sensores de deporte (ver Figura 2.11) que están conectados con ordenadores personales o teléfonos móviles.



Figura 2.11 – Red Wibree de sensores de deporte

Wibree trabaja en la banda de 2,4GHz, soporta una tasa de envío de datos de 1Mbps y la distancia de los enlaces puede estar comprendida entre 5 y 10m. Wibree está diseñado para trabajar con Bluetooth. La combinación de Bluetooth y Wibree consigue dispositivos más pequeños y más eficientes a nivel energético. Wibree se dio a conocer en octubre del 2006.

2.5 Protocolos de comunicación más usados en los dispositivos de las WSN's

2.5.1 Protocolo SPI

El protocolo SPI (Serial Peripheral Interface) como ya indica su nombre, es un sistema de comunicación serie entre periféricos, es un sistema de bajo coste y baja velocidad para comunicaciones de corta distancia, como por ejemplo, entre pequeños procesadores y sus periféricos.

Es un sistema full duplex, muy fácil de implementar entre dos hosts. Si este sistema se utiliza para más de dos hosts, empieza a perder sus ventajas pero el sistema de bus, basado en el protocolo I2C, funciona mejor. Sin embargo, SPI es capaz de ofrecer ratios mucho mayores, pudiendo llegar a las decenas de megahercios.

El protocolo SPI suele constar de un componente maestro y uno o más componentes esclavos. El maestro, se define normalmente como un microcomputador provisto de un reloj SPI (SPI clock), y los esclavos, como cualquier circuito integrado, reciben el reloj SPI del maestro. El ASIC (Application-Specific Integrated Circuit) en los productos de la tecnología VTI siempre opera, como componentes esclavos en modo de operación maestro-esclavo.

SPI, es una interfaz serie síncrona de 4 hilos, la comunicación de datos se activa mediante una señal baja, aplicada en la entrada Slave Select (SS) o en el Chip Select (CSB). Los datos, se transmiten mediante 3 conexiones: conexión para la entrada de datos serie (MOSI), la de salida de datos (MISO) y la señal de reloj (SCK) (ver Figuras 2.12, 2.13 y tabla 2.1).

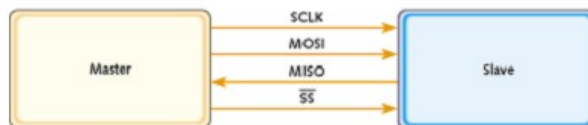


Figura 2.12 – Configuración Master y Esclavo

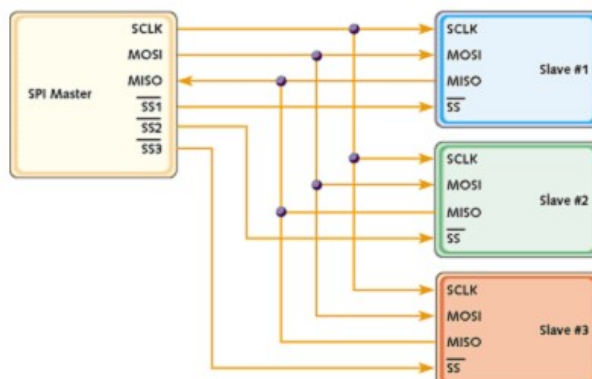


Figura 2.13 – Configuración Master y varios Esclavos

Signal Name	Description
SCLK	Clock
MOSI	Master out - Slave in
MISO	Master in - Slave out
SS	Slave select

Tabla 2.1 – Descripción de las señales del Bus SPI

El master, genera una señal de reloj y la envía a los esclavos. La línea de selección, slave line, se utiliza para indicar con que esclavo se está intentando comunicar el Master.

Como se puede ver, todas las señales excepto la de selección de esclavo son comunes a todos los esclavos, por eso, el master debe indicar cuál de los esclavos esta activo durante la comunicación.

Se puede imaginar, que con más de uno o dos esclavos, el número de salidas del master y el número de pistas en la placa, serian demasiadas para justificar el uso de este protocolo.

SPI usa un par de parámetros llamados clock polarity (CPOL) y clock phase (CPHA), para determinar, cuando los datos son validos dependiendo del señal de reloj. Estos deben establecerse, tanto en el dispositivo maestro como en los esclavos, para que la comunicación funcione correctamente. Mediante estos dos parámetros, se determina cual es el momento idóneo de muestreo.

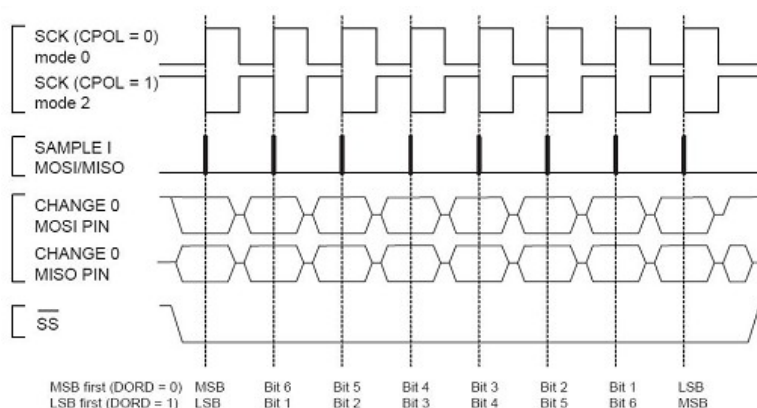


Figura 2.14 – Transferencia de datos bus SPI

En este ejemplo (Figura 2.14), se puede observar que tanto el MSB como el LSB se pueden transmitir en primer lugar, dependiendo de cómo está configurado el hardware SPI.

En nuestro caso, nos interesará sobretodo la comunicación entre Master y Esclavo ya que es el modo en que funcionará la comunicación entre la SD y el uC, el funcionamiento es el siguiente:

1. El Master pone en nivel bajo la línea slave select del esclavo con el que quiere interactuar. Esto indica al esclavo, que debe prepararse para iniciar la comunicación.
2. El master, genera la señal de clock de acuerdo con el modo SPI. Tanto el master como el esclavo transmiten un bit por ciclo de reloj.
3. Después de un byte, el Master pone la línea slave select en nivel alto.

Se han descrito las partes más importantes del protocolo SPI. No hay más conocimiento o estándar de comunicaciones avanzado, que se pueda implementar por encima de esta capa. Lo que ha sido descrito aquí, es todo lo que el protocolo SPI proporciona. No hay ninguna conformidad de conexión, o reconocimiento o cualquier norma de comunicación avanzada asociada con ello. Así aquellos tendrían que ser puestos en práctica sobre esta capa de ser necesario. SPI es un fabuloso y pequeño protocolo, para aquellas situaciones que requieran una simple comunicación entre dos hosts.

2.5.2 Protocolo I2C

La interfaz de comunicación I2C es una comunicación serie, por tanto sólo usa dos líneas, "Serial Data Line" (SDA) y "Serial Clock" (SCL). Los voltajes usados son +5 V o +3.3V. En nuestro caso usaremos 3.3V puesto que es la salida que proporciona el microcontrolador.

El diseño de referencia de I2C usa 7 bits para el espacio de direcciones, con 16 direcciones reservadas, lo que hace un total de 112 nodos conectados al mismo bus. Las velocidades comunes del bus de datos son de 100 kbit/s en modo standard y de 10kbit/s en modo de baja velocidad. Hay revisiones más recientes del bus I2C que pueden conseguir velocidades de hasta 3.4 Mbit/s en modo de alta velocidad. Sin embargo puesto que en nuestro caso solo se enviarán 42 Bytes para la lectura o escritura de los registros del RTC, la velocidad no es apenas relevante. En nuestro caso configuraremos el bus del puerto I2C de nuestro microcontrolador a 100 kHz.

Existirán dos tipos de nodos dentro del bus I2C:

- Nodo maestro: Nuestro microcontrolador.
- Nodo esclavo: El Reloj en tiempo real.

Podrá haber varios nodos maestros en el bus y además, los roles de nodos maestros y esclavos pueden cambiar durante la comunicación entre dispositivos. Hay cuatro posibles modos de operación, a pesar de que la mayoría de dispositivos usan un solo rol y sus dos modos de operación:

- El Nodo maestro transmite: El nodo maestro envía datos al esclavo.
- El Nodo maestro recibe — El nodo maestro recibe datos del esclavo.
- El Nodo esclavo transmite — El nodo esclavo transmite datos al maestro.
- El Nodo esclavo recibe— El nodo esclavo recibe datos del maestro.

El maestro entra inicialmente en modo de transmisión maestra enviando un bit de start seguido de 7 bits que indican la dirección del esclavo con el que se desea comunicar, seguido a continuación de un bit que indica si el maestro desea leer o escribir en el esclavo.

Si el nodo esclavo se encuentra conectado al maestro, responderá con un bit de ACK (activo a bajo) para esa dirección. El maestro a continuación continúa en modo de transmisión o recepción (en función del bit enviado tras la dirección) y el esclavo continúa en el modo contrario al maestro.

La dirección y los bits de datos se envían con el bit más significativo primero. El bit de start se indica con una transición de alta a baja de la línea SDA, permaneciendo la línea SCL en alta. El bit de stop es indicado con una transición de baja a alta de la línea SDA, permaneciendo la línea SCL en alta.

Si el nodo maestro desea escribir en el esclavo, entonces manda repetidamente la dirección del registro a leer al cual el esclavo responderá con sendos ACK's. (en este caso, el nodo maestro estaría en modo de transmisión maestra y el esclavo en modo recepción esclava). Si el nodo maestro desea leer del esclavo, entonces el maestro recibirá repetidamente un byte del esclavo, al cual el maestro responderá con sendos ACK's. (en este caso, el nodo maestro estaría en modo de recepción maestra y el esclavo en modo transmisión esclava).

El nodo maestro finalizará la transmisión con un bit de stop, o enviará de nuevo un bit de start si desea obtener el control del bus para otra transferencia.

2.6 Comunicaciones inalámbricas

Una red inalámbrica es aquella cuyo medio de transmisión son las ondas electromagnéticas, siendo la transmisión y la recepción a través de antenas.

Las ventajas mas destacadas en este tipo de comunicación son:

- Rápida instalación de la red sin la necesidad de usar cableado.
- Permiten la movilidad y tienen menos costes de mantenimiento que una red convencional.

El amplio crecimiento de las tecnologías inalámbricas ha permitido simplificar bastante las tareas cotidianas, mejorando el bienestar de las personas.

Dependiendo del alcance, se pueden clasificar las diferentes tecnologías inalámbricas en:

- Redes inalámbricas de Area Personal (WPAN). Son redes cuya finalidad es la comunicación entre dispositivos cercanos al punto de acceso debido a que tienen un alcance pequeño.

Diferentes estándares desarrollados para las comunicaciones inalámbricas de tipo WPAN son bluetooth y ZigBee.

Para muchos, bluetooth y ZigBee están todas agrupadas en la misma categoría, cuando en realidad representan muy diferentes etapas de desarrollo y ofrecen varios niveles de funcionalidad.

- Redes Inalámbricas de Area Local (WLAN). Tienen un alcance superior a las WPAN, superando un mayor numero de obstáculos. Son muy populares para acceder a Internet, y el estándar mas conocido en este tipo de redes es el WiFi.

A continuación se va a realizar un análisis de cada tipo de estándar para hacer una elección de cual es el que mejor cubre las necesidades de este proyecto:

- · Baja transmisión de datos.
- · Bajo consumo.
- · Bajo precio.
- · Pequeño tamaño de los módulos.

2.6.1 WiFi

WiFi es un conjunto de estándares para redes inalámbricas basado en las especificaciones IEEE. 802.11. Se creo para ser utilizada en redes locales inalámbricas, pero es frecuente que en la actualidad se utilice para el acceso a Internet.



Figura 2.15 – Logotipo de WiFi

WiFi es una marca de la WiFi Alliance (anteriormente la Wireless Ethernet Compatibility Alliance), su logotipo se muestra en la Figura 2.15, la organización comercial que prueba y certifica que los equipos cumplen los estándares IEEE 802.11x.

WiFi o WLAN es una red que requiere la actividad casi ininterrumpida de los dispositivos en la red. La ventaja de este estándar es la cantidad tremenda de datos que se pueden transferir de un punto a multi-puntos, pero se puede ver que la corriente en transmisión es alta.

Funciona a una frecuencia de 2.4 GHz, al igual que la tecnología inalámbrica bluetooth, sin embargo, en la versión 1.2 y mayores del estándar bluetooth se ha actualizado su especificación para que no haya interferencias en la utilización simultanea de ambas tecnologías.

Este tipo de sistemas requieren de unos 0.6 Watos de potencia. La máxima potencia de la señal que se transmite puede ser superior a 20 dBm.

Uno de los problemas mas graves a los cuales se enfrenta actualmente la tecnología WiFi es la seguridad. Un muy elevado porcentaje de redes son instaladas por administradores de sistemas y redes por su simplicidad de implementación sin tener en consideración la seguridad y, por tanto, convirtiendo sus redes en redes abiertas, sin proteger la información que por ellas circula.

2.6.2 Bluetooth

Fundamentalmente, bluetooth, cuyo logotipo aparece en la Figura 2.16, vendría a ser el nombre común de la especificación industrial IEEE 802.15.1, que define un estándar global de comunicación inalámbrica que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia segura, globalmente y sin licencia de corto rango.



Figura 2.16 - Logotipo de Bluetooth

Los principales objetivos que se pretende conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar cables y conectores entre estos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

Bluetooth proporciona una vía de interconexión inalámbrica entre diversos aparatos que tengan dentro de si esta tecnología, como teléfonos, cámaras, ordenadores portátiles, impresoras, etc. El alcance que logran tener estos dispositivos es de 10 metros. Para mejorar la comunicación es recomendable que nada físico (como una pared) se interponga.

La especificación de bluetooth define un canal de comunicación de máximo 720Kb/seg con rango óptimo de 10 metros.

La frecuencia de radio con la que trabaja esta en el rango de 2.4 a 2.48GHz con amplio espectro y saltos de frecuencia con posibilidad de transmitir en Full Duplex con un máximo de 1600 saltos/seg.

Los saltos de frecuencia se dan entre un total de 79 frecuencias con intervalos de 1MHz; esto permite dar seguridad y robustez.

La potencia de salida para transmitir a una distancia máxima de 10 metros es de 0dBm (1 mW), mientras que la versión de largo alcance transmite entre -30 y 20dBm (100 mW).

2.6.2 ZigBee

Esta tecnología es la más nueva en este ambiente. ZigBee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal y se representa con el logotipo de la Figura 2.17.



Figura 2.17 – Logotipo de ZigBee

ZigBee está diseñado para operaciones de baja potencia. Un dispositivo ZigBee puede dejarse en standby por un periodo largo de tiempo sin necesidad de volver a cargar la batería de ese dispositivo.

Esto elimina la necesidad del operador de recargar la batería frecuentemente.

ZigBee también ofrece mayor rango que bluetooth. Este está diseñado para dar servicio a dispositivos con baja transmisión de datos frente a dispositivos que requieren banda ancha para transmitir video y gráficos.

ZigBee es similar al bluetooth pero con algunas diferencias:

- Menor consumo eléctrico.
- Menor velocidad de transferencia.

Las comunicaciones ZigBee se realizan en la banda libre de 2.4GHz. A diferencia de bluetooth no utiliza FHSS (Frequency hopping), 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 el tipo de antenas utilizadas (cerámicas, dipolos, etc.). El alcance normal con antena dipolo en visión directa suele ser aproximadamente (tomando como ejemplo el caso de MaxStream, en la versión de 1mW de potencia) de 100m y en interiores de unos 30m.

La velocidad de transmisión de datos de una red ZigBee es de hasta 256kbps y su consumo es de aproximadamente unos 120 mW.

Por último decir que una red ZigBee la pueden formar, teóricamente, hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos. La realidad es menor, siendo, de todas formas, de miles de equipos.

Una vez analizados los detalles principales de los estándares WiFi, bluetooth y ZigBee, se ha realizado un cuadro resumen, recogido en la Tabla 2.2, donde se puede ver más detalladamente las características de cada sistema.

Los parámetros mostrados en la Tabla 2.2 son el estándar del MAC, la máxima velocidad de transmisión, el consumo de corriente típica en transmisión en standby, los requisitos de memoria de programa para un dispositivo típico, aplicaciones y opciones de conexión de la red, y así ver cual cumple con las especificaciones que se necesitan en este proyecto: bajo consumo, bajo alcance, y bajo coste.

COMPARATIVA PRINCIPALES ESTÁNDARES DE COMUNICACIÓN		
ZigBee	Bluetooth	WiFi
Estándar 802.15.4.	Estándar 802.15.1.	Estándar 802.11.
250 kbps	1 Mbps	Hasta 54 Mbps
TX 35 mA	TX 40 mA	TX > 400 mA
Standby 3µA	Standby 200 µA	Standby 20 mA
32-60 KB memoria	100+ KB memoria	100+KB memoria
Iluminación, sensores, control remoto...	Telecomunicaciones, audio...	Internet...
Red en malla, punto a punto o punto a multipunto	Punto a multipunto	Punto a multipunto

Tabla 2.2 – Diferencias entre ZigBee, Bluetooth y WiFi

De la comparativa entre los diferentes estándares de comunicación más populares que comparten la banda de 2,4 GHz sin licencia que muestra la Tabla 2.2, solamente ZigBee brinda la flexibilidad de la conexión de redes en malla.

También se pueden ver los reducidos requisitos de memoria de programa de ZigBee. Las aplicaciones ZigBee son típicamente muy simples. El consumo de potencia es reducido por el hecho de que los dispositivos "end point" de ZigBee pueden "dormir" mientras que se mantienen asociados a la red.

Uno de los puntos clave de esta tabla es mostrar que los estándares inalámbricos están basados en lo que se llaman "modelos de uso" o "aplicaciones". Ningún estándar cubre

todos los requerimientos de todos los “modelos de uso”. Los diseñadores deben escoger el estándar que cubre mejor sus requisitos de aplicación.

Para elegir el tipo de tecnología inalámbrica que se adapta a nuestras necesidades nos hemos fijado en unos parámetros específicos de cada estándar como son: cobertura, consumo, precio, frecuencia de trabajo y el protocolo de comunicación.

Tras este análisis, se ha decidido usar la tecnología ZigBee mediante el uso de módulos XBee por diversos motivos:

- Estos módulos sirven para hacer redes sensoriales.
- Tienen una tasa de transferencia baja, de Kbs, el consumo es reducido y su funcionamiento es por radio frecuencia (2,4 GHz).
- Posee E/S digitales.
- Convertidor A/D con 4 entradas analógicas o 4 E/S digitales.
- Puede incorporar una clave para las comunicaciones.
- Con él se pueden unir redes.
- Alcance de 100 metros en campo abierto y de 30 metros con paredes.
- Ancho de banda de Kbs.
- Para su uso, solo es necesario un PC, el cable de comunicación con el PC, y la alimentación del modulo que vaya unido al emisor.
- Software gratuito. DIGI es el proveedor del software, el nombre de este software es X-CTU, que permite configurar los parámetros del modulo y es como un hyperterminal.

Explicaremos ahora más específicamente en que consiste la comunicación ZigBee.

El termino ZigBee describe un protocolo inalámbrico normalizado, de bajo coste, para la conexión de una Red de Área Personal o WPAN. Este protocolo surge de la alianza de 25 empresas con el objetivo de ayudar al desarrollo de la tecnología inalámbrica.

Las características mas importantes del estándar IEEE 802.15.4 son la flexibilidad de la red, el bajo coste y el bajo consumo de energía; este estándar se puede utilizar para muchas aplicaciones domóticas e industriales, donde se requiere una baja tasa de transmisión de datos.

Los sistemas domóticos pueden utilizar una red cableada o inalámbrica. Los principales sistemas cableados son las líneas telefónicas, modems por cable y líneas de transmisión de energía eléctrica.

Cada uno de ellos ofrece ventajas y desventajas que dependen principalmente de su capacidad de ancho de banda, instalación, mantenimiento y costo, entre otros. Las redes inalámbricas implican un gran intercambio de información con un mínimo de esfuerzo de instalación.

ZigBee se ha implementado en la banda mundial de 2.4GHz, sin necesidad de licencia, las frecuencias 868/900 MHz y 2.4 GHz tienen buena penetración tanto a través de paredes y de techos, pero tienen un rango limitado. La limitación de rango es realmente deseable para reducir las interferencias.

El hardware y las redes ZigBee deben permitir una transferencia de datos de 10 a 115 kbps.

Se pueden poner hasta 100 redes de ZigBee funcionando conjuntamente. Los dispositivos de la red, sensores inalámbricos, monitores o controladores pueden funcionar hasta 2 años con baterías del tipo AA y AAA. En la Tabla 2.3 se describen las principales características de una red ZigBee:

CARACTERÍSTICAS DE UNA RED ZigBee	
BANDA DE FRECUENCIA Y RANGO DE TRANSMISIÓN DE DATOS	868 MHz → 20 Kb/s 915 MHz → 40 Kb/s 2,4 GHz → 250 Kb/s
ALCANCE	10 – 30 m
LATENCIA	Por debajo de 15 ms
CANALES	868/915 MHz → 11 canales 2,4 GHz → 16 canales
MODOS DE DIRECCIONAMIENTO	Todos los chips tienen 64 bits IEEE de direccionamiento
CANAL DE ACCESO	CSMA-CA
SEGURIDAD	128 AES
RED	Hasta 2^{64} dispositivos
RANGO DE TEMPERATURA	-40 a +85 °C

Tabla 2.3 – Características de la tecnología ZigBee

Algunas de las características más destacadas de esta tecnología inalámbrica, que se muestran en la Tabla 2.3 son:

- Diferentes bandas de trabajo: 2.4 GHz (16 Canales), 915 MHz (10 Canales), 868 MHz (1 Canal).
- Tasas de transferencia: 250 Kb/s, 40 Kb/s, 20 Kb/s.
- Topologías: estrella y p2p (punto-a-punto).
- Direccionamiento MAC recortado (16 bits) y extendido (64 bits).
- Bajo consumo energético.
- Gran densidad de nodos por red.
- Radio medio de alcance: 50 m (hasta 500 m, dependiendo del entorno).
- Direccionamiento a nivel de red (16 bits).
- Soporte para enrutamiento de paquetes.
- Permite topología de malla, gracias a las posibilidades de enrutamiento.
- Dispositivos FFD (Coordinador, router y dispositivo final) y RFD (dispositivo final).

Las redes sensoriales están formadas por un conjunto de nodos con un sensor acoplado que pueden monitorizar parámetros del entorno. Algunos de estos valores son: temperatura, humedad, humo, contaminantes del aire, dióxido de nitrógeno, luz, sonido, movimiento, etc.

Los nodos tienen la capacidad de comunicarse entre sí con el protocolo ZigBee y retransmitir la información adquirida a través de la red hasta un punto de control que registre los valores observados e incluso tome decisiones consecuentemente. Además, son los propios nodos los que se organizan automáticamente cada cierto tiempo para ver cuáles son las rutas de comunicación disponibles.

La escalabilidad es muy importante dada la naturaleza de los escenarios en los que pueden ser instaladas. Por un lado, hace posible incrementar el área monitorizada simplemente añadiendo nodos y sin necesidad de configuración alguna, ya que serán incorporados a la red dinámicamente.

Por otra parte, el hecho de que uno de ellos desaparezca no perjudica el funcionamiento global de la red ya que la información puede seguir siendo transmitida por caminos alternativos. Esto último representa una gran ventaja frente a diseños centralizados ya que proporciona un servicio de alta disponibilidad.

La alimentación de la red sensorial se realiza de manera autónoma a través de baterías, lo que permite su instalación en sitios en los que no haya corriente eléctrica como campos de cultivo, contenedores, bosques, invernaderos, jardines, etc.

Los nodos de la red sensorial no necesitan emitir continuamente, basta con que tomen una muestra cada cierto periodo de tiempo. De esta manera, se puede conseguir alimentar cada nodo durante meses con solo una batería.

Una red ZigBee la forman básicamente 3 tipos de elementos. Un único dispositivo coordinador, dispositivos Routers y dispositivos finales (end points o end devices).

El coordinador es el nodo de la red que tiene la única función de formar una red. Es el responsable de establecer el canal de comunicaciones y del PAN ID (identificador de red) para toda la red. Una vez establecidos estos parámetros, el coordinador puede formar una red, permitiendo unirse a él a dispositivos Routers y End Points. Una vez formada la red, el coordinador hace las funciones de Router, esto es, participar en el enrutamiento de paquetes y ser origen y/o destinatario de información.

El Router es un nodo que crea y mantiene información sobre la red para determinar la mejor ruta para transmitir un paquete de información. Lógicamente un router debe unirse a una red ZigBee antes de poder actuar como Router retransmitiendo paquetes de otros routers o de End points.

Los dispositivos finales son los encargados de recoger la información de los sensores y transmitirlas al medio para que estos datos sean gestionados por el coordinador. Normalmente estos equipos van alimentados con baterías. El consumo es menor al no tener que realizar funciones de enrutamiento.

Las ventajas que se pueden obtener de las redes sensoriales con ZigBee son:

- Escalabilidad.

La capacidad de auto-organización de la red permite que la monitorización de mayores espacios pueda hacerse fácilmente añadiendo nuevos nodos. Si se añaden nuevos comunicadores, estos son reconocidos automáticamente por el resto, los cuales actualizarán sus tablas de rutas en consecuencia.

- Alta disponibilidad.

La misma característica de auto-organización permite que el área de monitorización también pueda reducirse. En caso de que un nodo deje de funcionar, el resto de valores pueden seguir siendo transmitidos por caminos alternativos gracias al algoritmo de enrutamiento dinámico, que se encarga de actualizar las rutas. De esta manera, se garantiza siempre el funcionamiento del resto de la red, al contrario de lo que sucedería con un modelo de comunicación centralizado.

- Fácil implantación.

El hecho de que la comunicación sea inalámbrica, eliminando así la necesidad de trasladar cables de red, y que la alimentación se realice por medio de baterías, hace que la instalación sea fácil y posible en cualquier entorno, abriendo así un gran abanico de aplicaciones que pueden beneficiarse de esta tecnología.

Para poder operar en las redes 802.15.4, ZigBee, se tiene que tener integrados una serie de módulos inteligentes.

Los módulos utilizados en el proyecto son de la familia DIGI RF, anteriormente conocida como MaxStream, y son los llamados módulos XBee y XBee PRO.

Los módulos XBee son transeptores de radio frecuencia que trabajan en la banda de 2.4 GHz con protocolo de comunicación 802.15.4.

El modulo XBee, representado en la Figura 2.18, presenta una potencia de salida de 1mW, y la sensibilidad del receptor es de -92dBm. Esto le permite operar hasta a unos 100m en espacios abiertos, y hasta 30m en espacios urbanos.

El modulo XBee PRO presenta una potencia de salida de 60mW (18dBm), y la sensibilidad del receptor es de -100dBm. Esto le permite operar a más de 1km en espacios abiertos, y hasta 300m en espacios urbanos.

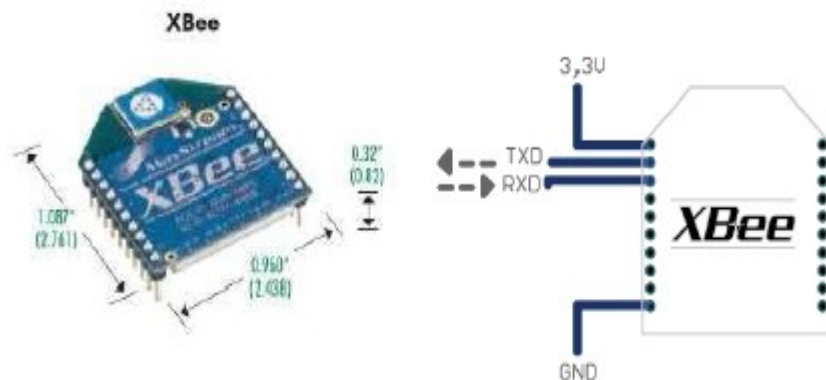


Figura 2.18 - Módulos XBee

La característica fundamental que ha destacado a los productos MaxStream es la sensibilidad de sus receptores. En vez de elevar la potencia del transmisor, con un aumento de consumo, para lograr mayor alcance, los productos MaxStream incorporan receptores mas sensibles, lo que les permite lograr un mayor alcance que otros productos similares, manteniendo un consumo reducido.

La corriente de operación de estos dispositivos ronda los 50mA, tanto para transmisión (215mA a máxima potencia para el XBee PRO) como para recepción, mientras que en el modo de bajo consumo se reduce a tan solo 10 μ A. La tensión de operación es 3 a 3,3V.

Las opciones de antena son: antena chip integrada en el modulo, antena de cuarto de onda en el módulo, conector MMCX para antena externa al modulo

El diseño exterior esta basado en un formato de algo menos de 27x25mm, la conexión a la PCB se realiza mediante dos conectores de 10 pines de paso 2mm separados 22mm entre si.

Los módulos XBee tienen 5 modos diferentes de operar:

- **Modo Recibir/Transmitir.**
Se encuentra en estos modos cuando el modulo recibe algún paquete RF a través de la antena (modo Receive) o cuando se envía 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 por el modulo durante un periodo de tiempo y es enviada solo cuando la dirección de destino la solicita.
- **Modo de bajo consumo.**
El modo de Bajo Consumo o modo Sleep, corresponde cuando el modulo entra en un estado de bajo consumo de energía. Esto depende de la configuración en la que se encuentra. Se debe indicar que el ahorro de energía depende enormemente del voltaje de alimentación.
- **Modo Comando.**

Este modo permite ingresar comandos AT al modulo 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.

- **Modo transparente.**
En el modo transparente, el modulo envía al remoto configurado como destinatario los mensajes que recibe por su puerto serie, y presenta en este los mensajes que recibe del modulo remoto. Los cambios de configuración se realizan escapando a modo comando e ingresando comandos AT.
- **Modo API.**
En el modo API, especialmente recomendado para el modulo que debe cumplir la función de coordinador, tanto comandos como mensajes y respuestas viajan dentro de un framing documentado, permitiendo mandar y recibir mensajes a y de múltiples remotos de forma sencilla, en una única interfaz serie.

Poner en marcha por primera vez uno de estos módulos es sencillo, un programa gratuito (XCTU) nos permite configurarlo de forma clara y simple, mediante una interfaz que nos deja ver los comandos AT sin necesidad de recordarlos para poder usarlos.

El direccionamiento de los módulos XBee puede ser de 16 o de 64 bits. Si se utiliza el direccionamiento de 64 bits, el número de elementos que formen la red es mayor (2^{64}) que en el de 16, pero las direcciones de cada modulo no pueden ser definidas arbitrariamente por el usuario como ocurre con el direccionamiento de 16 bits y hay que utilizar el numero serial que viene de fabrica.

El número máximo de direcciones de red que se pueden asignar a los dispositivos es de $2^{16} = 65535$, que coincide con el número máximo de elementos que pueden formar una red ZigBee cuando se utilizan direcciones de 16 bits.

En la red diseñada para este proyecto se van a direccionar los módulos con 16 bits ya que no se necesita un número elevado de elementos.

Los módulos XBee, pueden ser ajustados para usarse en redes de configuración punto-a-punto o punto-a-multipunto. Teniendo en cuenta que los módulos XBee Serie 2 permiten hacer redes mesh, y la serie 1 no.

Hay varias tipos de redes que soporta XBee:

- **Conexión Punto a Punto.**
Un modulo se conecta con otro como se muestra en la Figura 2.19.



Figura 2.19 – Conexión Punto a Punto

- **Conexión Punto a Multipunto.**

Permite transmitir información desde la entrada serie de un módulo a uno o varios módulos conectados a la misma red de manera más controlada, ya que se necesitan las direcciones de los otros módulos, por lo que existe mayor seguridad.

Todos los módulos que tengan idéntico PAN ID, pertenecerán a la misma red.

En la Figura 2.20 está representada este tipo de red punto a multipunto y como se configura. La elección del canal debe ser cuidadosa, ya que otras tecnologías como WiFi o bluetooth utilizan el mismo espectro de frecuencias, por lo que se podría producir interferencias.

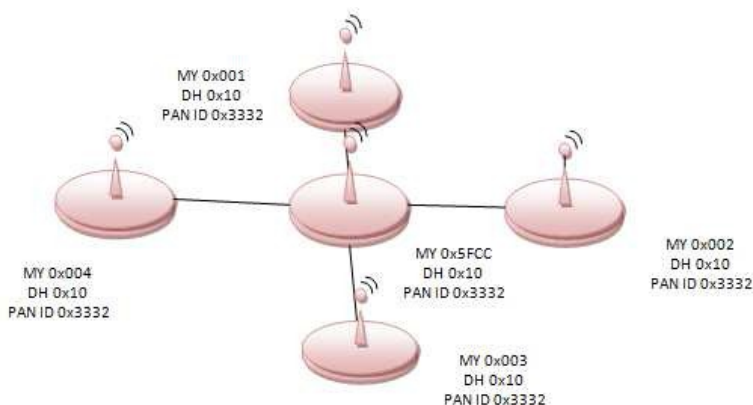


Figura 2.20 – Conexión Punto a Multipunto

- **Conexión NonBeacon. Peer to Peer.**

Una red peer-to-peer permite que todos los módulos se conecten con todos, es decir, se crea una conexión de par en par con cada uno de los módulos de la red. El modo de conexión Non-Beacon es la configuración por defecto y permite establecer una red peer-to-peer donde cada modulo puede hacer las funciones de maestro o esclavo.

La configuración de red Non-Beacon, se refiere a que cada nodo primario (o nodo central maestro) se mantiene despierto siempre. Por lo que los demás dispositivos que se conectan a el, pueden entrar en modo SLEEP (ahorro de energía), y solo despertarse cuando sea necesario para enviar datos. En una red Beacon, los dispositivos enrutadores (o nodos primarios) están siempre en modo SLEEP, y envían señales de su existencia (llamadas Beacon) cada cierto intervalo al resto de la red. Así para poder comunicarse, deben estar

todos los dispositivos totalmente organizados, ya que de no ser así, existe la posibilidad de perder la señal Beacon y no poder enviar hasta la próxima entrega. La ventaja de las redes Beacon, es el ahorro de energía. Por este motivo las redes Non-Beacon están pensadas para dispositivos que posean una alimentación segura, mientras las Beacon, para alimentación autónoma, como baterías. Los módulos XBee Series 1, solo soportan redes Non-Beacon.

- **Conexión NonBeacon con coordinador.**
Es básicamente lo mismo que una red punto-multipunto con la diferencia de que existe un modulo central que posee ciertas propiedades y características que le permiten administrar la red. En esta red, el modulo central es llamado coordinador, mientras que el resto de módulos son llamados Dispositivos Terminales (End Device) como se representa en la Figura 2.21. Un mismo modulo XBee puede ser configurado para funcionar como coordinador o como Dispositivo Terminal.

Para configurar esta red, todos los módulos deben tener el mismo canal y la misma PAN. El modulo coordinador se configura como ATCE=1 (ATCE1), mientras que todos los demás, los cuales serán llamados Dispositivos Terminales, se configuran como ATCE=0 (ATCE0).

Este tipo de configuración se usa cuando se necesita una unidad central para enviar mensajes a varios módulos, o juntar información proveniente de varios Dispositivos Terminales, asignar canales o ID de redes PAN.

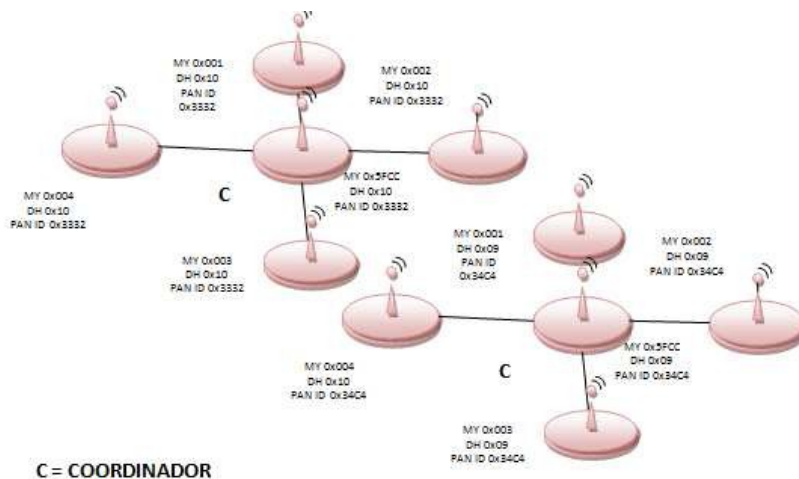


Figura 2.21 - Red NonBeacon con coordinador

En este proyecto se ha optado por una conexión Punto a Punto. Los dispositivos

Gestión y monitorización de contenedores intermodales mediante tecnología inalámbrica

de la red serán los módulos XBee emisores, y el coordinador será el modulo XBee receptor que se encuentra unido al PC y es el encargado de procesar toda la información que le llegue.

3. DISEÑO e IMPLEMENTACIÓN

3.1 Diseño

Ahora se dará una visión del diseño del sistema implementado en este proyecto.

A la hora de la creación de la base de datos, primero se realizó el modelo entidad-relación y posteriormente, el modelo relacional.

El modelo entidad-relación, ver figura 3.1, se realizó de la siguiente manera:

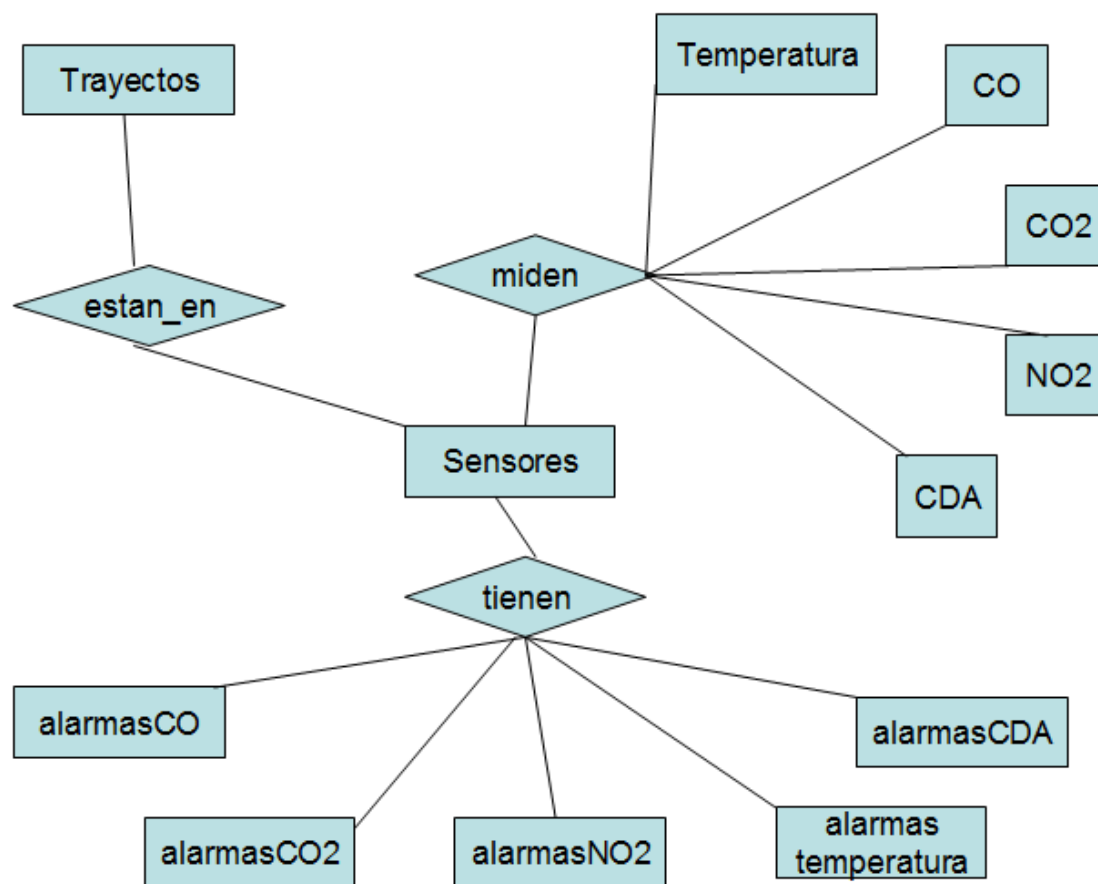


Figura 3.1 – Modelo Entidad-Relación desarrollado

Como podemos apreciar en la imagen, la tabla “Sensores” se relaciona con las demás tablas.

Mediante la relación “están_en”, la tabla “Sensores” y la tabla “Trayectos” se relacionan. Esta relación, la necesitamos para poder saber que sensores pertenecen a qué trayectos. De esta manera, podemos saber en todo momento los sensores que pertenecen a los trayectos y así poder obtener las medidas referentes a un determinado trayecto. Además, puesto que los sensores están localizados en una cierta posición, podemos saber la posición exacta en la que se encuentra un sensor en un trayecto.

Mediante la relación, “miden”, tenemos relacionado las tablas de “temperatura”, “monóxido de carbono”, “dióxido de carbono”, “dióxido de nitrógeno” y “contaminantes del aire”. En estas tablas almacenamos los datos recogidos por los sensores. Teniendo esta relación, podemos saber que sensores tienen que medidas.

En las tablas “alarmasco”, “alarmaco2”, “alarmasno2”, “alarmastemperatura” y “alarmascda” tenemos las diferentes alarmas que podemos crear de un determinado sensor. Estas alarmas pueden ser para que se nos avise cuando una temperatura es mayor, menor o igual de unos grados centígrados, cuando la medida de CO2 es mayor, menor o igual que una cierta cantidad establecida por el usuario de la aplicación,...

Decir que además de las tablas mostradas en la Figura 3.1, tenemos la tabla “usuarios” en donde almacenamos el usuario, contraseña, nombre, apellido, DNI, domicilio, población, provincia y correo electrónico de los usuarios de la aplicación. Además, tenemos una tabla auxiliar llamada “eleccion_comparar” que la utilizamos para realizar las comparaciones entre varios sensores, varios trayectos y varias posiciones.

Una vez realizado el Modelo Entidad-Relación, se realizó el modelo relación. Las diferentes tablas que nos quedan en el proyecto realizado son las siguientes:

- alarmascda(id,alarma,cardinal)

Tiene 3 atributos: “id”, “alarma” y “cardinal”.

El “id” hace referencia al identificador del sensor, “alarma” al dato que mete el usuario de aplicación y el “cardinal” puede ser mayor, igual o menor.

- alarmasco(id,alarma,cardinal)

Tiene 3 atributos: “id”, “alarma” y “cardinal”.

El “id” hace referencia al identificador del sensor, “alarma” al dato que mete el usuario de aplicación y el “cardinal” puede ser mayor, igual o menor.

- alarmascodos(id,alarma,cardinal)

Tiene 3 atributos: “id”, “alarma” y “cardinal”.

El “id” hace referencia al identificador del sensor, “alarma” al dato que mete el usuario de aplicación y el “cardinal” puede ser mayor, igual o menor.

- alarmasnodos(id,alarma,cardinal)

Tiene 3 atributos: “id”, “alarma” y “cardinal”.

El “id” hace referencia al identificador del sensor, “alarma” al dato que mete el usuario de aplicación y el “cardinal” puede ser mayor, igual o menor.

- alarmastemperatura(id,alarmas,cardinal)

Tiene 3 atributos: “id”, “alarma” y “cardinal”.

El “id” hace referencia al identificador del sensor, “alarma” al dato que mete el usuario de aplicación y el “cardinal” puede ser mayor, igual o menor.

- cda

Tiene 3 atributos: “id”, “medida” y “fecha”.

El atributo “id” guarda el identificador del sensor que ha realizado la medida. La “medida” es el dato que ha tomado el sensor y en “fecha” guarda el día, mes, año, hora, minutos y segundos en la que se realizó la medida.

- co

Tiene 3 atributos: “id”, “medida” y “fecha”.

El atributo “id” guarda el identificador del sensor que ha realizado la medida. La “medida” es el dato que ha tomado el sensor y en “fecha” guarda el día, mes, año, hora, minutos y segundos en la que se realizó la medida.

- codos

Tiene 3 atributos: “id”, “medida” y “fecha”.

El atributo “id” guarda el identificador del sensor que ha realizado la medida. La “medida” es el dato que ha tomado el sensor y en “fecha” guarda el día, mes, año, hora, minutos y segundos en la que se realizó la medida.

- nodos

Tiene 3 atributos: “id”, “medida” y “fecha”.

El atributo “id” guarda el identificador del sensor que ha realizado la medida. La “medida” es el dato que ha tomado el sensor y en “fecha” guarda el día, mes, año, hora, minutos y segundos en la que se realizó la medida.

- temperatura

Tiene 3 atributos: “id”, “medida” y “fecha”.

El atributo “id” guarda el identificador del sensor que ha realizado la medida. La “medida” es el dato que ha tomado el sensor y en “fecha” guarda el día, mes, año, hora, minutos y segundos en la que se realizó la medida.

- eleccion_comparar

Tiene diversos atributos que se utilizan para realizar las comparaciones entre trayectos, posiciones y medidas totales de temperatura, monóxido de carbono, dióxido de carbono,...

- sensores

En esta tabla se guarda la información relativa al identificador del sensor (atributo “id”), posición en el trayecto (atributo “posición”), trayecto al que pertenece (atributo “trayecto”) y la posición “x”, “y” y “z” del sensor, entre otros.

- trayecto

En esta tabla tenemos los siguientes atributos: id, origen, destino, hora_inicio, hora_fin, modo_transporte y responsable.

En “id” guardamos el identificador del trayecto, en “origen” y “destino” guardamos el origen y el destino del trayecto, en “hora_inicio” y “hora_fin” se guarda la hora de salida y la hora de llegada, en “modo_transporte” almacenamos el transporte, es decir, si es en tren, en avión, en barco,... y en “responsable” guardamos el nombre y apellido del responsable del trayecto.

- usuarios

En la tabla usuarios, almacenamos la información relativa a los usuarios de la aplicación Web. Estos son el usuario y la contraseña, el nombre y los apellidos, el DNI, el domicilio, población y provincia y el correo electrónico del usuario.

En la figura 3.2, se muestra la WSN y las conexiones entre los diferentes End Device, Coordinador y PC.

Existe una conexión direccional entre los End Device y el coordinador. En esta conexión se mandan los datos recogidos.

Entre el coordinador y el PC existe una conexión física cableada que sirve para transmitir los datos recibidos al ordenador.

En la figura 3.2 se muestran 3 End Device pero en la realidad podría haber el numero que se quisiese.

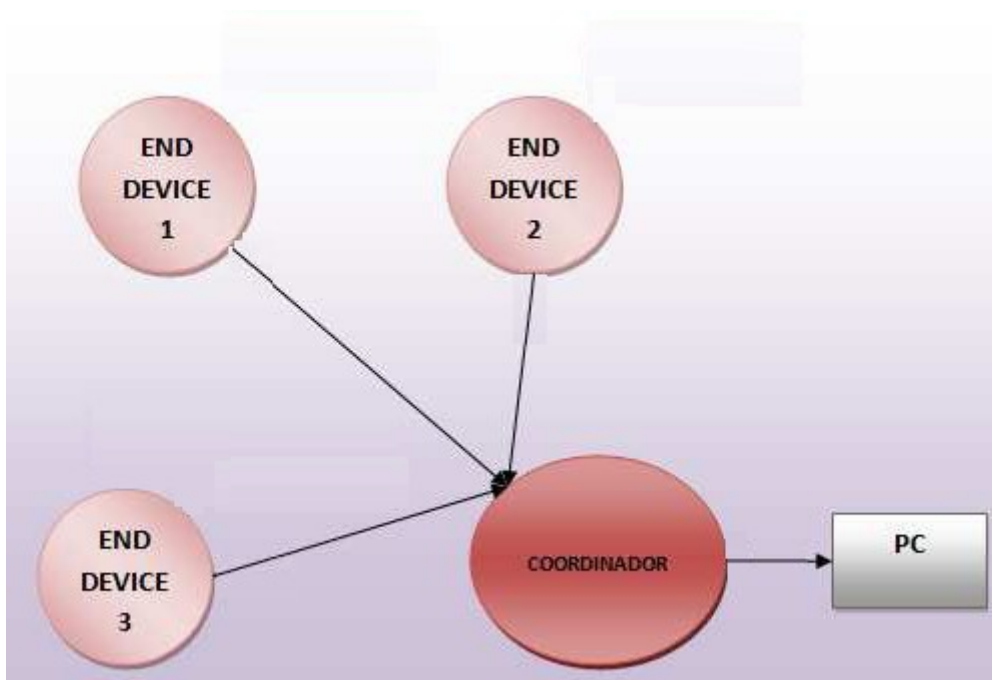


Figura 3.2 – Esquema parecido del sistema implementado

El sistema esta dividido en tres bloques:

- Bloque emisor: Los dispositivos End Device se encarga de transmitir las medidas realizadas.

Estos dispositivos cuentan con un modulo XBee y una pila de batería.

- Bloque receptor: El coordinador recibe los datos del bloque emisor y los gestiona con el PC.
- Bloque de procesado de datos: El PC al que va unido el coordinador se encarga de recibir los datos del coordinador y de insertar los mismos en una base de datos.

3.2 Implementación

Se explicará ahora el funcionamiento, los métodos y medidas llevados a cabo para realizar el proyecto.

Entre los diversos motes comerciales se ha elegido Waspote.

Se explicará ahora los diferentes motes existentes en el mercado y el porque de elegir Waspote.

3.2.1 Motes comerciales

3.2.1 Micaz

La plataforma de sensores MICAZ (ver Figura 3.3) es comercializada por Crossbow [17]. Estos dispositivos, trabajan en la banda de frecuencias de 2400 MHz a 2483.5 MHz. La familia MICAZ usa el Chipcon CC2420, que cumple con la normativa IEEE 802.15.4 y tiene un transceptor de radio frecuencia Zigbee integrado con un micro controlador Atmega 128L. Dispone de un conector de expansión de 51 pines así como de 512 KBytes memoria flash no volátil (para el almacenamiento de datos) y 128 KBytes de memoria Flash para llevar a cabo la programación del dispositivo.

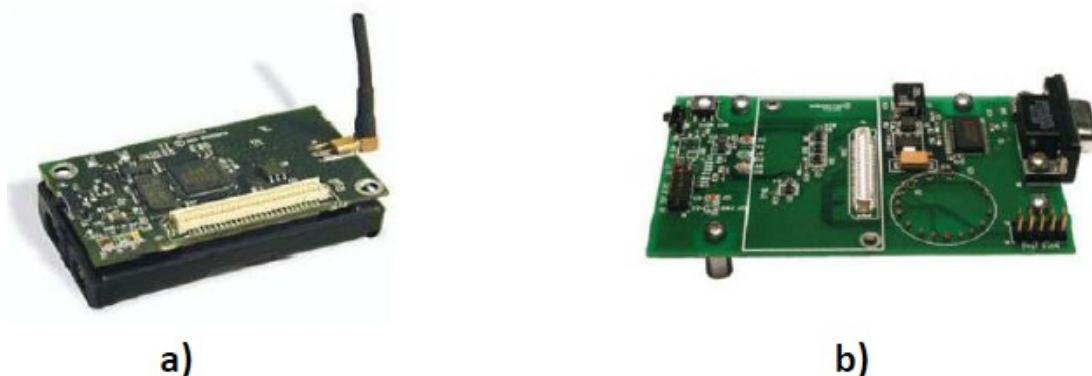


Figura 3.3 – (a) Micaz Mote. (b) Placa de desarrollo MIB510

Para la programación de estos motes se emplea la Placa de desarrollo MIB510 (Figura 3.3). Esta placa actúa como interfaz entre el PC y los motes a través del puerto serie, permitiendo la programación de los dispositivos acoplados.

3.2.2 Mica2 y Mica2dot

Diseñados por investigadores de la universidad de Berkeley, en California, existen dos modelos llamados Mica2 y Mica2dot. Su funcionalidad es similar pero son muy diferentes en factor de forma, como se puede apreciar en la Figura 3.4. Funcionan ambos a 4MHz con un microprocesador de 8 bits de la marca Atmel. Tiene 128KB de memoria para el programa y 4KB de memoria RAM. Su velocidad de transmisión radio es de 19,2Kb/s sobre un canal CSMA/CA y utiliza el estándar IEEE 802.15.4. También están equipados con una memoria externa de tipo Flash no volátil con 512KB que puede ser usada para guardar otros datos.

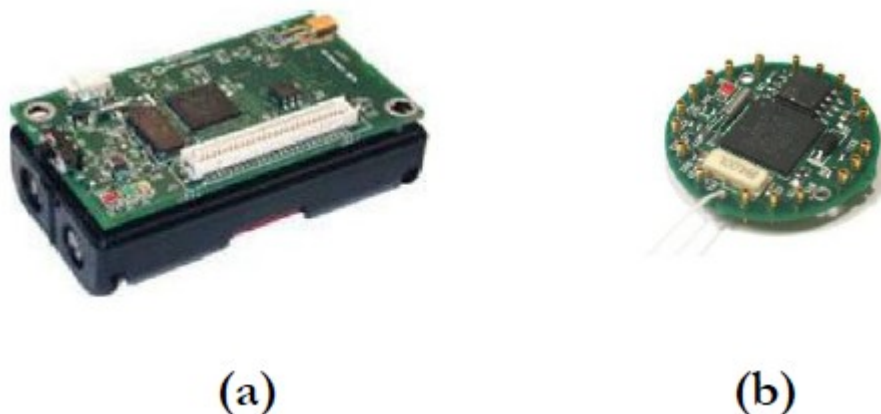


Figura 3.4 – (a) Mica2. (b) Mica2dot

A la placa principal del procesador se le pueden adherir placas con sensores/actuadores por medio de unos pines, en el caso del Mica2, o por medio de unos conectores, para el modelo Mica. Como por ejemplo, una placa con sensores de temperatura, luminosidad, un micrófono acelerómetro y un sensor magnético, entre otros.

3.2.3 Intel Mote 2

Intel Mote (ver Figura 3.5) surge de una colaboración [18] entre los laboratorios de Intel en Berkeley, la Universidad de Berkeley otros centros de investigación del resto de USA. Los Motes de Intel son pequeños, autónomos, alimentados por medio de baterías y con comunicación vía radio, de forma que son capaces de compartir información con otros y organizarse automáticamente dentro de una red AdHoc.

Uno de los principales objetivos del grupo de trabajo del Intel Mote es colaborar con la comunidad investigadora en la exploración de las potenciales aplicaciones de los

Motes y de las redes de sensores. Con este objetivo en mente, se han diseñado los Motes con una total compatibilidad con el sistema operativo TinyOS.

Las investigaciones de mejora se están centrando en tres aspectos:

- El trabajo con consumos de energía ultra bajos.
- Integración del sistema.
- Reconfiguración hardware.

Intel Mote 2 es una plataforma avanzada para la creación de una red de sensores wireless. La plataforma está construida alrededor del procesador de bajo consumo XScale PXA27x con comunicación radio 802.15.4 sobre la placa principal y una antena de 2.4GHz. Contiene 2 interfaces de sensores básicos en uno de los lados de la placa central y otros 2 interfaces avanzados para sensores en el otro lado.



Figura 3.5 – Intel Mote

El Intel Mote 2 es una plataforma modular y escalable. La placa principal contiene el procesador y el módulo radio y se le pueden agregar diferentes módulos sensores dependiendo de la aplicación final de la red de sensores.

El procesador que contiene la placa principal puede funcionar a bajo voltaje (0.85V) y una frecuencia también baja (13MHz), cuando se habilita el modo de bajo consumo, mientras que el valor máximo de operación es de 416MHz. También integra una memoria SRAM de 256KB dividida en 4 bancos de 64KB y diferentes opciones de E/S, lo que lo hace extremadamente flexible para soportar diferentes sensores A/D, opciones de radio. Estas E/S incluye I2C, 3 puertos serie síncronos, 3 puertos UART, E/S digitales, un cliente USB. Además, el propio procesador incluye diversos temporizados y un reloj en tiempo real.

Una de las características más importantes y que lo diferencian del resto de motes comerciales, es que existe una placa de expansión (ver Figura 3.6) que incorpora una cámara CMOS, sensor PIR, micrófono y altavoz capaz de reproducir sonidos con una alto nivel de detalle. Lo cual unido a las 64 MB de memoria disponibles y a su compatibilidad con TinyOS2.x.x y TinyOS 1.1.x (Sólo el altavoz, sensor PIR y micrófono), le permite desarrollar aplicaciones que son impensables en otros dispositivos comerciales.

El principal problema de este dispositivo, es su elevado precio, alrededor de 1000 € el Imote2 con la placa de expansión, se antoja muy elevado en la actualidad.

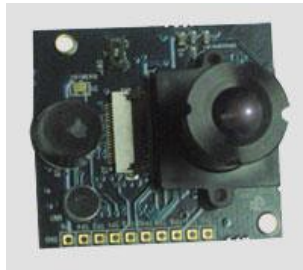


Figura 3.6 – Placa de expansión para Imote2 (cámara, altavoz, micrófono y sensor PIR)

3.2.4 TelosB

El dispositivo TelosB (ver Figura 3.7), también llamado Tmote Sky consta del microcontrolador MSP430 F1611. Este procesador RISC de 16 bits de bajo consumo está diseñado expresamente para el uso en redes de sensores inalámbricas.

Al igual que el Micaz, consta de un transceptor de radio Chipcon CC2420. Otra de las características del Telosb es que dispone de sensores de humedad, temperatura y luminosidad en la propia placa del dispositivo, lo cual permite realizar medir estas dos variables sin necesidad de incorporar una placa de expansión.



Figura 3.7 – TelosB

Una de las principales cualidades del mote TelosB es que totalmente compatible con los dos sistemas operativos más utilizados en redes de sensores inalámbricas: Contiki y TinyOS. Lo cual le ofrece una gran versatilidad a la hora de elegir cuál de ellos se adapta mejor a las características de la aplicación que se desee desarrollar.

3.2.5 Zolertia Z1

El mote Zolertia [19] (ver Figura 3.8) es un mote comercial que ha logrado hacerse un hueco en el panorama de las redes de sensores gracias a las características de las que hace gala.

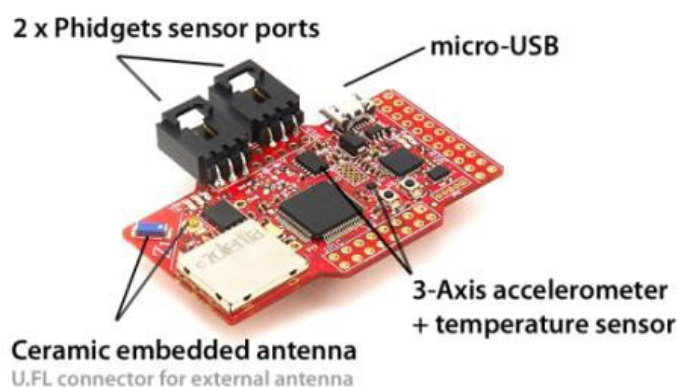


Figura 3.8 – Mote Zolertia Z1

El mote Z1 está equipado con un microcontrolador de bajo consumo MSP430F2617 de segunda generación, el cual dispone de una CPU RISC a 16MHz de frecuencia de reloj, 8 KBytes de memoria RAM y 92 KB de memoria Flash. Además incluye el transceptor CC2420 también de Texas Instruments, compatible con el estándar 802.15.4 y que opera a 2.4 GHz con una tasa efectiva máxima de 250Kbps. Además, el hardware de Z1 garantiza una gran eficiencia energética y robustez.

Este mote dispone a su vez de dos sensores integrados de fábrica. Un sensor de Temperatura y un acelerómetro, ambos digitales y listos para funcionar. Además es compatible con los sensores Phidgets y una gran variedad de sensores analógicos y digitales. Es programable a través del puerto USB. En lo que respecta a la alimentación ofrece un amplio abanico de posibilidades de conexión:

- Pilas AA o AAA.
- Pila de Botón (hasta 3.6V).
- Alimentación directa por USB.
- Conexión directa a una fuente de alimentación a través de dos cables (desde 4 hasta 5 V).

La característica más importante de este mote es que a pesar de ser relativamente reciente, es totalmente compatible tanto con TinyOS como con Contiki. Este hecho es esencial ya que el único mote de los vistos hasta ahora que era compatible con ambos sistemas operativos era el Telosb. Sin embargo, las características técnicas (tanto transceptor de radio como microcontrolador) del Telosb son muy inferiores a las del Z1. Por tanto el mote Z1 es una opción muy interesante tanto por sus características técnicas como por su compatibilidad software.

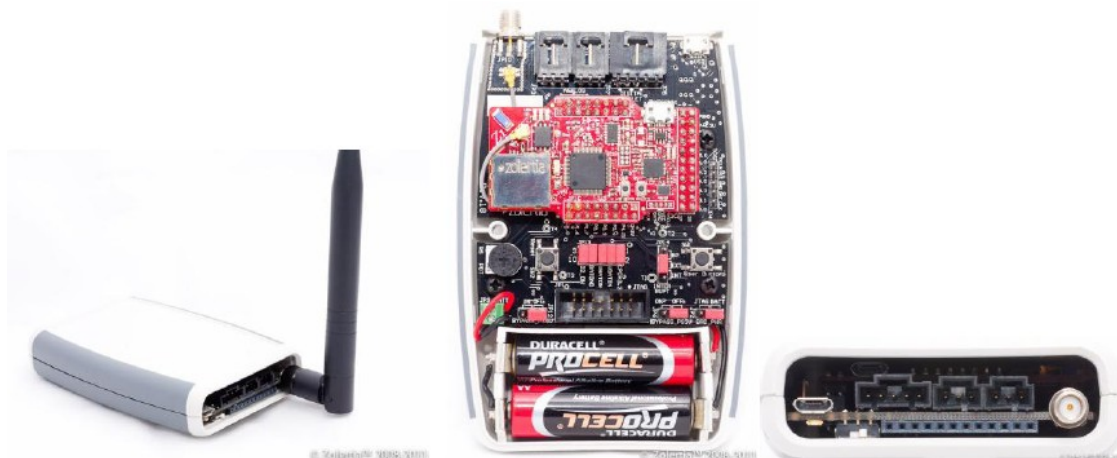


Figura 3.9 – Mote Z1 en su caja comercial y con su placa de conexionado (centro)

3.2.6 Wasmote

Wasmote es un dispositivo sensorial especialmente orientado a desarrolladores. Trabaja sobre diferentes protocolos (ZigBee, Bluetooth, GPRS) y frecuencias (2.4GHz, 868MHz, 900MHz) siendo capaz de obtener enlaces de hasta 12km. Cuenta con un modo de hibernación con un consumo de 0.7uA que permite ahorrar batería cuando no está transmitiendo. Están disponibles más de 50 sensores y un entorno de desarrollo (IDE) completamente open source (IDE + librerías + compilador) que hace realmente sencillo empezar a trabajar con la plataforma.

El módulo principal del Wasmote (ver Figura 1.23) de un microcontrolador ATmega1281 que funciona a 8 MHz y dispone de 8 KB de memoria RAM y 128 KB de memoria FLASH. En conjunto con el microcontrolador, la placa principal cuenta con un Reloj en Tiempo Real con su batería de respaldo y de un slot microSD.

La idea de Libelium es vender el resto de componentes del dispositivo en función de las necesidades del cliente. Es por ello que el dispositivo cuenta con diversos conectores de expansión que le permiten añadir diferentes elementos en función de las necesidades finales.

En lo que respecta a la conectividad vía radio, se le pueden añadir diferentes módulos XBee o bien un módulo Bluetooth. Además también permite añadir un módulo GSM/ GPRS para el envío y recepción de datos a través de la red GSM. También es posible añadir un módulo GPS con su correspondiente antena así como diferentes tipos de sensores destinados en su mayoría a la agricultura.

Libelium ofrece además un dispositivo gateway que permite comunicarse con los Wasmotes por radio, bluetooth o GSM/GPRS y hacer de pasarela hacia internet. De forma que a pesar de que los Wasmotes no soportan el protocolo IP, el gateway hace de pasarela y nos permite la comunicación a través de él, por ejemplo por medio de una red Wi-Fi.

El principal inconveniente del Wasmote es que no es compatible con Contiki ni con TinyOS. Lo cual limita en gran medida el uso del dispositivo. Ya que se depende totalmente del soporte del fabricante.

3.2.7 Comparativa

En la siguiente Tabla se puede apreciar una comparativa entre los diferentes motes comerciales, mostrando sus características más importantes.

Mote		MICA2	MICA2Dot	TelosB	Imote2	Zolertia Z1	Wasmote
MCU	Chip	ATmega128L		Msp430f1611	PXA271	MSP430F2617	ATmega1281
	Tipo	8 bits		16 bits	32 bits	16 bits	8 bits
	Memoria Programa (KB)	128		48	N/A	96KB	128KB
	RAM (KB)	4		10	32768	8KBytes	8KBytes
Almacenamiento Externo No Volátil	Chip	AT45DB014B		M25P80	N/A	N/A	N/A
	Conexión	SPI		SPI	N/A	N/A	SPI
	Tamaño (KB)	512		1024	32768	N/A	SD hasta 2000
Sistema de Alimentación	Tipo	2xAA	Coin Cell	2xAA	3xAAA	2xAA/A o Coin Cell	Variable
	Capacidad Típica (mAh)	2850	1000	2850	1100	1100	2000
RF	Chip	CC1000		CC2420	CC2420	CC2420	Variable (Xbee)
	Frecuencia	868/916, 433 ó 315 MHz		2,4GHz	2,4GHz	2.4GHz	2.4GHz
	datarate (Kbps)	38,4		250	250	250K	250
	Potencia Transmisión (dBm)	5		0	0	0	Hasta 18dBm
	Alcance Exterior (m)	152,4	152.4-304,8	75-100	~30	150	Hasta3200
Año comercialización		2002	2002	2005	2007	2009	2009

Tabla 3.1 – Características principales de diferentes motes

Referente a la programación de los sensores, se ha utilizado el lenguaje C y C++. El código de los “sensores cliente” y el de los “sensores servidores” es diferente.

Se explicará primero el código de los “sensores cliente”. En dicho sensor lo primero que se hace es iniciar el puerto USB, iniciar la SD, iniciar el ZigBee e iniciar el RTC. Esto se realiza mediante lo siguiente (ver Figura 3.10):

```
void setup() {  
  
  USB.begin();  
  
  SD.ON();  
  
  xbeeZB.init(ZIGBEE,FREQ2_4G,NORMAL);  
  
  xbeeZB.ON();  
  
  RTC.ON();  
  
}
```

Figura 3.10 – Iniciar USB, SD y ZigBee

Posteriormente, nos conectamos a la red creada por el “sensor servidor” (ver Figura 3.11)

```
connect();  
  
uint8_t connect(void)  
{  
  xbeeZB.setScanningChannels(0xFF,0xFF);  
  xbeeZB.setDurationEnergyChannels(6);  
  xbeeZB.getAssociationIndication();  
  
  if(!xbeeZB.associationIndication)  
  {  
    xbeeZB.getChannel();  
    xbeeZB.getExtendedPAN();  
  
    xbeeZB.getOperatingPAN();  
  
    xbeeZB.getOwnNetAddress();  
  
    xbeeZB.writeValues();  
  
    return 0x00;  
  }  
  else  
  {  
    return 0x01;  
  }  
}
```

Figura 3.11 - Conectarse a la red

Después, encendemos la placa de gases y configuramos el monóxido de carbono, el dióxido de carbono, el dióxido de nitrógeno y el contaminante del aire (ver Figura 3.12).

```
// Placa de gases
SensorGas.setBoardMode(SENS_ON);

// CO2
SensorGas.configureSensor(SENS_CO2, 1);
SensorGas.setSensorMode(SENS_ON, SENS_CO2);

// CO
SensorGas.configureSensor(SENS_SOCKET3B,1,0.5);

// NO2
SensorGas.configureSensor(SENS_SOCKET2B, 1, 0.5);
SensorGas.setSensorMode(SENS_ON, SENS_SOCKET2B);

// Contaminantes del Aire
SensorGas.configureSensor(SENS_SOCKET2A, 1 , 0.5);
SensorGas.setSensorMode(SENS_ON, SENS_SOCKET2A);
```

Figura 3.12 – Configuración placa de gases

Todo lo dicho hasta ahora se ha realizado en el void setup().

En el void loop() , lo que se hace es medir la temperatura, medir el valor de dióxido de carbono, del monóxido de carbono, del dióxido de nitrógeno y del contaminante del aire (ver Figura 3.13).

```
temperatura = RTC.getTemperature();

valorCO2 = SensorGas.readValue(SENS_CO2);

valorCO = SensorGas.readValue(SENS_SOCKET3B);

valorNO2 = SensorGas.readValue(SENS_SOCKET2B);

valorCdA = SensorGas.readValue(SENS_SOCKET2A);
```

Figura 3.13 – Mediciones

Cuando se miden los valores se mandan los mensajes (paquetes) al servidor. Mediante este código mandamos el paquete a la dirección MAC del servidor con el dato medido (ver Figura 3.14)

```
xbeeZB.send_to("0013A20040607364",paquete,0x51,NON_BLOCK);
```

Figura 3.14 – Envío del paquete

Este el código que se ha realizado en el cliente.

El código del “sensor servidor”, el que se encarga de recibir los datos, es el siguiente. Primero se inicia el puerto USB y se inicia el ZigBee. Esto se realiza de la siguiente manera (ver Figura 3.15):

```
USB.begin();  
  
xbeeZB.init(ZIGBEE,FREQ2_4G,NORMAL);  
  
xbeeZB.ON();
```

Figura 3.15 – Iniciar USB y ZigBee

Después, creamos la red con el PAN-ID (ver figura 3.16).

```
uint8_t panid[8]={8,7,6,5,4,3,2,1};  
  
xbeeZB.getChannel();  
  
xbeeZB.getExtendedPAN();  
  
xbeeZB.getOperatingPAN();
```

Figura 3.16 – Creación de la red

Una vez creada la red, esperamos a recibir los paquetes (ver Figura 3.17):

```
packet = xbeeZB.receive(WAIT_INF,0);
```

Figura 3.17 – Recepción de paquetes

Esté es el código tanto de los “sensores cliente” como del “sensor servidor”.

Referente a la base de datos, hemos creado 14 tablas que nos sirven para guardar datos de las medidas realizadas, datos de las alarmas creadas, datos de los usuarios, datos de los trayectos, datos de los sensores y datos auxiliares referentes a la comparación entre trayectos, posiciones y sensores.

La tabla donde guardamos la información a los sensores se llama “sensores” y tiene esta estructura (ver Figura 3.18).

Columna	Tipo
id	varchar(4)
medida	varchar(10)
fecha	varchar(30)

Figura 3.18 – Tabla “sensores”

En la “id” guardamos el identificador del sensor, en “medida” guardamos el dato medido y en “fecha” guardamos el día, mes, año, hora, minutos y segundos en que se realizo la medida.

La tabla donde guardamos la información relacionada con los trayectos se llama “trayecto” y esta implementada de la siguiente manera (ver Figura 3.19).

Columna	Tipo
<u>ID</u>	varchar(6)
Origen	varchar(70)
Destino	varchar(70)
Hora_Inicio	varchar(5)
Hora_Fin	varchar(5)
Modo_Transporte	varchar(10)
Responsable	varchar(60)

Figura 3.19 – Tabla “trayecto”

Ahora se mostrará la estructura de las tablas donde almacenamos las medidas realizadas. Estas tablas se llaman “temperatura” (ver Figura 3.20), “co” (ver Figura 3.21), “codos” (ver Figura 3.22), “nodos” (ver Figura 3.23) y “cda” (ver Figura 3.24).

Columna	Tipo
id	varchar(4)
medida	varchar(10)
fecha	varchar(30)

Figura 3.20 – Tabla “temperatura”

Columna	Tipo
id	varchar(4)
medida	varchar(10)
fecha	varchar(30)

Figura 3.21 – Tabla “co”

Columna	Tipo
id	varchar(4)
medida	varchar(10)
fecha	varchar(30)

Figura 3.22 – Tabla “codos”

Columna	Tipo
id	varchar(4)
medida	varchar(10)
fecha	varchar(30)

Figura 3.23 – Tabla “nodos”

Columna	Tipo
id	varchar(4)
medida	varchar(10)
fecha	varchar(30)

Figura 3.24 – Tabla “cda”

La tabla “usuarios” almacena diversa información de los usuarios de la aplicación Web (ver Figura 3.25). Como datos importantes hay que destacar el “user” y “pass” que servirán para poder acceder a la página.

Columna	Tipo
<u>user</u>	varchar(20)
pass	varchar(20)
nombre	varchar(40)
apellido	varchar(40)
DNI	varchar(9)
domicilio	varchar(60)
poblacion	varchar(20)
provincia	varchar(20)
correo	varchar(40)

Figura 3.25 – Tabla “usuarios”

Tenemos 5 tablas en donde guardamos los datos metidos por el usuario referente a las alarmas. Las tablas se llaman: “alarmastemperatura” (ver Figura 3.26), “alarmasco” (ver Figura 3.27), “alarmascodos” (ver Figura 3.28), “alarmasnodos” (ver Figura 3.29) y “alarmasca” (ver Figura 3.30).

Columna	Tipo
<u>id</u>	varchar(4)
<u>alarma</u>	varchar(10)
<u>cardinal</u>	varchar(8)

Figura 3.26 – Tabla “alarmastemperatura”

Columna	Tipo
<u>id</u>	varchar(4)
<u>alarma</u>	varchar(10)
<u>cardinal</u>	varchar(8)

Figura 3.27 – Tabla “alarmasco”

Columna	Tipo
<u>id</u>	varchar(4)
<u>alarma</u>	varchar(10)
<u>cardinal</u>	varchar(8)

Figura 3.28 – Tabla “alarmascodos”

Columna	Tipo
<u>id</u>	varchar(4)
<u>alarma</u>	varchar(10)
<u>cardinal</u>	varchar(8)

Figura 3.29 – Tabla “alarmasnodos”

Columna	Tipo
<u>id</u>	varchar(4)
<u>alarma</u>	varchar(10)
<u>cardinal</u>	varchar(8)

Figura 3.30 – Tabla “alarmasca”

Finalmente, tenemos la tabla auxiliar “eleccion_comparar” que utilizamos para lo descrito anteriormente. La estructura de dicha tabla se puede ver en la figura 3.31.

Columna	Tipo
tipo	varchar(4)
trayecto	varchar(4)
primertrayecto	varchar(6)
segundotrayecto	varchar(6)
dia	varchar(2)
mes	varchar(2)
ano	varchar(4)
dia1	varchar(2)
mes1	varchar(2)
ano1	varchar(4)
dia2	varchar(2)
mes2	varchar(2)
ano2	varchar(4)
posicion	varchar(6)
posicion1	varchar(6)
posicion2	varchar(6)

Figura 3.31 – Tabla “eleccion_comparar”

Estas son todas las tablas realizadas en el proyecto.

Para insertar los valores en la base de datos recibidos en el “sensor servidor”, se ha creado un programa realizado en Visual Basic 6, que se encarga de recibir los datos por el puerto serie y de realizar la conexión con la base de datos para insertar las medidas recibidas.

Al ejecutar el programa, abrirá el puerto en donde se encuentre conectado el sensor servidor y cargara el driver MySQL (ver Figura 3.32)


```

Dim conn As New ADODB.Connection
Dim rs As New ADODB.Recordset

Private Sub Form_Load()

Comm1.CommPort = 4
Comm1.RTSEnable = False
Comm1.RThreshold = 1
Comm1.Handshaking = 0
Comm1.Settings = "38400,n,8,1"
Comm1.PortOpen = True

conn.ConnectionString = "Driver={MySQL ODBC 3.51 Driver};Server=localhost;Database=proyecto;User=root;Password=toor;Option=3;"
conn.Open

If (rs.State <> adStateClosed) Then
    rs.Close
End If
rs.ActiveConnection = conn

End Sub

```

Figura 3.32 – Abrir puerto y cargar driver MySQL

Una vez abierto el puerto, ya podemos recibir los datos. Según el dato que recibamos, lo insertaremos en la tabla correspondiente. Este es el código que se encarga de recibir el dato y de insertarlo en la base de datos (ver Figura 3.33)

```

Private Sub Comm1_OnComm()
If Comm1.CommEvent = 2 Then
dato = Comm1.Input

    If tipo = "CO2" Then
        fecha = Date$
        hora = Time$
        rs.Open "INSERT INTO codos (id,medida,fecha) VALUES (' & identificador & ',' & medida & ',' & fecha & " & hora & "')"

    ElseIf tipo = "CO" Then
        fecha = Date$
        hora = Time$
        rs.Open "INSERT INTO co (id,medida,fecha) VALUES (' & identificador & ',' & medida & ',' & fecha & " & hora & "')"

    ElseIf tipo = "NO2" Then
        fecha = Date$
        hora = Time$
        rs.Open "INSERT INTO nodos (id,medida,fecha) VALUES (' & identificador & ',' & medida & ',' & fecha & " & hora & "')"

    ElseIf tipo = "CDA" Then
        fecha = Date$
        hora = Time$
        rs.Open "INSERT INTO cda (id,medida,fecha) VALUES (' & identificador & ',' & medida & ',' & fecha & " & hora & "')"

    ElseIf tipo = "T" Then
        fecha = Date$
        hora = Time$
        rs.Open "INSERT INTO temperatura (id,medida,fecha) VALUES (' & identificador & ',' & medida & ',' & fecha & " & hora & "')"

    End If
End If
End Sub

```

Figura 3.33 - Recibir datos e insertarlos

Referente a la implementación de la página Web, los lenguajes que se han utilizado para la realización son: HTML, PHP y CSS.

Mediante CSS se ha conseguido una interfaz más amigable al usuario. Parte del archivo css utilizado es el siguiente (Figura 3.34):

```
ul.menu {
    display:inline;
    list-style-type:none;
    font-weight:400;
    font-family:"Trebuchet MS", "Tahoma";
}

ul.menu a:link, ul.menu a:visited, ul.menu a:active {
    float:center;
    padding:5px 12px;
    display:inline-table;
    background:#0000b2;
    border:#0000b2 2px solid;
    color:white;
    text-decoration:none;
}

ul.menu a:hover {
    float:center;
    padding:5px 12px;
    display:inline-table;
    background:#013ADF;
    border:#0000ff 2px solid;
    color:white;
    text-decoration:none;
}

.styled-select select {
    background: transparent;
    width: 200px;
    padding: 5px;
    font-size: 16px;
    border: 1px solid #ccc;
    height: 34px;
}
```

Figura 3.34 – Estilo CSS

Mediante HTML y PHP hemos realizado toda la página Web. Una vez iniciada la sesión nosotros veremos la siguiente pantalla (Figura 3.35):

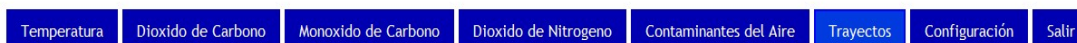


Figura 3.35 – Pantalla con sesión

Antes de nada, deberemos de elegir el sensor a monitorizar. Para ello iremos a la pestaña configuración en donde seleccionaremos el sensor. Dentro de dicha pestaña, podremos también ver, crear y borrar alarmas, crear y borrar sensores y ver el tiempo que hace que obtuvimos un dato de las diferentes medidas de dicho sensor (ver Figura 3.36).



Figura 3.36 – Pantalla de configuración

Una vez elegido el sensor podremos ver, dentro de cada pestaña de Temperatura, Dióxido de Carbono, Monóxido de Carbono y Contaminantes del Aire, los datos de hoy, de los últimos 7 días, del mes y todos los datos de dicho sensor. Junto a los datos se podrá ver una gráfica en donde se mostraran los datos para su mejor visualización de los mismos.

Además, podremos realizar comparaciones entre diferentes sensores.

Aquí se puede apreciar cómo queda la pestaña de Temperatura (Figura 3.37) y un gráfico con las medidas correspondientes al día de hoy (Figura 3.38), a los últimos 7 días (Figura 3.39), al último mes (Figura 3.40) y a la del año actual (Figura 3.41).



Figura 3.37 – Pantalla de pestaña Temperatura

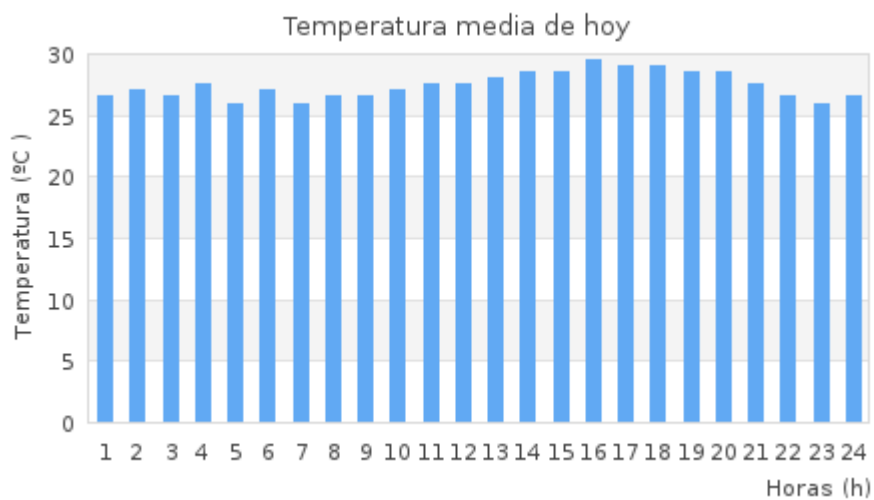


Figura 3.38 – Gráfica con las medidas de hoy

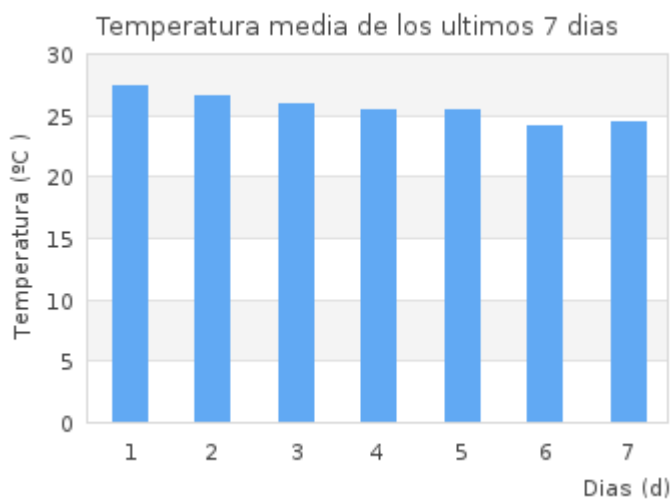


Figura 3.39 – Gráfica con las medidas de los últimos 7 días

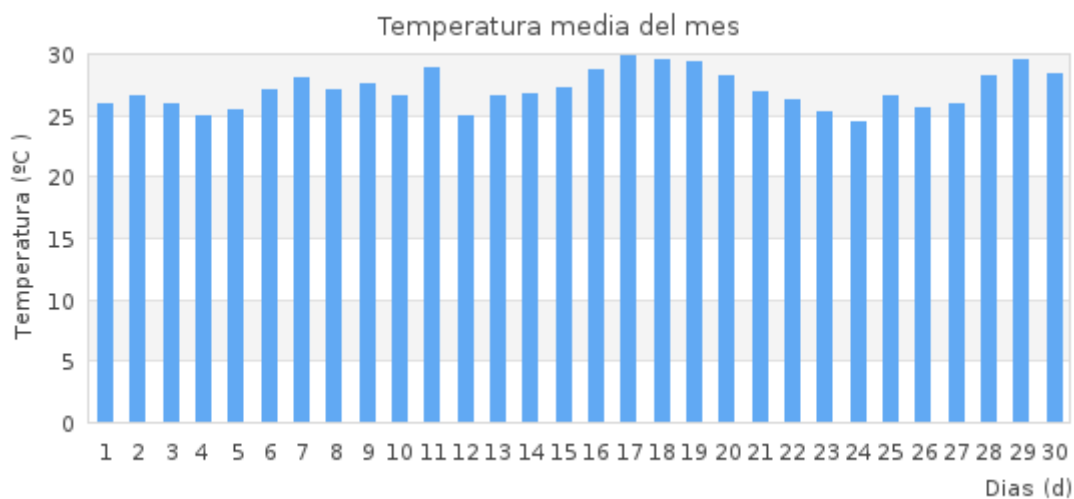


Figura 3.40 – Gráfica con las medidas del mes

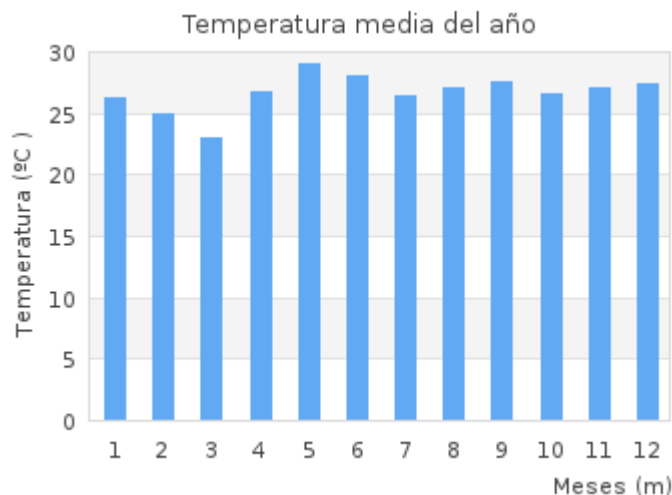


Figura 3.41 – Gráfica con las medidas del año

Además de esto se pueden realizar comparaciones entre varios sensores (ver Figura 3.42).

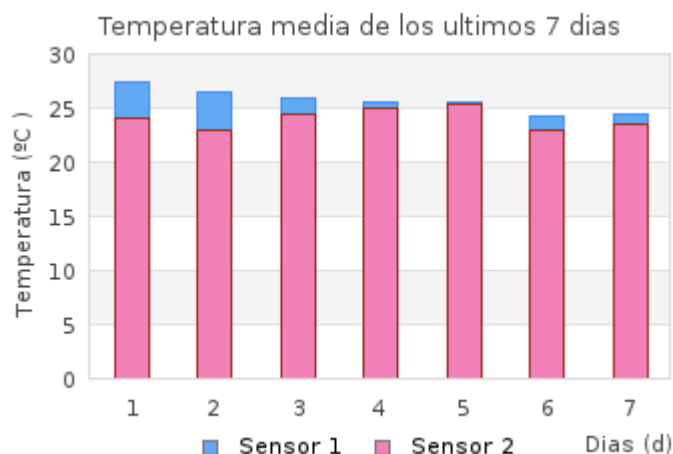


Figura 3.42 – Comparación entre sensor 1 y 2 de la temperatura de los últimos 7 días

Para el dióxido de carbono, monóxido de carbono, dióxido de carbono y contaminantes del aire, la interfaz es la misma.

Además de todo lo expuesto anteriormente, tenemos una pestaña llamada Trayectos. En dicha pestaña podemos crear y borrar trayectos (ver Figura 3.43), asignar sensores a un trayecto (ver Figura 3.44), ver sensores de un trayecto (ver Figura 3.45), ver medidas de un trayecto (ver Figura 3.46), ver medidas de una posición (ver Figura 3.47), comparar trayectos (ver Figura 3.48), comparar sensores de una determinada posición (ver Figura 3.49) y asignar una posición a un sensor (ver Figura 3.50).

Gestión y monitorización de contenedores intermodales mediante tecnología inalámbrica

Temperatura	Dioxido de Carbono	Monoxido de Carbono	Dioxido de Nitrogeno	Contaminantes del Aire	Trayectos	Configuración	Salir
Crear/Borrar Trayecto	Asignar Sensor a Trayecto	Ver Sensores de Trayecto	Medidas en un Trayecto	Medidas en una Posicion	Comparar Trayectos	Comparar Posicion	Asignar Posicion

Trayectos existentes	Elige opciones del trayecto	Crear/Borrar trayecto
"Origen " Pamplona " - L	Origen: <input type="text"/> Destino: <input type="text"/> Hora Inicio: <input type="text"/> Hora Llegada: <input type="text"/> Transporte: <input type="text"/> Responsable: <input type="text"/>	Crear trayecto Borrar trayecto

Figura 3.43 – Pantalla crear y borrar trayectos

Temperatura	Dioxido de Carbono	Monoxido de Carbono	Dioxido de Nitrogeno	Contaminantes del Aire	Trayectos	Configuración	Salir
Crear/Borrar Trayecto	Asignar Sensor a Trayecto	Ver Sensores de Trayecto	Medidas en un Trayecto	Medidas en una Posicion	Comparar Trayectos	Comparar Posicion	Asignar Posicion

"ID" 1 - "Trayecto" 1 - "F"

"ID " 1 "Origen " Pamplona " - Destino" Madrid

Asignar Trayecto

Figura 3.44 – Pantalla asignar sensor a un trayecto

Temperatura	Dioxido de Carbono	Monoxido de Carbono	Dioxido de Nitrogeno	Contaminantes del Aire	Trayectos	Configuración	Salir
Crear/Borrar Trayecto	Asignar Sensor a Trayecto	Ver Sensores de Trayecto	Medidas en un Trayecto	Medidas en una Posicion	Comparar Trayectos	Comparar Posicion	Asignar Posicion

"ID " 1 "Origen " Pamplona " - Destino" Madrid

Ver Sensores

"ID " 1

Figura 3.45 – Pantalla ver sensores de un trayecto



Figura 3.46 – Pantalla ver medidas de un trayecto



Figura 3.47 – Pantalla ver medidas de una posición



Figura 3.48 – Pantalla comparar trayectos



Figura 3.49 – Pantalla comparar posición



Figura 3.50 – Pantalla asignar posición

3.3 Pruebas

Las pruebas que se han llevado a cabo han sido realizadas en un laboratorio de la Universidad Pública de Navarra.

La primera prueba que se ha realizado ha sido la siguiente. Se han situado los sensores en diferentes puntos del laboratorio, simulando así las diferentes posiciones en donde se encuentran los sensores en un determinado trayecto.

Dichos sensores, se han dejado durante varios días realizando medidas, pudiendo así ver en la página Web las medidas realizadas.

Mediante la aplicación Web, hemos podido ver que sensores estaban en el trayecto, ver las medidas de un trayecto y de una posición determinada y comparar las medidas entre las diferentes posiciones.

La segunda prueba se realizó en dos laboratorios contiguos, simulando así dos trayectos diferentes.

Se dejaron durante una semana realizando medidas. Mientras se realizaban las medidas y también después de pasar la semana, se podía realizar diversas acciones en la aplicación Web.

Se podía comparar las medidas entre los diferentes trayectos, ver los sensores de uno de los dos trayectos, ver las medidas de uno de los dos trayectos, ver las medidas realizadas por las diferentes posiciones en cualquiera de los trayectos y comparar las medidas de las posiciones de los trayectos.

Además de esto, podemos ver en las pestañas de Temperatura, Monóxido de carbono, Dióxido de carbono y Contaminantes del aire, las diferentes medidas tomadas el día de hoy, las de los últimos 7 días, las del mes actual y el total de las medidas.

También, podemos realizar comparaciones entre diferentes sensores teniendo en cuenta bien la temperatura de hoy, la de los últimos 7 días, la del mes o la del año. Esta comparación se muestra en una gráfica de barras (ver Figura 3.51).

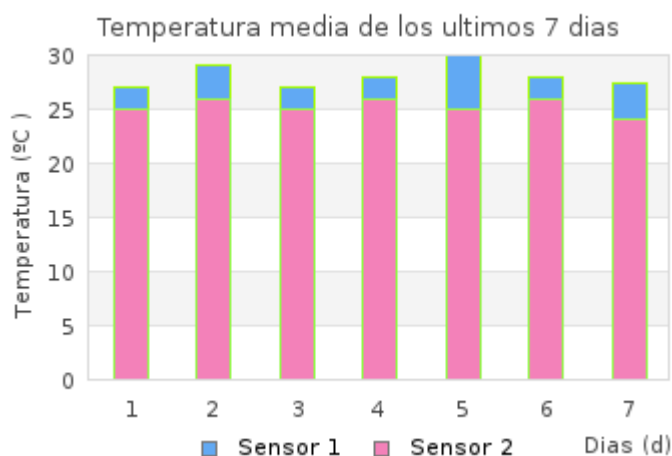


Figura 3.51 – Comparación de los últimos 7 días entre dos sensores.

4. CONCLUSIONES Y LÍNEAS FUTURAS

La conclusión principal que se puede obtener al finalizar el proyecto es que se ha realizado una planificación acorde a los planes iniciales, que se tenían al iniciar el proyecto, y que las replanificaciones que se han tenido que realizar no han supuesto grandes trastornos en la evolución del proyecto, gracias a la importante labor de comunicación entre el tutor y yo.

Por ello, se debe destacar la importancia de realizar una correcta gestión en todas las fases del proyecto, puesto que permite que los problemas que surjan a lo largo del proyecto, habituales en proyectos con un alto grado de complejidad, sean solucionados con el menor impacto posible y que los planes de mitigación sean los más oportunos. Realizando este control durante todo el proyecto se aprende de los problemas encontrados, sirviendo decisiones tomadas en el pasado para problemas que se presenten en futuras ocasiones.

Por otro lado, el realizar reuniones de manera periódica, además de favorecer la comunicación entre el tutor y el alumno, asimismo produce una detección rápida de los problemas.

Además, podemos decir que tras haber estudiado a fondo las redes de sensores inalámbricos observamos que son una mejora bastante evidente frente a las tradicionales redes wireless, debido a varios factores como son: la durabilidad del tiempo de vida de las baterías, lo cual permite una mayor portabilidad de los nodos sensores y que estos puedan registrar mayores sucesos al poder permanecer más tiempo en ciertos lugares, los protocolos de encaminamiento de las redes de sensores permiten que aparte de ganar en cuanto a durabilidad también se gane en eficiencia a la hora de evitar colisiones entre paquetes, lo que asegura también un menor número de tráfico innecesario en la red, y en cuanto al precio cada vez más se está equiparando el desembolso para la construcción de una red wireless frente a una red de sensores inalámbricos, por lo que es una competencia directa.

Este tipo de redes actualmente, está llevando una revolución tecnológica similar a la que tuvo la aparición de Internet, pues las aplicaciones parecen ser infinitas, además se habla de redes de vigilancia global del planeta capaces de registrar seguimiento de personas y mercancías concretas, monitorizar tráfico, y varias iniciativas y proyectos de investigación han despertado gran interés para ser aplicados en la práctica. Claro todo esto parecen grandes ventajas en un futuro, pero la mala utilización podría acabar con la privacidad de mucha gente porque se están llegando a miniaturizar tanto los nodos que pueden ser tan pequeños como para estar en cualquier lugar sin ser detectados y funcionando sin ningún tipo de problemas, por eso esperemos que el desarrollo sobre estas redes sirva solo para la mejora de nuestras vidas.

Cuando se habla de mantener las mercancías en las mejores condiciones, todas las horas empleadas en desarrollar técnicas y/o herramientas son pocas, o si no pocas, al menos, no son muchas.

Como conclusión final del presente proyecto, y dado el estado del arte al que se ha podido acceder, este innovador sistema podría incorporarse al equipamiento

tecnológico de las empresas que se dedican a esta labor. Mediante esta incorporación, se conseguiría una gran mejora en la protección de la mercancía.

A continuación se pasa a describir posibles alternativas que se podrían llevar a cabo en un futuro cercano, en un presente, para conseguir una poderosa herramienta de monitorización de mercancía.

El sistema futuro podría tener las siguientes líneas de evolución:

- Posible ampliación de la red con más módulos XBee.
- Incorporar más sensores aparte del monóxido de carbono, dióxido de nitrógeno... para tener más medidas y poder así realizar una mejor monitorización o detectar con más facilidad un problema.
- Mandar una alerta en el momento en el que se produjese una situación alarmante.
- Realizar una aplicación para dispositivos móvil para que los responsables de la mercancía puedan ver los datos de los sensores.
- Permitir la posibilidad de que el usuario interactúe más con la aplicación gráfica.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] G. Huang. —Casting the Wireless Sensor Netl. July/August 2003. MIT Technology Review, pp. 51-56.
- [2] Vellidis, G., Tucker, M., Perry, C., Kvien, C., Bednarz, C., 2008. —A real-time wireless smart sensor array for scheduling irrigationl. Comput. Electron. Agric. 61, 44–50.
- [3] Ryan Dickey, Timothy Franklin, Jake Harmon 2004. "Nuclear, Biological and Chemical (NBC) Communications network". Systems and Information Engineering Design Symposium
- [4] CodeBlue, University of Harvard. Harvard Sensor Network Lab. <http://fiji.eecs.harvard.edu/CodeBlue>
- [5] IEEE, Institute of Electrical and Electronics Engineers, 2006. —IEEE Standard for Information technology-Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) l. IEEE Standard 802.15.4-2006, Institute of Electrical and Electronics Engineers, New York.
- [6] Baronti, P., Pillai, P., Chook, V.W., Chessa, S., Gotta, A., Fu, Y.F., 2007. Wireless Sensor Networks: a survey on the state of the art and the 802.15.4 and ZigBee standardsl. Comput. Commun. 30, 1655–1695.
- [7] Drew Gislason. —ZigBee Wireless Networkingl. Newnes. 2008. ISBN-10: 9780750685979.
- [8] ZigBee Alliance, especificación de la revisión ZigBee 2007. Disponible on-line en: <http://www.zigbee.org>.
- [9] King, B.A., Wall, R.W., Wall, L.R., 2000. —Supervisory Control and Data Acquisition System for Closed-Loop Center Pivot Irrigationl. ASAE Technical Paper No. 00-2020. The American Society of Agriculture Engineers, St. Joseph, Michigan, USA.
- [10] HART. The Logical Wireless Solution: <http://www.hartcomm2.org/hart_protocol/wireless_hart/hart_the_logical_solution.html>.
- [11] ISA100.11a: <<http://www.isa.org/MSTemplate.cfm?MicrositeID=1134&CommitteeID=689>>.
- [12] 6LoWPAN, <<http://6lowpan.net/>>.
- [13] G. Mulligan, L.W. Group. —The 6LoWPAN architecturel. Proceedings of the EmNets, Cork, Ireland, 2007.
- [14] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler. —Transmission of IPv6 packets over IEEE 802.15.4 networks l. RFC 4944 (2007).
- [15] IEEE Standard 802.15.3. —Wireless medium access control (MAC) and physical layer (PHY) specifications for high rate wireless person area networks (WPANs)l. September 2003.
- [16] Wibree, <<http://www.wibree.com/>>.
- [17] Crossbow : < www.crossbow.com >
- [18] The intel mote, disponible on-line en: http://www.ieee.or.com/Archive/intel_mote.htm
- [19] Zolertia : <<http://www.zolertia.com/ti>>
- [20] Libelium Company, disponible on-line en: <<http://www.libelium.com/>>
- [21] Texas Instruments , disponible on-line en: <www.ti.com>
- [22] Digi Wireless Solutions, disponible on-line en: <<http://www.digi.com/>>

- [23] Tinyos , disponible online en: <<http://www.tinyos.net/>>
- [24] Microsoft .NET Micro Framework, disponible en:< <http://www.microsoft.com/en-us/netmf/default.aspx>>
- [25] MantisOS, disponible on-line en: <<http://mantisos.org/index/tiki-index.php.html>>
- [26] eCos , disponible on-line en: <<http://ecos.sourceware.org/>>
- [27] Mirrium, disponible on-line en: <<http://mirrium.com/page/home>>
- [28] Contiki Operative System, disponible on-line en: <www.sics.se/contiki/>
- [29] SimpleOS, disponible on-line en: <<http://sos.enix.org/fr/PagePrincipale>>
- [30] NanoRK, disponible on-line en: <www.nanork.org>
- [31] Jennic, disponible on-line en:<<http://www.jennic.com/>>