Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros Industriales

Departamento de Automática, Ingeniería Electrónica e Informática Industrial

Máster Universitario en Electrónica Industrial

PLATAFORMA DE INTEGRACIÓN HARDWARE-SOFTWARE PARA TESTBED DE REDES DE SENSORES INALÁMBRICAS

Autor: Gabriel Noe Mujica Rojas

Director: Jorge Portilla Berrueco

Codirector: Víctor Roselló Gómez-Lobo

Marzo 2012

Trabajo Fin de Máster

Este proyecto está dedicado a Delida y Numan,
por su apoyo incondicional, comprensión y por creer siempre en mí.

Índice

Capítulo 1

INTRODUCCIÓN Y OBJETIVOS DEL PRESENTE PROYECTO

I. Introducción.

El campo de las redes de sensores inalámbricas ha cobrado gran importancia en esta última década ya que se han abierto diversas líneas de investigación con el fin de poder llevar a la práctica los conceptos y definiciones que envuelven el potencial de esta tecnología, y que está llamada a ser el futuro en la adquisición de datos de cualquier entorno físico de aplicación, mediante una herramienta basada en la autogestión y desatención durante largos periodos de tiempo, capacidad de tomar muestras cuando sea necesario a través de nodos sensores que se caractericen por el ahorro de energía y que puedan ser capaces de trabajar de forma autónoma durante meses, y que el carácter inalámbrico de la red a desplegar facilite las tareas de instalación y mantenimiento.

Ello requiere que las condiciones para que una red de sensores inalámbrica sea la forma más viable de monitorizar un determinado entorno se base en ciertos requisitos de diseño, como lo es la baja tasa de transferencia de datos por parte de los nodos (estos deben ser capaces de transmitir la información recolectada desde los sensores y luego permanecer dormidos hasta una nueva adquisición), *hardware* enfocado al bajo consumo de energía con el fin de evitar cambios en la fuente de energía (baterías) durante largos periodos de tiempo, adaptabilidad al entorno de aplicación, flexibilidad y escalabilidad de la red si la aplicación hace necesario la inclusión de nuevos nodos o la modificación de los ya existentes, sin que ello suponga mayores dificultades en su desarrollo e implementación.

Desde la aportación de la universidad de California, Berkeley, llevada a cabo en 2003 por Jason Hill [Hill'03] la cual representa una base en la filosofía y diseño de nodos inalámbricos y que sería referencia para los nuevos desarrollos en esta tecnología, han sido muchos los centros de investigación y empresas que se han adentrado en este nuevo campo de investigación y en sus respectivas sub-ramas. Con la posterior creación por parte de los diseñadores de la universidad de Berkeley del nodo TelosB [Polastre'05], que pasaría a ser uno de los más extendidos en la comunidad de investigación y en el cual se exploran aspectos como el bajo consumo y la integración de las comunicaciones y el procesamiento en una tarjeta de desarrollo, se han definido ramas de estudio como el *hardware* de los dispositivos, el *software* embebido para el control de los elementos que componen el *hardware*, algoritmos para la gestión del nodo inalámbrico y los protocolos de comunicación que permitan establecer la conexión entre los dispositivos de la red.

En este último caso, y con el fin de crear un protocolo que cumpla con el concepto y los requisitos de las redes de sensores inalámbricas, se han diseñado estándares de comunicación como el IEEE 802.15.4 [802.15.4] enfocado a la baja tasa de

transferencia de datos de redes inalámbricas de área personal (*LR-WPAN*, *Low-rate Wireless Personal Area Network*), y que proporciona las capas básicas para dar servicio a este tipo de redes a través de la definición del nivel físico y el control de acceso al medio (capa física y capa MAC del protocolo).

En función de este estándar, ha surgido una alianza entre diversas organizaciones y empresas alrededor del mundo con el fin de definir una especificación de comunicación de alto nivel para aplicaciones que requieran bajo consumo de energía, baja tasa de transferencia de datos de forma segura, y que sean fácilmente integrables. De ello nace la especificación ZigBee [ZigBee 1], la cual se nutre del IEEE 802.15.4 para los niveles bajos de la pila de comunicación y define los niveles de red y aplicación. Hoy en día existen gran variedad de fabricantes que proporcionan módulos de radio con este protocolo integrado, con el fin de que terceros puedan implementar en sus desarrollos la tecnología ZigBee.

Con un protocolo de comunicaciones ya extendido y dispositivos *hardware* disponibles en el mercado, son muchos los desarrolladores que optan por adquirir nodos ya prediseñados y enfocan sus líneas de investigación en la gestión y control de la red inalámbrica, o bien en algoritmos específicamente diseñados para determinadas aplicaciones, siempre en función de la tecnología adquirida y las limitantes que ello supone. Por otro lado, existen desarrolladores que parten de la filosofía de las redes de sensores inalámbricas para crear su propia tecnología, con el fin de poder contar con un mayor margen de control a bajo nivel y adaptar sus diseños más fácilmente al entorno de aplicación en el que se desea desplegar la red se sensores.

El Centro de Electrónica industrial de la Universidad Politécnica de Madrid se incluye dentro de este último grupo, donde se ha diseñado una completa plataforma *hardware* para redes de sensores inalámbricas [Portilla'07], con el fin de investigar las potencialidades, dificultades y retos que supone el realizar un despliegue de nodos inalámbricos en cumplimiento de características primordiales como autonomía, flexibilidad y escalabilidad de la red, además de la autogestión de los dispositivos que forman parte de ella.

Esta plataforma es de carácter modular, lo que quiere decir que está basada en una arquitectura por capas, en la que cada una se encarga de realizar una función dentro del nodo inalámbrico. Desde su creación, ha sido bautizada como Plataforma Cookies, y sus detalles se presentan a continuación.

I.1 Plataforma Cookies.

La plataforma *hardware* modular Cookies está basada principalmente en cuatro

capas, las cuales están interconectadas mediante un bus común que permite intercambiar señales entre cada una de ellas, tales como líneas de alimentación, datos de los sensores o control de comunicaciones, entre otros aspectos. Este bus de comunicación está dividido en dos partes, un conector a un lado de las capas, que es denominado bus digital puesto que contiene las señales de sensores digitales, interfaces de comunicación, entradas y salidas de propósito general, entre otras; y un conector en el lado contrario denominado bus analógico, que se encarga de transferir las señales de sensores analógicos hacia convertidores analógico-digitales, señales de control de actuadores, entradas y salidas de propósito general, UART, entre otras.

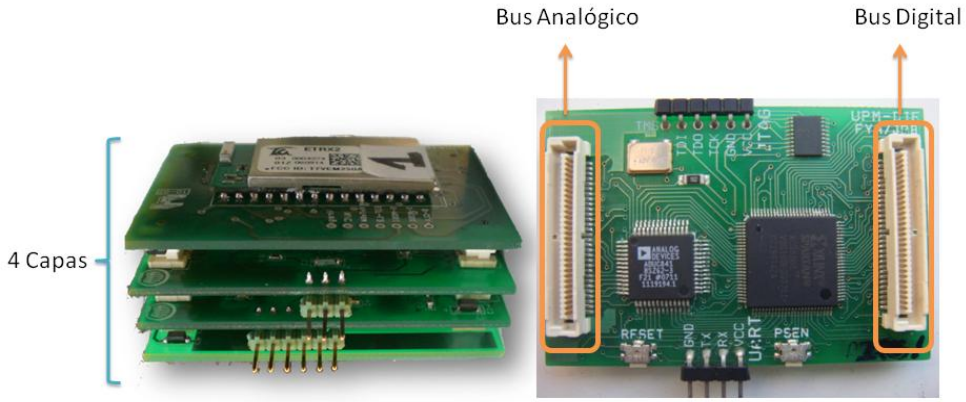


Figura 1.1: Estructura general de la plataforma Cookies.

Las cuatro capas fundamentales de la plataforma han sido diseñadas atendiendo a la clasificación de las cuatro funcionalidades que como mínimo debe poseer el nodo inalámbrico, que son alimentación, procesamiento, sensado y comunicaciones.

La flexibilidad y versatilidad que aporta el bus de interconexión permite agregar más capas adicionales a las 4 fundamentales, como por ejemplo una capa de memoria, una de depuración o una de procesamiento, en caso de ser necesario. Del mismo modo, este diseño modular permite reutilizar la plataforma *hardware* en diferentes entornos de aplicación, puesto que, al ser las capas intercambiables, en caso de necesitar realizar capturas de diferentes tipos de sensores sólo sería necesario cambiar la capa de sensado. También es posible, por ejemplo, utilizar la misma plataforma para dos tipos de protocolos de comunicación, con sólo realizar el cambio en la placa de comunicaciones del nodo inalámbrico. Del mismo modo ocurre en el caso de la capa de alimentación, la cual se puede adaptar según la fuente de energía que se utilice, siempre manteniendo la estructura global del nodo.

Todas estas características fueron diseñadas e implementadas con el fin de obtener una plataforma robusta, basada en la fiabilidad y al mismo tiempo en la sencillez del nodo inalámbrico, con el fin de obtener un prototipado rápido, flexibilidad y adaptabilidad del diseño a diversos entornos de aplicación de redes de sensores inalámbricas.

A continuación se detallan cada una de las capas base que forman parte de la plataforma Cookies y la aportación que realizan al conjunto global del nodo inalámbrico.

I.1.1 Capa de procesamiento.

Esta capa representa el cerebro de la plataforma modular, puesto que se encarga de procesar toda la información de las restantes capas, recolectar las medidas físicas del entorno mediante los sensores, transmitirla a través de la comunicación inalámbrica, y tomar las acciones necesarias para el ahorro de energía por parte del nodo.

Siguiendo con la filosofía de flexibilidad y versatilidad de la plataforma, la capa de procesamiento ha sido diseñada para contener dos elementos de procesamiento. En primer lugar un microcontrolador (uC), que funciona como núcleo principal de procesamiento de información y encargado de capturar los datos de los sensores analógicos a través del convertidor analógico-digital incluido en la mayoría de los uCs comerciales, controlar el intercambio de datos con la red inalámbrica mediante la gestión de la conexión UART con el módulo de comunicaciones, y disponer de señales de propósito general para realizar posibles actuaciones sobre periféricos incluidos en el nodo. Por otro lado, se encuentra la implementación de un elemento *hardware* como lo es la FPGA, que se encarga de tareas relacionadas con la adquisición de datos provenientes de sensores digitales, gestión de los protocolos que envuelven el control de los mismos y el intercambio de información con el uC, actuando como un dispositivo de coprocesamiento asociado. Si bien es cierto que la plataforma Cookies y la filosofía de la misma permite la inclusión de ambos elementos, es posible realizar implementaciones sin uno u otro elemento, lo que dependerá del tipo de aplicación y de los requisitos a nivel de procesamiento que esta exija.

La capa base de procesamiento (que será la utilizada en el presente proyecto) está compuesta por un microcontrolador ADuC841, de Analog Devices [ADUC841], que posee una memoria flash de programa de 62 KB, 4KB de memoria flash de datos y 2KB de RAM. Incluye un convertidor analógico-digital de 12 bits, 4 puertos de entrada/salida con señales de propósito general así como pines dedicados para conexiones UART, SPI, I²C y modulación PWM. La FPGA incluida en esta capa es del fabricante Xilinx, específicamente una Spartan 3 XC3S200 [Xilinx], con 200000 puertas equivalentes. Ya que la familia de FPGAs Spartan 3 está basada en memoria RAM, ha sido necesario incluir de forma adicional una memoria flash XCF01S de

Xilinx, con el fin de cargar la configuración en el caso de que ocurra un apagado y encendido del nodo.

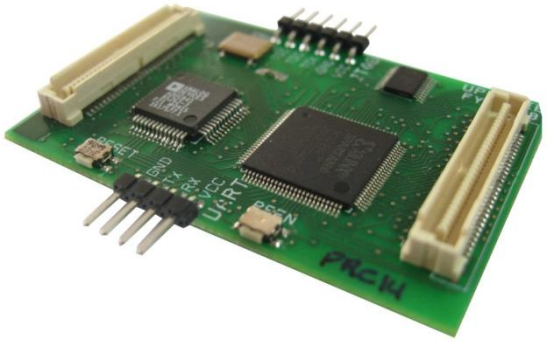


Figura 1.2: Capa de procesamiento de la plataforma Cookies.

I.1.2 Capa de comunicaciones.

La capa de comunicaciones es la encargada de llevar el módulo que implementa el protocolo a utilizar para el intercambio de información de los dispositivos inalámbricos. Como se ha comentado anteriormente, la especificación ZigBee ha sido creada con el fin de reunir los requisitos de las redes de sensores inalámbricas, que son la baja tasa de transferencia de datos, fácil integración y bajo consumo de energético. Es por ello que en la plataforma Cookies se ha decidido incluir un dispositivo que implemente la pila ZigBee y que la interfaz entre la capa de procesamiento y dicho dispositivo sea lo más sencilla e intuitiva posible. Para ello, existen en el mercado módulos de comunicaciones que pueden ser controlados a través de comandos AT vía UART, lo cual facilita las tareas de gestión de las comunicaciones inalámbricas y agiliza el prototipado de aplicaciones basadas en tecnología ZigBee.

El módulo incluido en esta capa es el ETRX2 de la empresa Telegesis, el cual está basado en un chip de Ember (EM250) con un microcontrolador que incluye la pila ZigBee y la electrónica relacionada con la radio [Telegesis 1]. Este módulo permite no sólo realizar la gestión a nivel de red mediante la conexión UART con el uC, sino que también dispone de registros de entrada y salida para configuración del dispositivo, diversas funcionalidades para la generación de eventos y pines de entrada y salida de propósito general para ser utilizados externamente. Posee una gran variedad de parámetros de configuración, 4 modos de consumo de energía con el fin de alargar la vida de las baterías del dispositivo inalámbrico, y posibilidad de incluir una antena externa, aunque por defecto lleva una antena integrada.

Figura 1.3: Capa de comunicaciones de la plataforma Cookies.

I.1.3 Capa de Alimentación.

Esta capa se encarga de generar las tensiones necesarias para la alimentación de todas las capas restantes a partir de la fuente de energía a la que esté conectado el nodo. En este sentido, se generan tensiones de 3.3V, 2.5V y 1.2V, que son valores que permiten cubrir la alimentación de un amplio rango de dispositivos incluidos en la plataforma *hardware*. La versión que se ha utilizado en el presente proyecto permite alimentar el nodo a través de baterías, o bien mediante una conexión USB además de implementar un chip FTD232 de la empresa FTDI que permite convertir las señales USB a UART, lo que posibilita realizar tareas de depuración en el caso de que dicha conexión se realice desde un ordenador.

Figura 1.4: Capa de alimentación de la plataforma Cookies.

Además de ello, la capa de alimentación lleva integrada el circuito para la monitorización del consumo de energía del nodo, a través de una resistencia *shunt*

asociada a la fuente de alimentación de entrada, y cuya señal es transmitida por el bus analógico al convertidor analógico-digital del uC.

I.1.4 Capa de Sensado.

Esta capa es la encargada de contener los sensores que recolectarán las magnitudes físicas del entorno de aplicación para que luego sean decodificados a través de la capa de procesamiento y, si es necesario, enviar los datos por la red inalámbrica. Dependiendo de la interfaz del sensor (ya sea analógica o digital), las señales irán o al bus analógico o al digital, pero dichas señales deben transferirse de forma adecuada a la capa de procesamiento, lo que quiere decir que la capa de sensado debe llevar los circuitos de acondicionamiento necesarios para que los valores puedan ser capturados correctamente por el uC o por la FPGA, según el caso.

Esta capa dota de gran versatilidad a la plataforma, puesto que se puede utilizar un mismo tipo de nodo para diversas aplicaciones, sólo siendo necesario el cambio de esta capa en el caso de requerir tomar medidas de otras magnitudes físicas distintas a las implementadas.

Entre las versiones diseñadas para la plataforma inalámbrica se encuentran una capa con sensores de temperatura y humedad (sensor SHT11 con interfaz digital I²C), y un sensor analógico de nivel de luz (LDR, *Light Dependent Resistor*). Otra de las capas está basada en un acelerómetro en dos ejes, el ADXL213 de *Analog Devices*, con interfaz PWM y un sensor de temperatura con salida codificada en el periodo de la señal. Por otro lado se encuentra una capa basada en una galga extensiométrica para la medida de deformaciones, con su respectivo circuito de acondicionamiento de la señal. Aparte de estas tres capas básicas creadas para realizar pruebas de funcionamiento de la plataforma *hardware*, se han incluido posteriormente otras relacionadas con sensores de PH, CO y CO2.

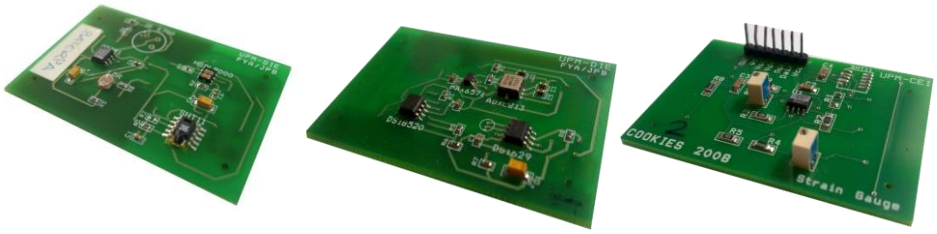


Figura 1.5: Capas de sensores de la plataforma Cookies.

II. Objetivos y alcance del proyecto.

El hecho de contar con una plataforma propia, como se ha comentado anteriormente, permite tener un mayor margen de diseño y modificación a bajo nivel para cumplir con los requisitos relacionados con un despliegue de nodos inalámbricos. Sin embargo, son muchos los aspectos a tener en cuenta para que una red de sensores cumpla con los objetivos para la cual ha sido concebida, más allá de la importancia intrínseca del *hardware* con el que estén compuestos los nodos.

Dichos aspectos van desde la correcta gestión de los recursos del nodo, diseño de los controladores de los periféricos que poseen, la adecuada decodificación de los datos recibidos desde los sensores, la integración entre los núcleos de procesamiento incluidos en el dispositivo y la comunicación entre ellos, entre otros factores críticos para el funcionamiento del nodo. Esto desde el punto de vista local (gestión propia del nodo), pero también es necesario controlar la interacción hacia el exterior, es decir, como se relaciona el nodo con los dispositivos restantes de la red inalámbrica, no sólo en términos de nivel de red (que se realiza mediante el protocolo utilizado), sino las reglas y directivas necesarias para el entendimiento entre ellos a nivel de aplicación.

Por otro lado está también la forma en que los nodos se comunican con el nodo coordinador de la misma, de modo que es necesario definir el comportamiento de este último ya que hasta el momento se han comentado las características de los nodos sensores remotos. Es evidente que el coordinador de la red inalámbrica cumple una función primordial puesto que, además de ser el responsable de administrar todos los aspectos relacionados con la gestión y el mantenimiento de la conexión entre los dispositivos, hace de puente entre la estación central en el que se encuentre conectado y el despliegue realizado, con el fin de que la red pueda ser monitorizada en tiempo real por un operador y, si es necesario, actuar sobre ella para realizar los cambios oportunamente.

Estos aspectos relevantes crean la necesidad de contar con una herramienta que permita gestionar y controlar los nodos inalámbricos desde todos los puntos de vista de operación, es decir, desde el más bajo nivel de abstracción (sistema embebido en los nodos) hasta lograr una total abstracción por parte el usuario final de la aplicación y, por tanto, facilitar las tareas de despliegue y mantenimiento de una red de sensores inalámbricas independientemente del entorno de aplicación.

El presente trabajo de investigación se centra en cubrir estas necesidades, por lo que su principal objetivo es la creación de una plataforma de integración *hardware-software* que permita explotar todas las potencialidades de la arquitectura Cookies a través de una herramienta que facilite el despliegue, control y mantenimiento de una red de sensores inalámbrica, con el fin último de contar con un sistema total para el prototipado rápido de aplicaciones, soporte de pruebas de nuevos desarrollos y la

posibilidad de implementación de dicha plataforma en cualquier entorno real, siendo sólo necesario realizar pequeños ajustes desde el más alto nivel de abstracción para que el sistema sea capaz de adaptarse por sí solo.

Para cumplir tales propósitos y lograr una completa integración del sistema conjunto, ha sido necesario fijar principalmente tres líneas de trabajo que se enmarcan dentro de los objetivos específicos del presente proyecto, las cuales se detallan a continuación:

Bibliotecas Software modulares: Basada en la filosofía de modularidad y flexibilidad de la plataforma *hardware*, se hace imprescindible primeramente contar con una plataforma *software* para el control de todos y cada uno de los elementos que componen al nodo Cookie, a partir de bloques funcionales que permitan gestionar desde el núcleo de procesamiento principal todas las características de la plataforma. Esto permitirá asegurar el control de los recursos *hardware* y facilitar la utilización de la plataforma desde un nivel más alto de abstracción, sólo con la configuración de parámetros estandarizados para el funcionamiento de la misma.

Perfil de aplicación Cookies: Después de contar con bloques *software* que permitan controlar las características de bajo nivel del nodo inalámbrico, es necesario crear una herramienta para la estandarización de la forma en la que se comunican los dispositivos a nivel de aplicación, con el fin de gestionar las características y atributos de los nodos sensores de forma remota y facilitar el entendimiento entre ellos. Para ello, es necesario fijar ciertas directivas y reglas que permitan homogeneizar la gestión de tareas asociadas a los nodos Cookies, a través del diseño de un perfil de aplicación.

Testbed para redes de sensores: Como resultado de las dos líneas anteriores de trabajo, la idea es contar con un instrumento que permita realizar pruebas reales haciendo uso de la plataforma de integración HW-SW, a partir de la gestión de todas las características y potencialidades que ofrece el perfil de aplicación creado y así facilitar el desarrollo de prototipos para aplicaciones basadas en redes de sensores inalámbricas, de forma rápida y eficiente. En este sentido, la idea es contar con un banco de pruebas basado en un despliegue de nodos Cookies que pueda ser controlado desde un ordenador central a través de una interfaz de usuario, desde el cual se lleva a cabo la monitorización y actuación sobre la red inalámbrica.

Con el fin de lograr todos los objetivos planteados, ha sido necesario realizar un exhaustivo estudio de la plataforma *hardware* descrita anteriormente con el fin de conocer la forma en la que interactúan cada uno de los elementos incluidos en los nodos, así como la arquitectura y filosofía de los mismos, para poder llevar a cabo la integración con el *software* y, como se verá más adelante, realizar ajustes en el *hardware* para poder implementar correctamente las funcionalidades diseñadas.

Por otro lado, ha sido necesario analizar las características de la especificación

ZigBee y, sobre todo, las propiedades que posee el módulo de comunicaciones que incluye la plataforma *hardware*, el ETRX2, con el fin de poder realizar una configuración y gestión adecuada de los nodos a través de la red inalámbrica, aprovechando las posibilidades y recursos que ofrece dicho módulo.

III. Organización del documento.

Tal y como se presentará en el capítulo 2, se han estudiado algunos de los desarrollos encontrados en el estado del arte para el control de dispositivos en redes de sensores inalámbricas, así como los *testbeds* más relevantes que se han creado en la comunidad científica, sus características y la relación que tienen con los objetivos del presente proyecto.

Por otro lado, en el capítulo 3 se describirán las bibliotecas *software* creadas para el completo control del nodo Cookie, posteriormente en el capítulo 4 se detallará el perfil de aplicación diseñado para explotar las funcionalidades de la plataforma inalámbrica, así como el *testbed* implementado para el soporte de pruebas y desarrollos dentro del marco de las redes de sensores inalámbricas.

Por último, en el capítulo 5 se expondrán los resultados obtenidos a través de diversos entornos prácticos de aplicación y la plataforma final implementada, y en el capítulo 6 se comentarán las conclusiones y líneas futuras de este proyecto de investigación.

Capítulo 2

ESTADO DEL ARTE

I. Introducción.

En este capítulo se presentarán las principales aportaciones dentro del campo de las redes de sensores inalámbricas que se pueden encontrar en la comunidad científica, desde el punto de vista de los sistemas de control del nodo inalámbrico, las herramientas de gestión de las redes de sensores y los bancos de pruebas enfocados al desarrollo de aplicaciones basadas en esta tecnología. Desde la contribución realizada por la Universidad de California a principios de la década pasada, así como el planteamiento de los retos y desafíos que supone el control de dispositivos inalámbricos [Blumenthal'03], han sido muchos los esfuerzos por parte de centros de investigación y organizaciones enfocados al diseño de sistemas para redes de sensores inalámbricas, tanto desde el punto de vista *hardware* de los dispositivos, hasta el control de forma remota de los mismos, todo ello con el fin de disponer de una herramienta eficiente, sencilla y fiable que permita monitorizar diferentes entornos de aplicación.

En función de los objetivos y el alcance del presente proyecto de investigación, se ha enfocado la presentación del estado de la técnica en tres grandes aéreas de estudio. Una primera en la que se comentan algunos trabajos por parte de investigadores en los que presentan tanto el desarrollo de una plataforma *hardware* propia, así como también las herramientas enfocadas en el control de la misma, con el fin de disponer de un *testbed* para redes de sensores inalámbricas enfocados al desarrollo de aplicaciones específicas. Por otro lado se comentan algunos de los sistemas operativos más importantes creados para redes de sensores inalámbricas hasta el momento y que poseen gran popularidad en la comunidad científica, así como también se expondrán algunos desarrollos encaminados al control de nodos inalámbricos comerciales y muy extendidos, entre ellos los nodos TelosB. Finalmente se presentarán los *testbeds* que están enfocados al uso de toda la comunidad internacional para realizar experimentos reales sobre los mismos, basados en despliegues de nodos inalámbricos a gran escala.

II. Plataformas Hardware-Software especificas.

Dentro de los desarrollos que se encuentran en el estado del arte en redes de sensores se pueden distinguir aquellos en los que los investigadores parten de dispositivos comerciales y enfocan sus esfuerzos en las herramientas de gestión de los mismos y, por otro lado, aquellos que realizan un diseño propio de *hardware* y posteriormente implementan los controladores necesarios con el fin de adaptar la plataforma a requisitos específicos. En este apartado se presentan algunos de los desarrollos relacionados con este último caso, sus principales características y aportaciones.

II.1 PAVENET.

En este proyecto, desarrollado en la Universidad de Tokio [Saruwatari'05], se plantea la creación de una completa plataforma para redes de sensores a partir del diseño de un nodo inalámbrico basado en Dual-CPU, denominado U³, como alternativa a los dispositivos disponibles ya que proponen las dificultades de utilizar nodos basados en un solo núcleo de procesamiento para el desarrollo de aplicaciones que necesiten grandes recursos de procesamiento disponibles.

La filosofía del nodo diseñado posee ciertas similitudes con la plataforma propuesta en el presente proyecto, ya que se basa en capas modulares que definen cada funcionalidad dentro del nodo inalámbrico. Dicho nodo se divide en cuatro tarjetas de desarrollo, una denominada *System Board* la cual posee uno de los núcleos de procesamiento basado en un microcontrolador PIC18F452. Por otro lado se encuentra *Communication Board* que implementa un transceptor RF TR3001 para las comunicaciones inalámbricas, y un microcontrolador similar al de la capa anterior. Así mismo, posee una tarjeta destinada a la alimentación del nodo (*Power Supply Board*), a partir de tres baterías AAA, y una tarjeta que contiene los sensores y actuadores según el tipo de aplicación (*Device Board*). La plataforma posee también una tarjeta adicional para conectar el nodo vía USB a otro dispositivo de control y, como opción alternativa, posee un IrDA para comunicaciones infrarrojas con PCs.

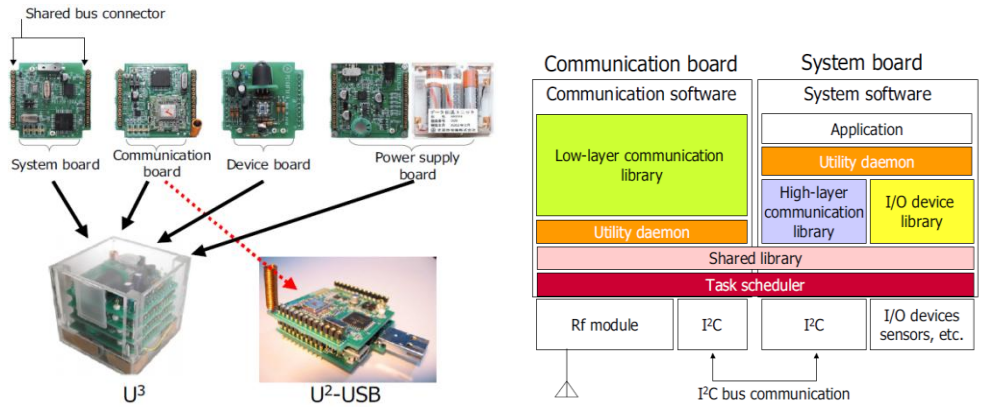


Figura 2.1: nodo U³ y software U² de la plataforma PAVENET.

El nodo implementa una plataforma de control denominada U² SDK a través de los dos microcontroladores, tal y como se muestra en la figura 2.1. La gestión del nodo se realiza mediante bibliotecas que permiten abstraer las comunicaciones y el control de los sensores mediante la definición de funciones de entrada y salida de datos.

La plataforma ha sido probada mediante aplicaciones relacionadas con el cálculo de detección de líneas de movimientos y dirección de personas en supermercados, así

como experimentos relacionados con la medición de la efectividad de protocolos de comunicación de redes inalámbricas. Del mismo modo, ha sido concebida como un *testbed* para dar soporte a prototipos y desarrollos dentro del grupo de investigación de la Universidad de Tokio.

II.2 REALnet.

Este proyecto, desarrollado por la Universidad de Cataluña, se basa en el despliegue de una red de sensores inalámbrica en el campus de la universidad con el fin de monitorizar parámetros medioambientales, y contar con una plataforma genérica que sirva para futuros desarrollos dentro del grupo de investigación. El trabajo presentado en [Albesa'07] está basado en tres nodos, un sensor de temperatura y nivel del agua en un estanque dentro del campus, un nodo *router* y el coordinador de la red. La principal característica es que utilizan módulos ETRX2 de Telegesis para las comunicaciones inalámbricas, y al ser una aplicación sencilla desde el punto de vista de procesamiento, el nodo sensor sólo utiliza un ATtiny2313 de bajo consumo del fabricante Atmel, mientras que el *router* está basado únicamente en el módulo ZigBee.

II.3 Real-time Sensor Monitoring.

En este trabajo de investigación, llevado a cabo en el *Electronics and Telecommunications Research Institute* de Corea, se propone un *testbed* para redes de sensores inalámbricas enfocado en la monitorización de medidas de sensores de forma remota a través de un servidor mediante conexión TCP/IP o UDP. La base del proyecto presentado en [Hong'09] se centra en la transmisión de alta velocidad de las medidas de los sensores, mediante el uso de comunicación SPI entre el microcontrolador que adquiere los valores y el procesador que implementa el protocolo de comunicación remota, tal y como se ilustra en la figura 2.2.

La capacidad de acceso a la información mediante conexión TCP/IP permite realizar la monitorización de los nodos desde lugares distintos al sitio de despliegue. Para ello, se plantea una herramienta para la gestión de los dispositivos desde un *Testbed Management Server*.

Aunque el objetivo es la monitorización de sensores a alta velocidad, la plataforma también está enfocada para soportar pruebas de sistemas operativos y experimentos relacionados con protocolos de comunicación, rutado de datos y localización, ya que el *testbed* está pensado para despliegues tanto en entornos *indoors* como *outdoors*.

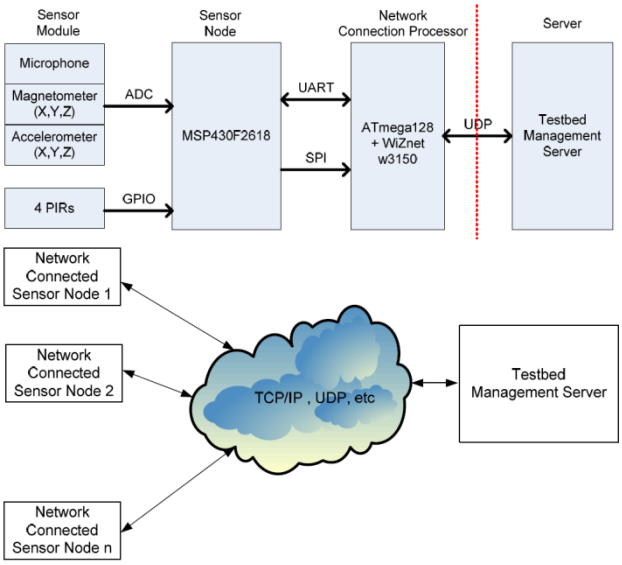


Figura 2.2: Arquitectura del testbed para la monitorización remota.

II.4 Heterogeneous Sensor Networks Testbed.

Este proyecto está enfocado en proporcionar una herramienta heterogénea para pruebas de redes de sensores inalámbricas, a través de la implementación de nodos de alta velocidad basados en WLAN, y nodos de media velocidad basados en IEEE 802.15.4. La idea se centra en realizar experimentos bajo la convivencia de ambos protocolos y el intercambio de información entre ambas redes. En tal sentido, en [Liu'10] se plantea una arquitectura en la que se parte de una estación base conectada a un punto de acceso WLAN con el fin de establecer la conexión con los dispositivos de alta velocidad, para que luego estos establezcan la conexión con los dispositivos de media velocidad, tal y como se ilustra en la figura 2.3.

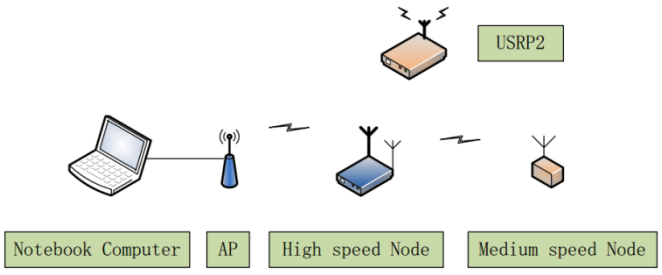


Figura 2.3: Testbed heterogéneo para redes de sensores inalámbricas.

El principal elemento de esta arquitectura es el nodo de alta velocidad (EZ270-2), ya

que implementa tanto WLAN como también IEEE 802.15.4, con el fin de hacer el puente con los nodos sensores finales. La arquitectura de dicho elemento se muestra a continuación:

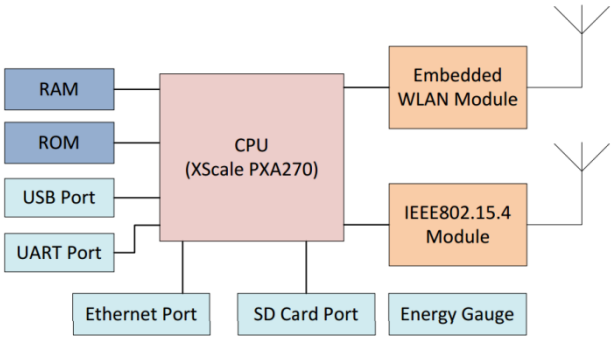


Figura 2.4: Arquitectura de EZ270-2 con WLAN y IEEE 802.15.4.

Además de ello, incluyen un dispositivo para generar interferencias y ruido (USRP2), con el fin de comprobar el funcionamiento de las redes en dichas condiciones. La plataforma está pensada para soportar gran cantidad de nodos y realizar experimentos relacionados con topologías, ruta de datos, tasas de transferencia de datos e intercambio de información entre las redes.

II.5 WSNTB.

El trabajo presentado en [Sheu'08] proporciona una completa infraestructura como banco de pruebas heterogéneo para redes de sensores inalámbricas en el que se plantea el diseño *hardware* y *software*, desde del nodo final sensor, hasta la herramienta en el servidor remoto para la gestión de todos los elementos que forman parte de dicha infraestructura, con el fin de contar con una completa plataforma para llevar a cabo experimentos.

Desde el punto de vista del nivel *hardware*, el *testbed* se compone de 34 nodos inalámbricos que implementan el protocolo IEEE 802.15.4. Se pueden diferenciar dos tipos de nodos, el primero denominado Optopus I basado en un AVR Atmega128L, el segundo conocido como Optopus II basado en un MSP430 de Texas Instruments. Ambos dispositivos llevan implementado un transceptor CC2420. Con el fin de crear un canal paralelo de comunicación entre el servidor y los dispositivos para tareas de programación y depuración, los diseñadores plantean una conexión Ethernet a cada nodo, para lo cual se utilizan convertidores Ethernet-RS232 en el caso del Optopus I, y un convertidor Ethernet-USB en el caso del Optopus II. Este tipo de arquitectura permite añadir cualquier dispositivo inalámbrico con conexión RS-232 o USB al *testbed*. Desde el punto de vista *software*, se ha creado una herramienta web para la

gestión de experimentos desde el servidor o bien vía internet de forma remota. Por otro lado, la plataforma da soporte a sistemas operativos para redes de sensores inalámbricas, tales como el LOS [Sheu'06], que es un sistema basado en algoritmos optimizados para reducir el consumo de recursos de memoria, a partir del sistema operativo TinyOS, el cual será repasado en el siguiente apartado.

III. OS para redes de sensores inalámbricas.

Otra de las líneas de investigación relacionada con la tecnología de redes de sensores es la creación de sistemas operativos para el control de los nodos, enfocados principalmente en la sencillez y facilidad de uso e implementación. Existen diversos sistemas operativos que son referencia en este campo de aplicación, aunque la dificultad radica en la extensa variedad de dispositivos y arquitecturas creadas, lo que hace que no sea siempre fácil realizar un *porting* de dichos OS a todas las plataformas. Aunque el objetivo del presente proyecto no es entrar en los detalles de los sistemas operativos disponibles, sí que se expondrán de forma general los más importantes en el estado del arte. A continuación se presenta el más extendido y de mayor uso en la actualidad, y luego se comentarán otros no menos importantes que se encuentran en la comunidad científica.

III.1 TinyOS.

Este sistema operativo ha sido otro de los aportes realizado por la Universidad de California en Berkeley [TinyOS], en cooperación con Intel Corporation, a las redes de sensores, ya que se centra en proporcionar una herramienta para la gestión de nodos inalámbricos enfocado en las limitaciones de memoria que caracterizan a los mismos. Está basado en el lenguaje nesC, que es un dialecto de C especialmente optimizado para ser usado en arquitecturas de bajos recursos de memoria. La filosofía de TinyOS está orientada al manejo de componentes, que son los que se encargan de realizar las tareas de bajo nivel. Dichos componentes ofrecen interfaces para ser utilizadas desde niveles superiores a través de comandos o eventos característicos del componente. También se pueden asociar tareas a un componente. Existen componentes predefinidos (denominados primitivos) pero el usuario puede a su vez definir nuevos componentes, o hacer uso de los ya existentes para crear otros.

TinyOS es un sistema basado en código abierto y que se encuentra en constante actualización y edición, por lo que existen gran variedad de componentes definidos para distintos núcleos de procesamiento, aunque es necesario destacar que desde sus inicios ha sido muy usado con los nodos desarrollados por la Universidad de California: el TelosB, y las MicaZ. Al estar estos dispositivos muy extendidos, son muchos los componentes y desarrollos que se pueden utilizar con ellos.

III.2 Contiki

Este sistema operativo es de código totalmente abierto y ha sido creado para sistemas embebidos y muy especialmente para dispositivos de redes de sensores inalámbricas, principalmente para aquellos microcontroladores que poseen bajos recursos de memoria. Una configuración típica de este OS necesita de sólo 2Kbytes de RAM y 40 Kbytes de ROM.

Desde su creación [Contiki] su programación está enfocada a eventos, por lo que no implementa complejos algoritmos para el procesamiento de tareas. Está centrado en la sencillez y los mínimos recursos que generalmente poseen los nodos inalámbricos. Su funcionamiento se basa en que cada vez que ocurre un evento, éste es asociado a su respectiva tarea la cual es procesada. Otra de sus características es que implementa el protocolo TCP/IP para microcontroladores de 8 bits, denominado uIP, además de otras pilas como Rime, que es un protocolo ultra liviano para la comunicación mediante *unicast* y *broadcast* unisalto y multisalto. La estructura general de este sistema operativo se aprecia mediante la figura 2.5.

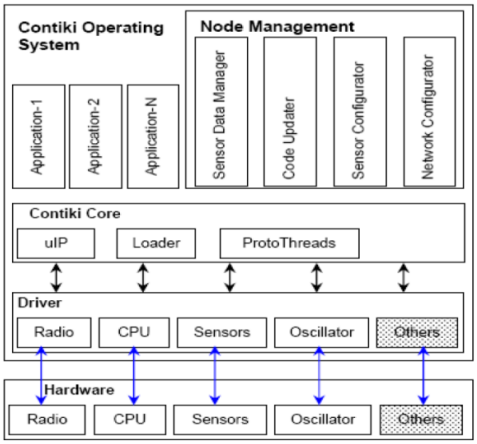


Figura 2.5: Arquitectura de Contiki OS.

III.3 MANTIS OS

La principal característica de este sistema operativo es que posee un modelo de programación basado en multi-hilos, en el que se fijan prioridades en la ejecución de las tareas de procesamiento. Este OS implementa un programador de tareas que se encarga de gestionar la memoria con el fin de estructurar una tabla de hilos de ejecución (*threads*). Cada hilo posee atributos tales como puntero actual de la pila, nivel de prioridad del hilo, puntero al inicio de la función del hilo y puntero al siguiente hilo.

Tal y como lo plantean los diseñadores en [Mantis] la arquitectura basada en capas que ofrecen cada uno de los servicios proporcionados por el sistema operativo, tal y como se ilustra en la figura 2.6.

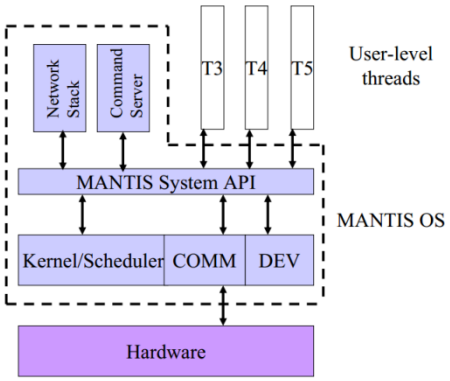


Figura 2.6: Arquitectura de MANTIS OS.

Este sistema operativo está enfocado a aplicaciones de bajo consumo ya que cada hilo de ejecución puede hacer llamado a la función *sleep ()*, la cual permite configurar el microcontrolador en modo de bajo consumo. Este sistema operativo esta desarrollado en lenguaje C y ha sido probado en plataformas *hardware* extendidas como Mica2, MicaZ y TelosB.

III.4 LiteOS

Este sistema operativo, desarrollado en la universidad de Illinois [LiteOS], ha sido especialmente creado para programadores que quieran desarrollar aplicaciones para redes de sensores inalámbricas sin tener mayores conocimientos de la plataforma *hardware*, ya que se basa en la filosofía de trabajo UNIX, así como programación enfocada a hilos de ejecución, y programación orientada a eventos. La idea se basa en proporcionar una herramienta para la gestión de los nodos inalámbricos, tales como configuración de sensores, transmisión y recepción de datos e instalar aplicaciones en los dispositivos.

Debido a que es un OS ligero, ha sido implementado principalmente en los nodos MicaZ, que disponen de 128 bytes de memoria flash de programa, y 2 Kbytes de memoria RAM. Actualmente está siendo implementado en la serie AVR de microcontroladores Atmel.

IV. Testbeds de uso público.

En este apartado se presentarán algunos de los *testbeds* más importantes y extendidos en la comunidad internacional, desde el punto de vista de su escala y propósito de aplicación. En esta selección entran aquellos bancos de pruebas que han sido concebidos para fomentar experimentos de forma abierta por parte de investigadores, en los cuales se despliega una gran cantidad de nodos inalámbricos con el fin de poder probar nuevos desarrollos en un entorno real de aplicación. Por lo general, al ser de uso extendido, se basan en las plataformas que más comúnmente son utilizadas por la comunidad científica, tales como nodos Telos y el sistema operativo TinyOS. Es importante destacar que son muchos los *testbeds* con estas características que se pueden encontrar en el estado de la técnica, aunque en este apartado se presentarán algunos de los más relevantes.

IV.1 MoteLab.

Motelab es un *testbed* para redes de sensores inalámbricas creado en la Universidad de Harvard [Werner-Allen'05] con el fin de dar soporte a experimentos de forma remota a través de una herramienta web como interfaz de usuario, en la que se pueden realizar tareas de reprogramación de los nodos, monitorización de datos, programación de tareas, entre otras acciones. Se basa en un repertorio de nodos desplegados permanentemente en el Electrical Engineering and Computer Science de Harvard. Inicialmente el proyecto incluía 26 nodos Mica2 desplegados, los cuales fueron actualizados por 30 MicaZ. Posteriormente, se han desplegado 190 nodos Tmote Sky que consisten en un microcontrolador MSP430 de Texas Instruments y un chip de radio CC2420. Los nodos incluyen sensores de temperatura, humedad y luz. Además de ello, se encuentran permanentemente conectados a la red eléctrica, por lo que la fuente de alimentación está asegurada.

Es posible realizar tareas de depuración, reprogramación y registro de datos de forma remota mediante un canal adicional cableado (*backchannel*), ya que todos los nodos se encuentran conectados vía Ethernet al servidor central, aunque esta conexión es secundaria puesto que la comunicación entre los nodos de la red se realiza de forma inalámbrica. Todos los dispositivos incluyen TinyOS como sistema operativo.

Desde la página web del *testbed* [MoteLab] se pueden descargar archivos para el desarrollo de aplicaciones, programar tareas, obtener mapas de conexión entre los dispositivos y crear prototipos de aplicaciones para realizar experimentos en tiempo real, así como conocer el estado de cada uno de los nodos incluidos en la red inalámbrica.

IV.2 Tutornet.

Este *testbed*, desarrollado y desplegado en la Universidad de Southern California está compuesto por 91 nodos TmoteSky y 13 MicasZ conectados vía USB a un dispositivo *stargate* que implementa el protocolo IEEE 802.11 b para comunicarse con el servidor central. Este tipo de conexión permite realizar la programación de forma paralela mediante un *backchannel* cuando la red inalámbrica está en funcionamiento. Desde la página web del *testbed* [Tutornet] se pueden obtener las instrucciones para realizar experimentos sobre toda la plataforma. La arquitectura propuesta se muestra en la figura 2.7.

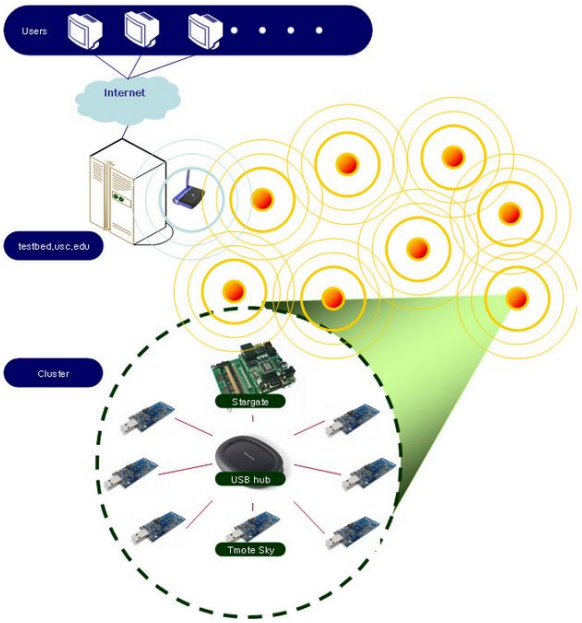


Figura 2.7: Arquitectura de Tutornet.

IV.3 Indriya.

Este *testbed* sigue una filosofía muy similar a la presentada en Motelab, haciendo uso de la herramienta web de ésta como interfaz para dar acceso a la plataforma. Se basa en 127 TelosB desplegados en School of Computing de la Universidad de Singapore [Chan'11], y permite cargar experimentos para realizar pruebas en tiempo real desde la interfaz web disponible a toda la comunidad científica [Indriya]. Al igual que MoteLab, los nodos incluyen TinyOS como sistema operativo, y se comunican mediante IEEE 802.15.4 a través del chip CC2420 implementado en el nodo. Del mismo modo, cuentan con la filosofía *backchannel* para realizar tareas de depuración y programación. Por otro lado, la plataforma implementa sensores infrarrojos,

magnetómetros, acelerómetros, temperatura, luz y acústica.

IV.4 TWIST.

Este proyecto, desarrollado por la Universidad Tecnológica de Berlín, representa una de las mayores plataformas a gran escala para realizar pruebas en tiempo real, ya que incluye más de 200 nodos desplegados en 1500 m2, dentro de instalaciones del campus de la universidad. La arquitectura del *testbed* propuesta en [Handziski'06] se basa en dos tipos de nodos sensores, los Tmote Sky y los eyesIFX. Actualmente existen 102 nodos por cada tipo, los cuales se conectan vía USB a unos "supernodos" que son los encargados de realizar la conexión puente Ethernet/WLAN a los servidores centrales y posteriormente a clientes remotos vía internet, tal y como se aprecia en la figura 2.8.

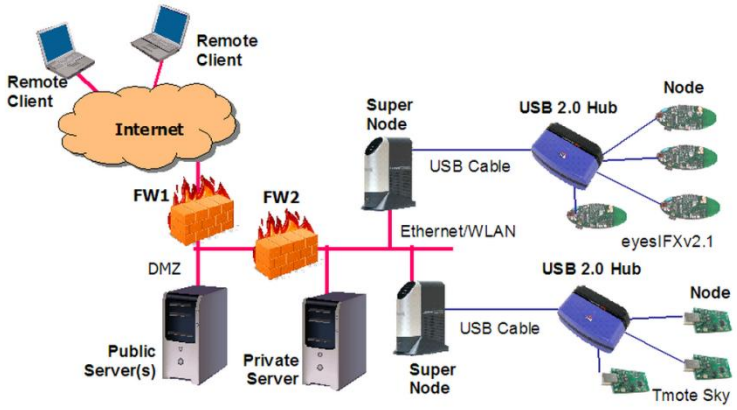


Figura 2.8: Arquitectura de TWIST.

Es posible acceder vía web a la plataforma [TWIST], así como configurar un perfil propio de aplicación para llevar a cabo experimentos en tiempo real. Del mismo modo, se pueden descargar archivos y códigos fuentes relacionados con actualizaciones de la misma.

IV.5 Imote2 Sensor Network.

Esta plataforma, basada en los nodos Imote2, es un *testbed* enfocado en la evaluación de algoritmos de localización y seguridad en la red. Los nodos se comunican entre sí vía inalámbrica y transmiten la información al nodo base (coordinador), el cual se encuentra conectado por USB a una estación de control que recibe todos los datos y en el que se encuentra implementada una interfaz de usuario para la gestión de la plataforma.

Aunque el *testbed* no dispone de interfaz remota, desde la magina web [Imote2] se pueden descargar las actualizaciones de los algoritmos de localización y la interfaz local de control.

IV.6 CitySense.

La filosofía de este *testbed* se diferencia de los presentados hasta el momento en que está enfocado en un despliegue de redes de sensores inalámbricas en un entorno totalmente *outdoor* con el fin de monitorizar parámetros medioambientales. Este proyecto, desarrollado por la Universidad de Harvard en colaboración con BBN Technologies, tiene como finalidad cubrir toda una localidad (Cambrige, MA) a través de 100 nodos inalámbricos, para monitorizar parámetros relacionados con los niveles de polución [Murty'08]. Específicamente, los investigadores podrán recolectar información acerca de los picos de polución durante el día, así como podrán realizar experimentos con algoritmos de localización de los dispositivos desde cualquier parte del mundo mediante acceso web a la plataforma.

En este caso los desarrolladores utilizan como dispositivos inalámbricos ordenadores embebidos con sistema operativo Linux y con comunicación IEEE 802.11 a/b/g, para ser desplegados en diferentes puntos de la ciudad, tal y como se muestra en la figura 2.9.

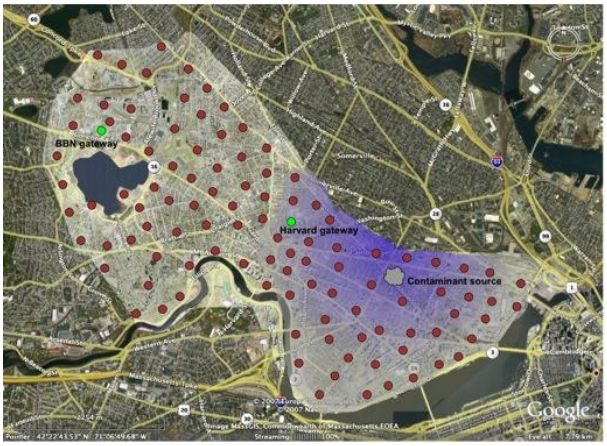


Figura 2.9: Mapa del despliegue de CitySence en Cambrige, MA.

La plataforma está abierta a la comunidad científica para configurar experimentos y aplicaciones relacionadas desde cualquier parte de forma remota [CitySense].

Capítulo 3

BIBLIOTECAS SOFTWARE PARA EL CONTROL DEL NODO INALÁMBRICO

I. Introducción.

Debido a la versatilidad de las redes de sensores inalámbricas y en especial de la plataforma Cookies, creada con el fin de cubrir un amplio rango de aplicaciones a través de una tecnología modular como la que representa, se hace cada vez más creciente la necesidad de contar con una herramienta que permita gestionar los recursos que posee dicha plataforma de forma que se pueda lograr un rápido y eficiente prototipado en cada aplicación en concreto. Actualmente existen herramientas de gestión para redes de sensores inalámbricas en desarrollo, como el bien conocido sistema operativo *TinyOS* [TinyOS], así como otros sistemas y aplicaciones mencionadas en el estado del arte, aunque cabe destacar que para cada una de ellas se hace necesario realizar un *porting* a la plataforma Cookies con el fin de obtener un correcto funcionamiento de dichos sistemas.

Aún cuando existen tales sistemas operativos disponibles, surge la necesidad de crear de forma alternativa una plataforma *software* que se adapte más adecuadamente a los recursos disponibles en los nodos Cookies, y a su filosófica modular, especialmente para la gestión de las comunicaciones inalámbricas y del procesamiento tanto de sensores analógicos como digitales, siendo necesario en este último caso una correcta gestión de las comunicaciones entre el núcleo principal (microcontrolador) y la FPGA, que es la encargada de procesar los sensores que proporcionan información de forma digital.

Se agudiza la necesidad de crear dicha plataforma por el hecho descrito anteriormente de que las redes de sensores pueden ser aplicadas en diferentes entornos, lo que conlleva a que la plataforma (tanto a nivel *hardware* como a nivel *software*) sea capaz de adaptarse fácilmente a los mismos, para lo cual se hace necesario converger el *hardware* de los nodos Cookies con una plataforma *software* fácil e intuitiva de utilizar para cualquier desarrollador de la aplicación, de modo que se garantice el prototipado rápido de la misma, que es el objetivo final de todo el sistema conjunto.

Esta necesidad da como resultado el diseño de bloques funcionales dentro del núcleo principal, en una arquitectura que, independiente de la capa de procesamiento a utilizar, siempre siga la filosofía de contar con los tres niveles paralelos de procesamiento, que son el módulo de comunicaciones y sus respectivos recursos, la FPGA y el microcontrolador, este último como contenedor de la plataforma *software* y núcleo principal del sistema conjunto.

A partir de todo lo expuesto, se plantea la creación de bibliotecas *software* modulares para redes de sensores inalámbricas, enfocadas en la plataforma Cookies y enmarcado en los requisitos de diseño de las mismas, para lo cual es necesario cumplir con ciertas condiciones y parámetros, los cuales se detallan a continuación.

II. Requisitos mínimos de diseño y funcionamiento.

Las bibliotecas *software* han de estar enmarcadas dentro de los requisitos de diseño que rigen a la plataforma Cookies, los cuales se desglosan a continuación:

- *Modularidad en la gestión de los recursos del sistema*: Dichas bibliotecas deben estar clasificadas según el tipo de recursos que gestionan, así como las funcionalidades a controlar, como por ejemplo gestión de comunicaciones, gestión de periféricos, gestión de sensores, entre otros aspectos. Cada clasificación debe tener campos distintivos que permitan gestionar características dentro de cada funcionalidad en específico.
- *Abstracción en la gestión de la aplicación*: Con el fin de lograr un prototipado rápido sin necesidad de entrar en los detalles de bajo nivel de la plataforma, se hace necesario crear funciones de tal forma que el nivel de abstracción para el desarrollador de la aplicación sea lo más alto posible, sólo siendo necesario conocer ciertos parámetros a configurar para el funcionamiento de los bloques.
- *Parametrización de los bloques funcionales*: con el fin de cumplir con la abstracción en el prototipado de diversas aplicaciones, es necesario que las funciones a desarrollar posean parámetros característicos de entrada y salida, así como ciertas directivas de funcionamiento.
- *Independencia de la plataforma*: Un aspecto indispensable para el desarrollo de las bibliotecas *software* modulares es la independencia de los bloques funcionales al *hardware* en las que irán embebidas, siempre que dicho *hardware* sea de carácter modular y, como se ha mencionado anteriormente, se base en la relación *Comunicaciones – FPGA – Microcontrolador*. Esto quiere decir que, aunque pueden haber funciones que sean dependientes de la tecnología utilizada (un tipo de microcontrolador, un tipo de módulo de comunicaciones), la filosofía, estructura y base de la función ha de ser la misma para una u otra, siendo *sólo* necesario pequeñas modificaciones a bajo nivel en el caso que se realice el *porting* a nuevas capas de procesamiento.

Estos requisitos han dado lugar a la creación de 4 bibliotecas base que dan soporte a toda la plataforma modular y que sirven de guía para la gestión de los principales recursos del sistema conjunto, las cuales se complementarán con las creadas posteriormente para la aplicación final a implementar. Estas bibliotecas están desglosadas por funcionalidades y cada una de ellas posee bloques específicos de procesamiento, tal y como se ha planteado en las condiciones de diseño.

Las bibliotecas han sido implementadas en lenguaje C y partiendo del *hardware* descrito en el capítulo I, pero al ser el concepto y la filosofía de las funciones

extensible a cualquier otro *hardware* que se desee implementar, la descripción de las bibliotecas y funciones se hará de forma global (bloques funcionales), sin entrar en los detalles de cada función en concreto. Se han seguido ciertas nomenclaturas, como el hecho de que los archivos de cabecera, en el que se encuentran las definiciones, valores y parámetros característicos de los bloques funcionales (interfaz entre la biblioteca y el desarrollador) y código fuente, en el que se encuentra definido el comportamiento de los dichos módulos, reciben el nombre de "*_CEI_funcionalidad.h*" y "*_CEI_funcionalidad.c*", respectivamente. Todas las funciones que componen las cuatro bibliotecas base de la plataforma se desglosan al final del capítulo.

III. Clasificación de las bibliotecas según los recursos del sistema a gestionar.

Como se ha mencionado anteriormente, el núcleo principal de procesamiento del nodo inalámbrico es el microcontrolador implementado en la plataforma Cookie, de modo que se hace necesario gestionar los recursos internos que este dispositivo posee de la forma más efectiva posible, como por ejemplo protocolos de comunicación, interrupciones internas y externas, convertidores analógicos-digitales, entre otras funcionalidades propias de cualquier microcontrolador comercial. Pero más allá de los recursos intrínsecos del mismo, es necesario realizar un control exhaustivo de diversos periféricos externos como es el caso de los módulos de sensores digitales implementados en la FPGA, o la comunicación con el módulo ZigBee, su correcta configuración y la gestión de los datos inalámbricos que se envían y reciben. Según el tipo de recurso a gestionar, se han clasificado las cuatro bibliotecas base como sigue a continuación.

III.1 Biblioteca "Queue".

Puesto que en los nodos pertenecientes a una red de sensores, sea el tipo que sea (Coordinador, *Full Functional Device* o *Reduce Functional Device*, como se verá más adelante), las tareas de procesamiento son de diversa índole, lo que quiere decir que el control de actividades y la gestión de datos se realiza ya sea por puertos de entrada y salida, lectura de convertidores, protocolos de comunicaciones y muy especialmente la gestión de la UART. Esto hace necesario la creación de un mecanismo para lograr en cierta forma el "paralelismo" (siempre tomando en cuenta el carácter secuencial del uC) de la gestión de los recursos, sobre todo por el hecho de evitar esperas activas innecesarias que se dan en ciertos casos, tomando en cuenta que existen diferentes frecuencias en el procesamiento y control de diferentes recursos.

Especialmente se hace necesario contar con un mecanismo para la gestión eficiente del puerto serie debido a que en éste recae una gran parte de la información y los

mensajes de comunicación a procesar, ya que el enlace entre el microcontrolador y el módulo ZigBee se realiza mediante la UART. En el caso de un nodo coordinador que está conectado a un ordenador, esta gestión se hace más evidente puesto que intervienen en la transmisión serie el módulo ZigBee, el Microcontrolador y el puerto serie del PC.

Todo ello da como resultado el diseño de un "buffer" basado en una cola estructurada con el fin de ir almacenando los datos entrantes/salientes del puerto serie del microcontrolador, apoyándose en los recursos de interrupción que éste proporciona, lo que permite gestionar los datos que son recibidos/transmitidos para su posterior procesamiento.

La estructura de la cola diseñada (ya sea en el caso de transmisión o recepción) posee tanto señales de datos como de control, es de carácter circular y, en términos generales, cumple con la arquitectura que se muestra en la figura 3.1.

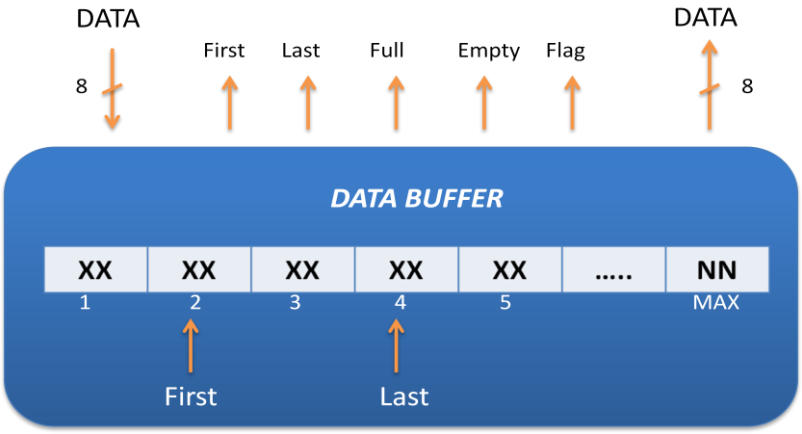


Figura 3.1: Estructura general de la cola de transmisión/recepción serie.

Cada posición de la cola está compuesta por 8 bits, tal y como se compone cada dato proveniente del puerto serie. Las señales de control asociadas a la cola se describen a continuación:

- *First y Last*: Señales que apuntan al primer y último elemento introducido en la cola y que se van actualizando independientemente en función de los elementos que entran y salen de la misma.

- *Full*: Bandera que indica que la cola está llena, por lo que no se pueden almacenar más elementos.

- *Empty*: Bandera que indica que la cola está vacía, por lo que no se pueden extraer más elementos.

- *Flag*: Bandera que indica que una cadena entera está lista para ser leída y procesada. Este bit de control es fundamental para detectar los casos en los que se ha recibido completamente un *prompt* del módulo ZigBee y por lo tanto realizar el respectivo procesamiento.

- *MAX*: Es el número máximo de bytes que se pueden almacenar en la cola. En el archivo de interfaz de la biblioteca (archivo de cabecera), este valor se encuentra parametrizado, junto con otros aspectos que permiten ajustar la cola en función de las necesidades de la aplicación.

Como los datos de transmisión y recepción funcionan de forma independiente, se han creado dos colas estructuradas para gestionar independientemente los datos que vayan a ser enviados y recibidos, ambas con parámetros para su modificación y control.

Siguiendo con las directrices planteadas anteriormente, los bloques funcionales que definen el comportamiento de la cola se encuentran en el archivo "*_CEI_queue.c*", y la interfaz entre la biblioteca y el nivel superior se encuentra en el archivo "*_CEI_queue.h*". Más adelante se describirá de forma conceptual el funcionamiento de los bloques que componen a esta biblioteca, que se considera como la base del procesamiento del nodo inalámbrico puesto que todos los datos, tramas y *prompts* del módulo ZigBee pasan por dicha biblioteca, además de ser utilizada también para la comunicación con el ordenador central de monitorización en el caso del nodo coordinador (COO).

III.2 Biblioteca "peripherals".

En esta biblioteca se encuentran los bloques funcionales encargados de gestionar los periféricos y módulos propios del microcontrolador, así como el control de módulos externos como es el caso de la FPGA, o la gestión de las comunicaciones de la UART mediante las interrupciones del puerto serie.

Esta biblioteca se puede destacar como la columna vertebral en el control de los bloques funcionales pertenecientes al *hardware* en el que se implemente la plataforma *software*. Aunque evidentemente existen funciones que son dependientes del *hardware*, como se ha dicho anteriormente la filosofía y metodología de los bloques funcionales debe ser la misma en uno u otro caso, con el fin de garantizar la interoperabilidad y portabilidad de la plataforma. Y en efecto, los archivos de interfaz con los niveles superiores (archivos de cabeceras) han de permanecer

inalterados (en cuanto a los módulos ya implementados) con el fin de garantizar dicha operatividad, haciendo en tal caso pequeñas modificaciones en el código fuente para adaptar las funciones a la plataforma *hardware* involucrada en la aplicación.

La estructura global de esta biblioteca se puede ilustrar a partir de la figura 3.2, en la que se aprecian los bloques funcionales involucrados en la gestión de los periféricos:

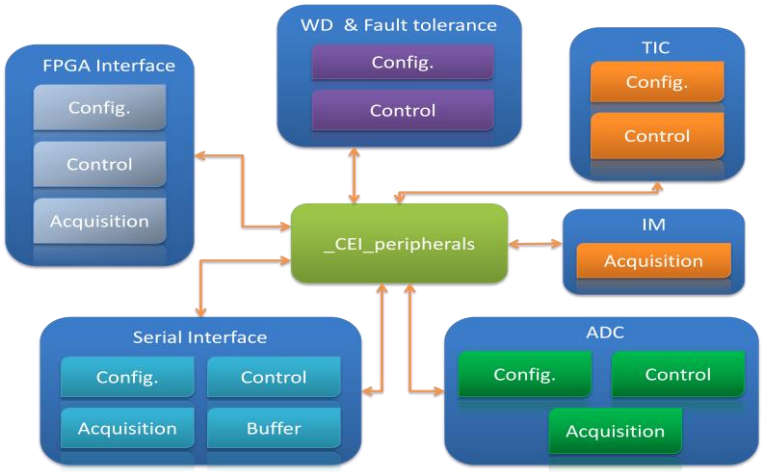


Figura 3.2: Estructura general de los módulos funcionales de _CEI_peripherals.

Como se ilustra en dicha figura, existen principalmente 5 bloques funcionales, encargados cada uno de la gestión de periféricos concretos. Más adelante se expondrán cada una de estas funcionalidades y el rol que tienen sobre el sistema global para la gestión del nodo inalámbrico. Dichos bloques funcionales abarcan los periféricos que son utilizados usualmente en la plataforma Cookies, sin que esto genere restricciones en cuanto a implementación de nuevos bloques que controlen nuevos periféricos, manteniendo la flexibilidad y modularidad de la plataforma para futuros desarrollos.

III.3 Biblioteca "Data Process".

Como se ha visto en el apartado anterior, y como se detallará más adelante en los módulos funcionales, la biblioteca *_CEI_peripherals* permite gestionar los periféricos que componen la plataforma Cookies, realizando principalmente un control más a bajo nivel en el sentido de que se realizan tareas cercanas al nivel *hardware*, como la manipulación de las interfaces de control y la adquisición de los datos provenientes

de los módulos externos. A partir de ello, se hace necesario subir de nivel con el fin de procesar de forma adecuada los datos obtenidos y enlazarlos con las tareas y acciones correspondientes, sobre todo en el caso de medidas de magnitudes físicas, puesto que por lo general los datos obtenidos de sensores han de ser transformarlos a magnitudes reales y estandarizadas, como es el caso por ejemplo de sensores analógicos, de los cuales se obtienen valores que estarán dentro de los rangos del convertidor analógico/digital del microcontrolador, por ejemplo 0 a 4095 (caso de un convertidor de 12 bits). En este caso, es palpable la necesidad de contar con módulos encargados de procesar los datos adquiridos, realizar cálculos numéricos y adaptar las magnitudes capturadas a valores estandarizados útiles, como por ejemplo valores de temperatura en grados Celsius.

Otro de los factores que influyen en la adquisición de datos provenientes de sensores es la necesidad en algunos casos de realizar una calibración de los mismos ya que en muchas ocasiones existen sensores que poseen cierta desviación típica, derivas, o son propensos a mostrar un comportamiento relacionado con la no calibración del mismo, por lo que se hace evidente la corrección pos-captura de las magnitudes físicas.

Todo esto hace necesario la creación de la biblioteca *_CEI_DataProcess*, la cual se encarga de las tareas de procesamiento y decodificación de los sensores, haciendo uso de los recursos de la biblioteca *_CEI_peripherals*. Además de ello, esta biblioteca se enmarca en la gestión de datos no necesariamente de carácter sensoriales, lo que incluye tareas útiles de conversión de tipo de datos, enmascaramiento de información para ser enviada por la trama ZigBee, decodificación de valores, entre otras funciones.

Hasta la fecha se encuentran implementados los bloques funcionales de los sensores que están presentes en la plataforma *hardware* de las Cookies, entre los que se encuentran sensores de temperatura, humedad relativa, LDR, acelerómetro en ejes Y-X, PH y temperatura del agua, CO, consumo de corriente y tensión del nodo inalámbrico, cuyo procesamiento se detallará más adelante. Del mismo modo, siguiendo con la filosofía de flexibilidad y modularidad de la plataforma, es posible agregar nuevas funciones en la medida en que se aumenten el número los sensores disponibles en trabajos futuros.

La arquitectura general de esta biblioteca se ilustra en la figura 3.3.

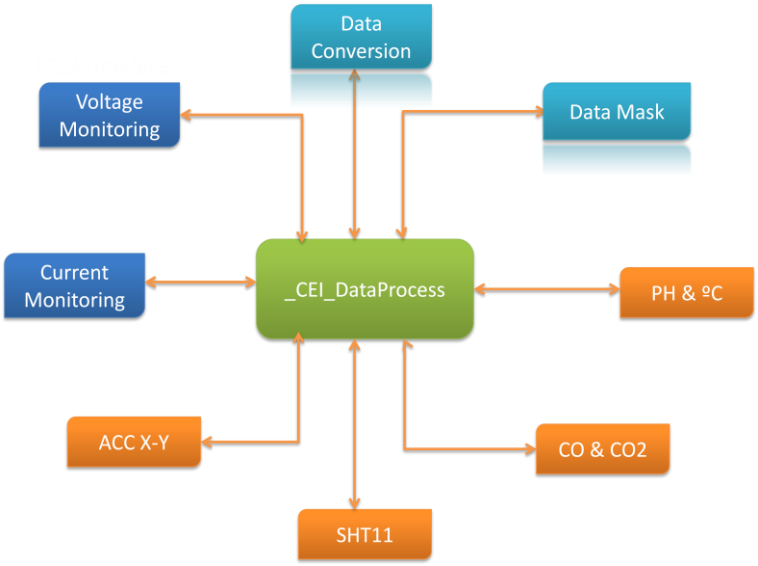


Figura 3.3: Arquitectura general de la biblioteca _CEI_DataProcess.

Más adelante se estudiarán los tipos de conversiones que se realizan según la adquisición de medidas de los sensores de la plataforma inalámbrica, y las peculiaridades que estos poseen para llevar a cabo su procesamiento. Se comentará a grandes rasgos el funcionamiento de los sensores cuando sea necesario con el fin de entender de forma más intuitiva la decodificación que se realiza en cada caso, aunque es necesario destacar que el estudio y desarrollo de estos sensores no entra dentro de los objetivos de este proyecto.

III.4 Biblioteca "ZigBee".

Desde el punto de vista de las comunicaciones, esta biblioteca es la más importante y una de las más complejas e interesantes a nivel de procesamiento, puesto que se encarga de gestionar completamente la capa de comunicaciones haciendo un extenso uso de la biblioteca *_CEI_queue*. Aunque sus bloques funcionales están enfocados principalmente para el control de los recursos del protocolo inalámbrico ZigBee, también se incluyen funcionalidades adicionales como la gestión de mensajes relacionados con la comunicación con el PC al ser ello también parte de la red en si misma (monitorización y control desde el ordenador central).

Es necesario destacar que esta biblioteca, denominada *_CEI_zb*, está desarrollada principalmente para el control del módulo de comunicaciones "de facto" implementado en la plataforma Cookies, que es el ETRX2 de la empresa *Telegesis,* tal

y como se ha descrito en el capítulo I. De hecho, los bloques funcionales están basados en los recursos de dicho módulo.

En este punto se hace necesario estudiar ciertos aspectos con el fin de responder a los requisitos de portabilidad y flexibilidad de la plataforma *software*. Aunque se trate de un módulo especifico de comunicaciones, la filosofía de los bloques funcionales de esta biblioteca se basa en determinados aspectos clave para el control del protocolo ZigBee, que son de aplicación común a cualquier módulo existente en el mercado puesto que finalmente se trata de un protocolo que posee características estandarizadas para su funcionamiento y gestión. Estos aspectos se enumeran de forma general a continuación, aunque su uso se extiende en toda la biblioteca:

- *Interfaz serie y control basado en comandos AT:* La mayoría de los módulos comerciales actualmente poseen firmwares basados en el control mediante los comandos AT, por lo que no es sólo una característica intrínseca del módulo ETRX2.

- *Gestión de parámetros de la Red:* Aspectos como direcciones *MAC* y *Short Address*, identificadores de red, conexión y reconexión a una red, configuración de claves de seguridad, canal de comunicación, entre otros parámetros básicos de ZigBee, se pueden encontrar en cualquier módulo de comunicaciones comercial en la actualidad.

- *Registros de configuración y control:* Tanto el ETRX2 como muchos otros fabricantes poseen la opción de modificar registros de configuración en los módulos, con el fin de poder variar parámetros relacionados con la comunicación serie, periféricos de entrada y salida, GPIO, claves de seguridad, entre otros parámetros básicos para el funcionamiento del *hardware*.

- *Implementación del ZigBee Device Object (ZDO):* el módulo ETRX2 proporciona funciones para gestionar el ZDO, lo cual es fundamental para controlar y depurar la red inalámbrica ZigBee, además de ser uno de los mecanismos que permite realizar la interoperabilidad entre módulos. El ZDO, como se comentará más adelante, es un estándar dentro de la pila ZigBee y por tanto ha de estar implementado en la mayoría de los módulos de comunicaciones comerciales.

Aunque es cierto que las instrucciones y los comandos utilizados para las funciones de la biblioteca están basados en el firmware del ETRX2, y la descripción de las características del módulo ZigBee se hará desde el punto de vista de este módulo, los parámetros de configuración han de ser los mismos para cualquier otro módulo que

se implemente, sólo realizando ajustes a bajo nivel para adaptar la configuración a otro dispositivo en concreto. De hecho, como se verá en el capítulo 5, la biblioteca fue implementada y probada en una aplicación real de despliegue con un módulo de comunicaciones de otro fabricante, el Zigbit de la empresa *Atmel* [ATMEL], pero la filosofía de los bloques funcionales y la interfaz con los niveles superiores era la misma, comprobando su correcto funcionamiento en condiciones reales de funcionamiento.

A continuación se muestra la estructura global de *_CEI_zb*, y los bloques funcionales incluidos en su arquitectura:

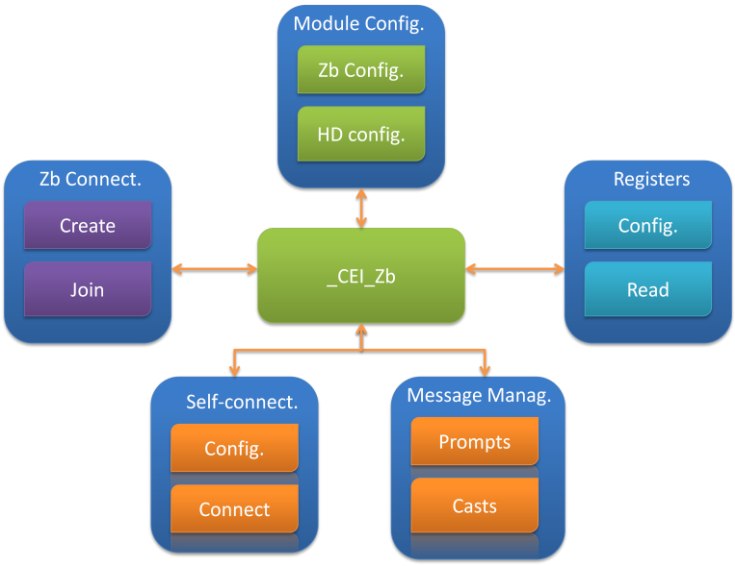


Figura 3.4: estructura general de la biblioteca _CEI_zb.

Como se verá más adelante, esta biblioteca permite tener un control total de las tareas relacionadas con el módulo de comunicaciones y las conexiones inalámbricas, sobre todo por el hecho de que facilita y ahorra un gran trabajo al desarrollador de una aplicación gestionar los parámetros de la red ZigBee, puesto que con unos simples cambios en la configuración se puede establecer y mantener una red totalmente operativa. Del mismo modo, la biblioteca permite gestionar tanto los mensajes que se envían y reciben a través del módulo ZigBee, pero además de ello, se pueden también gestionar los mensajes que se pretenden enviar a la estación central donde se encuentra la interfaz de monitorización del sistema conjunto (caso del COO) para así poder mostrar mensajes de información y alertas sobre los eventos

que ocurren en la red inalámbrica.

Como se ha comentado anteriormente, estos 5 bloques funcionales de *_CEI_Zb* hacen especial uso de la biblioteca *_CEI_queue*, para así poder realizar una gestión eficiente de los mensajes que circulan por la comunicación serie mediante la UART.

Hasta ahora se ha hecho una presentación general de las cuatro bibliotecas base que forman parte de la plataforma *software* y se ha comentado las principales características y funcionalidades típicas de cada una. En el siguiente apartado se entrará en los detalles funcionales de cada uno de los bloques de cada biblioteca, haciendo especial hincapié en aquellos que son críticos para el funcionamiento tanto de la plataforma *hardware*, como de las comunicaciones inalámbricas.

IV. Bloques funcionales dentro de las bibliotecas software.

En este apartado se describirán más detenidamente los bloques funcionales de cada biblioteca expuesta, así como las características y parámetros de interés para su correcto uso. Como se ha comentado anteriormente, la explicación de los bloques se hará de manera funcional, sin entrar en detalles de sintaxis o niveles de código de las funciones creadas para tal fin. Más allá de ello, se destacarán parámetros a tomar en cuenta a la hora de realizar configuraciones, así como los valores de entrada y salida de cada bloque funcional. Al final del capítulo se podrá obtener una síntesis con todas las funciones correspondientes a los bloques funcionales que se detallarán a continuación.

Como se verá a lo largo de este apartado, hay funcionalidades en las que están involucradas más de una función en específico, por lo que se hablará explícitamente de bloques funcionales, mientras que en otros casos se comentarán directamente las funciones que realizan una tarea en concreto.

IV.1 Funciones de la Biblioteca "queue".

Dentro de esta biblioteca se encuentran todas las funciones para la inicialización, control y modificación de la cola de transmisión y recepción de datos por el puerto serie, las cuales se detallan a continuación. En la tabla 3.1 se enumeran los parámetros modificables dentro de la interfaz de la biblioteca con niveles superiores:

PARÁMETRO	DESCRIPCIÓN
Tamaño del buffer	Modificación de la longitud del buffer de datos.

Carácter de fin de cadena	Carácter estándar que indica a los bloques funcionales cuál es el fin de cadena de entrada/salida.
Carácter de error	Carácter estándar a devolver en caso de detección de errores en lecturas/escrituras.
Estructura del Buffer	Aunque la estructura de la cola ya está íntegramente definida (figura 3.1), es posible añadir bits de control en trabajos futuros, cuando sea necesario.

Tabla 3.1: Parámetros modificables en el fichero de interfaz con la biblioteca "queue" (_CEI_queue.h).

IV.1.1 Función "iniqueue".

Esta función permite inicializar los valores de control de la cola tanto de transmisión y recepción, con el fin de poder utilizarla con valores predefinidos. Esta función ha de ser utilizada al principio de toda aplicación en la que se haga uso de esta cola, para así asegurar un correcto comportamiento de la misma.

IV.1.2 Función "newElement".

Esta función permite añadir un elemento (1 byte) a la cola de transmisión de datos, siempre y cuando la cola no se encuentre llena. Los parámetros de entrada y salida de dicha función se muestran en la tabla 3.2:

Parámetros de Entrada	Parámetros de Salida
Carácter de entrada (1 byte).	Resultado de la operación.
Cola a escribir.	

Tabla 3.2: Parámetros de entrada y salida de newElement.

IV.1.3 Función "newString".

Esta función permite añadir una cadena entera a la cola de transmisión de datos, siempre y cuando la cola no se encuentre llena. Los parámetros de entrada y salida de dicha función se muestran en la tabla 3.3:

Parámetros de Entrada	Parámetros de Salida
Cadena de entrada.	Resultado de la operación.
Cola a escribir.	

Tabla 3.3: Parámetros de entrada y salida de newString.

IV.1.4 Función "newString_Part".

Esta función realiza el mismo procedimiento que la anterior, pero se diferencia en el hecho de que copia sólo una parte de la cadena de entrada en la cola de transmisión. Los parámetros de entrada y salida de dicha función se muestran en la tabla 3.4:

Parámetros de Entrada	Parámetros de Salida
Cadena de entrada.	Resultado de la operación.
Posición de la cadena de entrada hasta donde se desea escribir.	
Cola a escribir.	

Tabla 3.4: Parámetros de entrada y salida de newString_Part.

IV.1.5 Función "getElement".

Esta función permite leer un elemento (1 byte) a la cola de recepción de datos, siempre y cuando la cola no se encuentre vacía. Los parámetros de entrada y salida de dicha función se muestran en la tabla 3.5:

Parámetros de Entrada	Parámetros de Salida
Cola de la que se leerá el carácter.	Carácter leído de la cola.

Tabla 3.5: Parámetros de entrada y salida de getElement.

IV.1.6 Función "getString".

Esta función permite leer una cadena entera de la cola de recepción de datos hasta encontrar el carácter de fin de cadena, siempre y cuando la cola no se encuentre vacía. Los parámetros de entrada y salida de dicha función se muestran en la tabla 3.6.

Parámetros de Entrada	Parámetros de Salida
Cadena en la que almacenará lo leído en la cola.	Cadena leída de la cola.
Cola a leer.	

Tabla 3.6: Parámetros de entrada y salida de getString.

IV.1.7 Función "getString_Part".

Esta función realiza el mismo procedimiento que la anterior, pero se diferencia en el hecho de que lee sólo una parte de la cadena de la cola de recepción. Los parámetros de entrada y salida de dicha función se muestran en la tabla 3.7.

Parámetros de Entrada	Parámetros de Salida
Cadena en la que almacenará lo leído en la cola.	Cadena leída de la cola.
Posición de la cola hasta donde se desea leer.	
Cola a leer.	

Tabla 3.7: Parámetros de entrada y salida de getString_Part.

IV.1.8 Función "space".

Esta función permite calcular el espacio libre disponible en una cola. Los parámetros de entrada y salida de dicha función se muestran en la tabla 3.8.

Parámetros de Entrada	Parámetros de Salida
Cola de la que se quiere calcular el espacio disponible.	Espacio disponible en la cola.

Tabla 3.8: Parámetros de entrada y salida de space.

Estas funciones descritas hasta ahora proporcionan una base para el tratamiento de los datos de entrada y salida del núcleo principal (microcontrolador), y son de gran utilidad en niveles superiores de procesamiento, ya que es utilizada en la mayoría de los bloques funcionales que se utilizan en la plataforma *software*.

IV.2 Funciones de la biblioteca "peripherals".

Tal y como se ha descrito anteriormente, esta biblioteca permite dar soporte para el control de los periféricos internos y externos del microcontrolador, por lo que en este apartado es necesario entrar en algunos detalles de bajo nivel con el fin de entender el funcionamiento de los bloques, sobre todo en las comunicaciones con la FPGA y la adquisición de medidas de sensores digitales. A continuación se explican los periféricos involucrados, las características a tener en cuenta para su funcionamiento, y el control que se realiza sobre los mismos. Ya que en este apartado intervienen más de una función para realizar diferentes actuaciones en cada periférico, se hablará de bloques funcionales y de las funciones que están incluidas dentro de estos, tal y como se ha ilustrado en la figura 3.2.

En la tabla 3.9 se enumeran los parámetros modificables dentro de la interfaz de la biblioteca con los niveles superiores:

PARÁMETRO	DESCRIPCIÓN
Carácter Inicio y Fin ZigBee	Permite definir los caracteres de inicio y fin estándar de cadena proveniente del módulo ZigBee.
Carácter de Inicio y Fin PC	Permite definir los caracteres de inicio y fin estándar de cadena proveniente del PC.
Variables Externas	Se externalizan variables útiles como el control del Watch Dog y los bytes de datos de la FPGA.

Tabla 3.9: Parámetros modificables en el fichero de interfaz con la biblioteca "perpherals" (_CEI_peropherals.h).

IV.2.1 Bloque funcional "ADC".

Este bloque se encarga de realizar la gestión del convertidor analógico-digital del microcontrolador con el fin de obtener las medidas físicas de los sensores analógicos de la plataforma *hardware*. Tal y como se puede obtener en detalle de la hoja de datos del microcontrolador, el ADC utilizado posee una resolución de 12 bits, 8 canales disponibles de conversión, una referencia interna de 2.5 V y varios modos de operación entre los que se incluyen *Single Conversion Mode* y *Continuous Conversion Mode*. Con el uso de la referencia interna de tensión y 12 bits de resolución, se obtiene una precisión de 2.5V/4096 = 0.61 mV por bit.

Los 12 bits correspondientes al resultado de la conversión se encuentran distribuidos en dos bytes, tal y como se muestra en la figura 3.5.

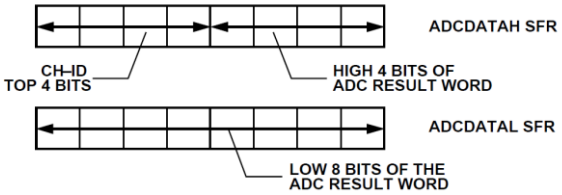


Figura 3.5: 12 bits de conversión del ADC distribuidos en 2 bytes de datos.

La gestión y adquisición de datos del ADC se realiza principalmente mediante 3 funciones, las cuales se detallan a continuación:

- *iniADC*: Esta función configura el modo de operación del convertidor y recibe como parámetro de entrada el canal a utilizar para la captura.

- *getADCvalue*: Permite obtener como parámetro de salida el valor en 12 bits del resultado de la conversión del ADC, con el fin de poder realizar operaciones con el valor total resultante, y no con los dos bytes separados.

- *ADCconversion*: Esta función se alimenta de las dos anteriores y permite realizar una captura completa del convertidor analógico digital, configurando el canal a utilizar, realizando la petición de captura y obteniendo el valor resultante en formato de 12 bits. Es la función más útil a nivel superior de abstracción para la gestión del ADC puesto que realiza todo el procedimiento necesario para la correcta utilización del mismo.

A parte de estas funciones que se consideran las más importantes del bloque, se

encuentran otras enfocadas a la depuración del convertidor, como por ejemplo conocer el estado de la conversión o gestionar la interrupción del mismo, entre otras opciones.

Como se verá más adelante, este bloque es especialmente útil y muy aplicado en la biblioteca *_CEI_DataProcess*, la cual se nutre de la biblioteca *_CEI_peripherals* para realizar las conversiones de las magnitudes físicas obtenidas de los sensores.

IV.2.2 Bloque funcional "TIC".

Este bloque funcional se encarga de gestionar el temporizador global de la aplicación, que se utiliza para gestionar interrupciones internas periódicas configurables, con el fin de realizar tareas de control y procesamiento de datos. Este módulo TIC (*Time Interval Counter*), es una característica muy importante que posee el microcontrolador *AduC841* puesto que permite llevar un contador global en horas, minutos y segundos, por lo que se puede establecer un reloj interno de aplicación. En el caso de utilizar este bloque funcional en otra plataforma, será conveniente realizar los ajustes necesarios al temporizador específico que posea dicho *hardware*, sin que ello requiera modificar la forma y la interfaz de las funciones.

El control del bloque TIC se realiza mediante dos funciones que se detallan a continuación:

- *iniTIC*: Esta función permite configurar correctamente el temporizador mediante los registros de control correspondientes y activar la interrupción asociada al mismo. Recibe como parámetros de entrada la base de temporización, entre las que están 1/128 segundos, segundos, minutos y horas, así como el periodo de interrupción asociado a dicha base de temporización. En otras palabras, si se selecciona como base de temporización segundos, y se especifica en periodo 2, la interrupción interna asociada al temporizador se ejecutará cada 2 segundos.

- *modeTIC*: Esta función permite configurar el comportamiento del temporizador, entre lo que se incluye la activación o desactivación del mismo y el tipo de interrupción (individual o periódica).

IV.2.3 Bloque funcional "Serial Interface".

Este es uno de los bloques funcionales de mayor relevancia ya que gestiona las comunicaciones del puerto serie, que son la base del intercambio de datos entre el microcontrolador y el módulo ZigBee, además de la conexión con la estación central de monitorización en el caso del nodo coordinador. A través de las funcionalidades

que se encuentran dentro de este bloque, se capturan los *prompts* provenientes del módulo ZigBee y se almacenan en la cola de recepción de datos, para su posterior procesamiento. Del mismo modo ocurre en el caso de la transmisión de mensajes.

Principalmente se encuentran definidas dos funciones que se encargan de controlar el puerto serie, las cuales se detallan a continuación:

- *iniSerial*: Esta función permite configurar la UART del microcontrolador, especificando como parámetro de entrada el tipo de *baudrate* al cual se quiere trabajar, entre los que se encuentran 9600, 19200, 38400, aunque es necesario destacar que las comunicaciones en la plataforma Cookies trabajan por defecto a 19200 b/s, por lo que es la velocidad a la que se debe configurar el puerto.

- *Serial_interrupt*: La gestión de los datos de entrada y salida del puerto serie se realiza mediante la interrupción asociada a la UART, para así cumplir con una filosofía mas enfocada a eventos y evitar esperas activas. Esta interrupción se llevará a cabo cuando ocurra un evento, bien sea en la transmisión de datos, o bien en la recepción, para lo que se divide la función en dos sub-bloques funcionales para gestionar cada caso.

Es necesario destacar que este bloque implementa la detección de mensajes provenientes del PC al que esté conectado el nodo Cookie, por lo que es necesario conocer la naturaleza del mensaje, es decir, si éste proviene de la red inalámbrica, o si proviene de la estación central de control. Esto es aplicable sobre todo en el caso del nodo coordinador que se encuentra conectado al PC. La estructura general de los mensajes en ambos casos viene definido tal y como se muestra en la figura 3.6.

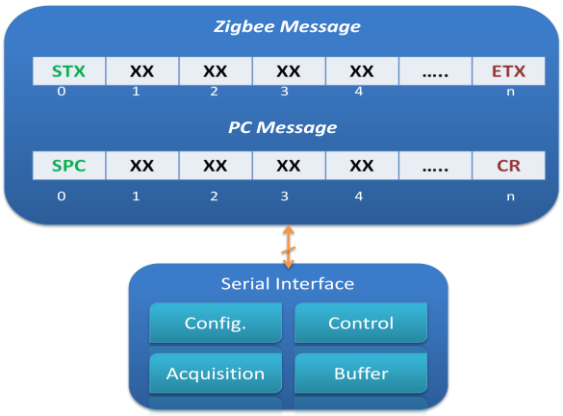


Figura 3.6: Trama de datos proveniente del módulo ZigBee y PC.

Las definiciones de los valores STX, ETX, SPC y CR se encuentran en los parámetros modificables de la interfaz de la biblioteca, en el caso de que sea necesario definir otros valores de control de cadenas diferentes a los ya implementados.

IV.2.4 Bloque funcional "Watch Dog".

Aunque la mayoría de las funciones criticas del sistema poseen valores de control de salida para conocer si se produce un error en el comportamiento de una función o si el resultado obtenido en algún procesamiento se considera dentro de valores no deseables, con el fin de aumentar la fiabilidad de la aplicación global se hace uso de la herramienta de *Watch Dog* o "perro guardián". Esta funcionalidad, que viene incluida en la mayoría de los microcontroladores comerciales, permite generar una interrupción en el momento en el que se desborde un temporizador que va asociado a dicha funcionalidad, y la principal ventaja es que, en el momento de generarse la interrupción, existen dos caminos de salto del contador de programa según la configuración realizada. La primera opción es saltar al vector de interrupción asociado al temporizador, o bien saltar al vector de inicio de programa, por lo que se estaría realizando en este último caso un reset de la aplicación embebida en el microcontrolador.

Esta función es especialmente útil en los casos en los que se necesita controlar el tiempo que tarda una función en procesar datos, o recibir información desde otro bloque funcional, lo que permite generar una alerta en el caso de detectarse un error o, en el escenario más crítico, reiniciar la aplicación. En tal sentido, aparte de los parámetros propios que proporciona el microcontrolador, se han agregado otros parámetros adicionales que pueden ser utilizados de forma externa por los demás bloques funcionales y que están definidos en la interfaz de la biblioteca, y los cuales se detallan a continuación:

- *_WD_timeCont:* Contador que permite generar un temporizador a partir del proporcionado por el *Watch Dog*, ya que el definido internamente tiene como valor máximo configurable 2 segundos por interrupción. Es por ello que, para extender este tiempo, se hace uso de este parámetro para lograr tiempos múltiplos de 2 segundos. En el caso de necesitar valores menores, se puede configurar para obtener un tiempo mínimo de 16 milisegundos por cada interrupción.

- *_WD_timeout*: Valor configurable que permite fijar el tiempo máximo de temporización antes de generar una alerta de error en la aplicación.

- *_WD_flag*: Bandera que indica que se ha producido un error al alcanzar el tiempo máximo configurado *(_WD_timeout_ = _WD_timeCont_).*

Estos valores se utilizan con la opción de interrupción al vector asociado, con el fin de llevar un control del número de interrupciones y cuya gestión se realiza en la propia función de interrupción.

Muchas de las funciones críticas del sistema, como las pertenecientes a la biblioteca ZigBee, implementan el *Watch Dog* con el fin de detectar fallos en el nodo inalámbrico. Por otro lado, es conveniente utilizar con atención y precaución esta funcionalidad, ya que un uso descontrolado puede llevar a generar alerta de errores cuando en realidad no ocurren (desbordamiento del temporizador antes de lo esperado por mala configuración, mantener habilitada la funcionalidad cuando no se necesita, etc). Es por ello que se han diseñado funciones para su fácil control y evitar estas situaciones. La metodología que se debe seguir para su uso es básicamente habilitarlo justo antes de su utilización, resetearlo dentro del proceso de ejecución cuando sea necesario, y deshabilitarlo cuando su utilización haya finalizado.

Las funciones asociadas al control del *Watch Dog* se detallan a continuación:

- *iniWD*: Permite configurar los parámetros del *Watch Dog*, e inicializar los valores característicos.
- *resetWD*: Permite realizar un reset del temporizador del *Watch Dog*.
- *stopWD*: Detiene la ejecución del *Watch Dog*.
- *Watchdog_interrupt*: gestiona el número de desbordamientos del temporizador del *Watch Dog*, y genera la activación de la bandera de advertencia en el caso de que se alcance el *timeout* configurado.

IV.2.5 Bloque funcional "FPGA".

Como se ha comentado en apartados anteriores, la plataforma modular es capaz de operar con sensores tanto analógicos como digitales. En el caso de los sensores analógicos la adquisición de los datos en el núcleo principal de procesamiento se lleva a cabo mediante el convertidor analógico-digital, por lo que la interfaz entre la medida física y el elemento de procesamiento está totalmente asegurada mediante un camino "estándar". En el caso de sensores digitales el concepto es diferente, puesto que cada fabricante impone un determinado protocolo de comunicación como interfaz entre el procesador de los datos y la medida física, y como es sabido existen innumerables protocolos disponibles en el mercado, lo cual no garantiza la estandarización en la adquisición que se desea en la plataforma modular.

Debido a este planteamiento, en el Centro de Electrónica Industrial se crearon una

serie de bibliotecas *hardware* con el fin de estandarizar el medio de comunicación entre el procesador principal de la aplicación y los sensores digitales [Portilla'07]. Dichas bibliotecas han sido implementadas para su uso en elementos *hardware* como las FPGAs, tal y como la que forma parte de la plataforma modular.

Estas bibliotecas permiten obtener una interfaz común para todos los sensores digitales, lo que los hace ver de forma genérica y por igual desde el punto de vista del microcontrolador, logrando la estandarización a un solo camino de adquisición, al igual que ocurre con los sensores analógicos. Aunque ni la explicación de dichas bibliotecas ni los protocolos de comunicación que han sido estandarizados entran dentro del alcance del presente documento, si es necesario destacar algunos detalles importantes de la interfaz genérica con el fin de entender claramente el control que se realiza desde el microcontrolador.

Hasta la fecha se han implementado las interfaces más extendidas en el mercado, que son: PWM, I²C, 1-Wire y periodo/Frecuencia. La interfaz común entre el microcontrolador y la FPGA está basada en *Triggers* o señales de disparo para la adquisición de los datos, señales de control y reconocimiento, y un bus de datos por el que se obtendrán las medidas finales de los sensores. La arquitectura global implementada se muestra a continuación:

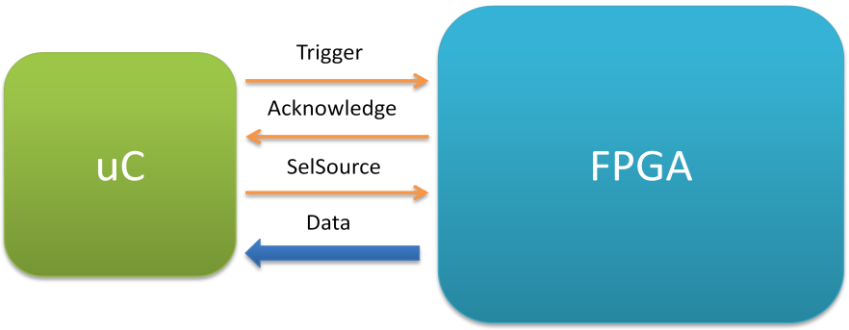


Figura 3.7: Interfaz de comunicación genérica entre el microcontrolador y la FPGA, para la adquisición de los datos de sensores digitales.

La interfaz común entre ambos dispositivos está compuesta de dos bytes, uno de datos y uno de control, cuyas señales se describen a continuación:

- *TrigByte1N*: El valor resultante de la adquisición de los datos de los sensores digitales está compuesto por 16 bits, pero al utilizar 8 bits de datos para la comunicación, es necesario hacer la petición de los bytes por separado, primero del byte menos significativo mediante esta señal, y luego del más significativo.

- *Ack1N*: Señal que indica que la transmisión del byte menos significativo se ha realizado con éxito y se puede solicitar el siguiente byte.

- *TrigByte2N*: Petición de byte más significativo del dato digital.

- *Ack2N*: Confirmación de la transmisión del byte más significativo del dato digital.

- *SelTrigger*: Puesto que en un mismo nodo puede haber más de un sensor digital, es posible que se implementen en la FPGA varias interfaces digitales, por lo que las señales de selección permiten elegir el sensor específico cuya adquisición se desea realizar.

- *Reset*: Permite hacer un reset de la interfaz genérica de comunicaciones entre el microcontrolador y la FPGA.

Debido a esta estandarización era evidente crear funciones desde el lado del procesador que controlaran la interfaz genérica de los sensores digitales, lo que ha dado lugar a la implementación del bloque FPGA, que cumple con el estándar de las bibliotecas *hardware*. Dicho bloque se basa fundamentalmente en dos funciones, que se detallan a continuación:

- *iniFPGA*: Permite inicializar el protocolo entre el microcontrolador y la FPGA, para iniciar la comunicación entre ambos dispositivos.

- *get_FPGAValue*: Permite capturar el valor de 16 bits perteneciente a la magnitud física adquirida del sensor digital. Esta función se encarga de generar los *triggers* necesarios y leer las señales de reconocimiento con el fin de obtener correctamente los valores de los sensores. El único parámetro externo necesario es la especificación del canal a utilizar, dependiendo del sensor cuya medida se va a adquirir.

IV.2.6 Bloque funcional "IM".

Esta funcionalidad, que está relacionada con la gestión de las interrupciones, más que un bloque representa el hecho de que todas las interrupciones de los periféricos involucrados en la plataforma se encuentran implementadas en esta biblioteca, por lo que los vectores de interrupción de los mismos se encuentran incluidos y, por tanto, las tareas de gestión relacionadas con las mismas.

IV.3 Funciones de la biblioteca "DataProcess".

Tal y como se ha comentado anteriormente, esta biblioteca se respalda principalmente en la biblioteca *peripherals*, con el fin de convertir las magnitudes de los sensores a las unidades estándares más comunes, como temperatura en grados Celsius, CO en partes por millón (ppm), entre otras. Es necesario recordar que la descripción de los sensores no está dentro de los objetivos del presente documento, por lo que no se entrará en detalles funcionales de los mismos, sólo indicando los parámetros de entrada y salida necesarios en cada caso.

En esta biblioteca cada conversión de medidas está asociada a una función en concreto (salvo en el caso de conversión y enmascaramiento de datos, que están compuestas por varias funciones), por lo que se hablará directamente de las funciones que relaciona cada sensor con su respectiva conversión. Se hace nuevamente este tipo de acotación puesto que en casos como la biblioteca *peripherals* se hace más conveniente hablar de bloques funcionales, ya que intervienen más de una función para la gestión de una determinada tarea, mientras que en los casos como *DataProcess* o *queue* se habla directamente de funciones. Lo más importante de cara a la implementación final son las funcionalidades (bloques y funciones) que posee la plataforma global.

Entre los parámetros característicos que se pueden modificar en la interfaz de la biblioteca se encuentran las estructuras de datos que están asociadas a aquellos sensores que poseen doble medida para su funcionamiento, como es el caso por ejemplo del sensor de PH que se relaciona a su vez con la temperatura en ^{o}C.

IV.3.1 Función "PHSensor".

Esta función calcula el valor de la temperatura en grados Celsius y reutiliza este valor para el cálculo de PH, el cual es dependiente de la temperatura. Los parámetros de entrada y salida necesarios para su funcionamiento se muestran en la tabla 3.10.

Parámetros de Entrada	**Parámetros de Salida**
Valores de PH y temperatura obtenidos previamente del *ADC*.	Cadena de datos con el valor de PH y temperatura en sus respectivas unidades.
Cadena que contendrá los valores resultantes	

Tabla 3.10: Parámetros característicos de la función PHSensor.

IV.3.2 Función "COSensor".

Esta función calcula el valor de CO en partes por millón (ppm) a partir del dato obtenido del ADC, por lo que en este caso se trata de un sensor analógico. Los parámetros de entrada y salida necesarios para su funcionamiento se muestran en la tabla 3.11.

Parámetros de Entrada	Parámetros de Salida
Valor del sensor de CO obtenido del ADC.	Valor del sensor CO en unidades ppm.

Tabla 3.11: Parámetros característicos de la función COSensor.

IV.3.3 Función "SHT11_conversion".

Esta función calcula el valor de la temperatura ambiente en grados Celsius y el porcentaje de humedad relativa a partir del sensor SHT11, que posee una interfaz I^2C, por lo que se trata de un sensor digital. Los parámetros de entrada y salida necesarios para su funcionamiento se muestran en la tabla 3.12.

Parámetros de Entrada	Parámetros de Salida
Selección del tipo de medida a convertir (temperatura o humedad).	Valor resultante en ^oC o %H.
Valor de la medida a convertir (a partir de la captura de la FPGA).	

Tabla 3.12: Parámetros característicos de la función SHT11_conversion.

IV.3.4 Función "ACC_conversion".

Esta función calcula el valor de aceleración en los ejes X e Y (unidades g), a partir del sensor *ADXL210E* de *Analog Devices*, cuyo sensor posee una interfaz PWM, por lo que se trata de un sensor digital con módulo implementado en las bibliotecas *hardware*. Los parámetros de entrada y salida necesarios para su funcionamiento se muestran en la tabla 3.13.

Parámetros de Entrada	Parámetros de Salida
Selección del tipo de medida a convertir (Acc eje Y o X).	Valor resultante en g.
Valor de la medida a convertir (a partir de la captura de la FPGA).	

Tabla 3.13: Parámetros característicos de la función ACC_conversion.

IV.3.5 Función "VoltageMonitoring".

Esta función permite monitorizar los niveles de tensión de alimentación en toda la plataforma modular, a partir de un circuito de medida de la tensión de entrada implementado en la capa de alimentación de la plataforma *hardware*. La medida es adquirida desde el ADC y transformada mediante esta función, cuyos parámetros de entrada y salida se muestran en la tabla 3.14.

Parámetros de Entrada	Parámetros de Salida
Valor obtenido del ADC.	Valor de la tensión alimentación en mV.

Tabla 3.14: Parámetros característicos de la función VoltageMonitoring.

IV.3.6 Función "CurrentMonitoring".

Esta función permite monitorizar el consumo de corriente de toda la plataforma Cookies, a partir del circuito basado en la medida de tensión de una resistencia *shunt* y calculando el valor de la corriente equivalente. Los parámetros de entrada y salida necesarios para su funcionamiento se muestran en la tabla 3.15.

Parámetros de Entrada	Parámetros de Salida
Valor obtenido del ADC.	Valor del consumo de corriente en mA.

Tabla 3.15: Parámetros característicos de la función CurrentMonitoring.

IV.3.7 Funciones para la conversión de datos.

Debido a que la plataforma *software* trabaja con diferentes tipos de datos, como cadenas de *strings* que poseen valores numéricos de interés, o medidas de sensores que están divididas en bytes, se hace necesario contar con mecanismos para la conversión de los mismos a formatos que sean convenientes para una tarea en especifico, como por ejemplo el hecho de tener una medida de temperatura (o cualquier otra magnitud) en 16 bits. Para poder enviar dicha información de forma inalámbrica es necesario separarla en dos bytes con el fin de poder transmitirla mediante el puerto serie al módulo ZigBee.

Con el fin de dinamizar estas conversiones, se han estandarizado funciones que implementan las mismas y las cuales se detallan a continuación:

- *ADCData2Int*: Permite concatenar dos bytes en un solo dato de 16 bits con el fin de poder realizar cálculos con el dato completo. Recibe como parámetros de entrada el byte más significativo y el menos significativo de la medida.

- *Int2String*: Permite convertir un valor entero de 16 bits, en valores imprimibles por pantalla. Por ejemplo, si se tiene el dato entero 2576, mediante esta función se convierte a la cadena "2576", teniendo a su vez la posibilidad de agregar una coma para expresar valores decimales por pantalla, por ejemplo "25.76". Recibe como parámetro de entrada el valor entero y la posición de la coma. Está hecha principalmente para mostrar resultados de sensores por pantalla.

- *string2num*: Permite convertir un *string* a un dato entero, útil para valores que se reciben por el puerto serie y que necesitan ser procesados.

IV.3.8 Funciones para el enmascaramiento de datos.

Puesto que existen caracteres de control de los mensajes y *prompts* que se envían y reciben por el puerto serie, como por ejemplo los caracteres de inicio y fin de cadena, carácter nulo, etc, se hace necesario realizar un enmascaramiento de la información que se envía por el puerto serie con el fin de evitar que un dato coincida con los caracteres de control. Para entender de una forma más intuitiva cual es el procedimiento de enmascaramiento de la información, se parte del ejemplo que se expone a continuación, el cual a su vez permite aclarar ciertos aspectos de la adquisición de los datos de sensores.

Se toma como base el querer enviar un dato relativo a temperatura ambiente en grados Celsius. La medida es enviada en tantos por cien, por lo que el dato sería, por

ejemplo, 2530 (25,30 ºC).

Dicho dato en binario se presenta tal y como se muestra en la figura 3.8.

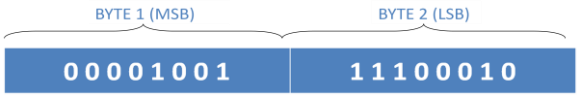


Figura 3.8: Separación de un dato entero en dos bytes.

Del byte 1 (MSB), los 6 primeros bits son los que contienen información (las medidas de sensores nunca sobrepasarán los 14 bits en el caso de sensores digitales, o 12 bits en el caso de sensores analógicos), mientras que en el byte 2 (LSB) se consideran más importantes los 7 bits más altos. Esto permite generar el enmascaramiento que se muestra en la figura 3.9.

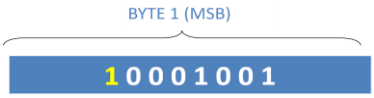


Figura 3.9: Enmascaramiento del byte más significativo de la medida.

En el caso del byte 2, los 7 bits más significativos se trasladan a la derecha, y se repite el mismo procedimiento que se ha hecho en el Byte 1, con lo que quedaría de la forma en la que se muestra en la figura 3.10.

Figura 3.10: Enmascaramiento del byte menos significativo de la medida.

Del lado del receptor de la información, dicho cambio debe ser desecho para así obtener el dato inicial.

Con el fin de que dicho procedimiento sea totalmente transparente para el desarrollador de la aplicación, se han creado las funciones necesarias para el enmascaramiento y desenmascaramiento de los datos según el procedimiento expuesto, las cuales son:

- *dataMask*: Enmascara el dato a enviar antes de ser transmitido por el puerto serie.

- *dataUnMask*: Desenmascara el dato recibido por el puerto serie.

En el capítulo 4 se ampliará la información del envío y la adquisición de los datos por la red inalámbrica a través del perfil de aplicación creado a partir de la trama de datos implementada para las aplicaciones basadas en la plataforma Cookies.

IV.4 Funciones de la biblioteca "ZigBee".

Como se ha comentado anteriormente, desde el punto de vista de las comunicaciones esta es la biblioteca más importante puesto que se encarga de realizar toda la gestión del módulo ZigBee para las comunicaciones inalámbricas. Aunque el propósito de este documento no es entrar en los detalles de bajo nivel asociados al protocolo ZigBee, si se aclararán algunos parámetros básicos para la correcta configuración de la red, así como aspectos a tomar en cuenta a la hora de utilizar las funciones contenidas en la biblioteca. Del mismo modo, aunque la plataforma está pensada para poder ser totalmente portable y utilizable con cualquier módulo de comunicaciones ZigBee, la explicación de los bloques se hará en función del módulo ETRX2.

En el archivo de interfaz con los niveles superiores (*_CEI_zb.h*), se encuentran los tres parámetros a modificar y que son mínimos para poder configurar correctamente la red inalámbrica, los cuales se muestran en la tabla 3.16:

PARÁMETRO	DESCRIPCIÓN
Dirección PAN larga.	Definición del identificador largo de la red inalámbrica a crear/conectar.
Dirección PAN corta.	Definición del identificador corto de la red inalámbrica a crear/conectar.
Canal a utilizar.	Elección del canal que se utilizará (11 – 26).

Tabla 3.16: Parámetros modificables en el fichero de interfaz con la biblioteca "ZigBee" (_CEI_zb.h).

En este sentido es conveniente aclarar ciertos aspectos relacionados con el funcionamiento de la red inalámbrica ZigBee [Daintree 1], y que se desglosan a continuación:

- *Direcciones PAN*: ZigBee permite configurar los identificadores de la red inalámbrica, que son la dirección larga (64 bits) y la dirección corta (16 bits). El módulo de comunicaciones es capaz de conectarse automáticamente a una red inalámbrica con la que posea mejor enlace y calidad de señal, en el caso de que no se especifique una PAN a conectar. En el caso de la plataforma Cookies los módulos se conectarán específicamente a la red configurada mediante la biblioteca ZigBee, con el fin de evitar errores en la conexión (por ejemplo si hubiese más de una red ZigBee en el entorno de aplicación).

- *Canal de comunicación:* Dependiendo de la frecuencia a la cual trabaje la aplicación y el módulo ZigBee empleado, es necesario configurar el canal de comunicación de la red inalámbrica. El módulo ETRX2 trabaja a una frecuencia de 2.4 GHz, en la que se puede seleccionar cualquiera de los 16 canales de comunicación (en [802.15.4] definidos del 11 al 26).

- *Direcciones MAC y ShortID:* Cada dispositivo ZigBee posee una dirección única MAC definida de fábrica y que está compuesta por 64 bits. Adicionalmente, el protocolo asigna una dirección corta "virtual" cada vez que un dispositivo se conecta a la red. Dicha asignación es de carácter aleatoria, por lo que cada vez que el nodo se conecta recibe una dirección corta diferente.

- *Registros de configuración*: Los módulos comerciales poseen registros de configuración con el fin de controlar parámetros tales como direcciones PAN, modificación de *prompts* de datos, configuración de los periféricos del módulo (UART, puertos I/O, SPI, temporizadores, etc), niveles de seguridad, entre otros aspectos [Telegesis 2].

- *Tipos de Dispositivos*: Se pueden configurar tres tipos de dispositivos dentro de la red ZigBee: Nodo Coordinador (COO), que es el nodo que crea y controla a red inalámbrica y debe existir uno por red; Nodo *Router* (*FFD, Full Functional Device*), que permite enlazar dispositivos finales con otros FFD y con el COO; Nodo Final (RFD, *Reduce Functional Device*), que son aquellos dispositivos que se conectan al coordinador a través de un nodo Padre (FFD), y no permite rutado de datos. Este último está pensado para nodos finales que requieran bajo consumo y enfocados a la adquisición de datos de sensores para luego ser transmitidos mediante su nodo padre.

- *Topologías:* Según la definición de los tipos de nodos, la topología de red será de tipo estrella, árbol o malla, siendo esta última la más utilizada a nivel de los nodos inalámbricos ZigBee.

A continuación se detallan los bloques funcionales que forman parte de la biblioteca ZigBee y que intervienen en la gestión de la red inalámbrica:

IV.4.1 Bloque funcional "Module Configuration".

Esta funcionalidad permite realizar configuraciones en el módulo ZigBee con el fin de garantizar su adecuado funcionamiento para la aplicación en la que será utilizado. Estas configuraciones no sólo tienen que ver con el nivel de comunicaciones, sino también con el nivel *hardware* del dispositivo. Está compuesto principalmente por la función *config*, que recibe como parámetro el tipo de nodo a configurar (COO, FFD o RFD), y permite realizar las siguientes tareas:

- *Canal de comunicación*: Selecciona el canal según el parámetro almacenado en *zb.h*.

- *Tipo de dispositivo*: configura el módulo para trabajar como COO, FFD o RFD, según el caso.

- *Direcciones PAN*: Configura la dirección PAN corta y larga de la red a la que se conectará el nodo.

- *Clave de red*: En caso de ser necesario, se puede cambiar la clave de seguridad de la red conformada por 128 bits. Por defecto el módulo se configura para que genere una clave aleatoria en el momento de establecer la red inalámbrica, la cual es recibida por los nodos en el momento de conectarse a la misma.

- *Puerto de E/S:* Configura los pines de entrada y salida del módulo ZigBee. En el capítulo 4 se verá la importancia que tiene el puerto externo del módulo ETRX2 en la gestión de la plataforma global.

- *Prompts:* Algunos mensajes de respuesta del módulo ZigBee pueden ser necesarios o innecesarios para la aplicación, por lo que son configurados los *prompts* que se desean recibir del módulo y los que no.

IV.4.2 Bloque funcional "Registers".

Este bloque permite modificar los registros de configuración del módulo ZigBee a través de dos funciones especificadas en la biblioteca. La primera, *modifySReg*, permite indicar el registro que se desea modificar y el parámetro a modificar del mismo, mientras que la función *send_SRegisters* otorga la posibilidad de crear una lista de registros a modificar con el fin de enviar una configuración en bloque al

módulo.

Esta funcionalidad dota de gran versatilidad a la comunicación con el módulo puesto que desde niveles superiores se puede especificar una lista completa de registros con parámetros a modificar y la función se encarga de realizar la configuración, logrando una mayor abstracción de nivel y automatización en el proceso.

IV.4.3 Bloque funcional "ZigBee Connection".

Este bloque permite realizar la conexión del módulo ZigBee a la red inalámbrica, según los parámetros configurados previamente en el mismo. Se basa en la función *connect*, que recibe como parámetro de entrada el tipo de conexión, lo que indica que existen dos opciones: conectarse a la red inalámbrica configurada (caso de FFD y RFD), o crear la red (caso de COO). En el caso de conexión a la red, la función configura el intervalo de tiempo en el que el nodo intenta conectarse a la red configurada, especialmente útil en los casos en los que se pierde conexión a la misma por alguna circunstancia como interferencias o problemas en la comunicación.

IV.4.4 Bloque funcional "Self-Connection".

Esta funcionalidad es una de las más importantes de la plataforma puesto que garantiza la conexión inalámbrica del dispositivo y su auto-reconexión en el momento en el que ocurra algún problema en la comunicación con la red. Se basa en la función *default_connection*, la cual se alimenta de los tres bloques anteriormente descritos y realiza una configuración y conexión automática simplemente con introducir como parámetro de entrada el tipo de nodo que se quiere configurar. Esta función detecta errores en la configuración y realiza la reconexión automática a la red cuando surge un problema en la conexión inalámbrica.

IV.4.5 Bloque funcional "Message Management".

Este bloque funcional permite realizar la gestión de los mensajes que se desean enviar y recibir del módulo ZigBee. Está compuesto por las funciones que se detallan a continuación:

- *sendMessage:* Se podría decir que, junto con la gestión de los buffers de transmisión y recepción de datos, es la función más extendida dentro de la plataforma, puesto que permite enviar mensajes al módulo ZigBee, comprobar que estos han sido recibidos correctamente en el destinatario, y detectar un tipo de respuesta según el mensaje enviado. Posee como parámetros de entrada el mensaje a transmitir y un valor indicativo del tipo de respuesta que se debe recibir, lo cual permite generar alertas de posibles

fallos en la transmisión o recepción de datos.

En el caso de enviar, por ejemplo, un mensaje CAST a un nodo especifico, las posibles respuestas son ACK (mensaje transmitido correctamente y recibido en el nodo receptor), NACK (mensaje no recibido por el nodo receptor) o ERROR (error en la transmisión). En los dos últimos casos se generaría una alerta a los niveles superiores con el fin de indicar que ha habido un problema durante el proceso de transmisión del mensaje.

Existe un repertorio de posibles respuestas según los datos a enviar, que se identifican con un determinado valor como parámetro (en el caso del ejemplo anterior el parámetro de entrada seria el valor 1). En el caso de no requerir una respuesta al mensaje enviado, el parámetro de entrada sería 0. Por otro lado, esta función también es útil para enviar mensajes al PC en el caso del nodo Coordinador, para lo que se establece un tercer parámetro que indica el destino del mensaje (ZigBee o PC).

- *sendUCAST*: Permite meter en la cola de transmisión la estructura de un mensaje UCAST, especificando como parámetros de entrada la información que se quiere enviar, y la dirección de destino del mensaje. Es una función especialmente útil para abstraer el envío de mensajes CAST de la tecnología que se esté utilizando.

V. Glosario de funciones de las cuatro bibliotecas base.

Hasta ahora se ha hecho una descripción de las cuatro bibliotecas base que conforman la plataforma *software* para la gestión del nodo Cookie, pero los detalles específicos de cada una de las funciones y los valores de entrada y salida de las mismas se pueden encontrar en [CEI 2012], que posee una descripción detallada de cada una de las funciones de las bibliotecas. A continuación, en las tablas 3.17, 3.18, 3.19 y 3.20, se muestra un desglose de las funciones que componen las bibliotecas descritas hasta el momento:

Funciones de la biblioteca qeueue	
_CEI_queue_iniqueue	*_CEI_queue_newElement*
_CEI_queue_newString	*_CEI_queue_newString_Part*
_CEI_queue_getElement	*_CEI_queue_getString*

_CEI_queue_space	

Tabla 3.17: funciones de la biblioteca queue.

Funciones de la biblioteca peripherals	
_CEI_per_iniADC	*_CEI_per_getADCvalue*
_CEI_per_ADCconversion	*_CEI_per_startADCconversion*
_CEI_per_ADCbusy	*_CEI_per_iniSerial*
_CEI_per_write2Serial	*_CEI_per_enableInts*
_CEI_Serial_interrupt	*_CEI_per_iniTIC*
_CEI_per_modeTIC	*_CEI_per_sleepuP*
_CEI_per_iniFPGA	*_CEI_per_get_FPGAValue*
_CEI_per_iniWD	*_CEI_per_resetWD*
_CEI_per_stopWD	*_CEI_Watchdog_interrupt*

Tabla 3.18: funciones de la biblioteca peripherals.

Funciones de la biblioteca DataProcess	
_CEI_dp_PHSensor	*_CEI_dp_COSensor*
_CEI_dp_SHT11_conversion	*_CEI_dp_ACC_conversion*
_CEI_dp_VoltageMonitoring	*_CEI_dp_CurrentMonitoring*
_CEI_dp_ADCData2Int	*_CEI_dp_string2num*
_CEI_dp_Int2String	*_CEI_dp_dataMask*

_CEI_dp_dataUnMask	

Tabla 3.19: funciones de la biblioteca DataProcess.

Funciones de la biblioteca ZigBee	
_CEI_zb_config	*_CEI_zb_connect*
_CEI_zb_default_connection	*_CEI_zb_configAsED*
_CEI_zb_modifySReg	*_CEI_zb_send_SRegisters*
_CEI_zb_sendMessage	*_CEI_zb_sendUCAST*
_CEI_zb_sendSN	*_CEI_zb_newShortMsg*

Tabla 3.20: funciones de la biblioteca ZigBee.

Capítulo 4

PLATAFORMA DE INTEGRACIÓN HARDWARE-SOFTWARE BASADA EN NODOS COOKIES

I. Introducción.

En el Capítulo anterior se han detallado las bibliotecas que permiten controlar y gestionar las características de los nodos Cookies, y se han presentado los bloques funcionales creados para tal fin. En este punto surge la necesidad de ir un paso más adelante y contar con una herramienta con mayor nivel de abstracción que sea capaz de utilizar dichos controladores para que el proceso de gestión del nodo y la red inalámbrica sea en la medida de lo posible transparente para el desarrollador de la aplicación, siendo necesario simplemente realizar algunos ajustes para el funcionamiento. Esto permitiría un prototipado mucho más rápido de los despliegues de redes de sensores inalámbricas, y su mantenimiento en el tiempo.

Es por ello que se hace evidente la creación de una plataforma que permita integrar las bibliotecas *software* implementadas con la plataforma *hardware* Cookies y que sirva de apoyo para futuros desarrollos de aplicaciones que estén enfocados en la utilización de estos nodos inalámbricos. Para tal fin, se ha diseñado lo que se denomina *Plataforma de Integración hardware-Software para Redes de Sensores Inalámbricas* y que se apoyará en una aplicación-despliegue tipo que será el banco de pruebas estándar (*testbed*) para dicha plataforma. Este *testbed* será la base para el prototipado de aplicaciones en laboratorio como preámbulo para los despliegues en entornos reales, y servirá como apoyo para realizar pruebas relacionadas con el desarrollo de nuevas capas *hardware* modulares, especialmente las nuevas capas de sensado que se desarrollen en el futuro. También permitirá dar soporte a nuevos desarrollos dentro las bibliotecas *software*. Este es un punto muy importante a tener en cuenta puesto que si en determinado momento es necesario realizar pruebas de una característica especifica (como por ejemplo una nueva capa de sensores para el sistema), la plataforma debe brindar un camino de pruebas rápido e intuitivo en el cual no sea un inconveniente la gestión conjunta de todo el sistema, sin importar el hecho de añadir un nuevo elemento. Esto requiere un diseño que se base en la flexibilidad, robustez y sencillez de la aplicación.

Para lograr tales objetivos, se han propuesto principalmente tres líneas de trabajo en el diseño e implementación de la plataforma, las cuales se desglosan a continuación:

- *Interfaz usuario-aplicación:* Esta rige la forma en la que se comunicará el desarrollador de la aplicación desde la estación central de control, con la plataforma inalámbrica.

- *Comunicación entre los nodos – nivel de aplicación:* Permitirá fijar ciertas directivas para la comunicación entre los nodos sensores y el nodo coordinador, desde el punto de vista de la aplicación (nivel superior al de

red). Esto da lugar a, como se verá más adelante, la creación de un perfil de aplicación propio para el intercambio de información entre los dispositivos.

- *Hardware del sistema:* Para la ejecución de las dos líneas anteriores es necesario explotar al máximo los recursos que ofrece la plataforma *hardware*, y realizar modificaciones cuando sean necesarias.

En la figura 4.1 se muestra la arquitectura global del banco de pruebas y la plataforma de integración basada en las tres líneas de diseño expuestas.

Figura 4.1: Arquitectura de la plataforma global.

A continuación se expondrá la arquitectura del *hardware* utilizado y los cambios que se han realizado para adaptar la plataforma *hardware* a las necesidades del banco de pruebas, sobre todo en el caso del nodo coordinador, el cual tiene que contener tanto los bloques funcionales para comunicación con la interfaz de usuario, como el perfil de aplicación creado para el intercambio de información entre los nodos.

II. Arquitectura de la plataforma de integración Hardware-Software.

La gestión del nodo inalámbrico se realiza mediante el trabajo conjunto de la capa de procesamiento y la capa de comunicaciones (Módulo de Comunicaciones – FPGA –

Microcontrolador). Dicho enlace entre los dispositivos permite aprovechar de forma más adecuada los recursos que poseen cada uno para que su aportación a la plataforma global sea mucho más efectiva.

Como se ha comentado anteriormente, el microcontrolador funciona como núcleo central de procesamiento, y es el que lleva embebido la aplicación para controlar todo el sistema. Las comunicaciones entre los tres dispositivos de procesamiento se llevan a cabo mediante la UART (caso del microcontrolador, el módulo de comunicaciones y el ordenador central), o mediante buses de datos (caso del microcontrolador y la FPGA, y el módulo de comunicaciones y la FPGA).

En el caso del nodo coordinador se plantean varios requisitos que son necesarios cumplir con el fin de lograr la flexibilidad deseada. En primer lugar, el hecho de ser capaz de realizar la gestión de los recursos del nodo haciendo uso de las bibliotecas, el perfil de aplicación para el intercambio de información con los nodos pertenecientes a la red, y la comunicación con el usuario a través del ordenador central. Por otro lado, el usuario debe poder comunicarse no sólo con la plataforma en si misma (nivel de aplicación) sino también con la red inalámbrica de forma directa (a través del módulo de comunicaciones) y por tanto tener acceso a los parámetros de configuración de la misma en el caso de ser necesarios ajustes en tiempo real (nivel de red). Esto plantea la necesidad de realizar un cambio en la filosofía de las comunicaciones entre todos los dispositivos de procesamiento de la plataforma *hardware*, especialmente para poder gestionar de forma más eficiente la interfaz UART que es la que permite el intercambio de información entre el PC y el nodo.

Para que esto sea posible se deben cumplir las siguientes condiciones de diseño:

- El usuario desde el ordenador central debe poder seleccionar que tipo de conexión desea establecer, es decir PC – uC (nivel de aplicación), o PC – Zb (nivel de red), y cambiarla cuando lo considere necesario.
- El microcontrolador, como elemento principal, debe ser capaz de seleccionar cualquier tipo de conexión cuando lo considere oportuno: PC – uC, Zb – uC o PC – Zb.
- En cada caso los cambios de comunicación deben ser reversibles, es decir, si se está en modo PC – uC y el usuario cambia a modo PC – Zb, el usuario debe ser capaz de volver al modo PC – uC en el momento que lo considere oportuno. Desde el punto de vista del nodo, esta condición requiere poder realizar el cambio o bien desde el uC, o desde el módulo ZigBee.
- Debe ser posible establecer un modo de depuración en el que sea posible la conexión entre los tres dispositivos. Esto quiere decir que, si por ejemplo se establece el modo PC – Zb, el microcontrolador ha de ser capaz de monitorizar la actividad entre ambos dispositivos, y tomar decisiones de

procesamiento cuando lo considere oportuno.

Estos planteamientos dan lugar al diseño de una arquitectura a nivel *hardware* para la gestión de las conexiones UART entre los dispositivos, que permita dotar de máxima flexibilidad a la plataforma. En tal sentido se ha propuesto aprovechar las ventajas de contar con un elemento *hardware* como es la FPGA para implementar un módulo de gestión UART en ella y así cumplir con los requisitos de funcionamiento. Para ello ha sido necesario realizar la implementación en VHDL y modificar algunas conexiones en el nodo Cookie para garantizar el correcto funcionamiento de dichos modos de funcionamiento.

II.1 Módulo de control UART.

El módulo de control de la comunicación serie, implementado en la FPGA mediante descripción en VHDL, se basa principalmente en dos señales de control que gobiernan el modo de conexión del puerto serie, y enlaza las líneas UART de los dispositivos que se pretenden conectar a través de las señales de transmisión y recepción de cada uno. Este control puede ser realizado desde el uC o desde el módulo ZigBee, por lo que se han implementado 4 señales de control, tal y como se muestra en la figura 4.2, en la que se aprecia de forma global la arquitectura de la descripción *hardware* implementada:

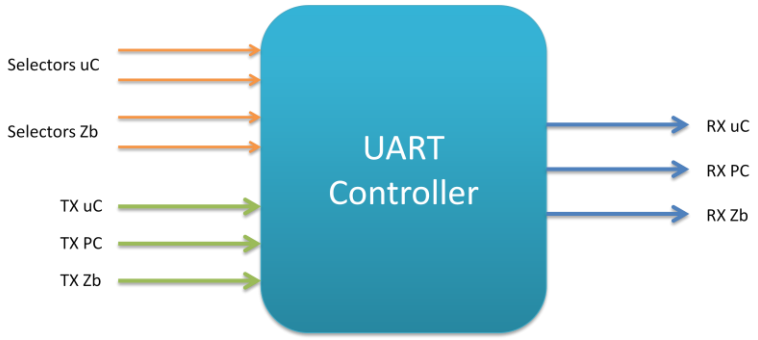


Figura 4.2: Señales de entrada/salida del controlador UART.

Los modos de comunicación no sólo permiten conectar la UART entre los dispositivos seleccionados, sino también realizar conexiones para la depuración de los datos que se transmiten vía serie, a través de conexiones adicionales "espías", tal y como se detalla en la tabla 4.1:

Selector	Conexión UART	Conexión de depuración (según implementación)
0 1	PC - Zb	Rx uC conectado a Tx Zb
		Rx uC conectado a Tx PC
1 0	uC - Zb	Rx PC conectado a Tx Zb
		Rx PC conectado a Tx uC
1 1	PC - uC	Rx Zb conectado a Tx uC
		Rx Zb conectado a Tx PC
0 0	PC – uC - Zb	Este es un modo adicional que hace una conexión circular entre los dispositivos, lo que permite tener una conexión activa en los tres elementos.

Tabla 4.1: Modos de funcionamiento del controlador UART.

Es necesario destacar el modo de funcionamiento PC – uC - Zb. En éste, el PC se encuentra conectado al módulo ZigBee (Tx PC – Rx ZB) con el fin de realizar tareas a nivel de red, mientras que este último está conectado al microcontrolador (Tx ZB – Rx uC) para así detectar y procesar mensajes relacionados con el perfil de aplicación. A su vez el microcontrolador está conectado al PC (Tx uC – Rx PC), con el fin de recibir datos en la estación central en el caso de que ocurra algún evento en la plataforma inalámbrica y, si es necesario por parte del uC, cambiar a otro modo a fin de actuar de forma oportuna. Todo esto permite crear una conexión circular entre los tres dispositivos, lo que proporciona un mayor control y flexibilidad a la plataforma.

Por otro lado, como se ha dicho anteriormente, las conexiones espías permiten realizar tareas de monitorización y depuración de los datos que circulan por la comunicación serie. Ejemplo de ello es el caso del modo uC – Zb, en el que se puede definir visualizar por pantalla en el ordenador central (a través de Rx PC) los mensajes que son transmitidos desde el uC o del módulo ZigBee, y así llevar un control de las actividades que se estén llevando a cabo y tomar las acciones oportunas. Como se detallará más adelante, se han implementado funciones para facilitar el control de los modos de operación entre los dispositivos.

II.2 Modificaciones hardware introducidas en la plataforma.

La arquitectura *hardware* propuesta para la implementación del módulo de control UART, junto con la plataforma global, se muestra de forma esquemática en la figura 4.3, en la que se aprecia el trabajo conjunto de los tres niveles de procesamiento de la plataforma de integración HW-SW, y las líneas necesarias para gestionar la comunicación serie de los tres dispositivos en la FPGA. Se han modificado las líneas de conexión UART de cada dispositivo para que estas estén conectadas a la FPGA, y se ha aprovechado las conexiones del uC y el módulo ZigBee con la FPGA para utilizar el módulo de control UART.

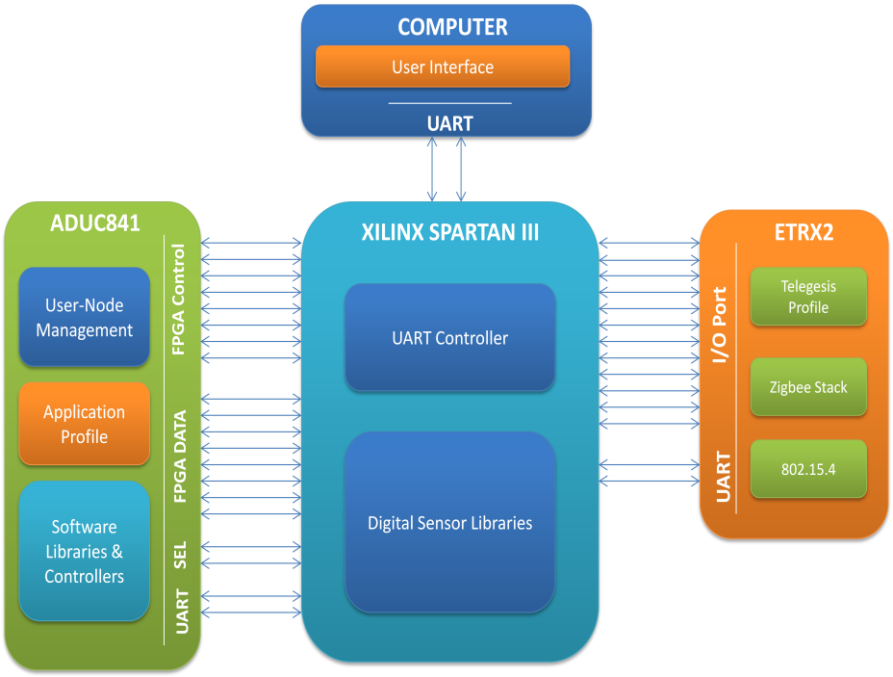


Figura 4.3: Esquema de la arquitectura de integración HW-SW.

La tabla 4.2 incluye las conexiones utilizadas para la comunicación entre todos los dispositivos del sistema conjunto, relacionando cada conexión con los pines físicos de la FPGA:

ADUC841	FPGA	ETRX2	PC
P3.7 (selector 1)	P27 (in)		

P3.6 (Selector 0)	P28 (in)		
TX	P72 (in)		
RX	P71 (out)		
FPGA DATA (7) (P2.7)	P23 (out)		
FPGA DATA (6) (P2.6)	P22 (out)		
FPGA DATA (5) (P2.5)	P21 (out)		
FPGA DATA (4) (P2.4)	P17 (out)		
FPGA DATA (3) (P2.3)	P16 (out)		
FPGA DATA (2) (P2.2)	P15 (out)		
FPGA DATA (1) (P2.1)	P14 (out)		
FPGA DATA (0) (P2.0)	P13 (out)		
FPGA TrigByte1N (P0.7)	P1 (in)		
FPGA ACK1N (P0.6)	P2 (out)		
FPGA TrigByte2N (P0.5)	P4 (in)		
FPGA ACK2N (P0.4)	P5 (out)		
FPGA SelTrigger (1) (P0.3)	P8 (in)		
FPGA SelTrigger (0) (P0.2)	P12 (in)		
	P53 (in)	I/O 0 (Selector 0)	
	P47 (in)	I/O 1 (Selector 1)	
	P64	I/O 2	
	P63	I/O 3	
	P62	I/O 4	
	P61	I/O 5	
	P60	I/O 6	
	P59	I/O 7	

	P80	I/O 8	
	P74	I/O 9	
	P79	I/O 10	
	P50	I/O 11	
	P68 (in)	TX	
	P67 (out)	RX	
	P39 (in)		TX
	P40 (out)		RX

Tabla 4.2: Conexiones entre los dispositivos de la plataforma HW-SW.

Tal y como se presentará más adelante, a parte de los cambios realizados en *hardware* para la implementación del control de la UART ha sido necesario realizar modificaciones adicionales para poder realizar operaciones de forma totalmente remota que, sin dichos cambios, no hubiese sido posible implementar. Los cambios están relacionados con la capacidad de poder realizar un reset de la plataforma inalámbrica así como programar el microcontrolador de forma remota vía módulo ZigBee con el fin de reconfigurar parámetros del sistema a nivel de código fuente cuando se considere oportuno, como se verá en apartados posteriores.

III. Diseño de un perfil de aplicación para la plataforma Cookies.

La necesidad de contar con una plataforma que permita un rápido prototipado para diferentes aplicaciones, así como la adaptabilidad a los requisitos que se planteen en un determinado despliegue de sensores inalámbricos, da lugar al diseño de una vía de estandarización de la manera en la que se comunican y se procesan los datos entre los dispositivos. El primer paso para dicha estandarización se ha llevado a cabo con el diseño de las bibliotecas *software* modulares que permiten contar con un repertorio de funciones para realizar diferentes acciones en la red de sensores, y sobre todo disponer de un completo set de controladores de sensores. Con ello la base está definida, pero el siguiente paso es la creación de ciertas directivas que permitan organizar de manera común la forma en que los nodos transmiten y reciben dicha información. Esto ha dado lugar a la creación de un perfil de aplicación especialmente diseñado para los despliegues tipo en los que se utiliza la plataforma Cookies.

Un perfil de aplicación es una colección de especificaciones y directivas enfocadas a la organización de un sistema para su adecuado funcionamiento, y facilitar el

desarrollo del mismo. Aunque este término es muy extendido y a la vez complejo puesto que incluye muchos conceptos y definiciones, en el caso de la plataforma Cookies se emplea para definir de forma estándar los parámetros que caracterizarán a las aplicaciones basadas en nodos Cookies y los atributos que se pueden utilizar y modificar dentro de dicha aplicación, valiéndose del repertorio de funciones disponibles en las bibliotecas *software*. Como se verá en posteriores apartados, esto conlleva a la creación de nuevas bibliotecas y funciones para facilitar el desarrollo final de una aplicación.

Con el fin de entender de forma clara los objetivos de la creación de un perfil de aplicación propio para la plataforma Cookies, es necesario comprender algunos conceptos claves dentro de este ámbito de estudio.

Por un lado existe el denominado *ZigBee Device Profile*, también conocido simplemente como Perfil del Dispositivo, y que tiene que ver con un repertorio de instrucciones con el fin de gestionar parámetros estándar de configuración y comportamiento de la red, como son *Device Descovery*, que permite determinar los dispositivos que son parte de la red, sus respectivas direcciones y los nodos que están asociados a los mismos. Por otro lado se encuentra el *Service Discovery*, que permite determinar los servicios que ofrecen los dispositivos respecto al perfil de aplicación implementado; *Binding Management*, que permite gestionar los vínculos entre los dispositivos; y el *Network Management*, que proporciona información útil y controles para realizar tareas como asociar o desasociar un nodo de la red, conocer la calidad de los enlaces entre los dispositivos, listas de rutado entre otras opciones [Daintree 2].

Este perfil es parte del estándar ZigBee [ZigBee 1], por lo que los dispositivos comerciales que implementan el protocolo han de proporcionar los comandos necesarios para hacer uso de estas funciones de gestión de la red inalámbrica. Este perfil es implementado mediante el *ZigBee Device Object* (ZDO) y, de hecho, el módulo de comunicaciones utilizado en la plataforma Cookies, el ETRX2, permite acceder a las funciones del perfil.

Muchas de estas características, como se verá en apartados posteriores, son utilizadas en la plataforma final implementada con el fin de proporcionar información de los caminos de rutado de datos, el enlace que se establece entre los nodos Cookies, la calidad de las comunicaciones, conexión y reconexión de los nodos inalámbricos, y la determinación de parámetros de red como direcciones físicas y los roles de los dispositivos dentro de la red ZigBee.

Por otro lado, *ZigBee Alliance* ha definido una serie de perfiles de aplicación enfocados principalmente en la interoperabilidad entre los dispositivos de diferentes fabricantes, facilitar el desarrollo de aplicaciones y estandarizar la forma en la que se

comunican los nodos dentro de una red inalámbrica. Entre los principales perfiles de aplicación definidos se encuentran el *ZigBee Home Automation*, el *ZigBee Smart Energy* y el *ZigBee Health Care.* Estos perfiles de aplicación se basan en un amplio repertorio de atributos que definen a la aplicación y las funcionalidades que se pueden utilizar, y que el estándar ZigBee Alliance los agrupa en lo que denominan *ZigBee Cluster Library* (ZCL) [ZigBee 2].

Si bien es cierto que dichos perfiles de aplicación han sido concebidos para proporcionar una herramienta que facilite el desarrollo de aplicaciones con esta tecnología y sobre todo enfocado en la interoperabilidad, en la actualidad no todos los dispositivos ZigBee comerciales están preparados para soportar dichos perfiles de aplicación. Esto es palpable, sobre todo, en los módulos de bajo coste, en los cuales los fabricantes implementan un perfil privado de aplicación que no permiten configurar los parámetros necesarios para conseguir la interoperabilidad con otros fabricantes.

Además, la necesidad de contar con un mecanismo sencillo pero eficiente para el intercambio de información, que sea capaz de adaptarse a los escenarios de aplicación de las redes basadas en nodos Cookies, ha dado lugar a la creación de un perfil que sirve de estándar dentro de la plataforma para los prototipos que se realicen con esta tecnología, facilitando el trabajo de desarrollo, pruebas e implementaciones finales, siempre enfocado en la flexibilidad y el prototipado rápido.

Para el diseño de dicho perfil de aplicación propio de la plataforma Cookies se ha partido de requisitos mínimos de funcionamiento. En primer lugar, la especificación ha de estar enfocada a los parámetros que comúnmente se tienen en cuenta dentro de una red basada en nodos Cookies (que surgen como consecuencia de las aplicaciones reales en las que ha sido probada la red, donde se han definido las necesidades que deben ser cubiertas por la plataforma), que son principalmente los que se desglosan a continuación:

- Identificación y clasificación de los dispositivos.
- Captura de medidas físicas de sensores.
- Consumo de energía de los nodos inalámbricos.
- Calidad del enlace entre los mismos.
- Mapa de conexión entre los dispositivos y ruta de datos.
- Configuración de los dispositivos y la red inalámbrica en sí misma.

Por otro lado, el diseño debe estar basado en la sencillez y efectividad de la plataforma, así como en la escalabilidad y flexibilidad para poder incluir nuevas características en desarrollos futuros. Del mismo modo, una de las principales funciones es lograr una abstracción de nivel que con sólo unos simples parámetros de configuración y sin necesidad de conocer todos los detalles de la plataforma, se logren obtener desarrollos adaptados a las necesidades de aplicación.

Este perfil se apoyará en el nivel de aplicación implementado en el módulo de comunicaciones, tal y como se ilustra en la figura 4.4, tanto de las funciones disponibles del ZDO, como también del perfil privado del mismo. Este último caso se utilizará principalmente la transmisión y recepción de mensajes CAST a través de la red inalámbrica, lo cual esta implementado en cualquier dispositivo ZigBee. Esto permitirá implementar el mismo perfil en el caso de hacer uso de otro módulo ZigBee de diferente fabricante.

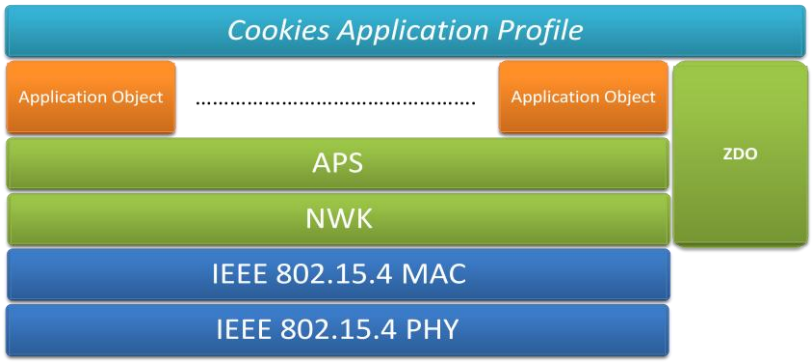


Figura 4.4: Implementación del perfil de aplicación a partir del estándar ZigBee.

Según los requisitos de diseño enumerados anteriormente, se ha decidido crear tres niveles funcionales que serán la base del perfil Cookies, los cuales se detallan a continuación:

- *Gestión de Datos*: Permitirá enviar y recibir información relacionada con los sensores de los nodos inalámbricos y realizar adquisiciones periódicas de datos.

- *Gestión del Nodo*: permitirá enviar y recibir información relacionada con la plataforma *hardware*, como consumo del nodo y realizar la programación de la plataforma *software*.

- *Gestión de la red*: Permitirá enviar y recibir información relacionada con la red inalámbrica, como identificación de dispositivos, enlace entre los nodos,

mapa de red, entre otros aspectos.

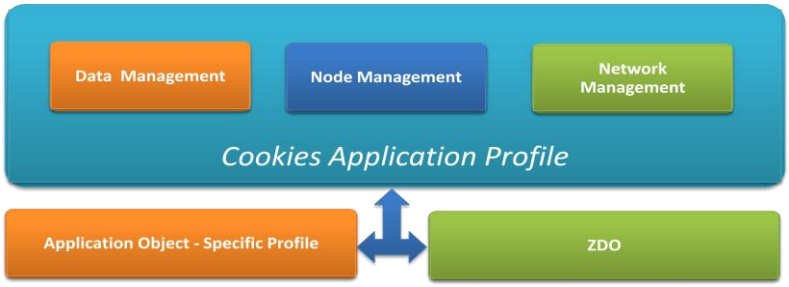


Figura 4.5: Arquitectura del perfil de aplicación Cookie.

III.1 Formato de mensajes del Perfil de aplicación.

El perfil creado para las aplicaciones de la plataforma Cookies se basa en una trama de datos contenida dentro de los mensajes CAST que se envían y reciben en la red inalámbrica ZigBee. Esta trama define las acciones a realizar según los tres niveles funcionales definidos y está compuesta por una cabecera de mensaje, seguida por los datos relacionados a dicha cabecera. El formato y la longitud de los datos dependerán directamente del tipo de acción contenida en la cabecera de mensaje. Por lo tanto, la cabecera de mensaje es la base para lograr el intercambio de información de forma organizada y estandarizada entre los dispositivos de la red. El formato definido de la cabecera de mensaje se detalla a continuación.

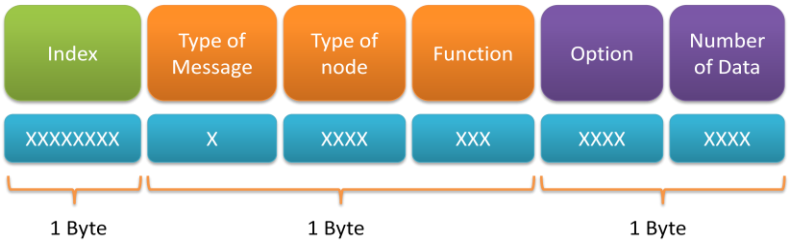


Figura 4.6: Cabecera de mensajes de la plataforma Cookies.

Tal y como se ilustra mediante la figura 4.6 la cabecera de datos está compuesta por 3 bytes. El significado de cada componente de la trama viene definido de la siguiente forma:

- *Index (1 byte):* Identificador de nodo a nivel de aplicación.
- *Tipo de mensaje (1 Bit):* El mensaje que se recibe puede ser o bien de petición de información (1), o transmisión de información (0).

- *Tipo de nodo (4 bits):* Indica el tipo de dispositivo dentro de la red inalámbrica.
- *Función (3 bits):* Indica el tipo de función, es decir **Gestión de Datos**, **Gestión de Nodo** o **Gestión de Red**.
- *Opción (4 bits):* Indica el tipo de acción a realizar dependiendo de la función seleccionada.
- *Número de datos (4 bits):* datos a recibir en función del tipo de nodo y la acción seleccionada.

Este formato cumple con los requisitos y el alcance que tienen actualmente las aplicaciones de la plataforma Cookies, y todos los componentes han sido dimensionados para que los desarrollos futuros puedan ser incluidos en esta trama de datos. Cabe destacar que el formato es totalmente flexible a modificaciones en tamaño y forma en el caso de que sean necesarias más opciones dentro de las aplicaciones, siempre y cuando se mantengan los parámetros básicos que definen a la plataforma.

Las características y detalles de cada componente de la trama de datos se definen a continuación.

III.1.1 Index y tipos de nodos.

El *Index* es un elemento muy importante dentro de la gestión de la información del perfil implementado, puesto que representa el identificador del nodo a nivel de aplicación y facilita el procesamiento de acciones. Según está definida la trama de datos, el *Index* puede ser un valor de 0 a 255 (8 bits), aunque el valor 0 está reservado para el nodo coordinador. Este identificador es asignado en cada nodo en el momento de realizar la programación de la aplicación en el dispositivo, y cada nodo debe tener uno propio.

Por otro lado se encuentran los tipos de nodos. Este campo define todos los posibles tipos de dispositivos que pueden estar dentro de una aplicación Cookies, y que se enumeran en la tabla 4.3. Aunque en el formato están definidos 4 bits (16 diferentes tipos de nodos), este campo puede ser ampliado en desarrollos futuros. La relación entre el tipo de nodo y el número de datos esta en todos los tipos de sensores que posee un determinado nodo y por tanto todas las posibles medidas que puede enviar. Según la definición del número de datos, un nodo puede tener hasta cuatro sensores diferentes (cada sensor se asocia a uno de los 4 bits del campo de Número de datos).

Valor	Tipo de Nodo	Número de Sensores
0000	Coordinador.	--
0001	Temp °C, Hum %H & LDR.	**3** (Temperatura, humedad y Luz).
0010	ACC X-Y.	**2** (Aceleración ejes X e Y)
0011	Strain Gauge.	**1** (Medida de Deformación).
0100	Router.	--
0101	Temp °C, PH.	**2** (Temperatura y nivel de PH).
0110	CO.	**3** (CO, CO2, NO).
0111 – 1111	A definir.	A definir.

Tabla 4.3: Tipos de nodos soportados actualmente por la plataforma Cookies.

Es necesario destacar que se asigna como tipo de nodo un dispositivo *router* en los casos en los que éste sólo es posicionado en la red inalámbrica para el rutado de datos, aunque también es posible que un nodo sensor funcione como *router* de datos. En ese caso el campo de Tipo de Nodo estaría relacionado con las medidas que éste realiza y no con el rutado de datos.

III.1.2 Función Gestión de Datos.

Esta función (valor 001) define las acciones que están relacionadas con la adquisición de medidas físicas de los sensores. El tipo de acción se determina a través del campo *Option*, y actualmente se pueden realizar las que se indican en la tabla 4.4.

Función	Opción	Tipo de acción
001 (Gestión de Datos)	0001	Adquisición Individual.
	0010	Adquisición Periódica.
	0011	Umbral.
	0100 - 1111	A definir.

Tabla 4.4: Tipos de acciones dentro de la gestión de datos.

Uno de los aspectos más interesantes de la gestión de datos es la capacidad de realizar la adquisición de cualquier medida de un sensor de forma particular o

combinación de ellas. Esta capacidad se puede ilustrar tomando como ejemplo el nodo sensor de temperatura, humedad y luz. Este sensor posee tres medidas, por lo que el campo de Número de Datos queda definido como se muestra en la tabla 4.5.

Tipos de Sensores	
0001	Temperatura.
0010	Humedad.
0100	Luz.

Tabla 4.5: Sensores definidos en el nodo 0001 (Temp, Hum & LDR).

Esto da lugar a la siguiente combinación de posibles de datos a adquirir en dicho nodo:

Tipos de Adquisición	
0001	Temperatura.
0010	Humedad.
0011	Temperatura y Humedad.
0100	Luz.
0101	Temperatura y Luz.
0110	Humedad y Luz.
0111	Temperatura, humedad y luz.

Tabla 4.6: Combinación de datos a adquirir del nodo de Temp, Hum & LDR.

En el caso de enviar, por ejemplo, la medida de temperatura y humedad, el orden de los datos que siguen a la cabecera sería primero el valor de la temperatura (2 bytes) y luego la humedad (2 Bytes), tal y como se definen en la tabla 4.6. Los tres tipos de adquisición de datos definidas se detallan a continuación:

Adquisición Individual: Permite obtener de forma puntual la medida de los sensores seleccionados, es decir, sólo cuando se realiza la petición de la misma desde el coordinador. En el ejemplo del sensor de temperatura, humedad y luz, la trama de datos quedaría definida tal y como se muestra en la tabla 4.7.

Index	Tipo de Mensaje	Tipo de Nodo	Función	Opción	Número de datos	Datos
1 byte	0	0001	001	0001	0011	4 Bytes

Tabla 4.7: Ejemplo de aplicación de la trama de datos con la acción Adquisición individual.

Adquisición Periódica: Esta funcionalidad permite obtener de forma periódica la medida de los sensores seleccionados, especificando para ello el valor del periodo a configurar. Este valor va expresado en un byte y la unidad de tiempo es segundos, por lo que se puede especificar un periodo de tiempo desde 1 segundo hasta 255 segundos. En el ejemplo del sensor de temperatura, humedad y luz, la trama de datos en el caso de solicitar datos periódicos quedaría definida tal y como se muestra en la tabla 4.8.

Index	Tipo de Mensaje	Tipo de Nodo	Función	Opción	Número de datos	Datos
1 byte	1	0001	001	0010	0011	1 Byte

Tabla 4.8: Ejemplo de aplicación de la trama de datos con la acción Adquisición periódica.

Valor de Umbral: Esta funcionalidad permite configurar un dispositivo para que realice adquisiciones de datos de los sensores de forma activa con el fin de monitorizar las variaciones de las magnitudes físicas, y así activar una alerta en el caso de detectar algún valor fuera de rango. El nodo realizará adquisiciones de datos de forma activa, con un periodo de 2 segundos por cada adquisición, procesará los datos de forma local y los comparará con el valor de umbral especificado. Los valores que deben ser configurados, por tanto, son el valor de umbral (2 bytes por medida) y el signo de la comparación (positivo [+] o negativo [-], un byte). El signo de comparación indica si la alerta se generará cuando el valor de umbral es sobrepasado por encima o por debajo.

Esta es una de las características más interesantes a nivel de sensado puesto que se puede configurar más de un umbral a la vez, es decir, se pueden asignar valores de umbrales a todos los sensores de un nodo o combinación de los mismos, para lo cual habría que especificar en la trama de datos 3 bytes por umbral (dos bytes de datos y uno de signo). En el ejemplo de la adquisición de medidas de temperatura y humedad, se podría configurar alertas para temperaturas menores a 25ºC, y humedad relativa mayor al 50%. En dichos casos el campo de datos quedaría expresado tal y como se muestra en la tabla 4.9.

Signo Temp. (-)	Umbral Temp. (25,00 ºC).		Signo Hum. (+)	Umbral Hum. (50,00 %).	
00101101	00001001	11000100	00101011	00010011	10001000

Tabla 4.9: Ejemplo de aplicación de la trama de datos con la acción Valor de umbral.

Al configurar el nodo con los sensores que se quieren monitorizar, este evaluará las medidas configuradas y se enviará de forma periódica la medida de dicho sensor al nodo que solicita la información en caso se sobrepasarse un umbral. Se pueden recibir al mismo tiempo todas las medidas que sobrepasen su respectivo umbral o combinación de estas, según sea el caso.

III.1.3 Función Gestión de Nodo.

Esta función, cuyo valor es 010, define las acciones que están relacionadas con la configuración y control del nodo inalámbrico, enfocadas principalmente a tareas de nivel *hardware*. El tipo de acción se determina a través del campo *Option*, y actualmente se pueden realizar las que se indican en la tabla 4.10.

Función	Opción	Tipo de acción
010 (Gestión de Nodo)	0001	Reset remoto de la aplicación.
	0010	Reprogramación remota del nodo.
	0011	Consumo de corriente del nodo.
	0100	Puesta en bajo consumo del módulo ZigBee.
	0101 - 1111	A definir.

Tabla 4.10: Tipos de acciones dentro de la gestión del nodo.

Reset Remoto de la aplicación: Esta funcionalidad permite realizar de forma remota un reset del núcleo central de procesamiento, en este caso el microcontrolador, y así reiniciar la aplicación en el nodo especificado. Tal y como se describirá en el apartado de las nuevas bibliotecas que dan soporte al perfil de aplicación, existe una función encargada de realizar las acciones necesarias para obtener el reset remoto,

para lo que también ha sido necesario añadir algunos ajustes a nivel *hardware* para que dicha función sea posible realizarla, como se verá más adelante.

Reprogramación Remota del nodo: Esta funcionalidad es una de las más importantes desde el punto de vista de la puesta en marcha y depuración dentro de la red inalámbrica, puesto que permite reprogramar el microcontrolador que contiene el perfil de aplicación creado. El principal potencial de esta funcionalidad es que la programación se realiza de forma inalámbrica a través de la red ZigBee, lo que supone un desafío a nivel de diseño y ejecución de dicha tarea. En el caso del *hardware* utilizado en la plataforma, existen diversas peculiaridades y detalles técnicos que hay que tomar en cuenta con el fin de poder realizar una efectiva programación remota del nodo inalámbrico. Para ello, y como se comentará en el apartado de las bibliotecas soporte del perfil de aplicación, se ha creado una función que realiza las configuraciones necesarias para realizar la tarea, y se han tenido que añadir modificaciones a nivel *hardware* con el fin de lograr el correcto funcionamiento de la misma.

Consumo de corriente del nodo: Esta funcionalidad permite monitorizar el consumo de corriente que está teniendo un nodo dentro de la red inalámbrica. En este caso, se utiliza el campo de Número de Datos para especificar el tipo de adquisición que se quiere realizar, como se muestra en la tabla 4.11.

Opción	Acción	Tipo de acción
0011 (Consumo de Nodo)	001	Adquisición Individual.
	010	Adquisición Periódica.
	011	Umbral.

Tabla 4.11: Tipos de adquisición dentro de la opción de consumo de corriente.

Tal y como se observa en la tabla 4.11, para la monitorización del consumo de corriente se pueden realizar las mismas tres adquisiciones que se realizan en el caso de los sensores de la plataforma. Sobre todo resulta muy útil la opción de Valor de Umbral puesto que permitirá detectar anomalías en el consumo de corriente de la plataforma inalámbrica. En la Adquisición Periódica es necesario especificar el periodo de muestreo de la medida de corriente, mientras que en Umbral se debe especificar tanto el signo como el valor de umbral a configurar.

Puesta en bajo consumo del módulo ZigBee: Esta funcionalidad permite configurar el módulo ZigBee en modo de bajo consumo, con el fin de alcanzar un ahorro en el consumo de la energía proveniente de las baterías, cuando el nodo inalámbrico se encuentra inactivo. Tal y como se puede obtener de [Telegesis 2], el módulo ETRX2

posee tres modos de operación de energía. El modo 3, que es con el que se alcanzan menores niveles de consumo, es el utilizado en esta funcionalidad. La tarea a realizar se basa en cambiar el modo de operación del módulo desde totalmente despierto a totalmente dormido, en el pasa a consumir 0.7 uA. Ya que en éste último modo el módulo no es capaz de atender la comunicación serie, la única forma de despertarlo es mediante una interrupción externa que se generará desde el microcontrolador. Para ello, en el momento de configurar la tarea de puesta en bajo consumo, es necesario especificar el tiempo en el que el módulo estará dormido antes de volver al modo despierto. En este caso se utiliza el campo de Número de Datos para especificar la base de tiempo del periodo de interrupción para despertar al módulo, como se muestra en la tabla 4.12:

Opción	Acción	Base de tiempo
0100 **(Puesta en bajo consumo)**	001	Segundos.
	010	Minutos.
	011	Horas.

Tabla 4.12: Configuración dentro de la opción de bajo consumo del módulo ZigBee.

Tal y como se aprecia en la tabla 4.12, el periodo en el que el módulo puede estar dormido va desde unos pocos segundos, hasta horas, lo cual es útil en los casos en los cuales existen nodos que son desplegados para tomar medidas en un determinado momento y permanecen inactivos por largos periodos de tiempo. Más adelante se comentará el proceso de configuración del modo de bajo consumo, y los ajustes necesarios para tal fin.

III.1.4 Función Gestión de Red.

Esta función (valor 011) define las acciones que están relacionadas con la red inalámbrica ZigBee, identificación y gestión de los dispositivos, conexión de los nodos, ruta de datos, entre otros parámetros. El tipo de acción se determina, al igual que en los casos anteriores, a través del campo *Option*, y actualmente se pueden realizar las que se indican en la tabla 4.13.

Función	Opción	Tipo de acción
011 **(Gestión de Nodo)**	0001	Identificación de Nodo.
	0010	Relación de *Index* con direcciones cortas y largas.

	0011	Tabla de enlace entre nodos.
	0100	Mapa de conexión entre nodos.
	0101	Relación entre *Index* y tipos de nodos.
	0110	Número de reconexiones de los nodos.
	0111 - 1111	A definir.

Tabla 4.13: Tipos de acciones dentro de la gestión de Red.

Es en este bloque funcional donde cobra mayor importancia la definición del *Index* puesto que es el valor que permite enlazar de forma rápida y efectiva diferentes atributos de los dispositivos y de la red inalámbrica en sí misma. Es el identificador de cada nodo a nivel de aplicación y la puerta para conocer diferentes detalles como ruta de datos, reconexiones, calidad de la red, enlace entre nodos, entre otras características. El *index* es una herramienta clave en la interfaz de usuario, puesto que, como se verá más adelante, es el principal parámetro a introducir por el usuario de la aplicación desde el ordenador central, para poder conocer cualquier atributo de un dispositivo en concreto.

La mayoría de las acciones que forman parte de la Gestión de Red están enfocadas para que sean utilizadas desde el ordenador central de monitorización y control, por lo tanto están definidas principalmente entre la interfaz de usuario y el nodo coordinador, como se verá en dicho apartado.

Cada dispositivo al conectarse a una red basada en la plataforma Cookies envía un mensaje de identificación con su correspondiente valor de *Index* al coordinador, el cual se encarga de guardar los parámetros característicos del mismo. En términos generales, a partir del *Index* se crea una base de datos completa en la que, para cada nodo perteneciente a la red, se guardan los valores de direcciones PAN, descripción del nodo, tipo de sensores que posee el nodo y número de reconexiones que realiza a la red inalámbrica. A partir de esta base de datos se extraerán los parámetros característicos del nivel de red: enlace entre los dispositivos conectados a la red, calidad de los enlaces, ruta de datos, y creación de un mapa de conexión. Si se tuviese por ejemplo una red inalámbrica con 5 dispositivos (1 coordinador, 1 *Router*, 2 nodos de temperatura, humedad y luz, y 1 nodo de aceleración en ejes X e Y), los datos almacenados relacionados con la aplicación y de los que se obtendrían los parámetros de red característicos, serían de forma esquemática como se muestran en la tabla 4.14.

Index	Dirección MAC	Dirección corta	Tipo de Nodo	Descriptor	Sensores	Reconexiones
00	000D6F0000512C97	F714	0000	Coordinador	0	0
01	000D6F0000F7C64	01D8	0100	Router	0	2
02	000D6F000042FC4	2189	0001	Temp ºC, Hum %, LDR	3	1
03	000D6F000084FC3	D841	0001	Temp ºC, Hum %, LDR	3	4
04	000D6F00008CF9	C87F	0010	Acc X, Y	2	3

Tabla 4.14: Valores característicos de nodos desplegados en una red Cookies.

Hasta ahora se han descrito las características que componen el perfil creado para el prototipado de aplicaciones basadas en nodos Cookies. Pero es necesario destacar que para su implementación ha sido necesario no sólo contar con las bibliotecas creadas y descritas hasta el momento para dar soporte a la plataforma *hardware*, sino que ha sido necesario crear nuevas bibliotecas que dieran total soporte a todos los bloques de gestión que componen el perfil diseñado. A continuación se detallan dichas bibliotecas y los bloques funcionales que forman parte de esta implementación, así como las modificaciones adicionales a nivel *hardware* que han sido imprescindibles para poder lograr ciertas funcionalidades importantes para la plataforma inalámbrica.

IV. Bibliotecas software soporte del perfil de aplicación.

En el capítulo 3 se han detallado las cuatro bibliotecas base para el control de todos los periféricos involucrados en la plataforma *hardware*, la captura de medidas físicas de sensores y la gestión de las comunicaciones inalámbricas. Sin embargo, el diseño de un perfil de aplicación propio para realizar cada una de las tareas descritas anteriormente ha supuesto la creación de nuevos bloques funcionales que permitan un nivel superior de abstracción con el fin de contar con un sistema global que se encargue de administrar todas las funcionalidades diseñadas dentro de dicho perfil de aplicación, así como disponer de plantillas de diseño destinadas a facilitar a los desarrolladores el prototipado rápido de aplicaciones sobre la plataforma.

En tal sentido, se han añadido 5 nuevas bibliotecas al repertorio de funcionalidades, 3 de ellas respaldan cada nivel del perfil, una como plantilla de edición de nodos, y una que contiene la descripción a más alto nivel de abstracción del perfil y que hace uso de todas las demás bibliotecas., tal y como se desglosan a continuación:

- *_CEI_tb_data*: Se encarga de funcionalidades relacionadas con la gestión de

datos del perfil de aplicación y tareas relacionadas con el tipo de adquisición de los sensores.

- *_CEI_tb_node*: Se encarga de las funcionalidades relacionadas con la gestión del nodo, y contiene importantes bloques funcionales como la gestión de la programación remota.

- *_CEI_tb_network:* Se encarga de las funcionalidades relacionadas con la gestión de la red inalámbrica desde el nivel de aplicación.
- *_CEI_tb_templates:* Esta biblioteca permite al usuario definir nuevos tipos de nodo desde el punto de vista de los sensores que posee, descripción del nodo y el tipo de medidas a adquirir.
- *_CEI_tb_global:* Contiene la descripción global del sistema conjunto, bloques funcionales relacionados con el perfil, y la administración de las demás bibliotecas creadas.

A continuación se describen los principales bloques funcionales que contienen dichas bibliotecas, y su aportación para el sistema conjunto. Se detallarán los bloques funcionales más importantes sin entrar en detalle de todas las funciones que componen cada una de las bibliotecas, ya que la filosofía de diseño es la misma que se ha aplicado en el caso de las anteriores bibliotecas.

IV.1 Biblioteca "tb_data"

Como se ha comentado anteriormente, esta biblioteca se encarga de gestionar los tres tipos de adquisición de datos diseñados: Individual, Periódica o Valor de Umbral. Existen funciones tanto para la adquisición de datos de los sensores y el envío por la red inalámbrica, así como la recepción de datos por parte del dispositivo que hace la petición y su correcta representación en el ordenador central, cuando se trata del nodo coordinador. Posee como parámetros de edición los comandos para la comunicación con el ordenador central, tal y como se verá en la descripción del bloque de gestión de la interfaz de usuario. A continuación se detallan los bloques funcionales más importantes desde el punto de vista del tratamiento de los datos de sensores.

sensor_send: Esta funcionalidad se encarga de decodificar el valor de Número de Datos proporcionado por la trama de datos con el fin de que, a partir del valor del Tipo de Nodo, obtener los valores de los sensores involucrados en el preciso orden en el que deben ser enviados, según la relación entre la combinación de valores a enviar y el número de valores, tal y como se ha visto en la descripción de la trama de datos del perfil de aplicación.

Recibe como parámetros de entrada el tipo de nodo y el número de datos a adquirir, y se encarga de obtener las medidas físicas, transformarlas a los respectivos valores estandarizados, enmascararlos y añadirlos a la cola de transmisión de datos para que sean enviados vía inalámbrica. Desde el punto de vista de procesamiento de la información, esta función realiza todas las tareas y pasos necesarios para la adquisición de datos de sensores.

sensor_get: Esta funcionalidad, diseñada para ser utilizada en el nodo coordinador de la red inalámbrica, realiza el procedimiento opuesto de la anterior función, es decir, a partir de la trama de datos recibida obtiene los valores resultantes del nodo sensor, desenmascara dicha información y relaciona el tipo de medidas con los sensores asociados a éstas, con el fin de retransmitir la información de forma adecuada al ordenador central de monitorización. Recibe como parámetros de entrada los valores correspondientes a la trama de datos recibida.

sensor_thhold: esta funcionalidad trabaja de forma similar a *sensor_send,* pero la principal diferencia está en que realiza la comparación de los valores de umbrales configurados y si se sobrepasa alguno de los valores, se envía dicha información al coordinador.

Es el corazón de la funcionalidad Valor de Umbral, ya que realiza todas las comparaciones necesarias según haya sido especificado por el usuario desde el ordenador. Se enviarán todos los datos que se sobrepasen, sea uno, todos o combinación de ellos en función del nodo a que se haga referencia.

Recibe como parámetros de entrada el tipo de nodo y el número de datos a comparar, así como una cadena de caracteres que contiene los valores de umbrales y los signos que han sido configurados por parte del usuario. Se encarga de adquirir las medidas físicas, transformarlas a los respectivos valores estandarizados, enmascararlos y añadirlos a la cola de transmisión de datos para que sean enviados vía inalámbrica, en caso de valores que sobrepasen el umbral.

thhold_request: Esta funcionalidad se encarga de obtener los valores de umbrales especificados desde el ordenador central, y transmitirlos de forma adecuada al nodo sensor, organiza la información según los sensores a configurar, valores de umbrales y signo de los mismos. Del mismo modo, enmascara y los incluye en la trama de datos que será enviada.

currentTime: Esta funcionalidad hace uso del bloque funcional TIC de *peripherals* con el fin de añadir la hora exacta en la que ha sido recibida una o más medidas de sensores por parte de un nodo, con el fin de visualizar dicha información en la interfaz de monitorización. Muestra el tiempo en horas, minutos y segundos (HH:MM:SS), después de los datos de sensores recibidos.

IV.2 Biblioteca "tb_node"

Esta biblioteca contiene tres de las funcionalidades más importantes de cara a la depuración de los nodos que forman parte de la red inalámbrica, las cuales son la capacidad de hacer un reset de cualquier nodo de forma remota; realizar la programación del microcontrolador a través de un enlace inalámbrico; y el control del módulo UART implementado en la FPGA. Ha sido necesario realizar cambios a nivel *hardware* de la plataforma Cookies, con el fin de obtener el correcto funcionamiento de dichos bloques funcionales. A continuación se describen cada una de las funcionalidades y los respectivos cambios realizados.

IV.2.1 Bloque funcional "reset".

Con el fin de realizar un Hard-reset, o reset *hardware* del microcontrolador de forma remota es necesario conocer cómo funciona dicho reset de forma local, es decir, como se realiza el mismo desde el nodo. Se hará la descripción desde el punto de vista del microcontrolador ADUC841, aunque en el caso de aplicarlo a otro microcontrolador habrá que analizar cómo funciona el reset del mismo para realizar los ajustes necesarios.

El reset de este microcontrolador es activo por nivel alto, y con el fin de lograr un correcto reinicio es necesario mantener el pin RESET durante 24 ciclos consecutivos de reloj a dicho nivel. En la placa de procesamiento del nodo se encuentra un pulsador que permite realizar esta acción de forma manual. Para lograr de forma remota esta acción, se ha diseñado un bloque funcional en la FPGA con el fin de generar la señal de reset, por lo que se ha conectado un pin desde la FPGA al pin RESET del microcontrolador, tal y como se muestra en la figura 4.7.

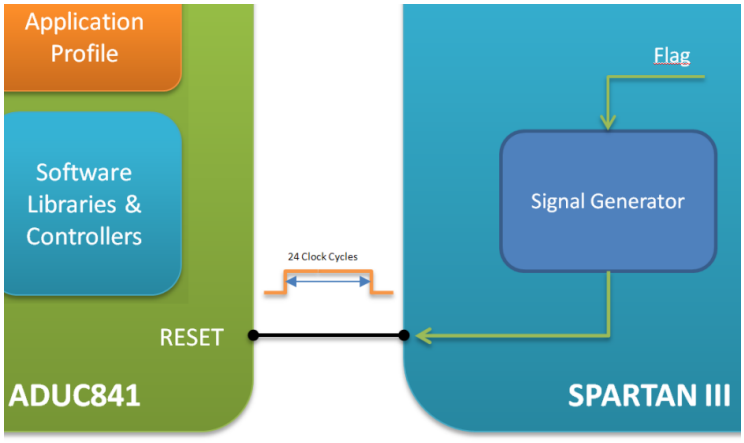


Figura 4.7: Esquema de conexión para la generación del reset remoto.

El bloque generador de reset implementado en la FPGA se basa en una señal de activación proveniente de un pin I/O del módulo de comunicaciones, y un pin de salida conectado al microcontrolador para así ejecutar el reset del mismo. En el diseño de esta funcionalidad se ha decidido realizar la activación del bloque de la FPGA desde el módulo ZigBee con el fin de poder generar la acción o bien desde el nivel de red (directamente mediante comandos de comunicación con el módulo), o desde el nivel de aplicación (funciones diseñadas para tal fin).

IV.2.2 Bloque funcional "programming".

Para realizar la programación remota del ADuC841 se hace necesario conocer en primer lugar cual es el proceso para la puesta en modo de programación del microcontrolador para luego cargar el fichero .hex en la memoria de programa del mismo. Esta puesta en modo de programación se hace mediante la combinación del pin PSEN del microcontrolador y el pin RESET. El procedimiento para llevarla a cabo es la siguiente: el pin PSEN está configurado por defecto como salida, pero en el momento de realizar un reset al microcontrolador este pin se configura como entrada momentáneamente durante el nivel alto de la señal de reset. Si en ese momento PSEN pasa a estar a nivel bajo, el microcontrolador entrará en modo de programación.

Para realizar la operación de forma manual, la capa de procesamiento cuenta con un pulsador conectado al pin PSEN y que permite conectarlo a masa, para así poder generar la secuencia de las señales, la cual se ilustra mediante la figura 4.8.

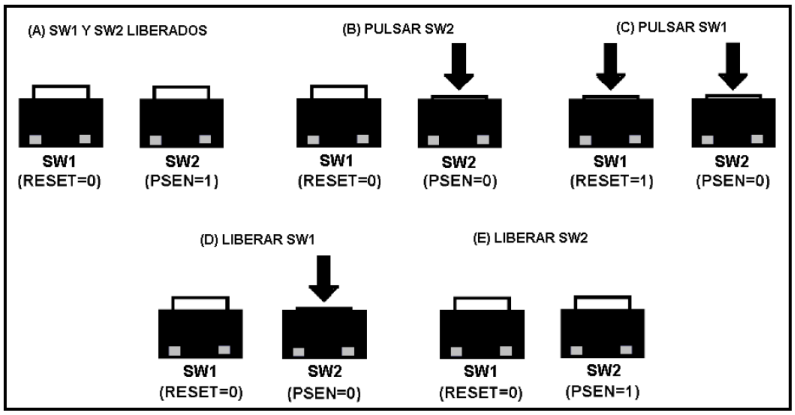


Figura 4.8: Secuencia necesaria para la puesta en modo de programación del microcontrolador.

Para poder realizar esta operación de forma remota ha sido necesario diseñar un bloque dentro de la FPGA que permita generar de forma sincronizada la secuencia de las señales, así como conectar el pin PSEN a una entrada de la FPGA y, junto con la conexión del reset, realizar la acción tal y como se muestra en la figura 4.9. Este

bloque implementado se activa mediante un bit de control que, al igual que en el caso del reset, está conectado a un I/O del módulo ZigBee para así poder realizar la acción directamente desde el mismo, o bien desde el microcontrolador.

Por otra parte, es necesario conocer como se carga el archivo de programación en la memoria del AduC841. Este microcontrolador permite realizar dicha acción a través de la UART, mediante un protocolo específico de transmisión y recepción de datos el cual se encuentra implementado en la herramienta *Windows Serial Downloader (WSD)*. Según se extrae de [Microconv], dicho protocolo plantea varias dificultades a la hora de realizar la programación del microcontrolador puesto que se basa en cambios en la velocidad de transmisión de datos del puerto serie. En primer lugar cambia la configuración del puerto serie del microcontrolador a 9600 bits/s y se realiza un intercambio de caracteres de reconocimiento y configuraciones. Luego de ello se cambia la velocidad a 115200 bits/s con el fin de programar a máxima velocidad para luego volver a 9600 bits/s y finalizar el proceso.

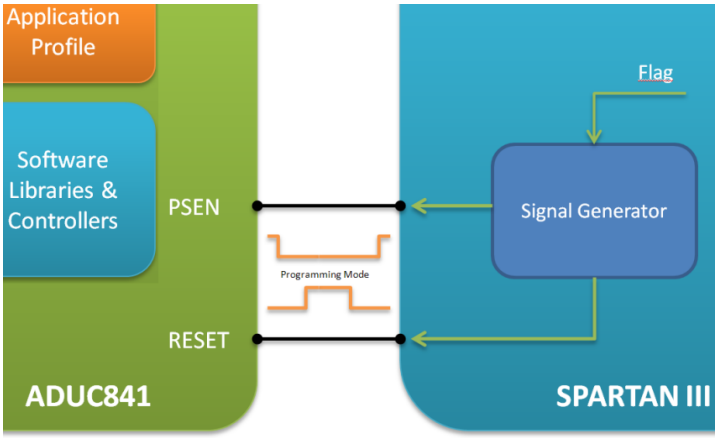


Figura 4.9: Esquema de conexión para la generación del modo de programación.

Con el fin de simplificar el proceso y asegurar la correcta programación del microcontrolador de forma inalámbrica, se ha decidido configurar el WSD para realizar la programación al mismo baudrate durante todo el proceso. Para ello ha sido necesario acceder a los archivos de configuración de la herramienta y cambiar los valores por defecto, tal y como se muestra en la figura 4.10.

```
CodeSize = 63488                        'size of code memory
DataSize = 1024                         'size of the data memory (pages)
QuickDownloadOption = True              'Allow Quick Download Protocol
FastBaudrateOption = False              'Allow change to Fast Baudrate
SecurityBitsOption = True               'Allow Download to Security Bits
BootloadOption = True                   'Allow Bootload Mode
VerifyAfterDownloadOption = True        'Allow Verify Code Command
T3CON = 000                             'baudrate config for 115200 (if allowed)
T3FD = 000                              'baudrate config for 115200 (if allowed)
T3CON9600 = 133                         'baudrate config for 9600
T3FD9600 = 008                          'baudrate config for 9600
XTAL = 2                                'default = 11.0592MHz crystal
fcore = 11059200                        'default
ClkAdjust = 2                           'For checking of max. kernel bauds
```

Figura 4.10: Parámetros de configuración interna del Windows Serial Downloader.

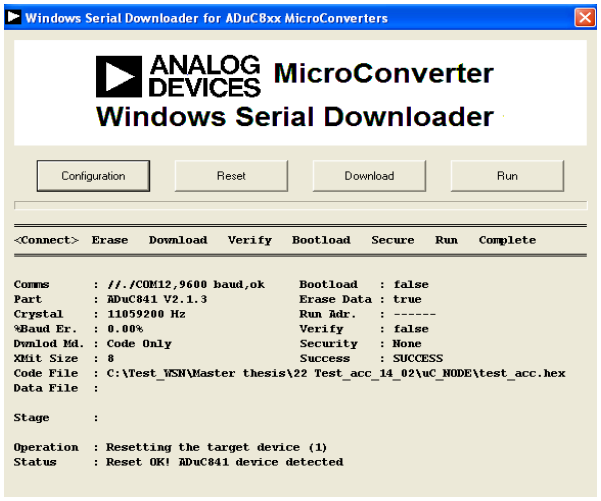


Figura 4.11: Interfaz de la herramienta usada para la programación del microcontrolador.

El siguiente paso es realizar la programación a través del módulo de comunicaciones de forma inalámbrica, desde el nodo coordinador hasta el nodo final. En el caso del módulo de comunicaciones ETRX2 de Telegesis, se cuenta con una característica muy útil la cual permite establecer un link directo entre dos dispositivos, generando un camino virtual de conexión entre las UART de los módulos. Dicha funcionalidad se lleva a cabo mediante el denominado *Data Mode* del módulo, en el que todos los datos que son transmitidos desde un punto (coordinador) son vistos desde el otro lado (nodo receptor) como si se tratase de una comunicación serie cableada.

Esta característica da como resultado el establecimiento de un enlace directo entre el ordenador central y el nodo que va a ser programado. La herramienta WSD ve al nodo final mediante el puerto de comunicaciones COM en el que está conectado el coordinador, y efectúa la programación de la misma forma como si estuviera conectado físicamente al ordenador.

Con el fin de realizar el proceso de programación de forma automática, se ha implementado el bloque funcional *programming* en el microcontrolador, el cual permite realizar todas las configuraciones necesarias para tal fin, desde la puesta en modo de programación, hasta la carga del programa fuente. El proceso que se lleva a cabo se ilustra de forma esquemática mediante la figura 4.12. Cabe destacar que, para que el proceso funcione correctamente, es necesario realizar ajustes de configuración tanto en el nodo coordinador como en el nodo final a programar, sobre todo por el hecho de que la plataforma trabaja a 19200 *bits/seg*, por lo que es necesario realizar el cambio en ambos dispositivos a 9600 *bits/seg*, realizar el proceso de programación y rehacer los cambios de configuración.

Figura 4.12: Pasos a realizar en el proceso de programación remota del microcontrolador.

Los pasos que están marcados en color verde son aquellos que se realizan directamente desde el ordenador central, mientras que los demás se realizan o bien desde el coordinador, o en el nodo final. En este punto es necesario destacar que existen dos posibilidades en la ejecución de dichas acciones. Una primera de forma indirecta, en la que el nodo coordinador le indica a nivel de aplicación al nodo inalámbrico que realice las configuraciones necesarias en su módulo ZigBee para

realizar la programación (cambio en el *baudrate* y puesta en modo de programación), o bien mediante la utilización de la configuración remota del módulo ZigBee (función ATREMS del módulo de comunicaciones, que permite realizar cambios en los registros de un módulo de forma remota a través de otro), en la que desde el microcontrolador del coordinador se configuran los registros del módulo ZigBee remoto. Ambas opciones son posibles y se ha comprobado el correcto funcionamiento en ambos casos, aunque la opción implementada ha sido mediante la utilización de la función ATREMS.

IV.2.3 Bloque funcional "UART_Mode".

Este bloque funcional se encarga de configurar el tipo de conexión UART, según las opciones detalladas en el apartado referido al controlador UART. Por lo tanto, el bloque recibe como parámetro de entrada el tipo de conexión (0, 1, 2 o 3), y realiza el cambio según el caso.

IV.2.4 Cambios hardware implementados para el bloque "programming".

Aparte de los cambios descritos anteriormente para poder realizar las acciones controladas por el bloque funcional *UART_Mode*, se han realizado ajustes adicionales en el conexionado para la implementación del reset y la programación remota. En la tabla 4.15 se muestran los pines incluidos, los cuales se añaden a la lista de conexiones de la plataforma de integración.

ADUC841	FPGA	ETRX2
RESET	P38 (out)	
PSEN	P75 (out)	
	P63 (in)	Reset Flag (I/O 3)
	P64 (in)	Programming Flag (I/O 2)

Tabla 4.15: Modificaciones adicionales realizadas en el conexionado de los dispositivos.

Con estos cambios y junto con las implementaciones llevadas a cabo en la FPGA, la estructura global de la plataforma se muestra en la figura 4.13.

IV.3 Biblioteca "tb_network".

Esta biblioteca define las características de la gestión de la red inalámbrica desde el punto de vista de la aplicación, genera las tablas de datos para almacenar la información relacionada con cada nodo, y haciendo uso de los *index* permite abstraer

al usuario del nivel de red. Junto con la biblioteca *ZigBee,* es la biblioteca más importante de cara a la utilización y aprovechamiento de los recursos más relevantes de la red inalámbrica, de tal forma que el usuario desde el ordenador central pueda monitorizar la actividad de los nodos a nivel de comunicación, como por ejemplo calidad de los enlaces, número de reconexiones, mapa de la red inalámbrica, entre otros aspectos.

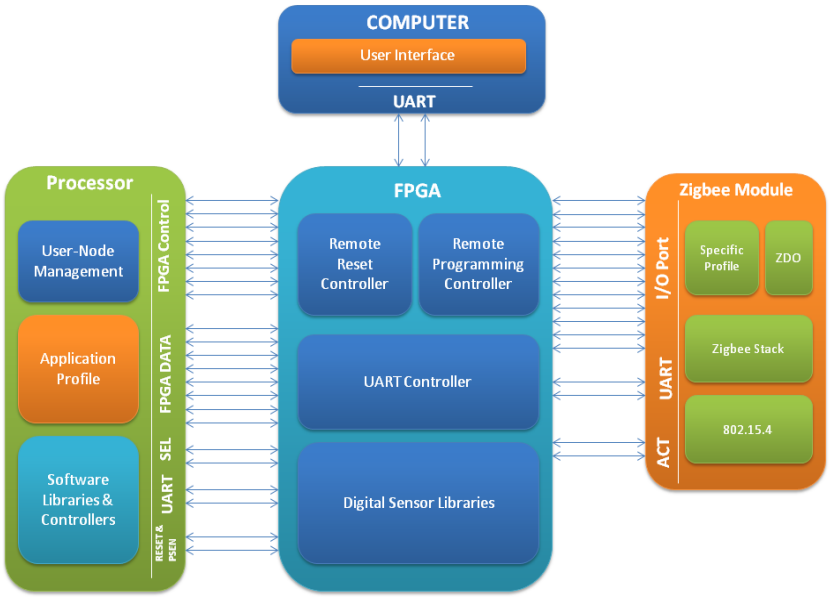


Figura 4.13: Plataforma global de integración HW-SW.

La tabla 4.16 muestra los parámetros característicos de la biblioteca *network* que pueden ser modificados y ajustados según la aplicación, a través del archivo de interfaz con los niveles superiores (*_CEI_tb_network.h)*:

PARAMETRO	DESCRIPCIÓN
T_NODES	Tabla que contiene la descripción de los tipos de nodos.
ID_NODES	Tabla que contiene las direcciones MAC de los nodos según su *index*.
SHORTID_NODES	Tabla que contiene las direcciones cortas de los nodos según su *index*.
TYPE_INDEX	Tabla que enlaza el tipo de nodo con

	su descriptor, según el *index*.
N_REC	Tabla que registra el número de reconexiones según el *index*.
N_NODES	Parámetro que define el máximo número de nodos.
MAX_TYPES	Parámetro que define el número máximo de tipos de dispositivos.

Tabla 4.16: Parámetros modificables de la biblioteca _CEI_tb_network.

Los bloques funcionales más importantes de esta biblioteca se detallan a continuación:

nodeNetwork: Permite registrar la conexión de un nodo inalámbrico a la red ZigBee. Cuando un nodo se conecta a la red, el nodo coordinador recibe un mensaje de identificación que contiene las direcciones físicas del dispositivo, el *index* y el tipo de nodo. Las características más importantes se detallan a continuación:

- El *index* permite ubicar la posición adecuada en las tablas de datos para almacenar las direcciones físicas, y enlazar el tipo de nodo con el descriptor del mismo. Al producirse una nueva conexión, se envía al ordenador central un mensaje de nuevo dispositivo, mostrando el *index* correspondiente, el descriptor y la hora en la que se ha producido la conexión.

- Esta funcionalidad también permite comprobar si se trata de una conexión nueva o si ha sido un nodo perteneciente a la red que se ha reconectado, en cuyo caso se mostraría por pantalla el *index* del nodo, y el número de reconexiones actuales. Por lo tanto, este bloque también controla la tabla de reconexión de los dispositivos.

- Recibe como parámetros de entrada la trama de datos recibida desde el dispositivo final.

coo_adress: Esta funcionalidad permite guardar al inicio de la aplicación los datos relativos al nodo coordinador, direcciones corta y larga, y asignar el *index* correspondiente (en este caso 0000 al ser coordinador).

AdressTable: Permite visualizar en el ordenador central la tabla completa de nodos conectados a la red inalámbrica, con su respectivo *index*, dirección MAC y dirección corta.

QualityTable: Este bloque permite conocer la calidad del enlace entre un dispositivo determinado y sus nodos vecinos, para lo cual hace uso de la funcionalidad del ZDO del módulo ZigBee para rastrear todos los nodos conectados a la red inalámbrica y con los cuales se tienen conexión directa, y se evalúa la conexión mediante el parámetro LQI. A partir de ello, la función permite mostrar una tabla que relaciona el *index* de los dispositivos con la calidad del enlace. Recibe como parámetro de entrada el *index* del dispositivo del que se quiere conocer la tabla de enlace.

NodeMap: Esta funcionalidad realiza una recopilación del camino de los datos dentro de la red inalámbrica con el fin de construir un mapa de la conexión entre todos los dispositivos pertenecientes a la misma. Hace uso de las rutas de datos que se pueden extraer de los parámetros ZigBee con el fin de crear una tabla estructurada en base al *index* de los dispositivos e indicando el número de saltos entre los dispositivos que tiene que dar un mensaje hasta llegar al nodo coordinador.

IndexDescriptor & NodeRec: bloques funcionales que permiten enviar al nodo coordinador la descripción de los nodos y el número de reconexiones realizadas, respectivamente.

pcUCAST: Esta funcionalidad proporciona la capacidad de enviar desde el ordenador central un mensaje a cualquier dispositivo de la red inalámbrica, sólo especificando el *index* del mismo. Es especialmente útil para abstraer a la aplicación de la forma con la que se envían los mensajes CAST en un módulo de comunicaciones de un fabricante u otro, sólo siendo necesario hacer los ajustes a bajo nivel sin que ello afecte la forma en que el usuario especifica la acción.

IV.4 Biblioteca "templates".

Esta biblioteca ha sido creada especialmente para el prototipado rápido de aplicaciones en las que, haciendo uso de la plataforma de integración, se puedan editar y crear nuevos tipos de sensores para que sus medidas sean enviadas y recibidas a través de la red inalámbrica. Está basada en los campos de tipos de nodos y número de datos de la trama del perfil de aplicación. Los tipos de sensores implementados hasta el momento están incluidos en esta plantilla de edición.

Se basa en dos funciones, una para la transmisión de la información, y otra para recepción. Cada una de ellas está dividida en bloques definidos como casos, que representan cada tipo de nodo definido. Por ejemplo, en el caso del nodo 0001 (temperatura, humedad y luz), este se encuentra definido en el caso 1, y el orden de los sensores es 1 (temperatura), 2 (humedad), y 3 (luz). De igual forma para el resto de nodos sensores establecidos. De la tabla 4.3 se puede deducir que la edición de los nuevos nodos se realizará a partir del caso 0111. Las dos funciones que definen esta biblioteca se describen a continuación:

sensor_send: Permite definir los sensores que forman parte de la plataforma, haciendo uso de las funciones proporcionadas por las bibliotecas para la adquisición de datos. El único requisito es que el valor resultante de una adquisición esté en formato de 2 bytes. La plataforma se encarga de realizar el resto de operaciones, enmascaramiento, trama de datos y transmisión de la información.

sensor_get: Este bloque define la forma en la que serán enviados y visualizados los datos recibidos de un nodo sensor en el ordenador central. El único requisito es especificar el descriptor de cada sensor a ser mostrado por pantalla. La plataforma se encarga de realizar el resto de operaciones, recepción de los datos, decodificación de la trama y desenmascaramiento de los datos recibidos.

Esta biblioteca permite abstraer al más alto nivel de la plataforma al usuario, puesto que con sólo unos ajustes puede realizar test de nuevos sensores implementados y comprobar su correcto funcionamiento con el sistema global.

IV.5 Glosario de funciones de las bibliotecas soporte creadas.

En el apartado anterior se ha hecho un repaso de los bloques funcionales que componen las bibliotecas soporte de la plataforma de integración HW-SW, aunque los detalles de bajo nivel no han sido incluidos y tomando en cuenta que la filosofía de diseño y requisitos de implementación son los mismos que en el caso de las bibliotecas base creadas. La descripción detallada de cada una de las funciones así como parámetros de entrada y salida puede ser encontrada en la documentación de las bibliotecas *software*. Por otro lado, los detalles de la biblioteca *global* se describirán más adelante ya que más que una biblioteca se trata del comportamiento del sistema completo y de cómo se administran todas las bibliotecas desde el nivel de abstracción más alto. A continuación se desglosan todas las funciones que componen este grupo de bibliotecas modulares:

Funciones de la biblioteca Data	
_CEI_data_keep_message	*_CEI_data_compare_commands*
_CEI_data_sensor_get	*_CEI_data_sensor_send*
_CEI_data_sensor_thhold	*_CEI_data_thhold_request*
_CEI_data_currentTime	

Tabla 4.17: Funciones contenidas en la biblioteca _CEI_tb_data.

Funciones de la biblioteca **Node**	
_CEI_node_reset	*_CEI_node_programming*
_CEI_node_UART_mode	*_CEI_node_power_mode*

Tabla 4.18: Funciones contenidas en la biblioteca _CEI_tb_node.

Funciones de la biblioteca **Network**	
_CEI_net_ini_nodeTable	*_CEI_net_pcUCAST*
_CEI_net_new_nodeType	*_CEI_net_new_nodeNetwork*
_CEI_net_keep_coo_adress	*_CEI_net_ AdressTable*
_CEI_net_ QualityTable	*_CEI_net_ NodeMap*
_CEI_net_ IndexDescriptor	*_CEI_net_ NodeRec*

Tabla 4.19: Funciones contenidas en la biblioteca _CEI_tb_network.

Funciones de la biblioteca **Templates**	
_CEI_template_sensor_get	*_CEI_template_sensor_send*

Tabla 4.20: Funciones contenidas en la biblioteca _CEI_tb_templates.

V. Testbed: Banco de pruebas para Redes de Sensores Inalámbricas.

La creación de un banco de pruebas para redes de sensores Inalámbricas basado en nodos Cookies es el objetivo final de la implementación de la plataforma de integración HW-SW diseñada y descrita hasta el momento, ya que será el soporte para exprimir las funcionalidades descritas y realizar pruebas dentro de un entorno real de aplicación. Es, por tanto, la consecución de los objetivos planteados de lograr integrar todas las características y requisitos fijados mediante la aplicación de un despliegue de nodos inalámbricos. Permitirá comprobar bajo un entorno real el conjunto de paquetes funcionales propuestos, y verificar el correcto funcionamiento del perfil de aplicación diseñado para aplicaciones basadas en los nodos Cookies.

En tal sentido, es de suma importancia poder establecer un camino de comunicación

entre el usuario de la aplicación y la plataforma inalámbrica, además de contar con una forma fácil y eficiente de gestionar el perfil de aplicación diseñado. Es por ello que se hace necesario crear una herramienta que sirva de interfaz entre el usuario final (desde el ordenador central) y la red de sensores (plataforma inalámbrica).

Los objetivos de dicha interfaz se basan en la capacidad de controlar todas las funcionalidades implementadas mediante el perfil de aplicación, poder realizar ajustes tanto desde el más alto nivel de abstracción así como poder configurar opciones a nivel de red en caso de ser necesario. Por otro lado, el usuario debe ser capaz de monitorizar todos los eventos que ocurren dentro de la red inalámbrica para tomar las acciones oportunas.

Del mismo modo, el banco de pruebas debe ser flexible y escalable (capacidad de poder incluir nuevos nodos a parte de los desplegados previamente), de fácil acceso y utilización por cualquier desarrollador, reutilizable (con el fin de aceptar diversas pruebas) y los nodos deben poder ser movibles para así realizar pruebas de performance en diferentes escenarios.

V.1 Interfaz de usuario.

Para el diseño de la interfaz entre el usuario y la aplicación se ha partido de la herramienta proporcionada por el fabricante Telegesis para controlar los Módulos ETRX2, denominada Telegesis Terminal [Telegesis 3]. Se ha decidido utilizar dicha herramienta puesto que posee una interfaz intuitiva y de fácil gestión, además de que posee opciones para la edición de controles, lo cual se adapta a las necesidades de la aplicación.

Más allá de las características de la herramienta, el diseño de la interfaz se basa en contar con una serie de directivas que se envían desde el ordenador central hacia el coordinador, éste las traduce y realiza las acciones oportunas en la red inalámbrica haciendo uso de los bloques funcionales embebidos en la plataforma. Para ello se ha definido un repertorio de comandos que se basa en parámetros de configuración, los cuales se han clasificado en tres grupos coincidentes con los campos del perfil de aplicación (Gestión de datos, Gestión de Nodo y Gestión de Red), más un cuarto grupo que se encarga de realizar la gestión de las comunicaciones serie mediante la UART. En la figura 4.14 se muestra una vista general de la interfaz de usuario así como los controles diseñados para la gestión de la aplicación.

Para facilitar la utilización de la interfaz y de los comandos de usuario, los controles diseñados poseen la trama relacionada con la acción a enviar al coordinador, por lo que sólo sería necesario introducir los parámetros referidos a dicha acción. Por ejemplo, si se desea realizar un reset remoto de alguno de los nodos de la red, al pulsar el botón "Remote Reset" el control introduce en la línea de comandos la trama

relacionada a dicho comando, y el usuario sólo debe introducir el *index* del nodo que se quiere reiniciar.

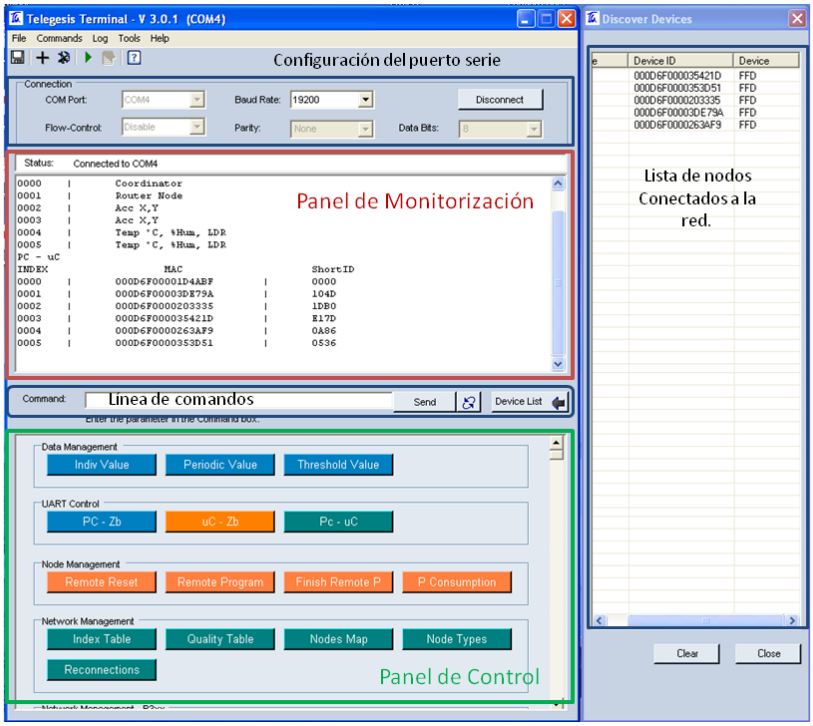


Figura 4.14: Interfaz de usuario del banco de pruebas.

- *Panel de control*: Se encuentran todos los botones que permiten realizar las acciones relacionadas con la gestión de la red de sensores.
- *Línea de comandos*: permite introducir el repertorio de instrucciones y los parámetros característicos de forma manual.
- *Panel de Monitorización*: Permite visualizar todos los eventos que ocurren en la red inalámbrica en tiempo real.
- *Configuración del puerto serie*: Permite realizar los ajustes de la comunicación del puerto serie del ordenador central.
- *Lista de nodos conectados a la red*: Permite visualizar las direcciones MAC de los nodos que están conectados a la red inalámbrica.

V.1.1 Repertorio de instrucciones.

Los comandos que permiten realizar las acciones sobre la red inalámbrica poseen la siguiente estructura:

<Tipo de gestión> <función>: <parámetros>

Tipo de gestión está relacionado con una letra según la acción a realizar, es decir, carácter "**d**" en caso de datos, "**n**" en caso de nodos, y "**w**" en caso de red. *Función* es el carácter que indica que tipo de acción se realiza dentro de la gestión seleccionada (1, 2, 3,…). *Parámetros* son los valores de configuración necesarios para realizar la acción seleccionada, y estos dependen de cada gestión. Tal y como se ha comentado anteriormente, los dos primeros campos son introducidos automáticamente en la línea de comandos al pulsar el botón correspondiente, sólo siendo necesario escribir los parámetros característicos. A continuación se describen cada uno de los comandos definidos con sus respectivos parámetros de configuración.

Valor individual (d1): Solicita el valor de un sensor dentro de un nodo, el de todos los sensores del nodo, o combinación de ellos.

d1: <Index>,<Data>	
<Index> [00-255]	Valor del *Index* del nodo sensor al cual se le solicitan los datos.
<Data>[00-15]	Magnitud de sensores requeridos, que pueden ser uno, todos, o combinación de ellos.
Ejemplo:	d1:02,04 (Nodo 2, solicitud de valor de LDR).

Tabla 4.21: Estructura del comando Valor Individual.

Valor Periódico (d2): Se solicita el envío periódico de un valor específico de un sensor dentro de un nodo, el de todos los sensores del nodo, o combinación de ellos.

d2: <Index>,<Data>,<Period>	
<Index> [00-255]	Valor del *Index* del nodo sensor al cual se le solicitan los datos.
<Data> [00-15]	Magnitud de sensores requeridos, que pueden ser uno, todos, o combinación de ellos.
<Period> [00-255]	Periodo de envío de datos en segundos.

Ejemplo:	d2: 02, 07, 05 (Nodo 2, envío del valor de Temperatura, humedad y LDR, cada 5 segundos).

Tabla 4.22: Estructura del comando Valor Periódico.

Valor de umbral (d3): Se solicita el envío periódico de un valor específico de un sensor dentro de un nodo, el de todos los sensores del nodo, o combinación de ellos, en el caso de que se supere un valor de umbral. Se especifica el valor de umbral, así como el signo del mismo, el cual indica si se envía el dato al sobrepasar positivamente el valor, o negativamente. Si se sobrepasa el valor de umbral, se enviarán los datos periódicamente cada 2 segundos.

d3: <Index>,<Data>,<Period>,<Sign><ThValue>	
<Index> [00-255]	Valor del *Index* del nodo sensor al cual se le solicitan los datos.
<Data> [00-15]	Magnitud de sensores requeridos, que pueden ser uno, todos, o combinación de ellos.
<Sign> [+-]	Indica si el umbral a superar es por arriba del valor especificado (+), o por debajo (-).
<ThValue>	Indica el valor de umbral a ser monitorizado (2 bytes como valor real).
Ejemplo:	d3: 02, 03,+3205,-4025 (Nodo 2, envío del valor de Temperatura si se superan los 32.05 ºC, y de humedad si esta baja del 40.25%).

Tabla 4.23: Estructura del comando Valor de Umbral.

Reset Remoto (n1): Permite realizar un reset de forma remota de un nodo específico, o el reset de toda la plataforma a nivel de aplicación.

n1: <Index>	
<Index> [00-255]	Valor del *Index* del nodo sensor que se pretende hacer el reset.
Ejemplo:	n1:02 (Se resetea en nodo con *Index* 2).
Ejemplo:	n1:00 (Si el *Index* es 0, se resetea toda la red inalámbrica a nivel de aplicación).

Tabla 4.24: Estructura del comando Reset Remoto.

Programacion Remota (n2): Establece el enlace serie inalámbrico entre el coordinador y el nodo especificado con el fin de programar el microcontrolador de forma remota con un archivo .hex especificado. Al realizar esta acción se debe desconectar la interfaz de usuario en el panel de configuración, y luego abrir el *Windows Serial Downloader* para poder cargar el archivo .hex.

n2: <Index>	
<Index> [00-255]	Valor del *Index* del nodo sensor que se pretende programar remotamente.
Ejemplo:	n2:02 (Se programa el nodo con *Index* 2).

Tabla 4.25: Estructura del comando Programación Remota.

Monitorización de Consumo (n3): Se solicita el valor del consumo de corriente en mili amperios del nodo especificado a través del *Index*. Existen 3 opciones, al igual que en el caso de los datos, para solicitar dicho valor. Se puede elegir entre valor individual, valor periódico y valor de umbral, cada una con sus respectivos parámetros. Los tres casos se muestran a continuación:

n3: <Index>,00	
<Index> [00-255]	Valor del *Index* del nodo sensor al cual se le solicitan los datos de consumo.
00	Indica que se trata de la opción Valor Individual
Ejemplo:	n3:02,00 (Nodo 2, solicitud de valor individual de consumo).

Tabla 4.26: Estructura del comando Monitorización de Consumo, en modo individual.

n3: <Index>,01,<Period>	
<Index> [00-255]	Valor del *Index* del nodo sensor al cual se le solicitan los datos de consumo.
01	Indica que se trata de la opción Valor Periódico.
<Period> [00-255]	Periodo de envío de datos en segundos.
Ejemplo:	n3: 02, 01, 05 (Nodo 2, envío del valor de consumo cada

	5 segundos).

Tabla 4.27: Estructura del comando Monitorización de Consumo, en modo periódico.

n3: <Index>,02,<Period>,<Sign><ThValue>	
<Index> [00-255]	Valor del *Index* del nodo sensor al cual se le solicitan los datos.
01	Indica que se trata de la opción Valor de Umbral.
<Sign> [+-]	Indica si el umbral a superar es por arriba del valor especificado (+), o por debajo (-).
<ThValue>	Indica el valor de umbral a ser monitorizado (2 bytes como valor real).
Ejemplo:	n3: 02, 01,+0852 (Nodo 2, envío del valor de consumo si se superan los 85.2 mA).

Tabla 4.28: Estructura del comando Monitorización de Consumo, en modo umbral.

Puesta en bajo Consumo (n4): Permite configurar el módulo ZigBee de cualquier nodo dentro de la red inalámbrica en modo dormido, durante un determinado periodo. Además del *index* del nodo, se especifica el periodo de tiempo en modo dormido, así como la unidad de tiempo (horas, minutos o segundos).

n4: <Index>,<Period>,<Unit>	
<Index> [00-255]	Valor del *Index* del nodo sensor al cual se configura el modo de bajo consumo.
<Period> [00-255]	Tiempo en el que el módulo ZigBee estará dormido.
<Unit [1–3]>	Unidad de tiempo del periodo especificado.
Ejemplo:	n4: 05, 04, 02 (Nodo 5, puesta en bajo consumo durante 4 minutos).

Tabla 4.29: Estructura del comando Puesta en Bajo Consumo.

Tabla de Direcciones (w1): Permite mostrar una tabla con todos los nodos que se encuentran conectados a la red inalámbrica con sus respectivos *index*, dirección MAC y dirección corta. Esta función no posee parámetros ya que directamente muestra la tabla correspondiente.

Enlace entre los dispositivos (w2): Esta opción permite obtener la calidad del enlace de datos que posee un nodo con sus respectivos vecinos, mostrando una tabla con los respectivos valores LQI de cada enlace.

w2: <Index>	
<Index> [00-255]	Valor del *Index* del nodo sensor del que se pretende conocer la tabla de enlace.
Ejemplo:	w2:02 (Se consulta el nodo con *Index* 2).

Tabla 4.30: Estructura del comando Enlace entre los dispositivos.

Mapa de Conexión (w3): Permite conocer la conexión existente entre un nodo especificado y el coordinador, y así conocer la ruta que siguen los datos entre ambos dispositivos. La acción permite mostrar cuantos saltos realiza el mensaje entre nodos, y por cuales nodos los realiza.

w3: <Index>	
<Index> [00-255]	Valor del *Index* del nodo sensor del que se pretende conocer la ruta de datos. Si se especifica como *index* 00, **se muestra un mapa completo de la red inalámbrica.**
Ejemplo:	w3:02 (Se consulta el nodo con *Index* 2).

Tabla 4.31: Estructura del comando Mapa de Conexión.

Tipos de nodo (w4): Esta opción visualiza en pantalla la descripción de todos los nodos conectados a la red, con sus respectivos *index*, lo que permite conocer el tipo de sensores que poseen y las medidas que realizan. Esta función no posee parámetros ya que directamente muestra la tabla correspondiente.

Número de reconexiones (w5): Permite visualizar en pantalla el número de reconexiones a la red que han realizado los nodos que forman parte de ella. Un nodo realiza una reconexión cuando, por ejemplo, pierde cobertura momentáneamente, se apaga y se enciende, o se realiza un reset sobre el mismo. Esta función no posee parámetros ya que directamente muestra la tabla correspondiente.

Por otro lado, en el caso de la gestión del la conexión serie, el control *PC – uC*

permite cambiar el modo de conexión para poder enviar comandos al microcontrolador, mientras que el comando *PC – Zb* recibe como parámetro 2 tipos posibles de conexión: Interfaz – ZigBee (1), o Interfaz – ZigBee – Microcontrolador (0).

V.1.2 Decodificación de instrucciones.

Los datos que son recibidos en el coordinador desde el ordenador central han de ser decodificados con el fin de poder enviar las instrucciones vía inalámbrica mediante la trama de datos del perfil de aplicación. Para ello ha sido necesario diseñar un bloque adicional en el microcontrolador que se encarga de realizar dicha tarea. Este bloque se encuentra implementado en la biblioteca *global,* y está dividido principalmente en tres sub-bloques, los cuales se detallan a continuación:

Verificación de comando: Este bloque hace uso de la función *compare_commands* que se encuentra incluida en la biblioteca *_CEI_tb_data,* la cual se encarga de comparar la trama recibida desde la interfaz de usuario con el repertorio de instrucciones predefinido y comprobar si forma parte del mismo. Si no es así, se genera un error hacia el ordenador central con el fin de indicar que el comando no existe.

Verificación de Parámetros: Se ha implementado un sub-bloque que se encarga de comprobar si, según la acción recibida, los parámetros introducidos son validos o no. Esto es útil en los casos en los cuales el usuario introduce valores fuera de rango, un *index* inexistente, o ha enviado el comando sin introducir un valor característico de la instrucción. En caso de ser así, se genera un aviso hacia el ordenador central con el fin de indicar que ha habido un error en el comando introducido. Esto permite, por tanto, filtrar posibles errores en la configuración de una acción.

Codificación de datos: Luego de comprobar que la trama recibida desde la interfaz es correcta, este sub-bloque permite enlazar los datos obtenidos desde el PC con la trama de datos perfil de aplicación, codificando los valores de forma adecuada para que puedan ser enviados a través de la red inalámbrica. En el caso de ser una acción que se realiza de forma local en el coordinador, se enlaza la acción recibida con la funcionalidad correspondiente para tal fin.

VI. Estructura del sistema embebido en el uC: Biblioteca "Global".

Esta biblioteca es la columna vertebral de todas las funcionalidades diseñadas hasta el momento, ya que es la encargada de enlazar las configuraciones y acciones recibidas desde la red inalámbrica o desde la interfaz de usuario, con los correspondientes bloques funcionales.

En tal sentido, esta biblioteca implementa un sub-bloque para la correcta decodificación de la trama de datos provenientes de la red inalámbrica y el sub-bloque de decodificación de instrucciones provenientes de la interfaz de usuario, en el caso del nodo coordinador. Además de ello, se incluye un módulo para la gestión de los mensajes de notificación provenientes de las demás bibliotecas, con el fin de tomar las acciones correctivas necesarias. En la figura 4.15 se muestra de forma esquemática todos los componentes incluidos en esta biblioteca, además de ilustrar la estructura general del sistema global implementado en el uC.

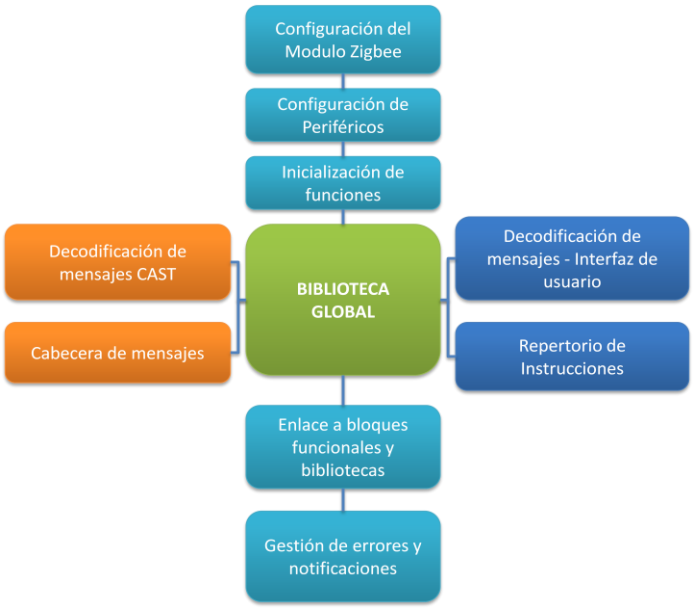


Figura 4.15: Estructura de la biblioteca _CEI_Global.

Capítulo 5

RESULTADOS OBTENIDOS

I. Evolución de la plataforma a partir de entornos reales de aplicación.

Durante el desarrollo de las bibliotecas *software* para el control de la arquitectura *hardware* y la plataforma de integración HW-SW, las implementaciones se han ido probando no sólo a nivel funcional dentro del laboratorio del Centro de Electrónica Industrial, sino que también se han utilizado para realizar pruebas de campo en el marco de proyectos relacionados con la plataforma Cookies para redes de sensores inalámbricas.

Dichas pruebas han resultado de gran utilidad no sólo para verificar el correcto comportamiento de los bloques funcionales que se iban diseñando dentro del presente proyecto, sino también para conocer los requisitos exigidos en entornos reales de aplicación, para así poder aproximar la plataforma de integración a condiciones reales y realzar las características que son mayormente prioritarias cuando se despliega una red de sensores.

Gracias a ello, el diseño de plataforma de integración HW-SW está enfocada y adaptada a las propiedades más importantes a tomar en cuenta en las implementaciones finales de nodos Cookies en aplicaciones reales, con el objetivo final de que el presente proyecto sirva como primera herramienta para pruebas de performance y funcionamiento de las mismas y, por lo tanto, sea el *testbed* soporte de desarrollos futuros.

Aunque han sido varios los proyectos en los que se han aplicado las funcionalidades que se han ido desarrollando durante el trabajo de investigación, en este apartado se explorarán dos casos prácticos que han dado aportaciones al diseño de los bloques funcionales finalmente concebidos, mientras que en el siguiente apartado se detallarán los despliegues realizados para pruebas con el sistema final creado e implementado. Del mismo modo, dichas pruebas permiten situar en contexto no sólo los bloques funcionales creados para el control del nodo inalámbrico, sino también el aporte real que estos proporcionan a la hora de desplegar una red de sensores inalámbricos en un entorno de aplicación.

A continuación se describen las dos pruebas más importantes de campo realizadas, sus características, peculiaridades y condiciones particulares a la hora de realizar el despliegue de los nodos Cookies.

I.1 Escenario 1: Estación Renfe de Cercanías Fuencarral.

Esta prueba, realizada en las vías férreas de la red de Cercanías Renfe de la ciudad de Madrid, específicamente en la estación de Fuencarral, se enmarca dentro del proyecto SafetyRail y consiste en realizar determinadas pruebas con los nodos

Cookies con el fin de cubrir los objetivos que se desglosan a continuación:

- Pruebas de alcance de los nodos inalámbricos de la plataforma Cookies con comunicaciones a la banda de Frecuencias 868MHz, con la implementación de antenas de corto alcance.
- Pruebas de vibraciones en la vía ferroviaria para la detección de trenes en aproximación.
- Aplicación de las bibliotecas *software* para el control del nodo inalámbrico.
- Prueba del *porting* de la biblioteca "*ZigBee*" para realizar el control del módulo de comunicaciones utilizado.

La conexión inalámbrica entre los nodos para la realización de estas pruebas se ha llevado a cabo mediante una capa de comunicaciones creada para albergar un módulo ZigBee que trabaje en las bandas 784, 868 y 915 MHz, que son bandas distintas a la utilizada por el módulo ETRX2, el cual trabaja a 2.4 GHz.
El módulo implementado en dicha capa es el ZigBit900 del fabricante Atmel [ATMEL], el cual tiene la posibilidad de transmitir entre -11 y +11 dbm. En las pruebas se ha decidido utilizar la potencia máxima para así aprovechar el campo de aplicación, al tratarse de un sitio abierto sin mayores obstáculos en el entorno.

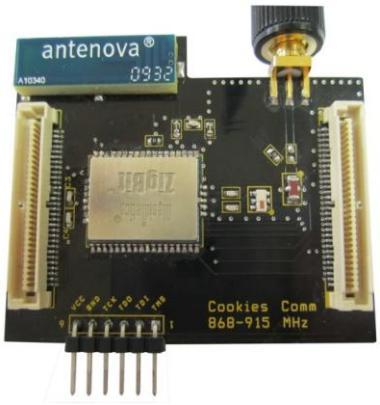


Figura 5.1: Placa ZigBee para la banda de frecuencias 784, 868 y 915 MHz.

I.1.1 Pruebas de alcance y porting de la biblioteca "ZigBee".

Luego de desarrollar una primera versión de las cuatro bibliotecas básicas detalladas en el capítulo 3, la idea consistía en probar el funcionamiento de los bloques diseñados mediante la gestión de esta capa de comunicaciones, para lo que ha sido necesario a su vez comprobar la flexibilidad y portabilidad de las bibliotecas a un nuevo *hardware*. En este caso ha sido necesario realizar el *porting* de los bloques

funcionales de la biblioteca _CEI_zb para así obtener el correcto funcionamiento de la biblioteca con el módulo Zigbit.

Tal y como se ha planteado en los requisitos de diseño de las bibliotecas *software* contemplados en el capítulo 3, la interfaz entre _CEI_zb y los niveles superiores ha permanecido inalterada, por lo que sólo ha sido necesario realizar algunos ajustes mas a nivel de nomenclatura, en función del repertorio de comandos AT disponibles para este módulo, pero la estructura de configuración y conexión se ha mantenido. Como resultado, se han utilizado las mismas funciones creadas para el módulo ETRX2, y se ha podido configurar y crear una nueva red inalámbrica, además de poder conectar de forma adecuada el nodo sensor a dicha red. Con ello, por tanto, se ha comprobado el correcto funcionamiento de los principales bloques funcionales de la biblioteca "ZigBee", en su versión inicial de desarrollo.

Por otra parte, se han probado las bibliotecas *_CEI_peripherals* y *_CEI_queue*, también en su versión inicial, comprobando la configuración adecuada de la conexión UART entre los dispositivos del nodo inalámbrico, y la transmisión y recepción de datos haciendo uso de los buffers creados para tal fin.

Como resultado del adecuado funcionamiento de los bloques funcionales probados, se han realizado pruebas de alcance de las comunicaciones ZigBee mediante el módulo de Atmel y con la utilización de antenas de corto alcance (antena integrada) para comprobar la cobertura que con estas se puede obtener sobre entornos abiertos. Dichas pruebas se resumen mediante la tabla 5.1, que muestra las condiciones de la prueba.

Antena	Integrada.
Potencia de Transmisión	+11 dBm.
Frecuencia de transmisión	868MHz
Entorno	Exterior, vías ferroviarias
Ambiente	Soleado
Interconexión	Conexión Nodo Final – Coordinador de la Red Inalámbrica.
Resultados obtenidos	Comunicación directa entre el nodo final y el coordinador, a **40 metros** de distancia.

Tabla 5.1: Condiciones de prueba de los módulos Zigbit de ATMEL.

Figura 5.2: Pruebas de alcance del módulo de comunicaciones Zigbit.

I.1.2 Pruebas de vibraciones.

La segunda prueba a realizar en la estación de trenes de Renfe consistía en la medición de vibraciones en la vía ferroviaria con el fin de detectar el paso de trenes por las mismas. Para tal fin, se ha utilizado la capa de sensores compuesta por el acelerómetro ADXL213, para medir dichas vibraciones colocando el nodo directamente en uno de los rieles de la vía y retransmitiendo la información del sensor al nodo coordinador para visualizar los datos por pantalla.

Desde el punto de vista de los bloques funcionales creados en el presente proyecto, se han podido realizar pruebas relacionadas principalmente con la biblioteca *_CEI_peripherals*, sobretodo del bloque que controla las comunicaciones con la FPGA al tratarse de un sensor digital, por lo que las muestras se tomaban a través de la interfaz genérica que la ella proporciona.

Como resultado de los bloques funcionales implementados, se ha podido realizar la correcta monitorización de las medidas de vibraciones cada vez que se aproximaba un tren al sitio donde se encontraba desplegado el nodo inalámbrico, percibiéndose la variación de los valores del sensor de aceleración en los ejes X e Y.

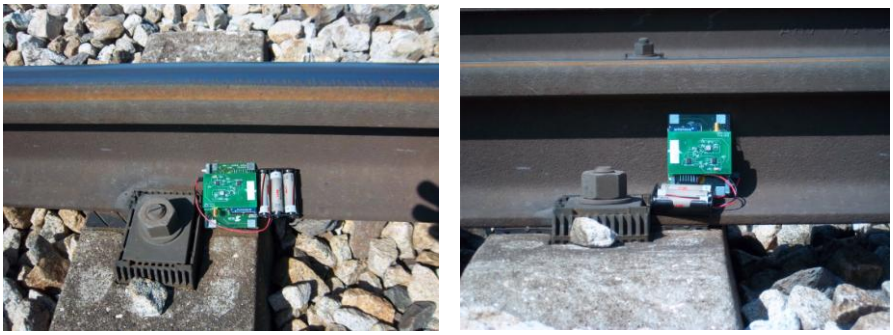


Figura 5.3: posición del nodo sensor durante las pruebas de vibraciones.

En la figura 5.4 se pueden observar los rangos de variación de aceleración partiendo de valores en reposo. En el caso del eje Y el valor proporcional de reposo del sensor (sin vibraciones) es de 4375, mientras que en el eje X es de 4983 (al ser las variaciones de aceleración muy pequeñas, se ha optado por visualizar los valores sin convertirlos a la correspondiente unidad en g, y así poder percibir numéricamente mayor rango de variaciones).

En el momento en el que el tren atraviesa la zona de medición del sensor, se detectan valores de vibración equivalentes, con sus respectivas oscilaciones relacionadas, antes de que el sensor vuelva a su estado de reposo. La frecuencia con la que han sido tomadas las medidas de aceleración en ambos ejes ha sido de 2 valores por segundo.

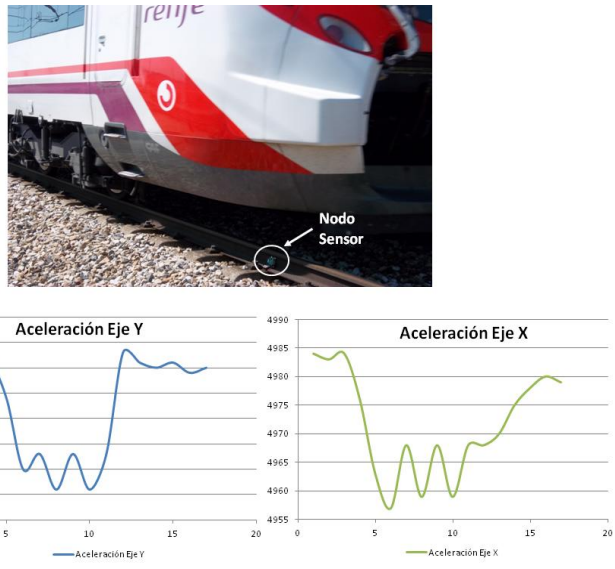


Figura 5.4: Medida de vibraciones en las vías de tren.

I.2 Escenario 2: Fabrica de café soluble.

Este segundo escenario práctico de aplicación ha supuesto un gran avance dentro del marco del presente proyecto de investigación ya que se han podido probar diversas funcionalidades importantes mediante un entorno real que suponía muchas dificultades desde el punto de vista de las comunicaciones inalámbricas y el procesamiento de los datos relacionados con la misma.

Las pruebas realizadas en este segundo escenario de aplicación se encuentran a su vez enmarcadas en el proyecto SustenTIC, que tiene como objetivos la monitorización en tiempo real de parámetros relacionados con el proceso de producción en fabricas que puedan generar efectos medioambientales, tales como la emisión de gases contaminantes o aguas residuales, además de buscar mejoras en el rendimiento de dichos procesos.

Dentro de dicho proyecto se han pautado realizar pruebas en una de las fábricas de café soluble más importantes de Europa ubicada en Palencia, España, con el fin de monitorizar las emisiones de gases y la calidad de las aguas residuales. Otro de los objetivos de dichas pruebas era realizar la monitorización de dichos parámetros mediante una herramienta sencilla, automática y de forma desatendida como lo es a través de una red de sensores inalámbrica basada en nodos Cookies, ya que dichas pruebas se realizan normalmente de forma manual por un operario, lo cual es contraproducente en términos de seguridad hacia los trabajadores. Por otra parte, mediante el método manual no se podría detectar en tiempo real una posible anomalía en los valores medidos, lo cual podría afectar de forma perjudicial el rendimiento de la producción.

Dentro de dichos objetivos, las pruebas en la fabrica se han destinado a cubrir principalmente dos aspectos clave, que son la monitorización continua de la red inalámbrica, lo cual incluye los valores de intensidad de señal entre los nodos y calidad de los enlaces, el consumo de energía de los nodos en tiempo real, mapa de conexiones entre los dispositivos, y el estado de los sensores de cada nodo; por otro lado el aspecto relacionado con las medidas de los sensores inalámbricos, con el fin de monitorizar los parámetros de calidad de las aguas residuales y la emisión de gases de la fábrica.

Además de ello, era de suma importancia conocer los valores de temperatura y humedad del ambiente en el que encuentran expuestos los nodos con el fin de conocer cómo afectan estos parámetros a la calidad de la señal de la red inalámbrica, y servir como referencia para la calibración de los sensores de gases.

El despliegue de nodos inalámbricos en este tipo de escenarios de aplicación supone varias dificultades y retos debido a las continuas interferencias que se pueden encontrar, tales como elementos metálicos, camiones de transporte, maquinaria de la

fábrica, entre otros factores. Es por ello que ha sido el entorno ideal para realizar pruebas de robustez de la red, sobre todo para comprobar el funcionamiento de algoritmos de conexión y reconexión de los nodos cuando ocurre algún problema con el enlace entre ellos, además de buscar mejores caminos de comunicación entre los mismos. Es de destacar que la comunicación entre los dispositivos se llevo a cabo mediante la utilización del módulo ETRX2, por lo que se ha realizado una comprobación rigurosa del correcto funcionamiento de la biblioteca "ZigBee", además de probar el performance de otros bloques funcionales relacionados con las comunicaciones, tal y como se describirá en líneas posteriores.

Las pruebas llevadas a cabo en la fábrica se han dividido en dos áreas en función del tipo de medidas a realizar y la superficie a cubrir mediante la red inalámbrica. Para cumplir con los objetivos planteados y obtener las medidas físicas del entorno, se han utilizado nodos inalámbricos con diferentes capas de sensado, en este caso una de medida de pH y temperatura del agua, una de medida de CO, y nodos con medidas de temperatura y humedad en el ambiente.

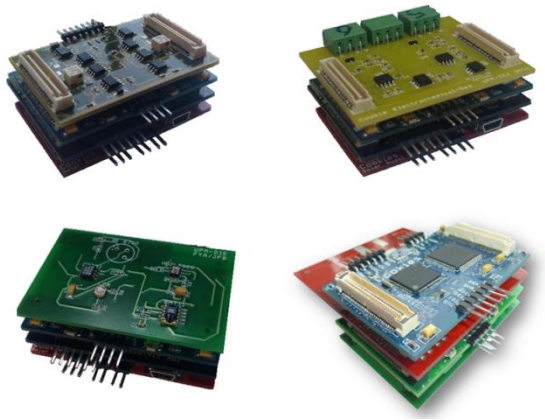


Figura 5.5: Tipos de sensores utilizados en el segundo escenario de aplicación.

A continuación se detallan los dos experimentos realizados en la fábrica objeto de estudio, y los elementos que intervienen en cada una de ellas.

I.2.1 Prueba Nº1: planta de aguas residuales y perímetro de la fábrica.

Este primer experimento se basaba en la monitorización en tiempo real de la calidad de las aguas residuales, cuya medida se llevo a cabo en la planta destinada para tal fin, como se aprecia en la figura 5.6. Por otra parte, el interés de la prueba se centraba en el estudio de las comunicaciones inalámbricas a través de parámetros de intensidad de radio (*RSSI, received signal strength indicator),* y calidad del enlace entre los dispositivos (*LQI, Link Quality Indicator*), y cómo afectan factores tales como la

temperatura y la humedad en el ambiente, así como el consumo de las baterías en el nodo inalámbrico. Tal y como se aprecia en la misma figura, el despliegue se ha llevado a cabo a partir de 4 nodos Cookies con el fin de cubrir todo el perímetro de la fábrica y conocer los parámetros de interés de la red en dicho escenario, los cuales se enumeran a continuación:

- *Nodo 1:* Nodo Sensor para la monitorización de la medida de pH y temperatura de las aguas residuales, así como el consumo de corriente del mismo.

- *Nodo 2:* Nodo Sensor/*Router* para la medida de temperatura y humedad ambiente, parámetros de calidad de la red inalámbrica (RSSI y LQI), y el consumo de corriente del mismo.

- *Nodo 3:* Nodo Sensor/*Router* para la medida de temperatura y humedad ambiente, parámetros de calidad de la red inalámbrica (RSSI y LQI), y el consumo de corriente del mismo.

- *Nodo 4:* Nodo coordinador de la red inalámbrica, el cual se encarga de recolectar los datos de los sensores para poder ser monitorizados en tiempo real.

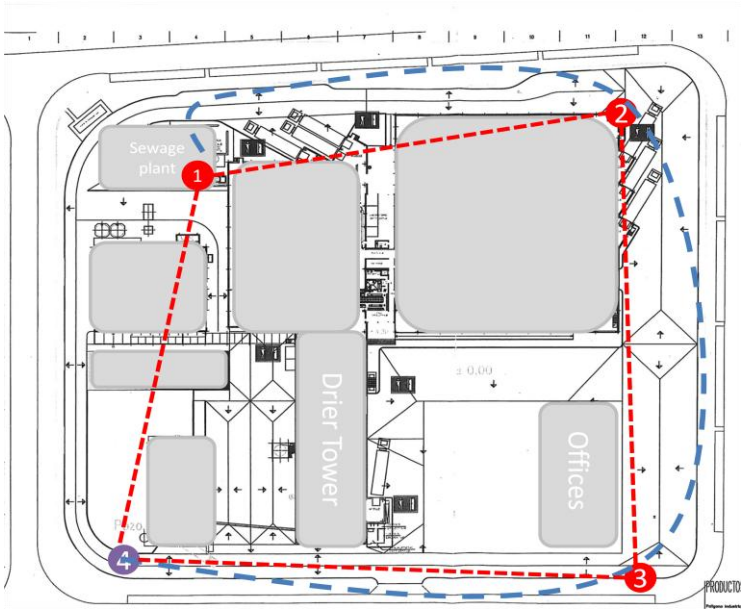


Figura 5.6: Despliegue realizado para la prueba Nº1.

Se ha podido comprobar el alcance de los nodos inalámbricos en estas condiciones, sobre todo tomando en cuenta que la distancia entre los mismos es de 130-160 metros, además de contar con la dificultad de obstáculos tales como camiones de carga y descarga que circulaban principalmente entre los nodos 2 y 3, por lo que se ha verificado la efectividad de la plataforma en los casos de pérdida momentánea de la red inalámbrica por dichos factores externos y el restablecimiento de la conexión remota, por lo que la autogestión de los nodos quedó totalmente asegurada.

En la figura 5.7 se puede apreciar tanto las variaciones en la medida del RSSI cuando había movimientos de camiones, así como la correcta adquisición de las medidas de pH, temperatura del agua y consumo de corriente.

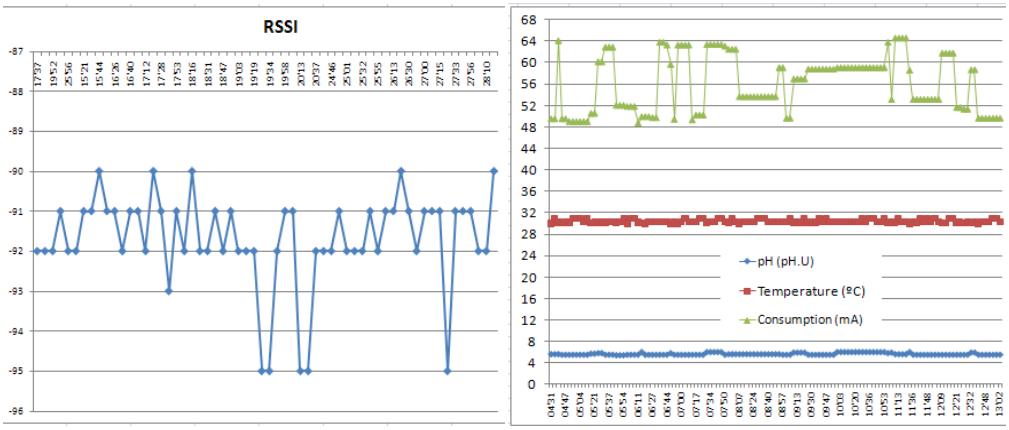


Figura 5.7: Medidas obtenidas por parte de los nodos de la red inalámbrica.

I.2.2 Prueba Nº2: Cobertura de todos los parámetros de medida.

Este segundo experimento se baso en la adquisición de todos los parámetros característicos que se deseaban tomar en cuenta durante el despliegue de la red de sensores en este escenario de aplicación. Esto quiere decir que, además de los parámetros medidos en la primera prueba, se ha añadido el sensor de CO en la chimenea de escape de gases, así como también se ha dado cobertura a las zonas de la fábrica donde no se había desplegado nodos en la anterior prueba, con el fin de monitorizar los datos de temperatura y humedad ambiental, así como la calidad de las comunicaciones entre los dispositivos. En tal sentido, se han utilizado 5 nodos para realizar el despliegue, distribuidos tal y como se detalla a continuación:

- *Nodo 1:* Nodo Sensor para la monitorización de la medida de PH y temperatura de las aguas residuales, así como el consumo de corriente del mismo.

- *Nodo 2:* Nodo Sensor/*Router* para la medida de temperatura y humedad ambiente, parámetros de calidad de la red inalámbrica (RSSI y LQI), y el consumo de corriente del mismo.

- *Nodo 3*: Nodo Sensor/*Router* para la medida de emisión de CO y temperatura en la chimenea de escape de gases, así como el consumo de corriente del nodo.

- *Nodo 4:* Nodo Sensor/*Router* para la medida de temperatura y humedad ambiente, parámetros de calidad de la red inalámbrica (RSSI y LQI), y el consumo de corriente del mismo.

- *Nodo 5:* Nodo coordinador de la red inalámbrica, el cual se encarga de recolectar los datos de los sensores para poder ser monitorizados en tiempo real.

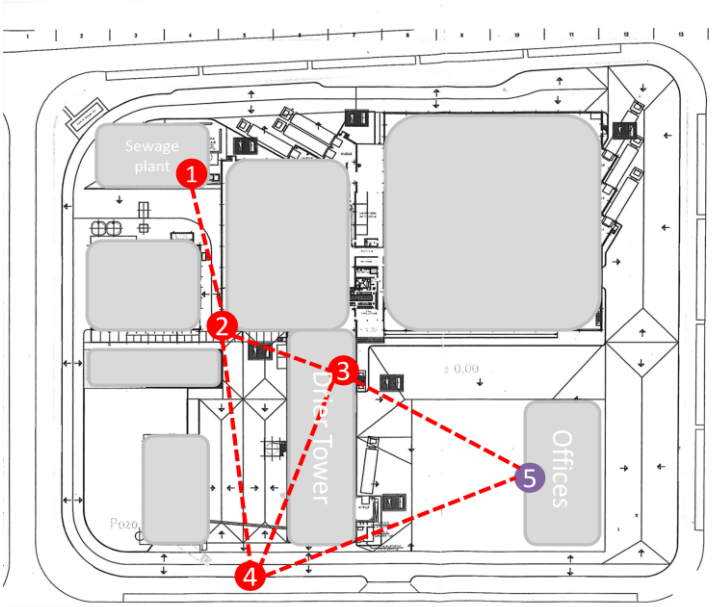


Figura 5.8: Despliegue realizado para la prueba Nº2.

Esta prueba ha resultado de gran utilidad para la plataforma de integración HW-SW puesto que ha supuesto la comprobación del funcionamiento del sistema conjunto de la red inalámbrica de forma robusta y fiable, ya que desde el momento en el que los nodos se conectaban a la red, la monitorización de los valores se llevaba a cabo de forma efectiva y, en el caso de pérdida momentánea de la conexión, la autogestión de

cada uno de los nodos se ha realizado sin la necesidad de intervenir en el sistema.

I.2.3 Resultados obtenidos y aportaciones desde el punto de vista de la plataforma de integración HW-SW.

El éxito en la realización de las pruebas de campo realizadas en el entorno de aplicación expuesto anteriormente ha supuesto un gran avance de cara a la implementación final de redes de sensores inalámbricas en aplicaciones reales que necesiten un sistema basado en la fiabilidad, sencillez, robustez y la autogestión de los dispositivos involucrados. Desde el punto de vista de la plataforma de integración presentada en el presente trabajo de investigación, cuyo aporte ha sido clave para el funcionamiento de las pruebas realizadas, se han verificado las siguientes características:

Biblioteca "ZigBee": para cumplir con los rigurosos objetivos planteados en este escenario de aplicación, se realizaron modificaciones desde la versión inicial probada con el fin de asegurar totalmente el nivel de conexión inalámbrica entre los dispositivos y la autogestión de los nodos en caso de detectar errores en las comunicaciones. Dichos cambios han servido de base para concebir la versión final de la biblioteca con la que se cuenta actualmente.

Biblioteca "DataProcess": Se ha podido verificar el correcto funcionamiento de los bloques funcionales que permiten decodificar los datos adquiridos desde la biblioteca "*peripherals*" para poder enviarlos vía inalámbrica. En este sentido se han probado gran parte de los bloques implementados, entre los que están las funciones para pH, CO, temperatura y humedad, consumo de corriente, enmascaramiento de la información y conversión de tipos de datos.

Perfil de aplicación: Para la correcta gestión de todo el sistema conjunto y el sostenimiento de la plataforma sin que sea susceptible de errores que ocasionen un colapso de la aplicación, ha sido necesario el diseño de las primeras funciones relacionadas con el perfil de aplicación que se detalla en el presente proyecto. Entre ello se encuentra el diseño de una trama de datos inicial que, aunque implementada en una primera versión de evaluación, ha permitido obtener resultados muy útiles de cara al diseño que se ha desarrollado e implementado en la versión final de la plataforma. Dentro de ello se encuentran los siguientes bloques funcionales, que fueron probados satisfactoriamente en los experimentos llevados a cabo.

Gestión de datos: Se verificó una versión inicial de la gestión de datos de los nodos a través de la primera aproximación a la trama final que sería implementada. Se comprobó el correcto funcionamiento de la gestión de datos periódicos de sensores para la monitorización en tiempo real.

Gestión de Nodo: Se verificó una versión inicial de la monitorización en tiempo real

del consumo de corriente por parte de los nodos.

Gestión de Red: Se verificaron tareas tales como la adecuada configuración y conexión de los nodos a la red inalámbrica, bloques funcionales para la detección de errores en la comunicación y reconexión en el caso de pérdida momentánea del enlace, así como una versión inicial del mapa de conexión entre los dispositivos, en el que se detectaban los saltos que requería realizar cada mensaje desde el nodo sensor hasta el nodo coordinador, a través de los nodos *routers*. Además de ello, se comprobó el funcionamiento de los bloques encargados de detectar la calidad de las comunicaciones entre todos los dispositivos perteneciente a la red inalámbrica.

Es interesante destacar que, aunque en el momento de realizar las pruebas de campo en la fábrica de café soluble el perfil de aplicación se encontraba en fase de desarrollo, se ha podido comprobar el correcto funcionamiento de los bloques diseñados hasta el momento, además de la total adaptabilidad y flexibilidad de la plataforma de integración al entorno de aplicación, lo que da lugar a la consecución de los objetivos propuestos y a los requisitos de diseño. Por otro lado se pudo comprobar la adecuada implementación de las bibliotecas *software* para el control del nodo inalámbrico, que en el momento de realizar el despliegue de la red inalámbrica se encontraban en una fase casi finalizada de desarrollo, sólo siendo necesario las respectivas optimizaciones y ajustes en función de las condiciones y requisitos finales de la plataforma de integración HW-SW planteada.

El éxito de las pruebas de campo realizadas y el aporte que la plataforma ha realizado para el correcto funcionamiento del despliegue de los nodos inalámbricos, ha dado lugar a que los resultados obtenidos hayan sido publicados en el articulo [Valverde'12], del cual el autor del presente proyecto es coautor, y en el que se detalla el procedimiento utilizado para el despliegue realizado, las condiciones de funcionamiento del sistema conjunto y cada uno de los dispositivos involucrados, los retos y dificultades que suponía realizar dichas pruebas en el entorno real de aplicación, y los objetivos finales alcanzados en dichos experimentos.

II. Resultados finales obtenidos de la plataforma de integración HW-SW y testbed para redes de sensores inalámbricas.

Luego de realizar el diseño y la implementación tanto de las bibliotecas *software* descritas en el capítulo 3 así como del perfil de aplicación y los bloques funcionales que soportan el *testbed* para redes de sensores inalámbricas detallado en el capítulo 4, y después de considerar el aporte obtenido en los campos de aplicación en los que ha sido probado la plataforma para su posterior optimización y rediseño en caso necesario, se ha obtenido una versión totalmente operativa del sistema conjunto que cumple con los requisitos y objetivos planteados en el presente proyecto.

Con el fin de situar al sistema conjunto en contexto mediante su aplicación dentro de

un entorno real, se han realizado pruebas de performance de la plataforma en el Centro de Electrónica Industrial, con el fin de comprobar el comportamiento de las funcionalidades desarrolladas.

La idea se centra en desplegar los nodos que forman parte del banco de pruebas creado (6 nodos en total, los cuales han sido correctamente adaptados a los cambios realizados a nivel *hardware* durante el desarrollo del presente proyecto), con el fin de tener la mayor cobertura posible de toda el área del Centro, y así realizar pruebas relacionadas con medidas de sensores, enlace entre los dispositivos, mapa de conexiones, programación remota, entre otros aspectos.

El Centro de Electrónica Industrial es un entorno *indoor* compuesto por diferentes salas y despachos (ver anexo I), además de contar con un número elevado de equipos electrónicos y personas trabajando en dicho entorno, por lo que se puede considerar como un área con un nivel considerable de interferencias y obstáculos. En tal sentido, se ha decidido realizar la monitorización de valores relacionados con el ambiente tales como temperatura, humedad y luminosidad en diversas áreas del Centro, así como comprobar cómo se comporta la red inalámbrica en dicho escenario de aplicación, verificando el mapa de conexión y calidad del enlace entre los dispositivos, además de probar la programación remota del microcontrolador a través del enlace ZigBee con el fin de estudiar los posibles inconvenientes que el entorno pueda producir al realizar dicha tarea.

Las pruebas se han dividido en dos partes, cada una en función de la posición de los nodos sensores (ver anexo I y II). Todos los nodos llevan implementado el perfil de aplicación desarrollado, y han sido configurados como FFD (sensores/*routers*), tal y como se muestra en la tabla 5.2.

TIPO DE NODO	MAGNITUD FÍSICA
Coordinador	------
Sensor/Router	Temperatura, humedad, Luz.
Sensor/Router	Temperatura, humedad, Luz.
Sensor/Router	Temperatura, humedad, Luz.
Sensor/Router	Temperatura, humedad, Luz.
Sensor/Router	Aceleración ejes X, Y.

Tabla 5.2: Nodos sensores desplegados en el CEI.

Se han configurado todos los nodos como FFD para no limitar la conectividad de los mismos y verificar el mapa de conexión que adopta la red inalámbrica en estas

condiciones. Se han utilizado 4 nodos sensores de temperatura, humedad y luz, lo que permite realizar pruebas con sensores tanto analógicos como digitales. Por otro lado, se ha utilizado una capa de sensado distinta a la anterior con el fin de comprobar el performance del nodo inalámbrico con varios tipos de sensado, para lo cual se ha elegido incluir el sensor de aceleración en los ejes X e Y, y detectar de forma remota una posible alteración de la posición del nodo. Por otro lado, se han utilizado los módulos ETRX2 para establecer la conexión entre los dispositivos y realizar la gestión de la red inalámbrica.

II.1 Despliegue Nº1.

Este primer despliegue cubre tanto el laboratorio principal donde se encuentran los estudiantes investigadores del centro, como también el pasillo que da acceso a los laboratorios de prácticas, taller y despachos. Tal y como se aprecia en el anexo I, se ha cubierto el laboratorio principal con dos sensores de medida de temperatura, humedad y luz, el pasillo con el sensor de aceleración y uno de temperatura, humedad y luz, y finalmente el laboratorio multiusos con un nodo sensor igual que el anterior.

Tal y como se ha especificado en el perfil de aplicación, cada nodo posee un identificador propio (*index*), por lo que al momento de conectarse a la red se muestra por pantalla, a través de la interfaz de usuario, los datos del nodo relacionados con los sensores que posee y la hora de conexión desde el inicio de la aplicación. En la figura 5.9 se muestra una captura de la interfaz de usuario donde se aprecia el momento de conexión de los 5 nodos sensores al nodo coordinador de la red, que es cuando han sido posicionados y encendidos en el lugar indicado. En dicho momento los nodos envían sus respectivos mensajes de identificación:

New Node: Acc X,Y	Index: 0003	00:00'00
New Node: Temp °C, %Hum, LDR	Index: 0005	00:01'16
New Node: Temp °C, %Hum, LDR	Index: 0004	00:03'36
New Node: Temp °C, %Hum, LDR	Index: 0007	00:04'54
New Node: Temp °C, %Hum, LDR	Index: 0002	00:06'00

Figura 5.9: Conexión de los nodos inalámbricos en las pruebas realizadas en el CEI.

Es interesante destacar que el nodo 3 ha sido el primero en encenderse y se ha conectado directamente al nodo coordinador, pero en el momento de encender el nodo 5, el nodo 3 ha utilizado como camino de conexión este último, permitiendo así tener un mejor enlace con el coordinador. Esto se aprecia mediante el empleo de la función de Mapa de Conexión, perteneciente a la Gestión de Red, en la cual se aprecia cómo se encuentran conectados los nodos inalámbricos entre si, como se ilustra también en el Anexo I.

HOPS	ROUTE
00	0002 → 0000
01	0003 → 0005 → 0000
01	0004 → 0005 → 0000
00	0005 → 0000
02	0007 → 0004 → 0005 → 0000

Figura 5.10: Mapa de conexión de los nodos obtenido desde la interfaz de aplicación.

En la figura 5.10 se aprecia que la columna de la izquierda muestra el número de saltos que realizan los mensajes entre la conexión de un nodo y el coordinador, mientras que la columna de la derecha se muestra el camino exacto de la comunicación, desde el nodo final pasando por los nodos *routers* hasta el nodo coordinador. Al estar configurados todos los nodos como FFD la versatilidad en las conexiones está asegurada, puesto que existen caminos auxiliares de comunicación entre los dispositivos en caso de pérdida momentánea de conexión.

A la vista del mapa de conexión entre los dispositivos, otro parámetro obtenido ha sido la calidad del enlace entre ellos, con el fin de conocer no sólo la forma en la que están conectados los nodos, sino también el nivel de calidad de dicha conexión. Para ello, se ha utilizado dicha función perteneciente a la Gestión de la Red, obteniendo los valores de LQI entre el nodo coordinador y los restantes nodos, tal y como se muestra en la figura 5.11.

Link Quality Table of Node 0000:

INDEX	LQI
0002	FF
0003	FF
0007	7E
0004	FE
0005	FF

Figura 5.11: Calidad de enlace entre el coordinador y los nodos de la red inalámbrica.

El valor indicativo del nivel de calidad de los enlaces se expresa en un rango de 0 a 255 (FF), donde este último valor representa el mejor nivel de calidad. Se aprecia como la conexión de datos entre los dispositivos en este primer escenario es bastante buena, sólo percibiéndose una disminución en el nodo que se encuentra más alejado del coordinador y, por lo tanto, con mayores obstáculos en la comunicación.

Dicho valor no se considera relevante cuando la tasa de transferencia de datos por segundos es baja, por lo que para el caso de la transmisión y recepción de mensajes relacionados con captura de sensores, registros de configuración, gestión de la red, etc., dicho nivel es más que suficiente para mantener el intercambio de información entre los dispositivos.

Sin embargo, en el caso de hacer uso de la funcionalidad de Programación Remota de microcontrolador, la calidad del enlace entre el coordinador y el nodo sensor es un valor crítico para que dicha tarea se lleve a cabo satisfactoriamente. Es por ello que se han realizado pruebas de programación de los dispositivos de forma remota para comprobar la eficiencia del proceso y el correcto funcionamiento del nodo inalámbrico luego de ser reconfigurado con un nuevo archivo fuente.

Se ha realizado primero una prueba de programación del nodo 2. Como se puede apreciar en el Anexo I, aunque la distancia entre el coordinador y dicho nodo no es corta, la calidad del enlace es muy buena, por lo que se ha podido reprogramar el procesador de forma exitosa y luego de ello se ha realizado un test de funcionalidad en la que se ha reiniciado de forma remota el nodo y se ha interrogado acerca de sus parámetros característicos, como conexión con la red inalámbrica, tipo de sensores que posee y captura de las magnitudes físicas. Como resultado de dicho test, se ha podido comprobar el correcto funcionamiento del dispositivo por lo que se ha verificado que todo el proceso de programación remota se ha llevado a cabo de forma adecuada.

Se ha realizado el mismo procedimiento con los nodos 5 y 3 y se ha comprobado el adecuado funcionamiento de los nodos inalámbricos. En el caso de los nodos 4 y sobre todo el nodo 7 la programación remota supone mayores dificultades. De hecho, se hace necesario repetir el proceso en varias ocasiones puesto que se detectan errores durante el proceso de reprogramación. Lo anterior permite concluir que el proceso de programación remota funciona de forma totalmente adecuada cuando el enlace entre los dispositivos es bueno o muy bueno, siendo estas, por lo tanto, las condiciones ideales de aplicación de dicha funcionalidad.

Por otro lado, se ha realizado la monitorización de los parámetros ambientes en las diferentes zonas del Centro de Electrónica Industrial con el fin de conocer los valores de temperatura, humedad y luz en el momento de realizar el experimento. En la figura 5.12 se muestra la captura de algunos de los valores más relevantes de monitorización obtenidos de cada nodo sensor, a través de la interfaz de usuario. Los valores de luminosidad son capturados a través de la resistencia dependiente de la luz incluida en la capa de sensores utilizada, y los valores son expresados directamente a partir de la medida obtenida del convertidor analógico digital, por lo que un valor de 4096 se considera como máxima luminosidad, mientras que 0 se considera como mínimas condiciones de luz.

Index:0002	Temp °C: 29,78	Hum %: 42,62	LDR: 3046	00:25'14
Index:0005	Temp °C: 30,28	Hum %: 35,40	LDR: 3349	00:25'51
Index:0004	Temp °C: 27,22	Hum %: 42,32	LDR: 3424	00:26'09
Index:0007	Temp °C: 27,00	Hum %: 51,46	LDR: 2818	00:26'17
Index:0002	Acc Y: 5452	Acc X:5810		00:26'33

Figura 5.12: Captura de valores obtenidos de los nodos sensores desplegados en el CEI.

Se puede apreciar que los valores de luminosidad son bastante buenos en todas las zonas de medida, mientras que existen algunas diferencias en los valores de temperatura y humedad. En el caso de la temperatura ambiente en el laboratorio principal el valor se encuentra entre 29 y 30°C debido principalmente a que posee mayor cantidad de equipos electrónicos y personas que en las otras áreas del Centro. En el caso del pasillo de despachos y el laboratorio multiusos los valores de temperatura se encuentran en torno a los 27°C. En este último caso la humedad relativa es la más equilibrada (entorno al 50%), mientras que en las demás zonas los valores son ligeramente inferiores.

Más allá del resultado de los sensores desplegados, se ha comprobado el correcto funcionamiento del bloque de Gestión de Datos, ya que se ha realizado de forma efectiva la adquisición de los datos de sensores analógicos y digitales, a partir de la configuración del tipo de medidas desde la interfaz de usuario y posteriormente la recepción de los datos en el panel de monitorización.

II.2 Despliegue N° 2.

En este segundo experimento se ha optado por posicionar los nodos en aquellas zonas de interés en las que no se había desplegado nodos en el experimento 1. Del mismo modo, se intento realizar una mayor cobertura de todo el Centro y así "dificultar" el escenario de aplicación con el fin de comprobar el performance del *testbed* en estas condiciones. Tal y como se puede apreciar en el Anexo II, se han desplegado nodos en el laboratorio de ordenadores, laboratorio de prácticas, taller, pasillo de despachos y laboratorio principal, respectivamente.

En el momento de posicionar los nodos y encenderlos nuevamente, cada uno de ellos envía el respectivo mensaje de identificación y reconocimiento hacia el coordinador, pero este último al detectar que dichos nodos ya forman parte de la red, retransmite un mensaje a la interfaz de aplicación indicando que ha habido una reconexión, junto con el *index* del nodo. Cada vez que ocurre una reconexión de un nodo perteneciente a la red inalámbrica, dicho mensaje es mostrado por pantalla, pero el usuario de la aplicación puede consultar en cualquier momento la Tabla de Reconexiones que

contiene la información del número de reconexiones realizadas por cada nodo miembro de la red hasta el momento, tal y como se muestra en la figura 5.13.

INDEX	RECONNECTIONS
0002	0002
0003	0001
0007	0003
0004	0001
0005	0002

Figura 5.13: Número de reconexiones de los dispositivos pertenecientes a la red.

También es necesario destacar que, aunque en el momento en el que un nodo se conecta por primera vez a la red se muestra la descripción del mismo, se puede consultar en cualquier momento la descripción de cada uno de los dispositivos conectados a la red inalámbrica con el fin de poder controlar a que *index* está asociado cada dispositivo. Por otro lado, desde la interfaz de usuario se puede conocer las direcciones MAC y *Short Adress* asociadas a cada nodo inalámbrico, con el fin de poder realizar tareas desde el nivel de red en caso de ser necesario. Estas opciones, detalladas en el capítulo 4, son visualizadas en el panel de monitorización de la interfaz de usuario tal y como se ilustra en la figura 5.14 y 5.15, respectivamente.

INDEX	TYPE OF NODE
0000	Coordinator
0002	Temp °C, %Hum, LDR
0003	Acc X,Y
0004	Temp °C, %Hum, LDR
0005	Temp °C, %Hum, LDR
0007	Temp °C, %Hum, LDR

Figura 5.14: Tipos de dispositivos conectados a la red inalámbrica.

En este segundo escenario de aplicación se ha podido comprobar que al aumentar el número de obstáculos entre los dispositivos (al estar más lejos los nodos entre sí, la incidencia de puertas, cristales y paredes era mucho mayor) ha hecho que el nivel de calidad del enlace entre los dispositivos y el nodo coordinador fuese más baja. Sin embargo, la conexión entre los dispositivos se mantuvo sin mayores inconvenientes, y se pudieron recopilar los respectivos valores de los sensores en cada laboratorio del Centro.

INDEX	MAC	ShortID
0000	000D6F00001D4ABF	0000
0002	000D6F00003DF160	023B
0003	000D6F00003DE79A	464D
0004	000D6F00003DE5A4	FBB3
0005	000D6F00003DEA8F	F646
0007	000D6F0000512C97	236C

Figura 5.15: Identificadores de los nodos pertenecientes a la red.

En las figuras 5.16 y 5.17 se muestra el mapa de conexión adoptado por los nodos inalámbricos y la calidad del enlace con el nodo coordinador. Se puede apreciar como el nodo 3 sirve de puente de conexión a los nodos 4, 5 y 7 al ser estos los nodos más alejados del coordinador (ver anexo II).

HOPS	ROUTE
00	0002 → 0000
01	0003 → 0002 → 0000
02	0004 → 0003 → 0002 → 0000
02	0005 → 0003 → 0002 → 0000
02	0007 → 0003 → 0002 → 0000

Figura 5.16: Mapa de conexión de los nodos en el despliegue N°2.

Link Quality Table of Node 0000:

INDEX	LQI
0002	FD
0003	DF
0007	46
0004	71
0005	6F

Figura 5.17: Calidad de enlace entre el coordinador y los nodos de la red inalámbrica.

Se ha podido monitorizar los valores de temperatura, humedad y luz en las zonas indicadas, obteniendo los resultados que se ilustran en la figura 5.18, capturados desde la interfaz de usuario de la aplicación.

Index:0002	Temp °C: 29,92	Hum %: 42,06	LDR: 2948	00:48'22
Index:0005	Temp °C: 28,22	Hum %: 40,64	LDR: 2010	00:48'51
Index:0004	Temp °C: 26,16	Hum %: 39,38	LDR: 2276	00:49'06
Index:0007	Temp °C: 23,84	Hum %: 59,24	LDR: 2238	00:49‘26

Figura 5.18: Captura de valores obtenidos de los nodos sensores en el despliegue N° 2.

Se ha podido comprobar cómo la medida de temperatura, humedad y luz proporcionada por el nodo 2 en el laboratorio principal es muy similar a las obtenidas en el primer despliegue realizado. Sin embargo, se puede apreciar como en el caso de los nodos 4 y 5 la medida de luz es diferente a las anteriores. Esto se debe a que, en el momento de realizar las pruebas, el laboratorio de prácticas y el taller no estaban siendo usados, por lo que las luces se encontraban apagadas. Sin embargo, estas dos salas son exteriores, por lo que poseen buena luminosidad, lo cual se comprueba a través del valor obtenido.

Así mismo, se ha monitorizado el laboratorio de ordenadores, que es una sala interior con solo luz artificial, lo cual se deduce del valor obtenido. Por otra parte, en el momento de realizar las pruebas el aire acondicionado de dicho laboratorio se encontraba activado, lo cual se aprecia mediante los valores de temperatura y humedad obtenidos, que son diferentes al resto de las salas monitorizadas.

Estas dos pruebas experimentales realizadas en el Centro de electrónica Industrial han servido para validar todas y cada una de las funcionalidades desarrolladas durante este proyecto de investigación, y se ha podido comprobar el correcto funcionamiento de la plataforma de integración HW-SW y el *testbed* para redes de sensores inalámbricas en un entorno real de aplicación, lo cual da como resultado la obtención de una herramienta eficaz para dar soporte a futuros desarrollos.

III. Pruebas adicionales realizadas.

Aparte de todos los experimentos presentados en los cuales se han comprobado las funcionalidades implementadas, el *testbed* ha sido utilizado para realizar pruebas relacionadas con otro proyecto de investigación desarrollado en el Centro de Electrónica Industrial. Dicho proyecto se basa en la creación de una herramienta para la reconstrucción de entornos *outdoors* e *indoors* con el fin de modelar todos los obstáculos presentes en los mismos y el efecto que puedan tener sobre el despliegue de redes de sensores inalámbricas. En tal sentido, se han realizado pruebas en el Centro de Electrónica Industrial con el fin de conocer el efecto de los obstáculos en entornos *indoors*, tales como paredes, cristales y puertas, en un despliegue de nodos inalámbricos y comparar estos datos con los proporcionados por la herramienta de

simulación. Las pruebas de basan principalmente en la medición de parámetros de intensidad de las señales de radio (RSSI) entre los dispositivos.

Se han realizado 3 tipos de despliegues en función de la posición de los nodos dentro del laboratorio y según los requisitos exigidos por dicho proyecto, tal y como se muestra en el Anexo III. En los primeros dos despliegues se han utilizado los 5 nodos y el coordinador del *testbed* para realizar las pruebas, mientras que en el caso del tercer despliegue la idea era ir variando la posición de uno de los nodos en función del coordinador con el fin de monitorizar la variación en la intensidad de la señal de radio, por lo que ha sido sólo necesario utilizar dos nodos.

En cada escenario se han obtenido las medidas de RSSI entre los dispositivos y el coordinador, los cuales han sido comparados con los valores calculados por la herramienta de simulación, obteniendo los resultados que se ilustran en el Anexo IV, en el que se puede comprobar las similitudes entre los datos teóricos y los prácticos obtenidos de forma experimental.

Los resultados de estas pruebas han sido incluidos en el articulo [He'12], del cual el autor del presente proyecto es coautor, y en el que se detalla el algoritmo utilizado para la simulación de entornos *indoors*, así como el procedimiento llevado a cabo para realizar las pruebas experimentales y el análisis de los resultados alcanzados.

Capítulo 6

CONCLUSIONES Y LÍNEAS FUTURAS

I. Conclusiones y aportaciones.

El presente proyecto de investigación se ha enmarcado dentro del diseño de una plataforma de integración *hardware-software* para un *testbed* de redes de sensores inalámbricas, cuyo principal objetivo es dar soporte a prototipos basados en nodos Cookies y facilitar el desarrollo de aplicaciones con dicha tecnología. Son muchos los avances que se pueden encontrar en la comunidad científica relacionados con la creación y mantenimiento de bancos de pruebas de nodos inalámbricos, con el fin de brindar una herramienta real como base para el estudio de diversas líneas de investigación en este campo, tanto desde el punto de vista *software*, como desde el nivel *hardware*.

Tal y como se ha descrito en el estado de la técnica, existen principalmente dos tipos de enfoques en el desarrollo de estos *testbeds* para redes de sensores. Por un lado aquellos en los que se parte de nodos comerciales muy extendidos, tales como los Telos o MicaZ, y se desarrollan aplicaciones en función de los recursos que dichos dispositivos proporcionan. Por otro lado se encuentran aquellos grupos de investigación que optan por diseñar sus propios nodos *hardware* a partir de los requisitos mínimos de las redes de sensores (flexibilidad, baja tasa de transferencia de datos, bajo consumo de energía, autonomía), para luego tener un mayor control de todo el sistema con el fin de adaptarse más fácilmente a los entornos de aplicación.

El Centro de Electrónica Industrial se incluye dentro de este último grupo, en el que se ha creado una completa plataforma *hardware* modular para redes de sensores inalámbricas basada en la flexibilidad, adaptabilidad y robustez de los nodos. Pero para poder explotar los recursos que esta plataforma posee, es necesario contar con una herramienta que permita gestionar las características de los nodos Cookies y que asegure la operatividad de los mismos dentro de los entornos de aplicación en los que serán desplegados.

Es por ello que en este proyecto de investigación se han diseñado y desarrollado los controladores necesarios para asegurar el correcto funcionamiento de todos los periféricos y funcionalidades incluidas en la plataforma *hardware*. En tal sentido, se han implementado bibliotecas *software* soporte del nodo Cookie, las cuales permitirán a los desarrolladores que trabajen con esta tecnología lograr un mayor nivel de abstracción y, con solo unos pocos ajustes, gestionar todos los recursos de la plataforma de forma rápida e intuitiva.

Además del aporte de dichas bibliotecas para el control del nodo inalámbrico, en este proyecto se ha expuesto la necesidad de ir un paso más adelante en dicha gestión. Se plantea la necesidad de contar con un mecanismo que permita estandarizar la forma en la que se comunican los nodos inalámbricos desde el nivel de aplicación y que fije las directrices para el entendimiento entre los dispositivos, con el objetivo final de

poder controlar de forma inalámbrica todos los recursos que estos poseen. Esto ha dado paso al diseño e implementación de un Perfil de Aplicación adaptado a las necesidades de la plataforma Cookies y con el que se logra una total gestión no solo desde el punto de vista del nodo, sino también de la red inalámbrica como un todo.

Para lograr tales objetivos ha sido necesario realizar ajustes no solo a nivel *software*, sino también desde el punto de vista *hardware* para poder realizar de forma efectiva las tareas de configuración y depuración remota. Funcionalidades tales como la programación de los dispositivos de manera inalámbrica, configuración remota, estado de la red, mapa de conexión, gestión de sensores y consumo de energía de los nodos han sido implementadas. El perfil de aplicación proporciona el medio con el que los nodos pertenecientes a la red inalámbrica pueden interactuar entre sí de manera efectiva, siendo capaces de intercambiar información relativa al entorno de aplicación y funcionar de forma conjunta para lograr las tareas de monitorización necesarias.

Gracias a estos desarrollos se ha logrado el objetivo final del presente proyecto, que es la creación de un *testbed* para redes de sensores inalámbricas basado en nodos Cookies, que permita explotar todas las características del perfil de aplicación diseñado y los recursos *hardware* disponibles. Para tal fin, se ha implementado una interfaz de usuario basado en un repertorio de comandos de ejecución desde una estación central de control, que permiten interactuar de forma rápida e intuitiva con la red inalámbrica.

Desde dicha interfaz se pueden utilizar todas las funciones desarrolladas en el perfil de aplicación para controlar los nodos inalámbricos, realizar configuraciones en tiempo real sobre la plataforma y monitorizar el entorno real de aplicación mediante los sensores implementados. Además de ello, es posible realizar ajustes a nivel de red en caso de ser necesario.

Se ha comprobado el correcto funcionamiento de todos y cada uno de los bloques funcionales diseñados no solo mediante el *testbed* implementado, sino también mediante proyectos en los que ha sido necesario realizar un control exhaustivo de la plataforma inalámbrica. Los entornos de aplicación en los que se han usado los desarrollos expuestos en este trabajo de investigación han representado grandes retos y dificultades que han contribuido a que finalmente se obtuviese un sistema totalmente robusto y adaptado a necesidades reales de aplicación.

Esta plataforma de integración *hardware-software* da lugar a facilitar el prototipado rápido de aplicaciones y la adaptabilidad de los nodos a diferentes entornos, con solo realizar unos pocos ajustes de configuración. El *testbed* proporciona un camino hacia la apertura de nuevas líneas de investigación dentro de las redes de sensores y el estudio de nuevos métodos para contribuir a la eficiencia en la monitorización de entornos reales de aplicación.

II. Líneas futuras.

La obtención de los objetivos planteados en el presente proyecto de investigación ha dado lugar a nuevos retos y propuestas dentro del ámbito de aplicación del *testbed* para redes de sensores inalámbricas. En primer lugar, se ha decidido diseñar una placa de procesamiento de ultra bajo consumo enfocada a aplicaciones en las cuales es necesario tener un equilibrio entre capacidad de procesamiento de datos y ahorro energético.

Esta placa tiene como objetivo dotar a la plataforma de una mayor flexibilidad, ya que su diseño está pensado en poder adaptarse a diferentes entornos de forma casi inmediata. Para lograr poner en práctica esta propuesta, la placa dispondrá de los dos dispositivos de procesamiento, FPGA y uC al igual que en el caso de la placa utiliza en este proyecto, pero con la ventaja de que será posible prescindir de cualquiera de los dos en el momento en que sea necesario en una determinada aplicación, optando por utilizar solo el uC o solo la FPGA, según los requisitos de procesamiento. Se han seleccionado tanto una FPGA como un uC de ultra bajo consumo para la implementación de la placa, y así proporcionar una plataforma con mucha más autonomía sin disminuir notablemente su capacidad de procesamiento.

Desde el punto de vista de la plataforma de integración *hardware-software* y el *testbed*, dicha placa llevará implementado los cambios realizados a nivel *hardware* en este proyecto, para que pueda ser totalmente compatible con los bloques funcionales de configuración y depuración de forma remota. La idea es realizar el *porting* de las funcionalidades desarrolladas y probar la efectividad de las mismas en la nueva placa, además de comprobar que dicho *porting* se puede realizar de forma rápida y fácil sin modificar la filosofía de diseño.

Por otro lado, se ha abierto una nueva línea de investigación relacionada con la conexión Ethernet de los nodos Cookies. En este sentido, la idea se centra en poder conectar los nodos que forman parte del *testbed* a una red paralela que sirva de respaldo para tareas de depuración y mantenimiento de la red principal inalámbrica.

En primer lugar, se plantea la opción de poder conectar el nodo coordinador mediante Ethernet para así poder tener acceso a éste de forma remota y poder monitorizar la red inalámbrica desde diferentes puntos.

En segundo lugar se propone como objetivo el poder conectar cualquier nodo que pertenezca a la red, a un punto de conexión Ethernet con el fin de contar con un medio alternativo distinto a la red principal para poder programar el nodo de forma inalámbrica, así como realizar acciones de configuración y mantenimiento. También se busca la utilización de dicha conexión como fuente de alimentación de los dispositivos, a través del denominado *Power Over Ethernet* (PoE), el cual permite obtener potencia de alimentación mediante el conector RJ-45. Este es un enfoque desde el punto de vista de pruebas de prototipos y experimentos en laboratorio,

puesto que ésta no sería una implementación viable en todos los entornos de aplicación.

Además de ello, se propone la creación de un nodo *Gateway* con implementación de WLAN con el fin de servir como puente de acceso entre redes basadas en IEEE 802.15.4 (ZigBee) y IEEE 802.11b (WI-FI). Con ello se dispondría de una red paralela inalámbrica, al igual que en el caso de Ethernet, para dar soporte a la red principal en tareas de programación, configuración y depuración de los dispositivos que forman parte del *testbed.* Del mismo modo, se tendría la capacidad de controlar a la red inalámbrica de forma remota desde diversos puntos de acceso.

Finalmente, estas implementaciones darían lugar a la creación de los respectivos bloques funcionales y bibliotecas para el control de los nuevos elementos a incluir, por lo que el desarrollo de las bibliotecas software es una línea de investigación que estará constantemente en evolución.

REFERENCIAS

[802.15.4] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification for Low-Rate Wireless Personal Area Networks (WPANs), IEEE Std. 802.15.4, 2003.

[ADUC841] Analog Devices, *http://www.analog.com/en/index.html*

[Albesa'07] J. Albesa, R. Casas, M. T. Penella, M. Gasulla, "REALnet: An Environmental WSN Testbed", Proceedings of the International Conference on Sensor Technologies and Applications, pp. 502-507, 2007.

[ATMEL] Atmel Zigit Product Manual. ATZB-900-B0 MHz Wireless Modules.

[Blumenthal'03] J. Blumenthal, M. Handy, F. Golatowski, M. Haase, D. Timmer-mann, "Wireless Sensor Networks - New Challenges in *Software* Engineering", Proceedings of 9th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), vol. 1, pp. 551-556, Portugal, 2003.

[CEI 2012] Cookies *Software* Libraries, User Manual & Start-up Guide. Centro de Electrónica Industrial, 2012.

[Chan'11] M. Doddavenkatappa, M. C. Chan and A. L. Amanda, "Indriya: A Low Cost, 3DWireless Sensor Network Testbed", Developed at School of Computing, National University of Singapore, Singapore, 2009.

[CitySense] Urban Scale WSN Testbed, *http://www.citysense.net/*

[Contiki] A. Dunkels, B. Gronvall, T. Voigt, "Contiki: a Lightweight and Flexible Operating System for Tiny Networked Sensors", In Proceedings of the 9th Annual IEEE International Conference on Local Computer Networks, Washington, DC, USA, pp. 455-462, October 2004.

[Daintree 1] Understanding ZigBee Application Framework. Daintree Networks, 2006.

[Daintree 2] Understanding IEEE 802.15.4 and ZigBee networking. Daintree Networks 2005.

[Handziski'06] V. Handziski, A. Kopke, A. Willig, A. Woliz, "TWIST: A Scalable and Reconfigurable Testbed for Wireless Indoor Experiments with Sensor Networks", Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing, pp. 63-70, 2006.

[He'12] D. He, G. Mujica, J. Portilla, T. Riesgo, "3D Indoor RF Propagation Simulation and Measurement for ZigBee Sensor Network", IEEE Globlecom 2012. Bajo revision.

[Hill'03] J. Hill, "System Architecture for Wireless Sensor Networks", PhD. Thesis, University of California at Berkeley, 2003.

[Hong'09] S. Hong, Y. Moon, S. Park, W. Kim, "Wireless Sensor Network Testbed for Real-time Sensor Monitoring", Proceedings of the Third International Conference on Sensor Technologies and Applications, IEEE Computer Society Washington, DC, USA, pp. 486-489, 2009.

[Imote2] Imote2 WSN Testbed, *http://www.ee.washington.edu/research/nsl/Imote/*

[Indriya] Indriya Testbed, *http://indriya.comp.nus.edu.sg/motelab/html/index.php*

[LiteOS] Q. Cao, T. Abdelzaher, J. Stankovic, T. He, "The LiteOS Operating System: Towards Unix Like Abstraction for Wireless Sensor Networks", In Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008), St. Louis, MO, USA, pp. 233-244, April 2008.

[Liu'10] Q. Liu, Z. Zhao, L. Cui, "A versatile heterogeneous sensor networks testbed", Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, New York, USA 2010.

[Mantis] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, H.R. Torgerson, "Mantis OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms", International Journal of Mobile Networks and Applications, vol.10, nº4, 2005.

[Microconv] Understanding the Serial Download Protocol. Microconverter Technical Note.

[MoteLab] Harvard Sensor Network Testbed, *http://motelab.eecs.harvard.edu*.

[Murty'08] R. Murty, G. Mainland, I. Rose, A. R.Chowdhury, A. Gosain, J. Bers, M. Welsh, "CitySense: A Vision for an Urban-Scale Wireless Networking Testbed", Proceedings of the IEEE International Conference on Technologies for Homeland Security, Waltham, pp. 583-588, May 2008.

[Polastre'05] J. Polastre, R. Szewczyk, D. Culler, "Telos: enabling ultra-low power wireless research", in Proc. IPSN, pp. 364-369, April 2005.

[Portilla'07] J. Portilla, A. de Castro, E. de la Torre, T. Riesgo, "A Modular Architecture for Nodes in Wireless Sensor Networks", Journal of Universal Computer Science (JUCS), vol. 12, nº 3, pp. 328-339, March 2006.

[Portilla'07] J. Portilla, T. Riesgo, A. Abril, A. de Castro, "Rapid prototyping for multi-application sensor networking", 12 November 2007, SPIE Newsroom. DOI: 10.1117/2.1200711.0851.

[Saruwatari'05] S. Saruwatari, T. Kashima, M. Minami, H. Morikawa, T. Aoyama, "PAVENET: A *Hardware* and *Software* Framework for Wireless Sensor Networks", Transaction of the Society of Instrument and Control Engineers, vol. E-S-1, no. 1, pp. 74-84, 2005.

[Sheu'08] J. P. Sheu, C. J. Chang, C. Y. Sun, W. K. Hu, "WSNTB: A Testbed for Heterogeneous Wireless Sensor Networks", Proceedings of the 1st IEEE International Conference on Ubi-Media Computing, pp. 338-343, 2008.

[Sheu'06] J.P. Sheu, B.K. Hsu, P.C. Lin, C.J. Chang, "Design and Implementation of A Lightweight Operating System for Wireless Sensor Networks", Proceeding of International Computer Symposium, Taipei, Taiwan, Dec. 2006.

[Telegesis 1] ETRX2 Module Product Manual. Telegesis Product Manual.

[Telegesis 2] TG-ETRXn-R302-AT-Commands. Telegesis Product Manual.

[Telegesis 3] Telegesis, *http://www.telegesis.com*

[TinyOS] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, "TinyOS: An Operating System for Sensor Networks", Available online: http://dx.doi.org/10.1007/3-540-27139-2_7.

[Tutornet] Tutornet Testbed, *http://enl.usc.edu/projects/tutornet/index.html.*

[TWIST] Indoor WSN Testbed, *http://www.twist.tu-berlin.de/wiki/WSN*

[Valverde'12] J. Valverde, V. Roselló, G. Mujica, A. Uriarte, J. Portilla, T. Riesgo, "Wireless Sensor Network for Environmental Monitoring: Application in a Coffee Factory", International Journal of Distributed Sensor Networks, vol. 2012, Article ID 638067, 2012. doi:10.1155/2012/638067.

[Werner-Allen'05] G. Werner-Allen, P. Swieskowski, M. Welsh, "MoteLab: A Wireless Sensor Network Testbed", Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, pp. 483-488, April 2005.

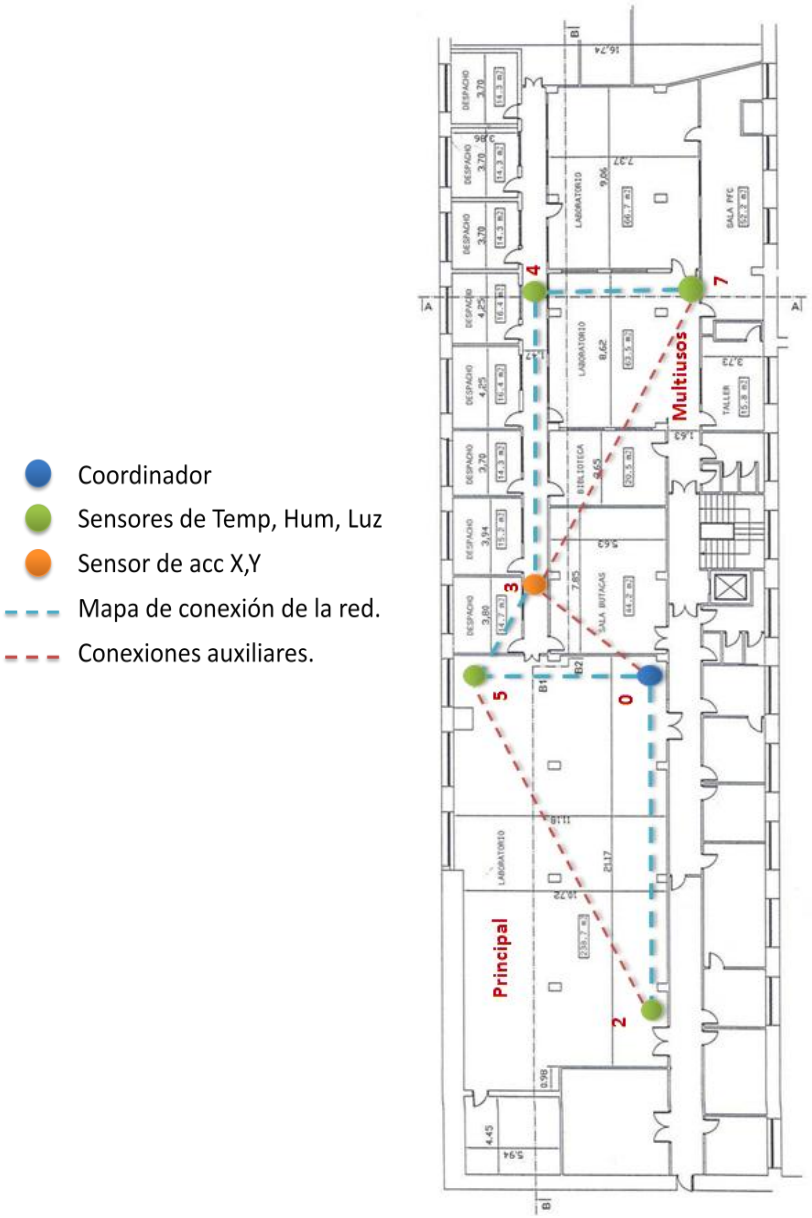
[Xilinx] Xilinx Inc, *http://www.xilinx.com/*

[ZigBee 1] ZigBee Pro specification, ZigBee Alliance 2007.

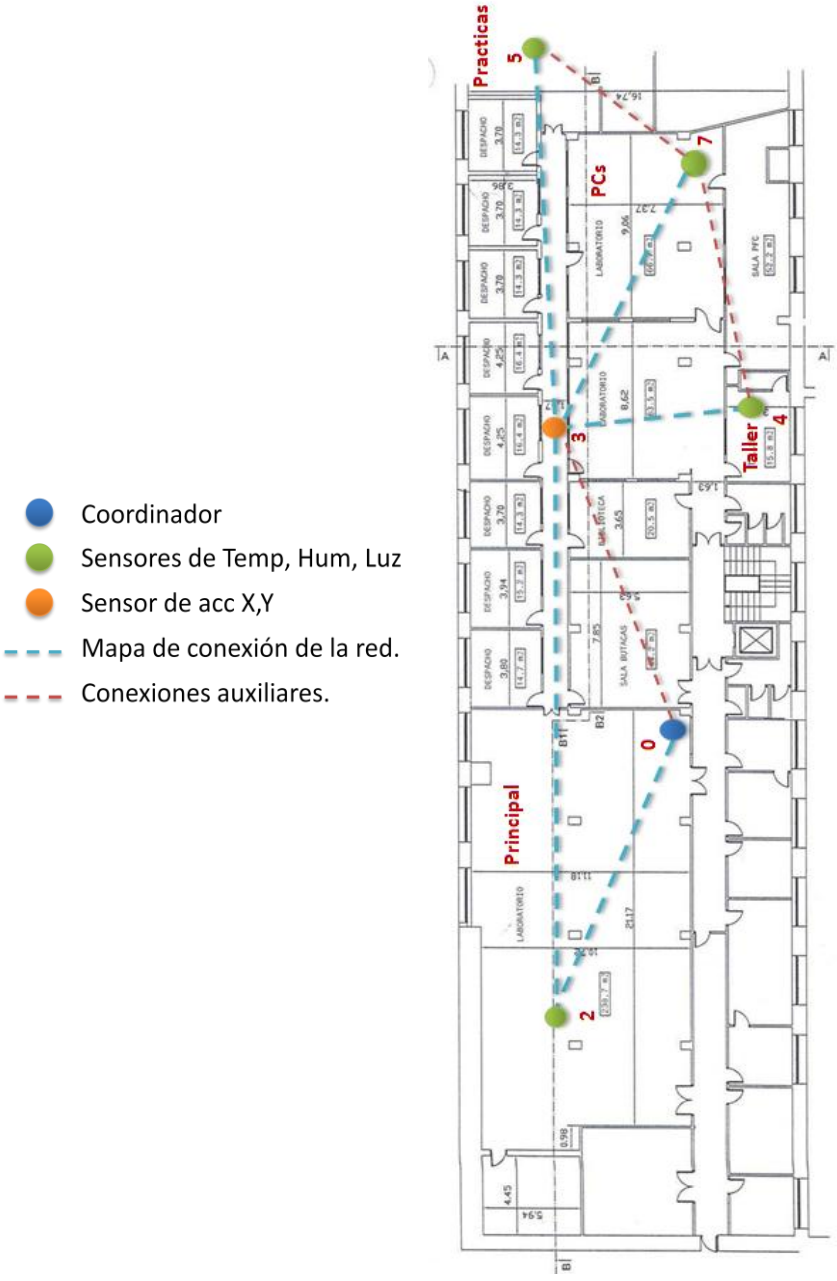
[ZigBee 2] ZigBee Cluster Specification. ZigBee Alliance 2008.

ANEXOS

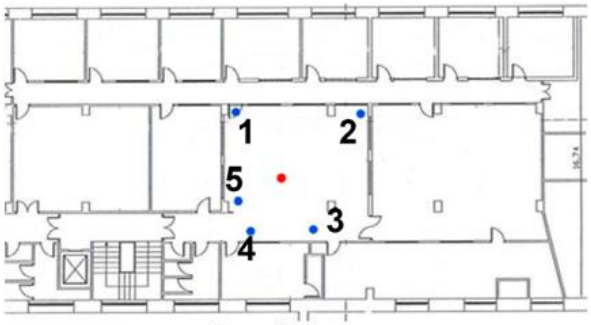
I. Despliegue N°1 realizado en el CEI.



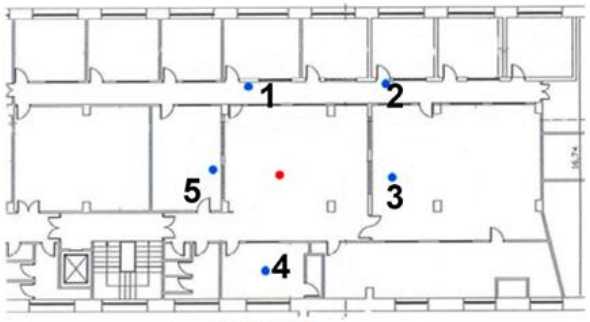
II. *Despliegue N°2 realizado en el CEI.*



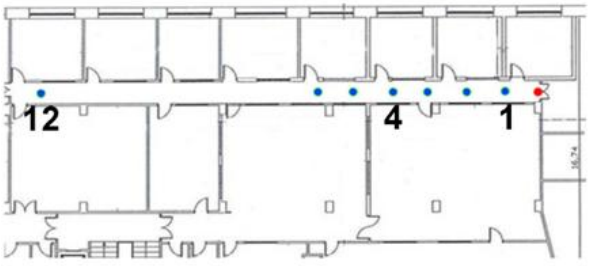
III. Despliegues para la medida de valores RSSI en el CEI.



Scenario A

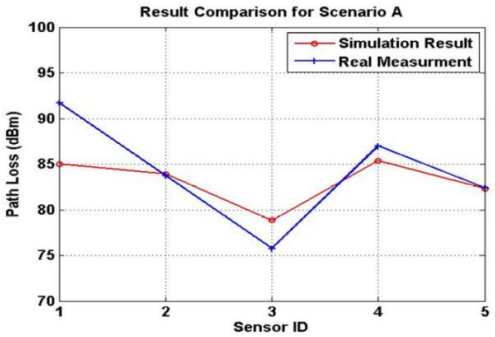


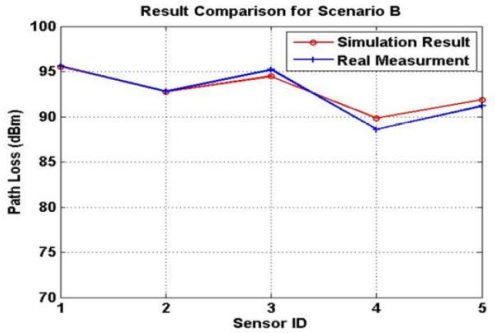
Scenario B

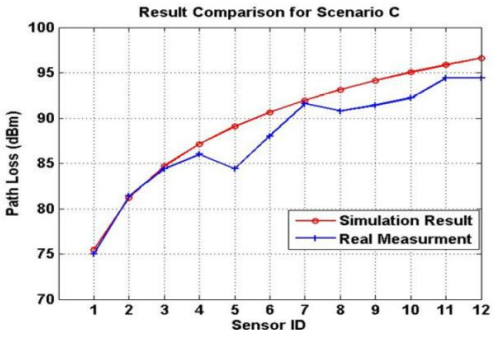


Scenario C

IV. Graficas comparativas de valores RSSI teóricos y prácticos.







ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS