

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



PROYECTO FIN DE CARRERA

**SISTEMA DE CONTROL
REMOTO PARA
APLICACIONES DOMÓTICAS A
TRAVÉS DE INTERNET**

Mario Rodríguez Cerezo

Octubre 2014

SISTEMA DE CONTROL REMOTO PARA APLICACIONES DOMÓTICAS A TRAVÉS DE INTERNET

AUTOR: Mario Rodríguez Cerezo
TUTOR: Nasib Fahim Fernández
PONENTE: Guillermo González de Rivera Peces

HCTLab
Dpto. Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Octubre 2014

Agradecimientos

En primer lugar, quiero empezar dando las gracias a mis padres, Antonio y María del Pilar, por todo lo que han hecho por mí durante toda su vida, por su apoyo constante y por haber estado a mi lado siempre. Asimismo, me gustaría dar las gracias al resto de mi familia, en concreto me acuerdo de mis abuelos Pedro (*in memoriam*) y Patricia, y de mis abuelos Antonio (*in memoriam*) y Pepita. También quiero dar las gracias a mis tíos, Eusebio y María Luisa, y a mis primos Patricia y Pablo. Gracias también a Tina y Paloma.

Muchas gracias a mi tutor, Nasib Fahim, y a Guillermo González de Rivera por la ayuda que me han prestado y por haberme dado la oportunidad de realizar este proyecto. También quiero agradecer al resto de personas que me han ayudado en determinadas fases del proyecto: Fernando López, Julio, Javi y David.

No quiero olvidarme de mis amigos de la ‘Urba’: Willy, Pablo, Victor, Daniel y Diego, a quien quiero agradecer la ayuda que me ha brindado durante la carrera.

Quiero agradecer de forma especial a mis compañeros y amigos de clase Guille, Porras, Ángel y Pencho por los grandes momentos compartidos durante estos seis años. Especial mención a Juanma y David, con los cuales, además de buenos momentos, he compartido muchas horas de prácticas.

Finalmente, me gustaría dar las gracias a todas las personas que han estado a mi lado durante el transcurso de mi vida.

Mario Rodríguez Cerezo

Madrid 2014

Resumen

En el presente proyecto fin de carrera se desarrolla un sistema domótico completo para el control de diferentes elementos de una vivienda y para la obtención de parámetros de interés. El control del sistema se realiza remotamente a través de Internet. El PFC se enmarca dentro de un proyecto del HCTLab para disponer de un sistema domótico avanzado que se vaya ampliando y perfeccionando con el tiempo.

En primer lugar, se realiza un análisis del estado del arte para conocer los tipos de sistemas domóticos existentes y poder seleccionar la alternativa más adecuada con el fin de cumplir los objetivos establecidos. Tras esto, se estudia el funcionamiento de las tecnologías y los equipos que se emplean a lo largo del proyecto.

A continuación se realiza el diseño del sistema, es decir, se crea la red domótica de control que conecta inalámbricamente todos los nodos que formen parte del sistema, estableciéndose los parámetros de comunicación y de configuración. Como elementos fundamentales para crear esta red inalámbrica se usan módulos de radiofrecuencia (XBee). El desarrollo del sistema completo implica, además, el diseño y construcción de un nodo de control de red y de otro nodo periférico, de tal manera que éste último se pueda conectar y pueda controlar elementos habituales en una vivienda (luces, ventiladores, calefacción, etc.) y que, a través de sensores, sea capaz de recoger información de interés para el usuario (temperatura, humedad, etc.).

El nodo de control va conectado a Internet y sirve de pasarela entre la red domótica inalámbrica y la red de internet. Se desarrolla una aplicación web que sirve de interfaz para que el usuario acceda al sistema y lo controle, permitiendo interactuar con cada uno de los dispositivos. El elemento fundamental del nodo de control es una tarjeta con microprocesador, denominada BeagleBone, que gobierna el sistema y donde se encuentra el software desarrollado.

Por último se evalúa el sistema verificando que funciona y cumple los objetivos establecidos al inicio del proyecto.

- **Palabras claves:** sistema domótico, módulos XBee, sensores, actuadores, sistema remoto, aplicación web, BeagleBone.

Abstract

The aim of this project is to develop a complete home automation system to control different elements of the home and to obtain parameters of interest. The system will be controlled remotely via the internet. It forms part of a project of the HCTLab group to provide an advanced home automation system which will be expanded and improved over time.

Firstly, we carry out research on the systems currently in use to obtain an overview of the home automation market and select the best alternatives in order to meet the objectives. After that, we will study the technologies and equipment to be used on the project.

Then, we will design the wireless home automation network, which is formed of the nodes which are part of the system. We will set up configuration and communication parameters. For key elements in creating the network we will use XBee wireless communication modules. The development of the entire system also involves the design and construction of a controller node network and a remote node, so that the latter can be connected and can control common elements in a home (lights, fans, heating, etc.) and which, through sensors, is able to collect information of interest to the user (temperature, humidity, etc.).

The controller node will have an internet connection and will serve as a gateway between the wireless home automation network and the Internet network. We shall develop a web application which provides a user interface to control the home automation system. The key element of the controller node is a card (called BeagleBone) with a microprocessor, where the developed software is running.

Finally, the system is evaluated to verify that it works and that the project's objectives have been met.

- **Key words:** home automation system, XBee modules, sensors, actuators, remote system, web application, BeagleBone.

Índice general

| | |
|--|----------|
| Agradecimientos | v |
| Resumen | vii |
| Abstract | ix |
| I Capítulos | 1 |
| 1. Introducción | 3 |
| 1.1. Motivación y Antecedentes | 3 |
| 1.2. Objetivos | 4 |
| 1.3. Marco de Trabajo | 4 |
| 1.4. Organización de la Memoria | 5 |
| 2. Estado del Arte | 7 |
| 2.1. Introducción | 7 |
| 2.2. Aplicaciones de la Domótica | 8 |
| 2.2.1. Confort | 8 |
| 2.2.2. Seguridad | 9 |
| 2.2.3. Comunicaciones | 9 |
| 2.2.4. Gestión de la Energía | 10 |
| 2.3. Componentes del Sistema | 10 |
| 2.3.1. Sensores | 11 |
| 2.3.2. Actuadores | 12 |

| | | |
|------------|--|-----------|
| 2.3.3. | Controladores | 12 |
| 2.3.4. | Otros dispositivos | 12 |
| 2.4. | Arquitectura | 13 |
| 2.4.1. | Arquitectura Centralizada | 13 |
| 2.4.2. | Arquitectura Distribuida | 14 |
| 2.4.3. | Arquitectura Mixta | 14 |
| 2.5. | Topologías | 15 |
| 2.6. | Medios de Transmisión | 16 |
| 2.7. | Tecnologías y Protocolos de Comunicación | 17 |
| 2.7.1. | LonWorks | 18 |
| 2.7.2. | Konnex | 19 |
| 2.7.3. | X10 | 21 |
| 2.7.4. | Insteon | 22 |
| 2.7.5. | Z-Wave | 22 |
| 2.7.6. | Otras tecnologías inalámbricas | 23 |
| 3. | Descripción general del sistema | 27 |
| 3.1. | Implementación del sistema de comunicaciones | 32 |
| 3.1.1. | IEEE 802.15.4 y Protocolo ZigBee | 32 |
| 3.1.2. | Red de ZigBee | 33 |
| 3.1.2.1. | Tipos de dispositivos | 33 |
| 3.1.2.2. | Topologías de Red | 34 |
| 3.1.2.3. | Red implementada. | 35 |
| 3.1.3. | Módulos Xbee | 35 |
| 3.1.3.1. | Selección de Módulo | 38 |
| 3.1.4. | Módulos XBee PRO Serie1 | 39 |
| 3.1.4.1. | Características eléctricas | 41 |
| 3.1.4.2. | Interfaces utilizadas | 41 |
| 3.1.4.2.1. | Comunicación serie (UART) | 41 |
| 3.1.4.2.2. | ADCs y DOs | 43 |
| 3.1.5. | Modos de comunicación | 44 |

| | | |
|------------|---|-----------|
| 3.1.5.1. | Modo transparente | 44 |
| 3.1.5.2. | Modo API | 46 |
| 3.1.6. | Tramas API | 48 |
| 3.1.6.1. | Estructura general de las tramas | 48 |
| 3.1.6.2. | Tipos de comandos API | 50 |
| 3.1.6.3. | Implementación | 53 |
| 3.1.6.3.1. | Activación de actuadores | 53 |
| 3.1.6.3.2. | Lectura de sensores | 56 |
| 3.1.7. | Configuración de los Módulos XBee | 59 |
| 3.1.7.1. | Tarjetas de desarrollo | 60 |
| 3.1.7.2. | Software X-CTU | 61 |
| 3.1.7.3. | Parámetros de Configuración | 65 |
| 4. | Unidades remotas | 69 |
| 4.1. | Hardware | 71 |
| 4.1.1. | Componentes | 71 |
| 4.1.1.1. | Sensor de temperatura | 72 |
| 4.1.1.2. | Sensor de humedad | 73 |
| 4.1.1.3. | Sensor de luminosidad | 74 |
| 4.1.1.4. | Relés | 76 |
| 4.1.1.5. | Regulador de tensión | 77 |
| 4.1.2. | Construcción del hardware | 79 |
| 4.1.2.1. | PCB del módulo Xbee | 80 |
| 4.1.2.2. | PCB de los sensores | 83 |
| 4.1.2.3. | PCB de los actuadores | 84 |
| 4.2. | Software | 85 |
| 5. | Unidad central de control | 87 |
| 5.1. | Hardware | 89 |
| 5.1.1. | Puertos y conectores de expansión | 93 |
| 5.1.2. | Placa adaptadora | 95 |

| | |
|--|------------|
| 5.2. Software | 97 |
| 5.2.1. Software esclavo | 100 |
| 5.2.1.1. Comunicación serie | 101 |
| 5.2.1.2. Paquetes API | 104 |
| 5.2.2. Software de control | 105 |
| 5.2.2.1. Servidor web | 106 |
| 5.2.2.2. Aplicación web | 107 |
| 6. Pruebas y Resultados | 113 |
| 6.1. Comunicaciones UART | 113 |
| 6.2. Sistema de comunicaciones | 114 |
| 6.3. Circuitos impresos | 117 |
| 6.4. Pruebas de Alcance | 118 |
| 6.5. Aplicación y sistema completo | 119 |
| 7. Conclusiones y trabajo futuro | 125 |
| 7.1. Conclusiones | 125 |
| 7.2. Trabajo futuro | 126 |
| Bibliografía | 128 |
| | |
| II Apéndices | 131 |
| | |
| A. IEEE 802.15.4 y Protocolo ZigBee | 133 |
| A.1. Capas | 136 |
| A.2. Seguridad | 141 |
| | |
| B. Modos de operación de un Xbee | 143 |
| | |
| C. Comandos API | 149 |
| C.1. Comando AT | 151 |
| C.2. Transmisión de datos y recepción de datos | 151 |

| | |
|---|------------|
| D. Configuración de los módulos de comunicación XBee | 155 |
| D.1. XBee coordinador | 157 |
| D.2. XBee dispositivo final | 158 |
| E. Especificaciones módulo XBee | 159 |
| F. Esquemáticos | 163 |
| G. Diagrama de secuencia | 169 |
| H. Presupuesto | 173 |
| I. Pliego de condiciones | 177 |

Índice de figuras

| | |
|--|----|
| 2.1. Campos de la Domótica | 8 |
| 2.2. Arquitectura Centralizada | 13 |
| 2.3. Arquitectura Distribuida | 14 |
| 2.4. Arquitectura Mixta | 14 |
| 2.5. Tipos de Topología | 16 |
| 2.6. Comparativa estándares de comunicaciones inalámbricas | 24 |
| 3.1. Esquema del sistema domótico desarrollado | 30 |
| 3.2. Diagrama de secuencia del sistema | 31 |
| 3.3. Topologías de ZigBee | 34 |
| 3.4. Tipos de antenas | 37 |
| 3.5. Diagrama esquemático de los pines del módulo Serie1 | 40 |
| 3.6. Conexiones mínimas requeridas | 42 |
| 3.7. Ejemplo de la transmisión en serie del byte 0x1F según la configuración establecida | 43 |
| 3.8. Estructura de la trama | 48 |
| 3.9. Estructura API específica | 49 |
| 3.10. Estructura API de Respuesta de Comando Remoto | 52 |
| 3.11. Estructura API de Respuesta de Comando Remoto | 53 |
| 3.12. Cabecera. Número de muestras y máscara del canal | 58 |
| 3.13. Cuerpo. Estados I/O digitales y ADC | 58 |
| 3.14. Tarjetas de desarrollo XBIB-U-DEV y XBIB-R-DEV | 60 |
| 3.15. Pantalla de inicio de software X-CTU | 62 |
| 3.16. Configuración general del módulo | 63 |

| | |
|--|-----|
| 3.17. Pestaña Terminal. Ejemplo de comunicación en Modo Transparente | 64 |
| 3.18. Pestaña Terminal. Ejemplo de comunicación API (Transmisión de datos) | 64 |
| 3.19. Pestaña Terminal. Ejemplo de comunicación API (Comandos Remotos) | 65 |
| 4.1. Esquema de la unidad remota desarrollada | 71 |
| 4.2. Esquemático del circuito de acondicionamiento del sensor de temperatura | 73 |
| 4.3. Esquemático del sensor de humedad | 74 |
| 4.4. Circuito de acondicionamiento de la fotorresistencia | 75 |
| 4.5. Esquemático del circuito de acondicionamiento para los relés | 77 |
| 4.6. Esquemático del circuito de alimentación y regulación | 79 |
| 4.7. Unidad remota construida | 80 |
| 4.8. PCB del Módulo Xbee | 81 |
| 4.9. PCB de los sensores | 83 |
| 4.10. PCB de los actuadores | 84 |
| 4.11. Diagrama de flujo de la unidad remota | 86 |
| 5.1. Esquema de la unidad central de control | 88 |
| 5.2. Diagrama de bloques del sistema | 91 |
| 5.3. Componentes y conectores de la tarjeta BeagleBone | 92 |
| 5.4. Placa adaptadora BeagleBone-XBee | 96 |
| 5.5. PuTTY | 99 |
| 5.6. Diagrama de flujo del funcionamiento general | 108 |
| 5.7. Aplicación web | 111 |
| 6.1. Paquetes API en la unidad central | 115 |
| 6.2. Plano de la vivienda donde se realizaron las pruebas | 119 |
| 6.3. Implementación del sistema | 120 |
| 6.4. Aplicación Web I | 121 |
| 6.5. Aplicación Web II | 121 |
| 6.6. Aplicación Web III | 122 |
| 6.7. Aplicación Web IV | 122 |
| A.1. Canales del estándar IEEE 802.15.4 | 135 |

| | |
|---|-----|
| A.2. Capas de IEEE 802.15.4 y ZigBee | 136 |
| A.3. Recorrido de los datos a través de las capas | 137 |
| A.4. Relación de la capa MAC con sus vecinas | 138 |
| A.5. Relación de la capa de Red con sus vecinas | 140 |
| B.1. Modos de operación del módulo XBee | 145 |
| C.1. Estructura trama comando AT | 151 |
| C.2. Estructura API de Transmisión de Datos | 152 |
| C.3. Estructura API de Recepción de Datos | 153 |
| C.4. Estructura API de Respuesta de Resultado de la Transmisión | 153 |
| F.1. Esquemático del PCB de los sensores | 165 |
| F.2. Esquemático del PCB de los actuadores | 166 |
| F.3. Esquemático del PCB que incorpora el módulo XBee | 167 |
| G.1. Diagrama de secuencia (1ª Parte) | 171 |
| G.2. Diagrama de secuencia (2ª Parte) | 172 |

Índice de tablas

| | |
|---|-----|
| 2.1. Comparativa de protocolos | 23 |
| 2.2. Comparativa Tecnologías inalámbricas | 25 |
| 3.1. Tabla comparativa Serie1 vs Serie2 (modelos de potencia regular) . . | 36 |
| 3.2. Pines de los módulos XBee Serie1 | 40 |
| 3.3. Características eléctricas del XBee | 41 |
| 3.4. Configuración de los parámetros del puerto serie del XBee | 43 |
| 3.5. Comandos API | 51 |
| 3.6. Configuración de los XBee de la red | 66 |
| 4.1. Sensores de temperatura candidatos a formar parte de la electrónica . | 72 |
| 4.2. Sensores de humedad candidatos a formar parte de la electrónica . . . | 73 |
| 4.3. Niveles de iluminación | 74 |
| 4.4. Reguladores de tensión candidatos a formar parte de la electrónica . . | 78 |
| 5.1. Pines del conector P8 | 94 |
| 5.2. Pines del conector P9 | 95 |
| 5.3. Conexiones BeagleBone-XBee | 97 |
| 5.4. Modos de operación del conector P9 | 102 |
| 5.5. Descripción de los bits de configuración | 103 |
| 6.1. Pruebas recepción de paquetes | 116 |
| 6.2. Mediciones del consumo y las tensiones | 117 |
| 6.3. Pruebas de Alcance | 118 |
| D.1. Parámetros de configuración XBee-unidad central | 157 |

| | |
|---|-----|
| D.2. Parámetros de configuración XBee-unidad remota | 158 |
| E.1. Tabla comparativa XBee vs XBee PRO | 161 |

Parte I

Capítulos

Capítulo 1

Introducción

1.1. Motivación y Antecedentes

El hombre siempre ha buscado controlar su entorno. Desde el sílex de las primeras herramientas de piedra hasta el silicio de nuestros ordenadores, el objetivo del hombre ha sido la supervivencia y mejorar su calidad de vida. La necesidad de cobijo y protección frente a un entorno hostil, ha hecho que la vivienda experimentara una gran evolución.

Con el desarrollo tecnológico de las últimas décadas nuestros hogares y oficinas han sufrido una transformación radical. Los avances en el mundo de los ordenadores, la electrónica y las comunicaciones han repercutido de forma directa en nuestra calidad de vida, haciendo que los edificios donde vivimos y trabajamos sean más seguros, cómodos y eficientes.

En los últimos años se ha buscado el desarrollo de tecnologías que permitan la gestión y el control de los distintos elementos que conforman una vivienda, creando el concepto de "Domótica". En la actualidad la domótica está en auge y su presencia en los hogares es cada vez mayor. En España con la actual crisis inmobiliaria se ha frenado la expansión de la domótica, sin embargo se espera que en los próximos años se incremente su ritmo de crecimiento [1].

La idea de este proyecto surge del interés del laboratorio HCTLab, perteneciente a la Escuela Politécnica (U.A.M), de disponer de un sistema domótico básico que pueda ser mejorado y ampliado en el futuro.

1.2. Objetivos

El objetivo de este proyecto ha sido el planteamiento, diseño, construcción y prueba de un sistema domótico que permita el control de una red de sensores y actuadores y que sirva de base para futuras ampliaciones. La red es flexible y fácilmente ampliable, teniendo la capacidad de ser desplegada en cualquier lugar de una casa. Su principal objetivo es la recogida de datos como la temperatura, la humedad o la luminosidad, y a su vez la capacidad de activar o desactivar elementos habituales de una vivienda como la calefacción, el riego del jardín, luces, persianas etc.

Para cumplir lo anterior, se ha analizado el estado del arte con el fin de conocer los tipos de sistemas domóticos existentes, los medios de transmisión y los protocolos de comunicación. Tras esto se ha seleccionado la alternativa más adecuada en función de las posibilidades de implementarla y del uso que se le iba a dar.

A continuación, se ha implementado la red domótica que conecta todos los nodos que forman parte del sistema, estableciéndose el modo y los parámetros de la comunicación. Se ha diseñado y construido el hardware necesario para disponer de estos nodos y al mismo tiempo se ha desarrollado el software que controla la red, es decir las instrucciones y los mensajes de comunicación entre nodos.

El sistema creado debe tener la posibilidad de ser controlado remotamente (mediante ordenador, smartphone u otros dispositivos con conexión a Internet), para ello el sistema debe estar conectado a la red. Además se ha desarrollado una aplicación sencilla que permite al usuario interactuar con los diferentes elementos de la instalación.

Finalmente se ha comprobado el buen funcionamiento del sistema y se ha desarrollado un mecanismo de protección frente a algunos posibles fallos.

1.3. Marco de Trabajo

El Proyecto se ha desarrollado en Grupo de Investigación HCTLab (Human Computer Technology Laboratory), perteneciente al departamento de Tecnología Electrónica y de las Comunicaciones de la Escuela Politécnica Superior (EPS) en la Universidad Autónoma de Madrid (UAM).

El laboratorio, fundado en Octubre de 2002, centra sus líneas de investigación en ámbitos multidisciplinares, principalmente en torno a tecnologías que relacionan a las personas con las máquinas o, más concretamente, con los sistemas basados en procesador.

El grupo de investigación lo componen varios profesores universitarios que desarrollan su actividad de I+D en áreas muy diversas. Éstas van desde el diseño de circuitos impresos y aplicaciones software basadas en sistemas embebidos (microcontroladores, FPGAs, etc.) hasta el desarrollo de tecnologías de accesibilidad.

1.4. Organización de la Memoria

La memoria de este proyecto fin de carrera se estructura en ocho capítulos:

- **Capítulo I:** Se explica la motivación que ha llevado a la realización de este proyecto, los objetivos que se aspiran a alcanzar y el marco de trabajo en el que se realiza el proyecto.
- **Capítulo II:** Se realiza un estudio del estado del arte de los sistemas domóticos, analizando desde las soluciones que ofrecen hasta los dispositivos que los forman o las tecnologías de redes que emplean.
- **Capítulo III:** Se realiza la descripción general del sistema, es decir, se presenta el sistema domótico que se implementará introduciendo las características fundamentales del sistema y los principales componentes que lo integran. Al mismo tiempo se detalla en profundidad todo lo relacionado con la tecnología empleada para crear la red domótica, así como con los módulos de radiofrecuencia (hardware, parámetros de configuración, paquetes de comunicación, etc. . .).
- **Capítulo IV:** Se detallan las características de hardware del nodo remoto y el proceso de construcción.
- **Capítulo V:** Se detallan las características de hardware de la unidad central y se describe el software desarrollado, incluida la aplicación de control.
- **Capítulo VI:** Se muestran y se analizan las pruebas realizadas para evaluar el sistema domótico desarrollado.
- **Capítulo VII:** Se presentan las conclusiones del proyecto y las posibles mejoras futuras.

Capítulo 2

Estado del Arte

En este capítulo se ofrece una panorámica de los sistemas domóticos existentes. Se presentan sus componentes, las arquitecturas y topologías posibles, los medios de transmisión que utilizan y las tecnologías y protocolos de comunicación que emplean.

2.1. Introducción

El término Domótica hace referencia etimológicamente a domus (que significa casa/hogar en latín). La Domótica se define como el conjunto de sistemas capaces de automatizar una vivienda. Cualquier sistema domótico se basa en una red integrada por diferentes elementos de un hogar que permite la supervisión y el control remoto de aparatos, sistemas eléctricos y electrónicos, todo ello con el fin de ofrecer al usuario nuevos servicios en el ámbito de su vivienda que le aporten ventajas en aspectos relacionados con la seguridad, el confort, la gestión de la energía y las comunicaciones.

Una instalación Domótica es un Sistema Inteligente de Control (S.I.C) formado por redes de comunicaciones -cableadas o inalámbricas- que conectan un conjunto de equipos y que permiten obtener información, procesarla y actuar sobre el entorno.

Conviene diferenciar entre la Domótica y la Inmótica, ya que en ocasiones se emplean indistintamente ambos términos erróneamente [2]. Como se ha dicho anteriormente la Domótica hace referencia al entorno del hogar (estando sus aplicaciones reservadas a la vivienda como elemento aislado), mientras que la Inmótica hace referencia a

sistemas de control y automatización de edificios, tanto de uso terciario como industrial (oficinas, bloques de viviendas, colegios, etc.). Aunque el ámbito de la Inmótica supera a la Domótica, esto no significa que los productos y las tecnologías que se usan sean diferentes, ya que, de hecho, la arquitectura del sistema es muy similar en ambos casos.

2.2. Aplicaciones de la Domótica

Una instalación Domótica puede ofrecer multitud de servicios que aportan valor añadido a una vivienda. Sus aplicaciones se suelen dividir en cuatro grandes áreas [3], aunque como se observa en la figura 2.1, es posible que una aplicación ofrezca servicios en más de un ámbito al mismo tiempo.



Figura 2.1: Campos de la Domótica

2.2.1. Confort

Las aplicaciones de esta área tienen como finalidad simplificar las tareas del hogar, creando automatismos o incrementando las posibilidades de control, para mejorar la comodidad en una vivienda. Algunas de estas aplicaciones podrían ser:

- Apagado general de toda la iluminación de una vivienda en un único pulsador, que permita asegurarse rápidamente que la iluminación queda apagada al salir de la casa.

- Activación de la iluminación en función de la luminosidad ambiental o por detección de presencia, permitiendo el encendido y apagado de las luces según el movimiento de las personas por la vivienda.
- Control de cualquier dispositivo electrónico de la vivienda a través de un único mando a distancia o a través de internet.
- Integración del portero automático en el teléfono o televisor, permitiendo al usuario no tener que levantarse a abrir.

2.2.2. Seguridad

La gestión de la seguridad y la vigilancia constituye un capítulo esencial en los servicios que puede ofrecer la Domótica. Las aplicaciones de esta área se pueden integrar en tres campos:

- Seguridad de Bienes: control de acceso con reconocimiento o identificación de usuarios, control de presencia, detección de personas no autorizadas, violación del espacio privado y vigilancia electrónica.
- Seguridad de Personas: teleasistencia y telemedicina de personas mayores o enfermos mediante el uso de pulsadores de pánico o de ayuda que generan alarmas locales y remotas.
- Alarmas Técnicas: detección de averías, posibles errores en la instalación o mal uso involuntario de los sistemas, aviso de escapes de agua, gas, humo, etc. Estos avisos pueden indicar el punto exacto de la vivienda en donde se encuentra la incidencia e incluso, si hay un escape de gas, cortar el suministro a través de la electroválvula del gas. Las alarmas también pueden detectar la pérdida de suministro eléctrico e iniciar su recuperación, un aviso de humo puede activar la apertura de ventanas y persianas, etc.

2.2.3. Comunicaciones

La gestión de las comunicaciones permite recibir, enviar, almacenar y procesar la información. Algunas de estas aplicaciones podrían ser:

- Control y gestión a distancia de la instalación domótica.
- Conocimiento en tiempo real del estado del hogar.
- Transmisión de alarmas, mensajes, alertas, etc.
- Comunicación exterior a través de servicios como internet (entretenimiento, teleasistencia, compras, telebanca, teleformación, correo electrónico, etc)

2.2.4. Gestión de la Energía

La gestión de la energía permite racionalizar los diferentes consumos energéticos (agua, gas natural y electricidad) en función de las tarifas energéticas, la potencia contratada, las condiciones ambientales, etc. La gestión de la energía se puede realizar mediante programadores, temporizadores, termostatos, etc. Algunas aplicaciones podrían ser:

- Programación de la climatización, permitiendo optimizar el consumo y la vez ofrecer una temperatura de confort en función de las zonas de la vivienda.
- Programación de equipos domésticos, facilitando la utilización eficiente de lavadoras, secadoras, hornos, etc.
- Gestión de tarifas, encendiendo los equipos domésticos en las zonas horarias con una tarifa más económica.
- Racionalización de la carga eléctrica, desconectando equipos no prioritarios en función del consumo en un determinado momento, lo que permite reducir la potencia contratada.
- Detección de averías en el los equipos domésticos, al observar un aumento en su consumo.

2.3. Componentes del Sistema

Al margen de los equipos de interconexión (repetidores, concentradores, puentes, etc.) y de la infraestructura de cableado (en el caso de no ser inalámbrica), una

instalación domótica necesita diferentes tipos de dispositivos [4]. A continuación se hablará de los tres dispositivos que tiene cualquier instalación, aunque existen otros que, dependiendo del tipo y de la envergadura de la instalación, podrían formar parte de ella como son las pasarelas residenciales o los interfaces.

2.3.1. Sensores

Un sensor es un dispositivo capaz de convertir una magnitud de una determinada naturaleza física, química o biológica en otra, generalmente eléctrica. Los sensores proporcionan la información que será posteriormente procesada. La calidad de un sensor viene determinada por su exactitud, fiabilidad, resistencia, sensibilidad y margen de error.

Existen numerosos tipos de sensores que se pueden agrupar en función de distintos criterios de clasificación.

1. Según su alimentación:

- a) Activos. Necesitan de una alimentación eléctrica determinada.
- b) Pasivos. No necesitan corriente eléctrica.

2. Según el tipo de señal implicada:

- a) Discretos o Detectores. Presentan un número finito de salidas posibles, que corresponden con estados de la variable a medir. Son sensores sencillos, baratos y muy habituales en las instalaciones domóticas. Ejemplos de este tipo son los sensores de detección de humo o de detección de gas.
- b) Continuos. Su salida varía en función de la variable medida. Ejemplos de este tipo son los sensores de temperatura, los de humedad, los de viento, etc.

Otro sistema de clasificación habitual es según el ámbito de aplicación, pudiendo ser de gestión climática, de gestión contra robo, de control de la iluminación, de gestión contra incendio, de control de presencia, etc.

2.3.2. Actuadores

Un actuador es un dispositivo que al recibir una señal eléctrica realiza una función física sobre un medio exterior; por ejemplo, recibe una instrucción de un regulador o controlador y genera una instrucción para activar un motor, un engranaje, una válvula, etc. Se puede decir que realizan la función inversa de los sensores.

Existen multitud de dispositivos que pueden considerarse como actuadores. Los **relés** actúan como un interruptor controlado por un circuito eléctrico y son capaces de conmutar un circuito de salida de mayor potencia que el de entrada, permitiendo encender y apagar bombillas u otros equipos. Los **dimmers** son dispositivos que regulan la potencia que llega a una carga, se usan para regular la intensidad de las bombillas, fluorescentes, etc. Las **electroválvulas** se utilizan para regular el fluido de líquidos y gases. También hay **motores eléctricos**, **contactores**, etc.

2.3.3. Controladores

Los controladores reciben las señales de los sensores, las procesan y las envían a los actuadores. En definitiva, son los encargados de realizar la gestión en una instalación domótica. Hay diferentes tipos dispositivos que pueden ejercer la función de controladores, dependiendo en gran medida del tipo de arquitectura de dicha instalación.

Algunos dispositivos son más característicos de sistemas distribuidos, como los **automatas programables** que tienen poca capacidad computacional pero que pueden informar y recibir órdenes de sistemas superiores o los **microcontroladores** que son fáciles de instalar y capaces de actuar sobre las luces, la calefacción o el gas.

Otros dispositivos son más idóneos en sistemas centralizados. Ejemplo de ellos son los **ordenadores** que disponen de rápidos y potentes microprocesadores y de una enorme capacidad de memoria lo que permite utilizar programas complejos y elaborados, además facilitan la conexión con los interfaces de usuario. También se incluye en este grupo los **sistemas embebidos**.

2.3.4. Otros dispositivos

Como ha sido mencionado, existen más dispositivos que pueden formar parte de una red domótica. Entre ellos se destacan las interfaces de usuario, los electrodomésticos

inteligentes y las pasarelas residenciales que actúan de frontera entre las redes internas de una vivienda y la red exterior.

También son necesarios dispositivos como filtros, reguladores, transceptores o amplificadores, que acondicionan las señales en función de los requerimientos o que actúan de interfaz entre el medio de transmisión y los elementos ya mencionados. Estos tipos de dispositivos suelen estar incluidos por los fabricantes junto a los actuadores, sensores y unidades de control, por lo que es habitual que no sean mencionados específicamente como elementos necesarios en una instalación domótica.

2.4. Arquitectura

Por arquitectura se entiende el lugar donde reside la inteligencia del sistema domótico, es decir, cómo y dónde se produce el control de la instalación. Cada tipo de arquitectura tiene sus ventajas e inconvenientes. Elegir un tipo u otro, depende en muchas ocasiones de las prestaciones del sistema o de motivos económicos. A continuación se citan los principales tipos de arquitectura.

2.4.1. Arquitectura Centralizada

Es aquella en la que existen una única unidad central de control que es la encargada de recibir la información de todos los sensores, procesarla según los criterios del usuario y seguidamente generar las órdenes oportunas que son enviadas al conjunto de actuadores repartidos por el hogar (ver esquema en la figura 2.2). La unidad de control es el dispositivo fundamental y del que depende todo el sistema para su correcto funcionamiento.

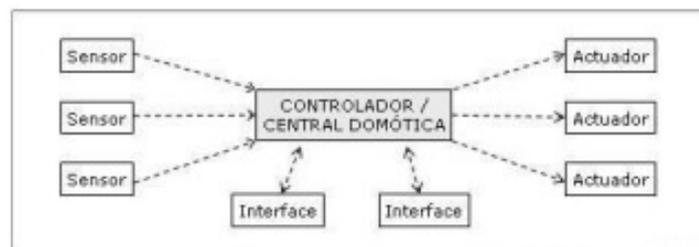


Figura 2.2: Arquitectura Centralizada

2.4.2. Arquitectura Distribuida

Es aquella en la que no existe una unidad central de control, si no que cada elemento del sistema dispone de inteligencia incorporando su propio controlador. Es habitual de los sistemas con cableado en bus a través del cual se comunican todos los elementos. En este caso, si un elemento fallase, el resto del sistema continuaría operando correctamente. En la figura 2.3 se puede observar el esquema de un sistema con este tipo de arquitectura.

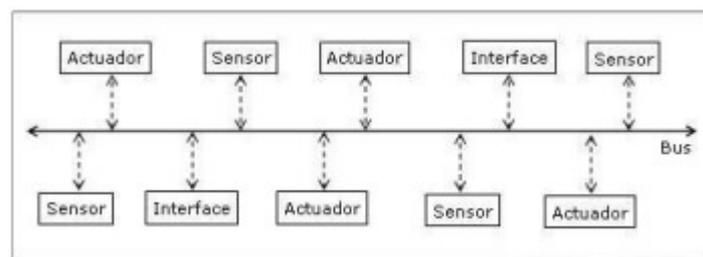


Figura 2.3: Arquitectura Distribuida

2.4.3. Arquitectura Mixta

Es un término medio entre las dos anteriores arquitecturas. Como se observa en el esquema de la figura 2.4 en este tipo de arquitectura se dispone de varias unidades de control capaces de adquirir y procesar información de varios sensores y comunicarse entre ellas.

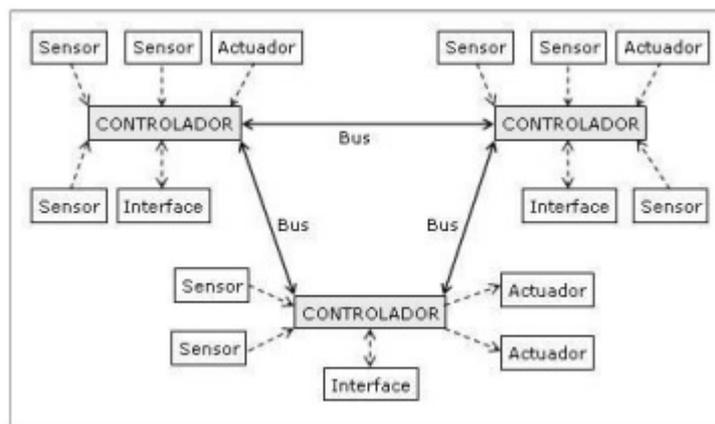


Figura 2.4: Arquitectura Mixta

2.5. Topologías

Por topología se entiende la forma en que está diseñada la red, ya sea desde el punto de vista físico o lógico. Aunque existen diversas formas de clasificar las topologías de red, la más general es la siguiente:

1. **Topología en Estrella:** cada nodo está conectado únicamente al controlador central. Su principal ventaja es la facilidad para añadir nuevos dispositivos y sobre todo el hecho de que un fallo de un nodo no compromete el funcionamiento del sistema en su totalidad. Por el contrario, un fallo del controlador central provoca la caída total de la red. Otro inconveniente es que admite una cantidad limitada de dispositivos o de lo contrario se podría saturar el controlador central.
2. **Topología en Anillo:** los dispositivos se conectan formando un anillo cerrado, por lo que cada nodo está conectado con los dos nodos colindantes. Su principal inconveniente es que la red es vulnerable a un fallo de cualquier elemento, además, resulta complicado ampliar la red con nuevos dispositivos.
3. **Topología en Bus:** todos los dispositivos comparten el mismo canal de comunicación (bus). Añadir nuevos dispositivos a la red resulta fácil y un error en algún nodo no implica problemas en la red. Sus principal desventaja es la necesidad de que los nodos tengan un cierto grado de inteligencia ya que no existe un controlador central y necesitan coordinar su acceso al bus principal.
4. **Topología en Árbol:** es una mezcla de la topología en estrella y la topología en bus. Existe una jerarquía entre los diferentes dispositivos de la red.
5. **Otras topologías:** existen otros muchos tipos de topologías como son, la mixta, doble anillo, malla, etc (ver figura 2.5).

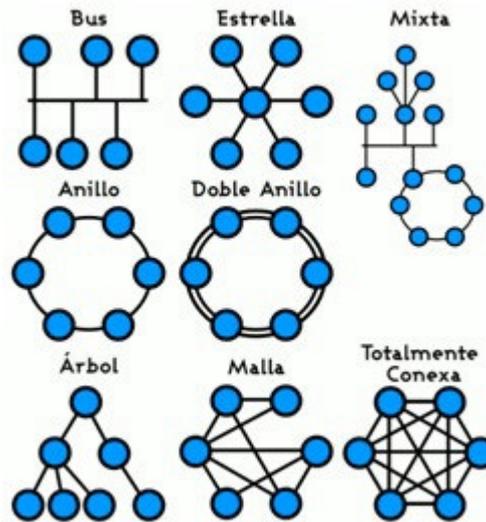


Figura 2.5: Tipos de Topología

2.6. Medios de Transmisión

Es el soporte físico a través del cual se produce el intercambio de información (envío de señales) entre los diferentes elementos de una red. Se suelen clasificar en dos grandes grupos: Cableados e Inalámbricos.

1. Cableados

- a) **Corriente Portadora (PL).** Utiliza como soporte físico las líneas eléctricas de las viviendas.
- b) **Par Metálico.** Utiliza como soporte físico cables formados por uno o varios conductores de cobre capaces de cubrir un amplio rango de aplicaciones. Este medio permite transportar voz, datos y la propia alimentación de los dispositivos.
- c) **Fibra Óptica (FO).** Utiliza como soporte físico un material dieléctrico transparente, por el que se envían pulsos de luz que constituyen los datos a transmitir.
- d) **Coaxial.** Estos cables están formados por un par de conductores concéntricos separados por un aislante dieléctrico. El conductor central trans-

porta los datos, mientras que la malla exterior actúa como retorno de la corriente y referencia de tierra.

2. Inalámbricos

- a) **Infrarrojos.** La comunicación entre dos dispositivos se realiza a partir de un diodo emisor que emite una onda en la banda de infrarrojo con la información. Es necesaria la presencia de un fotodiodo receptor el cual extrae la información de la señal recibida. Uno de los inconvenientes de este medio de transmisión, es que se necesita una línea visual entre el diodo emisor y receptor.
- b) **Radiofrecuencia.** La comunicación se realiza mediante el empleo de la banda del espectro electromagnético situada entre los 3 Hz y los 300 GHz. Es uno de los medios más utilizados por su flexibilidad, capacidad de transmisión y ausencia de cableado, pero es muy sensible a las perturbaciones electromagnéticas.

El medio de transmisión más idóneo depende de varios factores. Si la seguridad es el factor predominante, un sistema de cableado sería más apropiado que un sistema inalámbrico ya que dificultaría que se pudiera alterar la información desde el exterior. Si se busca que la red pueda ser desplegada fácilmente en cualquier vivienda, los medios más idóneos son las Corrientes Portadoras y los Sistemas Inalámbricos. Otros parámetros a tener en cuenta a la hora de decidir el medio, sería la velocidad de transmisión de datos y el tipo de datos a transmitir.

2.7. Tecnologías y Protocolos de Comunicación

Una de las principales características de un sistema domótico es el protocolo de comunicaciones que emplea, es decir, el conjunto de reglas y normas que usan dos dispositivos de la red para comunicarse entre sí y transmitir la información.

Las tecnologías empleadas en redes domóticas se pueden clasificar de varias maneras, siendo una de las más habituales agruparlas en sistemas o protocolos propietarios y en protocolos abiertos. Los primeros son aquellos que son propiedad de una empresa y sólo pueden ser utilizados por equipos y dispositivos de esa compañía o de

terceros bajo licencia. A este grupo pertenecen tecnologías como Amigo, Simon VIS, PLC, Plus Control, etc. Los segundos son aquellos de libre acceso y que pueden ser empleados por cualquier empresa.

Cada tipo de tecnología se asocia con uno o varios medios de transmisión, lo que conlleva que, en función del sistema elegido, se usará uno u otro medio. De forma análoga, el medio utilizado también determina aspectos como la velocidad máxima, el número de dispositivos, etc.

En la actualidad existen multitud de tecnologías diferentes [4][2][5][6], siendo este uno de los principales motivos que ha ralentizado la implantación de la domótica. La gran variedad de sistemas ha generado incompatibilidades entre equipos e instalaciones y ha provocado desconocimiento y dudas a los usuarios. A continuación se explicarán algunas de las tecnologías existentes más representativas, centrándose principalmente en sistemas de redes de control y automatización y dejando más de lado las tecnologías de interconexión de dispositivo, de redes de datos y de redes multimedia.

2.7.1. LonWorks

La tecnología Lonworks (Local Operating NetWork), desarrollada a principios de los 90 por la empresa norteamericana Echelon, es utilizada para implementar redes de control. Es una tecnología abierta a cualquier fabricante y utilizada para aplicaciones domóticas y para la gestión de edificios, aunque es en el ámbito de la automatización industrial donde ha tenido más éxito. Está recogida como estándar norteamericano (EIA 709.1 en ANSI) y, desde el año 2009, como estándar global (ISO/IEC 14908) para redes de comunicaciones.

LonWorks es un sistema de control distribuido, en el que la inteligencia se reparte por un conjunto de nodos independientes e interconectados entre sí. Se trata de una tecnología fiable, modular y flexible que puede soportar diferentes tipos de topología (bus, anillo, estrella, etc.). Esta tecnología es independiente del medio de transmisión, siendo los más usados el par trenzado y el de ondas portadoras. También es posible usar fibra óptica, cables coaxiales e incluso radio. Del canal de transmisión dependen parámetros como la velocidad de transmisión, la distancia máxima, el número de nodos, etc.

Actualmente la mayoría de los dispositivos que conforman estos sistemas incluyen

el microcontrolador Neuron Chip (fabricado por Cypress Semiconductor, Motorola y Toshiba), aunque pueden basarse en otros tipos de procesadores. Este microcontrolador tiene como características: tres procesadores (2 para comunicaciones y uno para aplicaciones), pines de entrada y salida para interactuar con los sensores y actuadores, memoria RAM, memoria EEPROM, etc. También cuenta con un puerto que actúa de interfaz del transceptor, el cual se encarga de adaptar las señales entre el Neuron Chip y el canal físico.

El protocolo de comunicaciones LonTalk, implementa todos los niveles (7 capas) del modelo de referencia OSI. Servicios como el reenvío automático de tramas tras una pérdida o la autenticación del emisor están integrados en este protocolo. LonTalk utiliza “CSMA predelectiva p-persistente” para reducir las colisiones en el acceso al medio compartido y los períodos de espera provocados.

Los dispositivos que forman una red de control LonWorks se comunican entre sí mediante el envío de tramas que contienen las direcciones de destino, información para el encaminamiento, datos de control y aplicación, etc. A la hora de direccionar cualquier nodo de la red se puede hacer uso del identificador único del NeuronChip, o bien de la dirección de dispositivo que se le asigna al incorporarlo a la red. Esta última manera es más eficiente, posibilita el reemplazamiento de nodos defectuosos y permite la creación de grupos, subredes, etc.

Existe una asociación formada por fabricantes, integradores y distribuidores de productos basados en LonWorks llamada LonMark, que tiene como objetivo fomentar y dar a conocer esta tecnología y sus ventajas, facilitar la integración de dispositivos de diferentes compañías, etc.

LonWorks ha tenido un gran éxito de implantación en oficinas, edificios y sobre todo en industrias, aunque en el ámbito doméstico ha tenido una acogida más modesta debido a la existencia de otras tecnologías más económicas y de prestaciones similares. Su implantación está más extendida en Estados Unidos que en Europa.

2.7.2. Konnex

Konnex (KNX) [7] surge a partir de la iniciativa de tres asociaciones europeas de sistemas domóticos: EIB (European Installation Bus), EHS (European Home Systems) y Batibus. Su objetivo es desarrollar un único estándar de control y automatización

de oficinas y viviendas, capaz de competir en calidad y prestaciones con otros sistemas como LonWorks o CEBus. La tecnología Konnex es un estándar abierto, capaz de ser implementado sobre cualquier plataforma de microprocesador y con la posibilidad de ser utilizado en instalaciones muy diversas, desde una gran industria a una pequeña vivienda.

KNX está aprobado como estándar europeo (CENLEC EN 50090 y CEN EN 13321-1), como estándar norteamericano (ANSI/ASHRAE 135) y como estándar internacional (ISO/IEC 14543-3).

Este sistema puede adoptar diferentes tipos de topología de bus y se caracteriza por un control descentralizado, es decir, los dispositivos se comunican sin la necesidad de una unidad central. Un sistema KNX puede funcionar sobre diferentes medios de transmisión, como son el par trenzado (KNX TP1), corrientes portadoras (KNX PL110), radiofrecuencia (KNX RF) e IP/Ethernet (IP KNX), el cual permite enviar paquetes KNX encapsulados en tramas IP. Dependiendo del medio de transmisión se alcanzan diferentes velocidades de transmisión.

El protocolo de comunicaciones KNX se basa en el modelo de referencia OSI. La comunicación entre los dispositivos se realiza mediante el envío de tramas que incluyen campos como la dirección de destino, la dirección de origen, datos, código de comprobación, etc. Como sucede con otras tecnologías, estos dispositivos tienen dos direcciones que operan de modo excluyente, una dirección física única y una dirección lógica que no tiene porque ser única y que, de hecho, es compartida por grupos de dispositivos que desempeñan la misma tarea (por ejemplo, el encendido de alarmas en varias habitaciones de forma simultánea).

KNX ofrece dos niveles diferentes de configuración para la realización de sus proyectos. El modo-E (easy mode) orientado a instaladores con baja cualificación (dispositivos con una función concreta) y el modo-S (system mode) orientado a instaladores con conocimientos avanzados, que permite aplicaciones más sofisticadas.

La Asociación Konnex es la encargada de promover esta tecnología, de realizar la certificación de productos y garantizar la interoperabilidad de los dispositivos basados en este estándar.

2.7.3. X10

La tecnología X10 fue desarrollada a finales de los 70 por la empresa escocesa Pico Electronics of Glenrthes con el objetivo de poder transmitir datos a través de la red eléctrica de baja tensión (230V a 50Hz en Europa y 120V a 60Hz en EEUU). Se trata de un protocolo abierto a cualquier fabricante que quiera producir dispositivos basados en esta tecnología.

Una de las características que ha provocado que X10 siga siendo de los sistemas más exitosos es su bajo coste, ya que al emplear la red eléctrica existente puede ser implementado sin hacer obras en la vivienda. Además, es una tecnología sencilla y de fácil manejo, lo que permite una instalación rápida sin grandes conocimientos. Otra de las características de esta tecnología es que es flexible y fácilmente ampliable, permitiendo añadir de manera sencilla nuevos dispositivos a la instalación. Su ancho de banda, sin embargo, es bastante reducido comparado con otras tecnologías actuales, lo que provoca que la transmisión de los datos se realice a muy baja velocidad (50bps en Europa y 60 bps en Estados Unidos).

El funcionamiento de X10 se basa en una sencilla modulación de la señal que circula por la red eléctrica. En la modulación, el transceptor aprovecha los puntos en que la señal sinusoidal de tensión alterna (señal portadora) presenta un valor nulo de potencia para poder insertar, o no, un impulso (señal moduladora) de 1ms de duración, de muy bajo voltaje y a una frecuencia de 120KHz, dando lugar a la señal modulada, la cual permite generar código digital. En cada ciclo de corriente alterna se transmite un bit, que es interpretado como un '1' binario cuando hay un pulso de 120KHz o como un '0' binario cuando hay ausencia de dicho pulso.

La transmisión de un mensaje estándar requiere de 11 ciclos de corriente alterna, que corresponden a: código de inicio (2 ciclos), código de casa (4 ciclos) y código de función o de unidad (5 ciclos). La reducida dimensión de estos mensajes hace que la cantidad máxima de dispositivos posibles sea pequeña (256) y que las opciones de funcionalidad también sean limitadas. Por lo tanto, el sistema X10 es una opción adecuada si se desea una instalación domótica reducida y poco compleja.

En la actualidad, X10 es uno de los sistemas más antiguos que se siguen empleando en aplicaciones domóticas y está presente en numerosos hogares de todo el mundo.

Por último, mencionar la existencia del protocolo UPB (Universal Powerline Bus),

que se basa en el concepto de X10. Emplea también la red eléctrica, pero presenta velocidades de transmisión superiores y mayor fiabilidad al tener menos problemas de interferencias con electrodomésticos. Sin embargo, su uso no está muy extendido y hay pocas empresas que fabriquen dispositivos de este tipo.

2.7.4. Insteon

La tecnología Insteon fue desarrollada a partir del año 2000 por la empresa SmartLabs con el objetivo de crear un sistema para la automatización y control de los hogares. Insteon es un protocolo de comunicación propietario, que se basa en el empleo de la red eléctrica y la radiofrecuencia al mismo tiempo. Su topología es del tipo malla, en la que todos los nodos actúan como repetidores, es decir, cada dispositivo retransmite los mensajes que recibe. Con esto se consigue que la red de control sea muy robusta y fiable.

Los dispositivos Insteon están muy enfocados al control de la iluminación y son compatibles con equipos basados en otras tecnologías como X10 y ZigBee. Su presencia en el mercado europeo es todavía pequeña.

2.7.5. Z-Wave

Z-Wave es un protocolo de comunicación propietario e inalámbrico, diseñado para la automatización y control de una vivienda.

El sistema Z-Wave emplea módulos RF de baja potencia para la transmisión de la información. Una de las diferencias entre Z-Wave y la mayoría de tecnologías inalámbricas para redes de control, es que la primera trabaja en la banda de frecuencia de los 900 MHz y por tanto, sufre menos interferencias que los sistemas que trabajan a 2.4GHz (banda más poblada). Por el contrario, la capacidad de transmisión de datos es limitada, siendo su tasa de transferencia inferior a otras tecnologías inalámbricas.

Las empresas pertenecientes a la Alianza Z-Wave son las encargadas de fabricar productos que usan esta tecnología y de garantizar la interoperabilidad entre ellos. Actualmente Z-Wave es una de las tecnologías con mayor crecimiento, siendo la opción elegida por muchos fabricantes de sistemas de seguridad domóticos.

En la tabla 2.1 se comparan las principales características de estos cinco protocolos.

| | LonWorks | KNX | X-10 | Insteon | Z-Wave |
|---------------------------------|------------------------------------|------------------------|---------|----------------|-------------|
| Medio | RE,RF,FO Coaxial, Par trenz. | RE,RF, Par trenz. | RE | RE, RF | RF |
| Tipo | Abierto, bajo licencia | Abierto | Abierto | Propietario | Propietario |
| Velocidad de transmisión | 1.25 Mbps | 9.6 Kbps Par trenz. | 60 bps | 38.4 Kbps | 9.6 kbps |
| Número de dispositivos | 32000 | 58000 | 256 | $16 * 10^{16}$ | 232 |

Tabla 2.1: Comparativa de protocolos

2.7.6. Otras tecnologías inalámbricas

Hay algunos tipos de tecnologías inalámbricas utilizadas para la interconexión de dispositivos y redes de datos que también pueden emplearse en redes de control y en aplicaciones domóticas. En esta categoría se analizarán 3 de estas tecnologías, como son Wi-Fi, Bluetooth y ZigBee.

1. **Wi-Fi (Wireless Fidelity)**. Es una de las tecnologías inalámbricas más usadas en la actualidad y está presente en multitud de hogares. Las redes Wi-Fi son redes WLAN (Wireless Local Area Network) que se basan en el estándar internacional IEEE 802.11. El empleo de esta tecnología permite conexiones rápidas y muy seguras ya que dispone de diferentes mecanismos de seguridad. La tecnología Wi-Fi, que trabaja en la banda ISM, permite velocidades de transferencia muy elevadas y un rango de alcance relativamente grande. Es usada principalmente para comunicar ordenadores, tablets o smartphones a un router y permitir así la conexión a internet de estos dispositivos.
2. **Bluetooth**. Es un estándar de comunicaciones típico de las redes WPAN (Wireless Personal Area Network) y permite enlaces de corto alcance y bajo consumo. Es una tecnología que opera en la banda de frecuencia ISM (alrededor de los 2.4 GHz) y asegura la transmisión de datos, voz o video. Está recogida como estándar internacional IEEE 802.15.1. Esta tecnología se utiliza principalmente para la sincronización automática entre dispositivos y sobre todo para la conexión de periféricos como teléfonos, manos libres, teclados, ratones, etc.

3. **ZigBee**. Permite una comunicación inalámbrica de baja velocidad y es típica de las redes LR-WPAN, opera en la banda ISM y está basada en el estándar internacional IEEE 802.15.4. Se trata de una tecnología simple, robusta y con un consumo de energía muy bajo, que, sin embargo, dispone de un ancho de banda reducido.

Cada uno de estos estándares se diferencian en aspectos como la tasa de transferencia, el tipo de modulación, el rango de alcance, la banda de frecuencia y otros. Por ejemplo se puede aprovechar que todos los teléfonos móviles incorporan la tecnología Bluetooth, para controlar una instalación domótica a través del móvil [8], al mismo tiempo sería una opción adecuada si se desea una comunicación eficiente a cortas distancias y con un consumo reducido, mientras que si se requiere de mayores distancias y mayores tasas de transferencia de datos Wi-Fi (WLAN) sería más apropiada. En la figura 2.6 y en la tabla 2.2 se comparan estas tecnologías inalámbricas en función de varios aspectos.

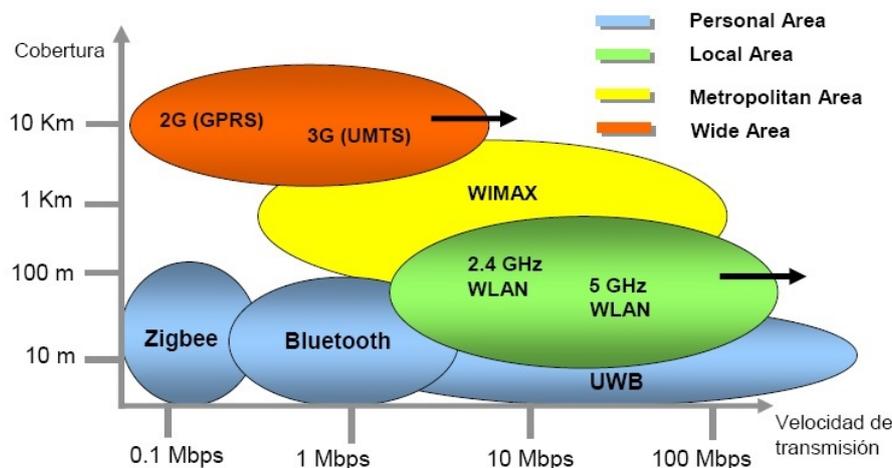


Figura 2.6: Comparativa estándares de comunicaciones inalámbricas

ZigBee es una tecnología ideal para redes sencillas que no necesitan de grandes velocidades, lo que permite un bajo consumo y un coste reducido [9]. ZigBee es más idónea para redes con muchos dispositivos que requieran una duración prolongada de las baterías. Todo esto ha hecho que sea una de las tecnologías más empleadas en el ámbito de la domótica y que más posibilidades ofrece para el control de temperatura, control de accesos, encendido o apagado de luces, etc [10][11]. Esta tecnología ha sido

| | WiFi | Bluetooth | ZigBee |
|---------------------------------|-------------|------------------|---------------|
| Estándar | 802.11.x | 802.15.1 | 802.15.4 |
| Medio | RF | RF | RF |
| Velocidad de transmisión | 54 Mbps | 1 Mbps | 256 kbps |
| Consumo | Alto | Medio | Bajo |
| Coste | Medio | Bajo | Bajo |

Tabla 2.2: Comparativa Tecnologías inalámbricas

la seleccionada al ser inalámbrica, de bajo coste y consumo y por ser abierta, lo que permite desarrollar el hardware y software que se desee. Se hablará de esta tecnología con más profundidad en el siguiente capítulo del proyecto.

Capítulo 3

Descripción general del sistema

A lo largo de este capítulo se proporciona una visión general del sistema domótico propuesto. Además se detalla la implementación del sistema de comunicaciones empleado para crear la red de control y automatización.

Tras analizar el estado del arte, se llegó a la conclusión de cómo debía ser el sistema y las características que debía tener.

El sistema propuesto puede ser controlado remotamente a través de internet y es capaz de cubrir las siguientes aplicaciones:

- Consultar el estado de diferentes tipos de sensores analógicos.
- Activar/Desactivar ciertos elementos como bombillas o motores.

Al mismo tiempo el sistema sirve de base para cubrir aplicaciones más complejas sin necesidad de cambiar su estructura ni sus características. Entre las aplicaciones que el sistema podría ser capaz de cubrir, se pueden citar:

- Fijar automatismos para activar o desactivar actuadores en función de si se alcanzan ciertas condiciones en el estado de los sensores.
- Emplear sensores digitales que cubran aplicaciones más complejas.

En cuanto a las características del sistema propuesto, se decidió que el sistema usase un protocolo de comunicaciones inalámbrico para la conexión de los diferentes dispositivos de la instalación. Al hacer uso de la radiofrecuencia como medio de transmisión

de la red, el sistema puede ser usado indistintamente tanto en viviendas de nueva construcción como en viviendas ya construidas, sin la necesidad de hacer ninguna obra. Además el despliegue resulta sencillo y rápido. De entre las tecnologías de este tipo analizadas en el estado del arte se concluyó que la tecnología ZigBee (concretamente el estándar IEEE 802.15.4) era la mejor opción, al garantizar bajos consumos y no necesitar grandes anchos de banda. En definitiva, este es el sistema de comunicación que se emplea en la instalación para conectar los distintos dispositivos que integran la red.

En segundo lugar, se eligió que el sistema tuviese una arquitectura centralizada, lo que se traduce en menor coste económico y mayor sencillez a la hora de implementar y diseñar la red. Al mismo tiempo, se decidió que la instalación fuese controlada por el usuario a través de una aplicación web, visualmente atractiva y de fácil manejo, alojada en la unidad central [12][13]. La unidad central de control del sistema está formada por un módulo de radiofrecuencia Xbee (que emplea la tecnología ZigBee/IEEE 802.15.4) y por un dispositivo de control (BeagleBone), el cual se tiene que conectar a la red de internet y en el que hay instalado un servidor web, que es lo que permite al usuario interactuar con el sistema. El usuario, desde dispositivos como ordenadores o tabletas con conexión a internet, accede al servidor y a la aplicación a través de un navegador web.

La plataforma de desarrollo BeagleBone es un sistema embebido con gran capacidad computacional y con una alta versatilidad de comunicaciones con el exterior, que es lo que permite que la unidad central de control actúe de pasarela entre la red de internet y la red de control inalámbrica (red domótica).

El tipo de topología de la red está condicionada por los dos aspectos anteriores, la tecnología empleada en el sistema de comunicaciones y el tipo de arquitectura. El sistema tiene una topología en estrella, que es la que resulta más idónea de entre todas las permitidas por ZigBee y que es coherente con la arquitectura centralizada seleccionada. La topología en estrella incrementa la flexibilidad del sistema y permite ir ampliando fácilmente la cantidad de nodos periféricos, así como que sea la base de un futuro sistema más amplio y diversificado.

A parte de la unidad central de control, el resto de los componentes de la red domótica son las unidades que se reparten por la vivienda. Estas unidades remotas o dispositivos finales integran un módulo para la comunicación inalámbrica de radio-

frecuencia (módulo Xbee) y un conjunto de sensores y actuadores. Como se detallará en el capítulo 4 se ha desarrollado un tipo de unidad remota estándar que dispone de tres sensores y de tres actuadores. Sin embargo, el sistema admite otros tipos diferentes de unidades remotas, por ejemplo que contengan un microcontrolador con el que se pueda realizar cualquier tarea o que dispongan de otra distribución, en tipo y cantidad, de sensores y actuadores.

La unidad remota diseñada incluye un sensor de temperatura, otro de luminosidad y otro de humedad. Con estos tres sensores se puede recoger información útil de la casa en general y de cada habitación en particular. Incluso se puede obtener información de interés para el cuidado del jardín. Los actuadores son genéricos, es decir, las unidades remotas disponen de tres relés a través de los cuales se puede activar o desactivar los diferentes elementos que el usuario conecte. Como se detalla en la sección (¡!!), es posible emplear estos sensores y relés gracias a la capacidad de los módulos XBee al disponer de conversores análogo-digitales y de salidas digitales.

El esquema general del sistema se puede apreciar en la figura 3.1, donde se observa la estructura del mismo y sus componentes fundamentales.

Tras haberse presentado la estructura y las características del sistema desarrollado, a continuación se exponen aspectos relacionados con el funcionamiento y el software del sistema.

La unidad central de control, en concreto la plataforma de desarrollo BeagleBone que es donde reside el software y que actúa como pasarela entre la red de Internet y red Domótica, se encarga de dos funciones. En primer lugar, ofrecer al usuario una interfaz para que interactúe con el sistema domótico a través de la red de Internet, de modo que se recojan sus instrucciones y se le presente la información. En segundo lugar, llevar a cabo el control y manejo de la red domótica, y en concreto, gestionar la comunicación inalámbrica con los nodos remotos repartidos por la red.

Como se detalla en la sección (5.2), el software desarrollado para la plataforma BeagleBone se ha dividido en dos bloques claramente diferenciados en función de lo comentado en el párrafo anterior. Por un lado, se encuentra la aplicación web que ofrece la interfaz al usuario desde la que interactúa con el sistema. Por otro lado, el denominado “software esclavo”, el cual es ejecutado siguiendo las instrucciones de la aplicación web, se encarga de crear los paquetes para la comunicación inalámbrica en la red domótica ZigBee/IEEE 802.15.4 y su envío desde la BeagleBone al módulo de

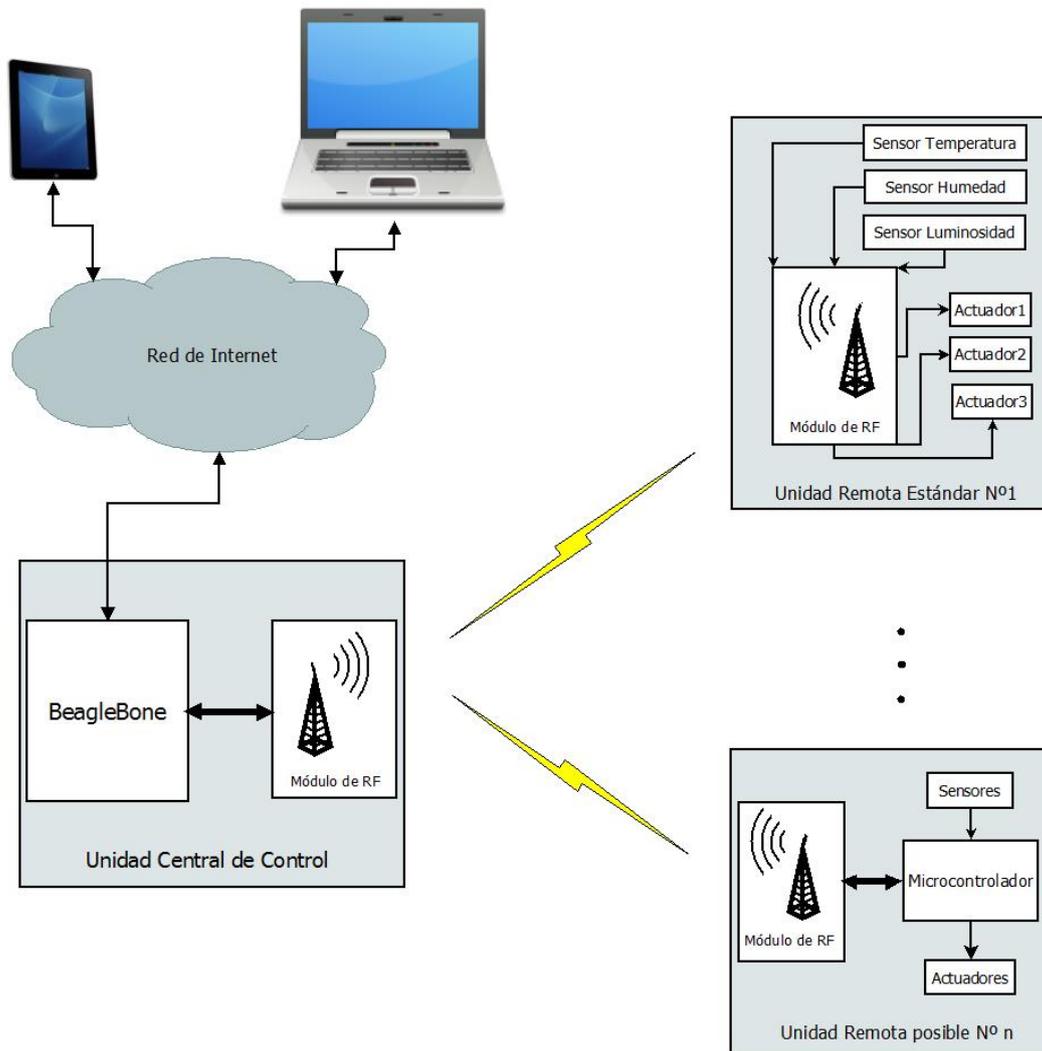


Figura 3.1: Esquema del sistema domótico desarrollado

RF. Dicho software también se encarga de la recepción de los paquetes de entrada y su procesamiento.

El funcionamiento general del sistema, desde que el usuario accede desde un navegador para realizar una acción sobre el sistema hasta que dicha tarea es realizada, se puede apreciar en el diagrama de secuencia de la figura 3.2. El diagrama representa el funcionamiento general del sistema para una acción genérica, sin entrar a considerar los detalles del software y sin especificar una acción en concreto.

Hay que tener en cuenta dos aspectos sobre este diagrama. Por un lado y para facilitar su comprensión, no se han diferenciado ni detallado las etapas de ejecución

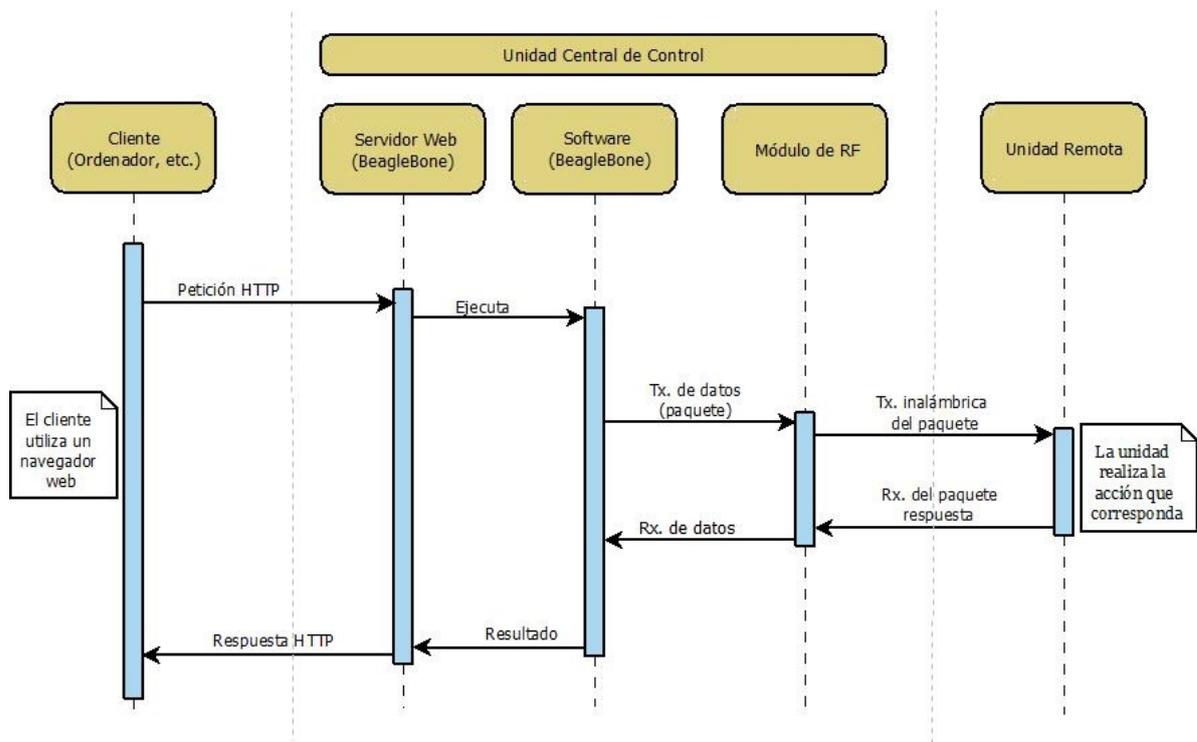


Figura 3.2: Diagrama de secuencia del sistema

de la aplicación web y ejecución del software esclavo, sino que se han integrado en una única etapa como ejecución del software. Por otro lado, hay otras situaciones en las que las peticiones http no requieren del envío de datos por radiofrecuencia a través de la red domótica (por ejemplo, el acceso a la página inicial de la aplicación). En el anexo G se muestra un diagrama de secuencia completo, incluyendo estas divisiones y sus detalles.

El funcionamiento del sistema se puede resumir de la siguiente manera:

1. El usuario a través de un navegador web (cliente) realiza peticiones http al servidor con las instrucciones oportunas.
2. El servidor web (instalado en la plataforma BeagleBone de la unidad central) atiende estas peticiones y ejecuta el software que, entre otras cosas, crea el paquete de comunicación que es enviado al módulo XBee.
3. El mensaje es transmitido, a través del módulo de RF, a unidad remota específica.

4. La unidad remota recibe el mensaje, realiza la acción que corresponda y transmite por radiofrecuencia el mensaje de respuesta a la unidad central.
5. El módulo de RF de la unidad central recibe el mensaje y lo envía automáticamente a la plataforma BeagleBone.
6. A continuación dicho software, entre otras cosas, procesa el paquete de respuesta y termina su ejecución generando el resultado que es enviado al servidor web.
7. Por último, el servidor envía la respuesta para que el cliente la visualice a través de su navegador.

3.1. Implementación del sistema de comunicaciones

A lo largo de esta sección se introduce la tecnología inalámbrica empleada en la red domótica. Se detalla el tipo de comunicación utilizada y como están estructurados los paquetes que se transmiten. También se analiza como se ha configurado la red y los módulos de radiofrecuencia elegidos para hacer uso de esta tecnología.

3.1.1. IEEE 802.15.4 y Protocolo ZigBee

Durante esta sección se presentará el estándar IEEE 802.15.4 y el protocolo ZigBee, el cual está basado en dicho estándar y es el más difundido entre todos los tipos de tecnologías 802.15.4.

El estándar IEEE 802.15.4 se definió para desarrollar redes inalámbricas de área personal con tasas de transferencia bajas (LR-WPAN, Low Rate-Wireless Personal Area Network). El estándar es ampliamente usado en redes de sensores inalámbricos (WSN), donde una tasa de transferencia baja no es un inconveniente y por el contrario se consigue una comunicación fiable y bajos consumos en los nodos sensores.

ZigBee es un protocolo de comunicaciones inalámbrico creado a principios de siglo por ZigBee Alliance, una organización abierta de más de 400 miembros entre reguladores, universidades y empresas (Texas Instruments, Philips, Digi, etc.) que proveen desde

chips de radio a herramientas de desarrollo y testeo. La misión de esta organización es el desarrollo y mejora de los estándares ZigBee.

Como se ha mencionado ZigBee es una ampliación del estándar IEEE 802.15.4, por lo que ambos comparten características. Se diseñaron para cumplir las siguientes especificaciones:

- Bajo coste en cuanto al despliegue y mantenimiento de la red.
- Velocidad de transmisión reducida.
- Bajo consumo de potencia por parte de los dispositivos, lo que permite prolongar el uso de las baterías.
- Instalación fácil y sencilla.
- La creación de redes ampliables y flexibles.

El protocolo ZigBee es analizado en el anexo A. En concreto se detallan las bandas de frecuencia en la que opera, se muestra la pila de protocolos (las capas que emplea) y finalmente se indican los mecanismos de seguridad que ofrece para proteger las comunicaciones.

3.1.2. Red de ZigBee

A lo largo de esta sección se analizan las especificaciones de las redes ZigBee, en concreto se presentan los tipos de componentes en una red y las topologías. Finalmente se indican las particularidades de la red implementada.

3.1.2.1. Tipos de dispositivos

Las redes 802.15.4 y en concreto ZigBee pueden estar formadas por tres tipos de elementos según la función que desempeñan en la red

- **Coordinador.** Existe un único coordinador por cada red. Es el encargado de crear, inicializar y controlar la red, estableciendo aspectos como el canal de comunicaciones y asignando parámetros como las direcciones de red. Una

vez establecida ésta, su función es la misma que la de un Router, encaminar los mensajes o recibirlos en el caso de ser el destinatario. Son dispositivos de funcionalidad completa (FFD) que almacenan información de la red y que se encargan de gestionar la seguridad de ésta.

- **Routers.** Son nodos de tipo FFD que se encargan de interconectar dispositivos, enrutando los mensajes hacia el nodo destino. Se emplean para extender la cobertura de la red y para crear nuevos caminos de comunicación que sean de utilidad en caso de congestión o caída de algún nodo.
- **Dispositivos Finales (End Devices).** Son nodos de funcionalidad reducida (RFD) que carecen de la capacidad de enrutar paquetes, limitándose a enviar mensajes a su nodo padre, ya sea el coordinador o un router. Estos dispositivos tienen una carga de trabajo y un consumo menor que los anteriores, permitiendo que sean alimentados con baterías.

3.1.2.2. Topologías de Red

Como se mencionó en la sección 2.5, una red domótica puede presentar diferentes tipos de topología. En el caso de la tecnología ZigBee es posible implementar redes de control y automatización con 3 tipos de topologías físicas (ver figura 3.3)

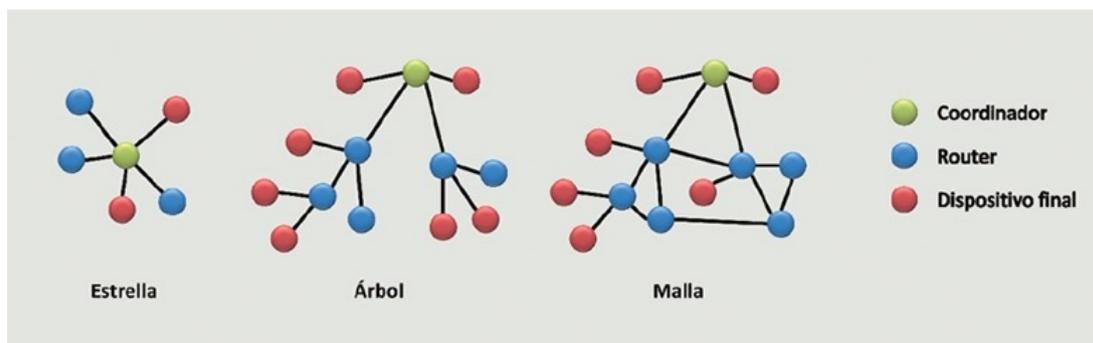


Figura 3.3: Topologías de ZigBee

- **Malla (MESH).** Estas redes están formadas por un coordinador, routers y dispositivos finales. Los nodos pueden establecer comunicación con cualquier otro dispositivo, mediante routers que actúan de repetidores. Se puede considerar que este tipo de topología es una extensión de las peer-to-peer.

La ventaja de este tipo de redes es su flexibilidad, al disponer de caminos alternativos y no depender de un único nodo. Tiene como inconveniente el incremento en la complejidad y requiere una mayor capacidad de computación al tener que elegir el camino más apropiado.

- **Árbol (Cluster Tree).** Los diferentes componentes de esta red están organizados en una estructura jerárquica. La red está formada por un coordinador y por un grupo de routers de los que pueden colgar, a su vez, otros dispositivos denominados hijos. Éstos pueden actuar como routers con más hijos o como dispositivos finales. Cada nodo sólo puede tener un único padre y los dispositivos finales no pueden tener ningún nodo hijo. Esta topología, al igual que la de Malla, permite expandir la cobertura de la red.
- **Estrella.** La red está compuesta por varios dispositivos finales conectados únicamente al coordinador central por el que pasan todas las comunicaciones. El alcance máximo de la red viene determinado por el alcance del coordinador, el cual necesita alimentarse a través de la red eléctrica. El resto de dispositivos finales pueden emplear baterías.

3.1.2.3. Red implementada.

Las redes de tipo estrella han sido consideradas las más adecuadas, ya que son sencillas de controlar, cubren satisfactoriamente cualquier aplicación domótica y no se necesitan más de dos nodos para implementar un sistema prototipo como el que se desarrollará en este proyecto. Por tanto ha sido la topología elegida.

Aplicando estos conceptos a la estructura del sistema presentado con anterioridad, se establece que la unidad central operará como coordinador de la red siendo un dispositivo de funcionalidad completa (FFD). Las unidades remotas operan como dispositivos finales siendo nodos de funcionalidad reducida (RFD). La red tendrá un coordinador y tantos dispositivos finales como se considere oportuno.

3.1.3. Módulos Xbee

Los elementos fundamentales de una red ZigBee son los dispositivos que se emplean para realizar la comunicación inalámbrica. Estos dispositivos son módulos de radiofre-

cuencia que constituyen los nodos de la red y que estarán repartidos por la vivienda. Si bien existen diferentes tipos de dispositivos, en este proyecto se utilizarán los denominados ‘Módulos XBee’ o ‘XBee radios’. Estos módulos, además de ser los más extendidos, han sido elegidos por sus características, su bajo precio y al disponer en el laboratorio HCTLab de las interfaces necesarias para configurarlos y probarlos.

Hay una gran cantidad de módulos XBee diferentes, aunque básicamente se pueden agrupar en dos tipos, los de la Serie 1 y los de la Serie 2 (ver tabla comparativa 3.1). Ambos son muy similares en cuanto a hardware, tienen el mismo footprint y la mayoría de los pines desempeñan las mismas tareas. Sin embargo, no es posible el uso de ambas series en una misma red ya que no son compatibles entre sí, utilizando cada una diferentes perfiles de aplicación. La elección de una u otra serie dependerá de las aplicaciones que se les vayan a asignar.

| | XBee Serie1 | XBee Serie2 |
|--------------------------------------|----------------------------|-----------------------------------|
| Alcance interior | hasta 30m | hasta 40m |
| Alcance sin obstáculos | hasta 100 m | hasta 120 m |
| Potencia de transmisión | 1 mW (0dbm) | 2 mW (+3dbm) |
| Tensión de alimentación | 2.8 - 3.4 V | 2.8 - 3.6 V |
| Sensibilidad del receptor | -92dbm (1% PER) | -98dbm (1% PER) |
| Corriente de apagado | 10 uA | 1 uA |
| Topologías de red | Punto a punto, estrella | Punto a punto, estrella, malla |
| Rango de temperatura | -40 to 85 C | -40 to 85 C |
| Nº Entradas/Salidas digitales | 8 | 10 |
| Nº entradas analógicas | 7 | 4 |
| Nºsalidas analógicas | 2 | Ninguna |

Tabla 3.1: Tabla comparativa Serie1 vs Serie2 (modelos de potencia regular)

Los módulos de la Serie 1, basados en el chipset de Freescasle, están pensados para ser utilizados en redes con topología punto-a-punto o en estrella. Traen instalado el firmware 802.15.4, aunque pueden ser actualizados para soportar otros protocolos (por ejemplo el protocolo propietario DigiMesh, similar a ZigBee Mesh).

Los módulos de la Serie 2 están basados en el chipset de Ember Netwoks y fueron diseñados principalmente para ser utilizados en aplicaciones que requieren de redes con topología de malla. Dentro de esta serie existen los módulos originales ZB y los módulos 2B, que permiten reducir el consumo de potencia, y que pueden incorporar

un microprocesador programable, etc.

De cada tipo de módulo XBee existen dos versiones (PRO y Regular) que se diferencian en la potencia de transmisión. Los módulos PRO (Serie 1 y 2) tienen mayor capacidad de alcance, permitiendo incluso doblar la distancia de transmisión. Los módulos PRO tienen sin embargo un mayor consumo.

Con independencia de su serie, cada módulo XBee permite emplear diferentes tipos de antenas para transmitir y recibir las señales (ver figura 3.4). Entre las opciones de antenas actualmente disponibles, destacan:

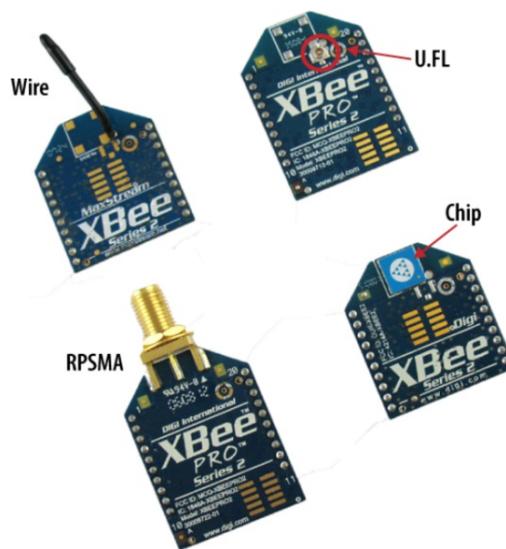


Figura 3.4: Tipos de antenas

- Chip antena. Es un pequeño y robusto chip cerámico que permite una radiación unidireccional cardioide (forma de corazón). Este tipo de antena radia o capta desde la parte frontal, teniendo mínima sensibilidad en su parte posterior donde se produce la atenuación.
- Wire antena. Es un sencillo cable que sobresale del módulo y que permite una radiación omnidireccional. Por lo tanto, cuando el cable se encuentra recto y perpendicular al módulo la distancia cubierta es prácticamente la misma en todas las direcciones. Su inconveniente es la fragilidad de la antena
- Conector U.FL. Es el conector más pequeño y frágil de las dos opciones posibles para introducir una antena externa. El uso de estas antenas es recomendable

para aplicaciones especiales en las que se requiere un diagrama de radiación específico, o cuando el módulo Xbee va a ser introducido en una caja metálica y es necesario que la antena sobresalga para evitar atenuaciones. Su inconveniente es que requiere de un pequeño cable entre el conector y la antena.

- Conector RPSMA. Es una variante más grande y voluminosa del conector U.FL, que permite montar la antena directamente sobre el módulo XBee sin necesidad de usar un cable de conexión.

3.1.3.1. Selección de Módulo

La elección del tipo de antena depende de la instalación que se quiera realizar. En este caso, se ha optado por la denominada ‘Wire Antena’ al ser más pequeña y barata que las antenas externas, siendo para la mayoría de las aplicaciones igual de útil. Por otro lado, el tener un diagrama de radiación omnidireccional nos permite un despliegue más sencillo y polivalente. La antena puede hacer frente a señales de cualquier dirección sin que el usuario tenga que preocuparse por la orientación de los módulos.

Para realizar este proyecto se ha optado por emplear módulos de la Serie1, ya que se pretende crear una red domótica de control y automatización con una arquitectura centralizada y una topología en estrella. Como se ha mencionado, el sistema está pensado para ser usado en el interior de la vivienda, por tanto las distancias serán relativamente reducidas y podrán ser cubiertas más que de sobra por el módulo coordinador, cuyo radio-alcance será suficiente para comunicarse con el resto de módulos distribuidos por la casa. No será necesario incrementar la profundidad de la red y usar nodos que actúen de repetidores (topología malla o árbol), por lo que no tendría mucho sentido emplear módulos de la Serie2, los cuales son más complejos de configurar. Además, como se ha mencionado anteriormente, llegado el caso, los módulos XBee Serie 1 pueden ser configurados con el firmware DigiMesh que tiene prestaciones similares a ZigBee Mesh.

Una vez elegida la topología en estrella, la elección de los módulos PRO garantiza el correcto funcionamiento incluso en viviendas de grandes dimensiones.

Por tanto, los módulos XBee que se emplearán en el desarrollo de este proyecto serán dispositivos de Digi, de la Serie 1, de tipo Pro y con antenas wire (dipolo de cable

flexible). Su identificador es XBP24-AW1-001.

A pesar de que se ha optado por trabajar con la Serie1-Pro, el sistema funcionaría correctamente con otros módulos, ya sea en la totalidad de la red o de forma combinada. Se podrían usar conjuntamente módulos Regulares y PRO, o bien módulos con diferentes tipos de antena; incluso, se podrían sustituir todos los dispositivos de la Serie 1 por módulos de la Serie 2.

3.1.4. Módulos XBee PRO Serie1

Como ya se ha mencionado, los módulos Xbee Serie1 son sistemas embebidos de radiofrecuencia que implementan el protocolo IEEE 802.15.4 para permitir la conectividad inalámbrica entre módulos [14].

Además de la antena y de otros elementos auxiliares, los Xbee de la Serie 1 incorporan un pequeño microcontrolador de 8 bits y un transceptor de RF. El microcontrolador que utiliza es el MC9S08GT60 del fabricante Freescale y el chip de radio es el MC13193 de Freescale. El microcontrolador es el encargado, entre otras cosas, de que los datos sean preparados para ser transmitidos según el protocolo IEEE 802.15.4. De la misma manera, se encarga de procesar los paquetes recibidos por el receptor para obtener los datos de la aplicación. En el anexo B se pueden observar los modos de operación de los XBee según la tarea que estén realizando en cada momento.

Entre las principales características de los Xbee Serie1 PRO destacan sus reducidas dimensiones, que operan en la banda libre ISM de 2.4 GHz, una tasa de transferencia de Radiofrecuencia de 256 Kbps, un alcance de cerca de 100 metros en interiores y una sensibilidad del receptor de -100 dBm. El resto de características se puede encontrar en el manual del módulo XBee [15] y su comparativa con los modelos de potencia regular se pueden observar en la tabla del anexo E.

Los módulos Xbee, además de ser capaces de transmitir y recibir datos mediante una comunicación inalámbrica, disponen de conversores ADC (Analog-to-Digital Converter), terminales digitales I/O, terminales de salida PMW (Pulse-Width Modulation) y de un puerto UART. Estas características resultan de gran utilidad ya que permiten emplear estos módulos para tareas sencillas, como la lectura de sensores o la activación de actuadores, sin necesidad de emplear microcontroladores externos, lo que se traduce en una reducción de costes y de consumo.



Figura 3.5: Diagrama esquemático de los pines del módulo Serie1

Como se observa en la figura 3.5 cada módulo dispone de 20 terminales para diferentes propósitos, incluidos dos para polarización. La función de los pines es configurable, es decir, un mismo terminal puede realizar diferentes tareas, sean de entrada o salida. Por ejemplo puede actuar como un ADC o como Digital I/O. En la tabla 3.2 se pueden observar todos los pines del módulo XBP24-AW1-001, sus nombres y sus descripciones

| Pin # | Name | Direction | Description |
|-------|------------------------|-----------|---|
| 1 | VCC | - | Power supply |
| 2 | DOUT | Output | UART Data Out |
| 3 | DIN / CONFIG | Input | UART Data In |
| 4 | DO8* | Output | Digital Output 8 |
| 5 | RESET | Input | Module Reset (reset pulse must be at least 200 ns) |
| 6 | PWM0 / RSSI | Output | PWM Output 0 / RX Signal Strength Indicator |
| 7 | PWM1 | Output | PWM Output 1 |
| 8 | [reserved] | - | Do not connect |
| 9 | DTR / SLEEP_RQ / DI8 | Input | Pin Sleep Control Line or Digital Input 8 |
| 10 | GND | - | Ground |
| 11 | AD4 / DIO4 | Either | Analog Input 4 or Digital I/O 4 |
| 12 | CTS / DIO7 | Either | Clear-to-Send Flow Control or Digital I/O 7 |
| 13 | ON / SLEEP | Output | Module Status Indicator |
| 14 | VREF | Input | Voltage Reference for A/D Inputs |
| 15 | Associate / AD5 / DIO5 | Either | Associated Indicator, Analog Input 5 or Digital I/O 5 |
| 16 | RTS / AD6 / DIO6 | Either | Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6 |
| 17 | AD3 / DIO3 | Either | Analog Input 3 or Digital I/O 3 |
| 18 | AD2 / DIO2 | Either | Analog Input 2 or Digital I/O 2 |
| 19 | AD1 / DIO1 | Either | Analog Input 1 or Digital I/O 1 |
| 20 | AD0 / DIO0 | Either | Analog Input 0 or Digital I/O 0 |

Tabla 3.2: Pines de los módulos XBee Serie1

3.1.4.1. Características eléctricas

En la Tabla 3.3 se describen las principales características eléctricas de los módulos XBP24-AW1-001. Se pueden observar los valores de tensión permitidos para la alimentación del módulo (idealmente 3.3 Voltios), las tensiones mínimas y máximas que soportan las entradas, las tensiones mínimas y máximas que ofrecen los pines de salida y los diferentes consumos del XBee.

| | Mínimo | Máximo | Unidad |
|--|-----------|------------|--------|
| Tensión de alimentación | 2.8 | 3.4 | V |
| Tensión de entrada nivel bajo (V_{IL}) | - | $0.35*VCC$ | V |
| Tensión de entrada nivel alto (V_{IH}) | $0.7*VCC$ | - | V |
| Tensión de salida nivel bajo (V_{OL}) | - | 0.5 | V |
| Tensión de salida nivel alto (V_{OH}) | $VCC-0.5$ | - | V |
| Tensión de referencia para ADC (V_{REF}) | 2.08 | VCC | V |
| Corriente en la transmisión RF (TX) | ~ 215 | | mA |
| Corriente en la recepción RF (RX) | ~ 55 | | mA |

Tabla 3.3: Características eléctricas del XBee

3.1.4.2. Interfaces utilizadas

De entre todos aquellos recursos de los que disponen estos módulos, a lo largo de este proyecto se ha hecho uso de terminales conversores analógico-digital, de terminales de salida digital y del puerto UART.

3.1.4.2.1. Comunicación serie (UART)

Los módulos Xbee, para poder comunicarse con otros dispositivos como microcontroladores u ordenadores, disponen de un puerto UART (Universal Asynchronous Receiver Transmitter) para la transmisión en serie de datos. Esta comunicación es asíncrona y full-duplex. Ha sido empleada en la unidad central de control.

Este tipo de comunicación puede realizarse de manera directa con cualquier equipo que disponga de UART compatible con la de la tecnología CMOS, como son la mayoría de los microcontroladores o la plataforma BeagleBone en este caso. También puede realizarse a través de un convertidor desde niveles del estándar RS-232 como sucede si se desea comunicar un módulo Xbee y un ordenador, tal y como se ha realizado para configurar estos módulos.

Los datos salen del módulo XBee a través del PIN 2 que corresponde a la señal de transmisión (Tx) y entran por el PIN 3 que corresponde a la señal de recepción (Rx). Hay que tener precaución con estas señales, para no confundirlas con los nombres que reciben en la UART de la tarjeta Beaglebone, ya que son denominadas justo al contrario. La mayoría de manuales de XBee las denominan señal OUT y señal IN, para no confundirse con las señales Tx y Rx que usa el microcontrolador y que se refieren a la antena (transmisión inalámbrica).

En la figura 3.6 se observan las conexiones establecidas para el módulo XBee de la unidad central, que de hecho, son las conexiones mínimas que debe tener un módulo Xbee para establecer una comunicación serie con otro dispositivo. No obstante, también es posible el uso de señales como RTS (Request To Send - pin 16) y CTS (Clear To Send - pin 12) para el Control de Flujo.

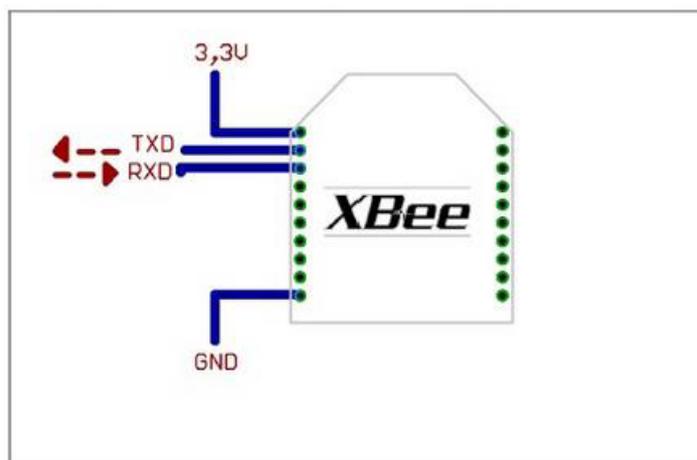


Figura 3.6: Conexiones mínimas requeridas

Como se ha mencionado la comunicación es serie y asíncrona, lo que implica respectivamente que los bits se transmiten individualmente de manera secuencial y que no se usa una señal de reloj, sino que la información de sincronización es extraída de los mismos datos.

La comunicación está fijada por un grupo de parámetros modificables como son: la velocidad en baudios, la cantidad de bits de datos, las opciones de paridad, la cantidad de bits de parada, etc. Para que la comunicación funcione correctamente es necesario que las UART de los dos dispositivos que se están comunicando tengan

estos parámetros configurados a los mismos valores. La configuración UART que se ha utilizado en los módulos XBee se indica en la tabla 3.4.

| | |
|-------------------------|----------------------|
| Velocidad | 9600 baudios |
| Bits de datos | 8 |
| Bits de stop | 1 |
| Control de Flujo | Sin control de flujo |
| Paridad | Ninguna |

Tabla 3.4: Configuración de los parámetros del puerto serie del XBee

En este caso cada byte transmitido por el Xbee a través del pin DOUT consistirá en un bit de inicio a bajo nivel, ocho bits de datos (los menos significativos delante) y un bit de parada a alto nivel (ver figura 3.7).

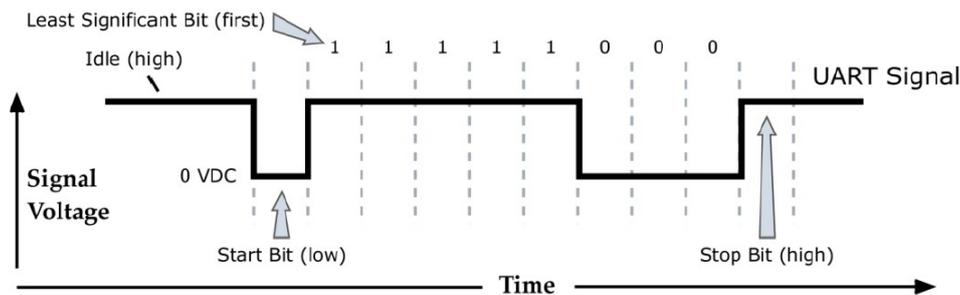


Figura 3.7: Ejemplo de la transmisión en serie del byte 0x1F según la configuración establecida

3.1.4.2.2. ADCs y DOs

Las salidas digitales y los conversores analógicos-digitales (ADC) serán empleados en los módulos de las unidades remotas para la lectura de sensores analógicos y la activación de actuadores.

Los ADC del módulo Xbee son conversores de 10 bits que ofrecen una resolución de 1024 niveles (0x03FF) y permiten leer un valor analógico y convertirlo a digital para ser transmitido. Existen un máximo de 7 pines ADC en cada módulo.

El rango de tensión admitido por los ADC depende del tipo de módulo Xbee, aunque nunca podrá superar los 3.3 voltios. En el caso de la Serie1 es necesario conectar el PIN 14 con un voltaje, que será el que establezca el límite de tensión superior del ADC. Se ha decidido fijar el valor máximo de 3.3 voltios para este pin.

Respecto a las salidas digitales (DO) hay 8 por módulo y una vez activadas, pueden presentar un valor high o low, lo que se traduce en una salida con tensión 3,3V ó 0V respectivamente. Se emplearán para conmutar relés y activar actuadores.

3.1.5. Modos de comunicación

Los módulos Xbee disponen de dos modos de trabajo para realizar las comunicaciones inalámbricas. Con estos dos modos, que se seleccionan por medio de los parámetros de configuración, se consigue adaptarse a un gran número de aplicaciones diferentes. A continuación se detallan cada uno de los modos de comunicación, con sus ventajas e inconvenientes, sus aplicaciones ideales y finalmente el modo seleccionado.

3.1.5.1. Modo transparente

Es el tipo de conexión que viene cargada por defecto en los módulos Xbee. Resulta la manera más sencilla de comunicarse y es la más fácil de configurar. Su funcionamiento es sencillo, todos los caracteres que son recibidos a través del terminal DIN (pin 3) pasan a la cola de transmisión de RF para que sean enviados al destino deseado. De manera inversa, todos los datos que llegan al módulo a través de RF son enviados por el terminal DOUT (pin2).

Los datos recibidos por el terminal DIN se guardan en el buffer de entrada. Dependiendo de cómo esté configurado el parámetro RO, se esperará a que llegue una cantidad mínima de caracteres, o bien, se esperará un cierto período de tiempo antes de que, finalmente, los caracteres sean integrados en paquetes RF y transmitidos.

Este modo de comunicación está destinado principalmente a la comunicación punto a punto y se utiliza frecuentemente para reemplazar la comunicación serie por cable. Se emplea también para comunicaciones punto a multipunto y para los denominados cables virtuales.

Punto a punto

Trabajar en modo transparente es una buena opción cuando se desea comunicar entre sí únicamente dos módulos. La configuración requerida es muy sencilla, basta con fijar las direcciones de destino de los módulos y unos pocos parámetros más:

- Se define arbitrariamente una dirección (MY) para el módulo A, que debe coincidir con la dirección destino (DL) del módulo B. Y lo mismo a la inversa.
- Los parámetros de Identificador de la Red (ID) y de Canal de Comunicaciones (CH) deben ser los mismos en los dos módulos.

Un caso en el que tendría sentido emplear este modo de operación, sería para realizar el control de un Robot o de un Vehículo Aéreo no Tripulado (o dron). Estos dispositivos móviles llevan uno o varios microcontroladores que se encargan del funcionamiento del autómatas y únicamente utilizan los módulos XBee para recibir instrucciones del controlador. Los microcontroladores se encargan de organizar la comunicación, creando los mensajes que serán enviados al terminal DIN del Xbee para que éste los transmita inalámbricamente. En el sentido contrario sucede lo mismo, el módulo Xbee únicamente envía por el terminal DOUT el mensaje recibido por RF, y será el microcontrolador el encargado de interpretarlo. Los microcontroladores son programados para crear mensajes de comunicación siguiendo una estructura de bytes definida por el usuario. Estos mensajes permiten el control de los dispositivos móviles, por ejemplo, la transmisión de las coordenadas espaciales que debe seguir un robot o la altitud y velocidad a la que debe volar un dron.

Esta opción fue descartada porque una comunicación punto a punto no es válida para los requerimientos de la red de este proyecto.

Punto a multipunto

Varios Xbee operando en Modo Transparente también permiten realizar comunicaciones punto a multipunto. Una vez todos los módulos estén configurados para que tengan el mismo canal de comunicación y el mismo identificador de red, se puede proceder de dos maneras.

1. La primera sería entrar en el Modo Comando antes de cada transmisión para fijar en la dirección de destino, la dirección del módulo al que se desea transmitir en cada ocasión. Esta opción se descartó debido a que en una red amplia de sensores y actuadores habría que estar cambiando constantemente el parámetro con la dirección de destino del coordinador.
2. La segunda manera es la denominada Broadcast, que consiste en el envío de la información desde un nodo al resto de nodos que estén en la misma red. El

inconveniente de este sistema y por el cual se descartó, es que los dispositivos finales tendrían que disponer de microcontroladores para poder interpretar una cabecera auxiliar que habría que añadir en el mensaje, la cual identificaría a qué módulo en concreto iba dirigido ese paquete.

3.1.5.2. Modo API

El objetivo del modo de comunicación API (Application Programming Interface) es transmitir los datos de una manera segura y estructurada, como alternativa al modo Transparente. Es más complejo pero ofrece mayores posibilidades en cuanto al control de la red y al aprovechamiento de las capacidades de los XBee [16][17]. Además, permite la configuración de los módulos Xbee y el ruteo de la información.

Para trabajar en este modo, el parámetro AP del módulo Xbee debe configurarse con el valor 1 o 2.

En el modo API, la información que entra y sale del XBee tiene que seguir una estructura específica y debe de estar empaquetada en tramas (frames), que definen operaciones y eventos dentro del módulo. Si el módulo se encuentra en modo API, todos los datos recibidos y enviados por la UART deben ser tramas de comando API. En el caso de que los datos que entran al XBee no cumplan las especificaciones de las posibles tramas API o que el checksum de la trama no sea correcto, los datos son descartados y no son enviados.

Hay cinco tipos de tramas, 2 para la transmisión y 3 para la recepción de datos y comandos:

- I. Frame de datos para transferir por RF
- II. Frame de comando, donde va contenido alguno de los comandos AT
- III. Frame de recepción de los datos
- IV. Frame de respuesta de un comando
- V. Frame de notificación de eventos

Cada frame contiene un campo con la dirección de destino, lo que permite que los datos puedan ser enviados a diferentes dispositivos sin tener que cambiar la dirección de destino mediante la configuración del módulo. En cuanto a la recepción de paquetes, es posible identificar al Xbee emisor, ya que la trama también incluye un campo con la dirección de éste. Estas dos características convierten al modo API en la opción idónea para aplicaciones informáticas que controlan una red de nodos con topología en estrella, como es el caso de este proyecto, y por lo tanto ha sido el primer motivo por el que ha sido seleccionado.

Por otro lado, el modo API es el único que permite controlar remotamente los dispositivos que forman los nodos de la red, lo que posibilita realizar lecturas y escrituras de los pines I/O de dichos dispositivos, siendo éste el segundo motivo por el que ha sido seleccionado. Por ejemplo, es posible activar los pines de salidas digitales y también realizar lecturas de los sensores conectados a los ADC del módulo remoto. Los resultados de estas lecturas son enviados al otro módulo a través de tramas API.

En resumen, esta capacidad de control remoto junto con la facilidad de comunicación con múltiples nodos (red estrella), han sido determinantes para la elección del modo de comunicación API para realizar este proyecto.

A modo de resumen, se podría decir que algunas de las ventajas de trabajar en modo API son:

- Permite identificar el origen y destino de las tramas.
- Sencillez para transmitir a diferentes destinos.
- Los datos pueden contener comandos AT.
- Permite configurar los módulos de manera remota.
- Permite recibir confirmación de cada paquete transmitido (ACK).
- Aviso de error en caso de fallo en la transmisión.
- Facilita información de la comunicación (potencia de la señal de recepción).

Por el contrario, podríamos decir que sus inconvenientes son:

- Mayor complejidad, ya que para transferir los datos es necesario estructurarlos en tramas.
- Es necesario emplear más bits para enviar la información.

Al ser API el modo en el que trabajan todos los Xbee del sistema domótico, en la siguiente sección se analizarán en detalle las tramas API.

3.1.6. Tramas API

En esta sección se hablará de la estructura de las tramas API y de los tipos de tramas existentes (llamados también tipos de comandos API). Finalmente se mostrará que implementación se ha dado a estas tramas.

3.1.6.1. Estructura general de las tramas

Las tramas API que se envían a través de la UART deben cumplir con una secuencia específica de bytes que indicarán al Xbee el tipo de operación a realizar. De igual manera, los módulos responderán enviando por el pin DOUT de la UART secuencias específicas que pueden ser desde datos recibidos de un dispositivo remoto hasta información acerca del estado del Xbee o del estado de la comunicación. En cualquier caso, todas las tramas deben cumplir una serie de especificaciones y tener la estructura que se indica en la figura 3.8.

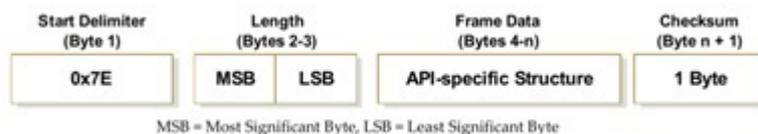


Figura 3.8: Estructura de la trama

■ Start Delimter

Cada trama API comienza con un byte de inicio, denominado “start delimiter”. Este es un único número que indica que se está al comienzo de una trama de datos y su valor es siempre 0x7E (en formato hexadecimal). Cuando se reciben datos desde el puerto serie de Xbee, lo primero que hay que hacer es buscar el

citado byte de inicio. Una vez recibido, se sabe que a continuación se encuentra el paquete de datos y el resto de la información..

■ Length Bytes

Los siguientes dos números después del byte de comienzo indican el tamaño en bytes del campo Frame Data. Esto permite conocer la longitud que se debe de leer para obtener la información.

Para la mayoría de las tramas, que no suelen tener un tamaño muy grande, es habitual que el segundo byte, denominado MSB (Most Significant Byte) sea cero.

■ Frame Data Bytes

Esta trama de datos es específica de cada tipo de mensaje recibido o enviado desde un módulo Xbee y es donde se encuentra contenida la información, así como los distintos tipos de estructuras de los comandos del modo API. Algunas tramas tendrán una gran cantidad de datos mientras que las más pequeñas pueden llegar a contener sólo 2 bytes.

Como se observa en la figura 3.9 este campo está compuesto por el byte “cmdID” que es el identificador del tipo de comando API que se envía o se recibe y por el subcampo “cmdData” donde van contenidos el resto de datos del comando API y que es de longitud variable. En la sección 3.1.6.2 se detallarán las diferentes estructuras del Data Frame en función de los posibles comandos API.

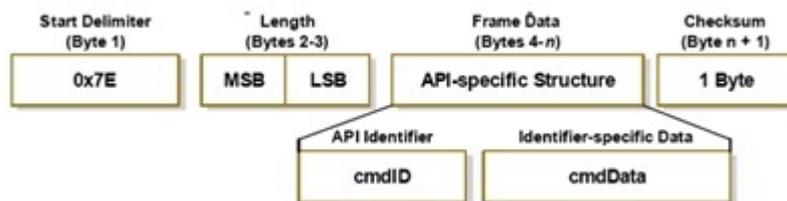


Figura 3.9: Estructura API específica

■ Checksum

El último byte de la trama es siempre el “checksum” y sirve para comprobar la integridad de la información recibida o enviada. Se calcula sumando todos los bytes de la trama (sin incluir los delimitadores ni la longitud) y quedándose

únicamente con los 8 bits menos significativos del resultado de esa suma, que serán restados a 0xFF. Para verificar el resultado, se suman todos los bytes (incluyendo checksum, pero excluyendo los delimitadores y la longitud) y el byte menos significativo del resultado debe ser igual a 0xFF. Este cálculo aritmético está diseñado para ser muy eficiente en los procesos de cómputo.

3.1.6.2. Tipos de comandos API

Hay diferentes tipos de comandos API que permiten mandar y recibir comandos AT normales y remotos, saber el estado del módulo, recibir lecturas de entradas digitales y analógicas, enviar y recibir mensajes de datos, etc.

El tipo de comando viene definido por el byte identificador (cmdID) e indica la función de la trama y cómo será la estructura de ésta. Permite seleccionar el modo correcto de extraer la información, es decir, ofrece la posibilidad de interpretar correctamente los bytes y los campos que forman la estructura específica, los cuales cambian en función del tipo de comando API.

En la tabla 3.5 se observan todos los tipos de comandos API soportados por los módulos Xbee Serie-1 y sus identificadores API. A continuación se examinan las estructuras específicas de los comandos API empleados en este proyecto. Además en el anexo C vienen detalladas las estructuras de los paquetes API Comando AT y de los paquetes API Transmisión de datos y Recepción de datos. Estos últimos, a pesar de no haber sido implementados en este proyecto, son de gran importancia para el futuro ya que serán los utilizados para la transmisión de datos específicos en unidades remotas con microcontrolador.

Comando AT remoto y respuesta de comando remoto

El modo API es el único capaz de controlar remotamente los dispositivos que forman los nodos de la red, lo que permite realizar lecturas y escrituras de los pines de entrada y salida de dichos dispositivos. Por ejemplo, es posible activar los pines de salidas digitales y también realizar lecturas de los sensores conectados a los ADC del módulo remoto. Los resultados de estas lecturas son enviados al otro módulo a través de tramas API. También sería posible que los módulos remotos operasen de forma automática enviando la lectura de sus sensores periódicamente.

| Identificador API | Tipo de comando |
|-------------------|---|
| 0x17 | Comando AT remoto |
| 0x97 | Respuesta de comando remoto |
| 0x08 | Comando AT |
| 0x09 | Comando AT (En cola de espera) |
| 0x88 | Respuesta de comando AT |
| 0x01 | Transmisión de datos (Direccionamiento de 16 bits) |
| 0x81 | Recepción de datos (Direccionamiento de 16 bits) |
| 0x00 | Transmisión de datos (Direccionamiento de 64 bits) |
| 0x80 | Recepción de datos (Direccionamiento de 64 bits) |
| 0x83 | Recepción de datos de entradas digitales y analógicas (Direccionamiento de 16 bits) |
| 0x82 | Recepción de datos de entradas digitales y analógicas (Direccionamiento de 64 bits) |
| 0x89 | Resultado de la transmisión |
| 0x8A | Estado del modem |

Tabla 3.5: Comandos API

Este tipo de comandos es de gran importancia ya que permite controlar remotamente cualquier Xbee y será el que se use en el sistema para comunicarse. El formato API permite acceder de manera remota a cualquier dispositivo en la red y es una característica permitida sólo en el modo API. Cualquier tipo de comando AT que pueda ser creado localmente puede ser enviado de forma inalámbrica, ya sea para leer entradas digitales y analógicas, escribir salidas o configurar el dispositivo remoto.

Los comandos AT remotos se pueden emplear para averiguar información de un módulo Xbee remoto, ya sea el valor al que está configurado uno de sus parámetros o el valor de los sensores que están conectados a los pines de entrada de dicho módulo. Estos comandos también se pueden utilizar para realizar cambios sobre el Xbee remoto, como por ejemplo modificar el valor de alguno de sus parámetros de configuración. En este segundo caso, la trama API que se envía debe contener un campo para el nombre del comando AT y otro campo para el valor del parámetro que se desea actualizar, mientras que en primer caso la trama API no incluye un campo para el valor.

La mecánica de funcionamiento es la siguiente: por la UART del Xbee emisor entra una trama API del tipo Comando AT Remoto (identificador API = 0x17) que contendrá las instrucciones para ser enviadas inalámbricamente a un nodo remoto. Una

vez que el Xbee remoto reciba por RF el paquete y realice lo que se ordena, enviará inalámbricamente un mensaje al modulo emisor con él resultado de la acción o con la información que se solicitaba. Cuando el Xbee reciba este mensaje, a través de su puerto serie saldrá una trama API del tipo Respuesta de Comando Remoto (con el identificador API=0x97), la cual contendrá el resultado de la instrucción que se envió de manera remota, ya sea una confirmación de que se realizó lo ordenado o los valores con la información que se solicitó.

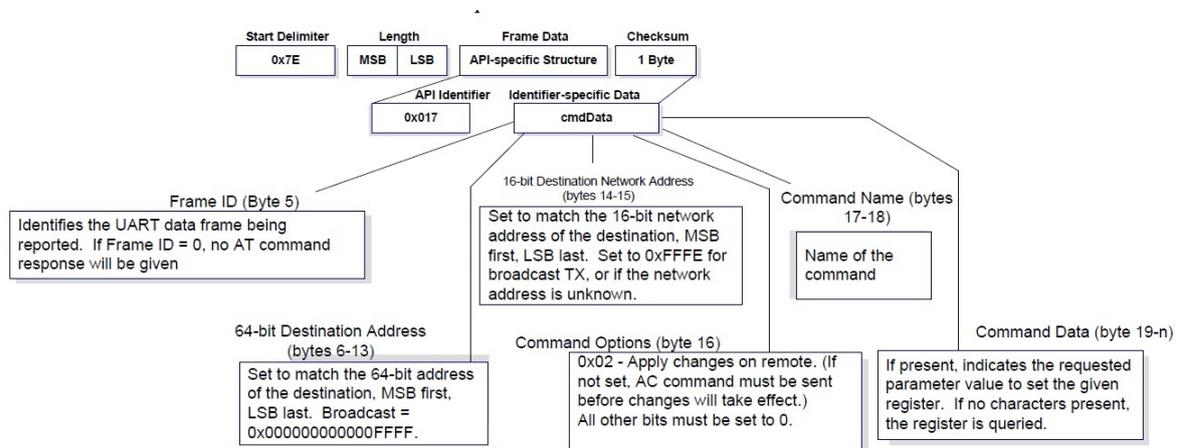


Figura 3.10: Estructura API de Respuesta de Comando Remoto

La estructura específica API en las tramas tipo Comando AT Remoto se divide en los siguientes campos: un byte para el identificador del tipo de comando API (0x17), el byte identificador del frame, 8 bytes para direccionar usando 64 bits, 2 bytes para usar direcciones de 16 bits, 1 byte de opciones, 2 bytes con el nombre del comando y un último campo opcional con el valor al que actualizar ese comando (ver figura 3.10). Al direccionar los paquetes se puede especificar la dirección de 64 bits o la de 16 bits. Para usar la dirección de 64 bits, en el campo de la dirección de 16 se ha de fijar el valor 0xFFFFE. Para usar la dirección de 16 bits, en el campo de la dirección de 64 se debe fijar el valor a cero.

La estructura específica API en las tramas tipo Respuesta de Comando Remoto se divide en los siguientes campos: un byte para el identificador del tipo de comando API (0x97), el byte identificador del frame, 8 bytes para direccionar usando 64 bits, 2 bytes para usar direcciones de 16 bits, 2 bytes con el nombre del comando, un byte para indicar el resultado del envío del Comando Remoto y un último campo opcional

con el valor de los datos del comando solicitado (ver figura 3.11).

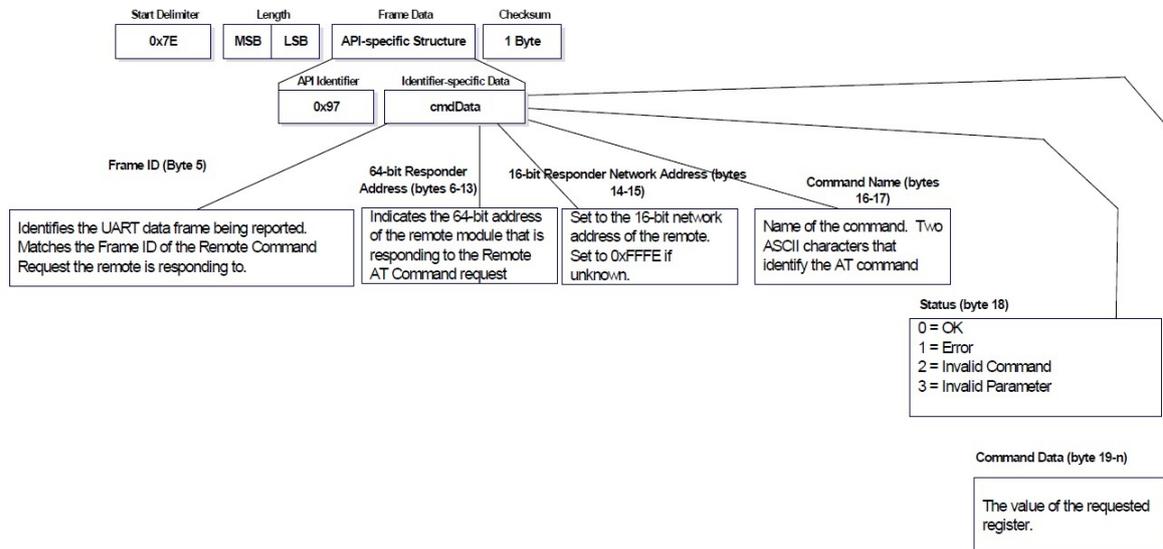


Figura 3.11: Estructura API de Respuesta de Comando Remoto

3.1.6.3. Implementación

A continuación se puede observar la implementación que se ha hecho de estas tramas API en el sistema, concretamente se muestran ejemplos de su utilización en la activación de actuadores y en la lectura de sensores, las cuales representan las dos facetas que cubre este sistema.

3.1.6.3.1. Activación de actuadores

La activación o desactivación de un actuador se realiza en dos pasos. En primer lugar se envía un paquete al módulo XBee remoto para que cambie el estado de la salida digital que corresponda, y en segundo lugar se envía otro paquete indicando al módulo que guarde en la memoria no volátil la nueva configuración anteriormente modificada.

Como se ha mencionado en la sección 3.1.4 los módulos Xbee disponen de gran cantidad de salidas digitales. Para la configuración de estos pines se usa el comando ATDn, donde 'n' representa el número de esa salida. Este comando permite configurar los pines de la siguiente manera:

- Dn=2; Pin establecido como entrada para Conversor Analógico-Digital.
- Dn=3; Pin establecido como entrada digital.
- Dn=4; Pin establecido como salida digital de nivel bajo.
- Dn=5; Pin establecido como salida digital de nivel alto.

Sin embargo, habrá que respetar las limitaciones de función de cada pin que se muestran en la tabla. Así por ejemplo, el comando ATD3=4, fijaría el pin17 (DIO3) como salida digital a nivel bajo.

Para guardar los cambios de configuración en la memoria no volátil de un módulo XBee se utiliza el comando 'WR'.

Durante este proyecto se utilizarán las salidas digitales de los Xbee para activar o desactivar los elementos que están conectados a los pines de estos módulos remotos. Para activar un actuador se fija el pin como salida digital de nivel alto, mientras que para desactivarlo se configura el pin como salida digital a nivel bajo.

A continuación se muestra un ejemplo de una trama API usada para configurar como salida digital de nivel alto un pin (DIO5) de un Xbee remoto. Esta trama deberá ser enviada por la UART al módulo Xbee coordinador, para que éste la transmita inalámbricamente al Xbee destino.

```
7E 00 10 17 01 00 00 00 00 00 00 00 00 01 02 44 35 05 66
```

El significado de los bytes es:

- | | |
|--------------------|--|
| 0x7E | Comienzo de trama. |
| 0x0010 | Longitud del Frame Data. |
| 0x17 | Identificador de Comando AT Remoto. |
| 0x01 | Identificador de trama (a elección del usuario). |
| 0x0000000000000000 | Dirección de 64 bits (en este caso todo a cero, ya que se emplearán direcciones de 16 bits). |
| 0x0001 | Dirección de 16 bits del Xbee remoto. |

| | |
|--------|---|
| 0x02 | Opción de comando para que se ejecute inmediatamente (equivalente a AC). Al activar esta opción, se ha librado de tener que enviar otra trama remota con el comando AC para que aplicase los cambios de configuración inmediatamente. |
| 0x4435 | Comando ATD5 (0x44 y 0x35 representan los caracteres D y 5 en Hexadecimal). |
| 0x05 | Parámetro al que se debe fijar el comando AT (en este caso Salida digital de nivel alto). |
| 0x66 | Checksum de la trama API. |

En el caso de que todo fuese correctamente, del puerto serie del Xbee coordinador saldría una trama tipo Respuesta de Comando Remoto como la siguiente:

```
7E 00 0F 97 01 00 13 A2 00 40 AA 5E A0 00 01 44 35 00 50
```

El significado de los bytes es:

| | |
|--------------------|--|
| 0x7E | Identificador del comienzo de trama. |
| 0x000F | Longitud del Frame Data. |
| 0x97 | Identificador de Respuesta de Comando Remoto. |
| 0x01 | Identificador de trama (el mismo en la trama anterior). |
| 0x0013A20040AA5EA0 | Dirección de 64 bits/ de hardware del módulo que responde al Comando AT Remoto. |
| 0x0001 | Dirección de 16 bits/corta/de red del Xbee que responde al Comando AT Remoto. 0x4435 Comando ATD5. |
| 0x00 | Estado correcto en la ejecución del comando remoto. |
| 0x50 | Checksum de la trama API. |

A continuación y como segunda parte del ejemplo, se muestra la trama que se envía al módulo Xbee remoto para indicarle que guarde en la memoria su nueva configuración.

7E 00 0F 17 01 00 00 00 00 00 00 00 00 01 00 57 52 3D

El significado de los bytes es:

| | |
|--------------------|--|
| 0x7E | Comienzo de trama. |
| 0x000F | Longitud del Frame Data. |
| 0x17 | Identificador de Comando AT Remoto. |
| 0x01 | Identificador de trama (a elección del usuario). |
| 0x0000000000000000 | Dirección de 64 bits (en este caso todo a cero, ya que se emplearán direcciones de 16 bits). |
| 0x0001 | Dirección de 16 bits del Xbee remoto. |
| 0x00 | Sin opción de comando especial. |
| 0x5752 | Comando ATWR (0x57 y 0x52 representan los caracteres W y R en Hexadecimal). |
| 0x3D | Checksum de la trama API. |

3.1.6.3.2. Lectura de sensores

Una de las posibilidades que debe ofrecer el sistema es averiguar el valor de los sensores analógicos conectados a los Xbee remotos, para realizar estas lecturas se utilizarán los pines de entrada analógicos de los Xbee.

Para poder leer un sensor analógico conectado a un módulo Xbee, un pin de éste debe ser configurado como entrada ADC mediante el comando ATDn=2. Por ejemplo el comando ATD1=2, fijaría el pin17 (AD3) como entrada analógica. A continuación se emplea 'IS' (Force Sample) que es un comando que lee y muestra el estado de todos los pines ADC/DIO activos en un módulo XBee.

A continuación se muestra un ejemplo de una trama API usada para averiguar el valor de un sensor conectado a un Xbee remoto. Previamente el pin conectado al sensor deber haber sido configurado para actuar como entrada ADC. Esta trama deberá ser enviada por la UART al módulo Xbee coordinador para que éste la transmita inalámbricamente al Xbee destino.

7E 00 0F 17 01 00 00 00 00 00 00 00 00 01 00 49 53 4A

El significado de los bytes es:

| | |
|--------------------|--|
| 0x7E | Comienzo de trama. |
| 0x000F | Longitud del Frame Data. |
| 0x17 | Identificador de Comando AT Remoto. |
| 0x01 | Identificador de trama (a elección del usuario). |
| 0x0000000000000000 | Dirección de 64 bits (en este caso todo a cero, ya que se emplearán direcciones de 16 bits). |
| 0x0001 | Dirección de 16 bits del Xbee remoto. |
| 0x00 | Sin opción de comando especial. |
| 0x4953 | Comando IS (0x49 y 0x53 representan los caracteres 'I' y 'S' en Hexadecimal). |
| 0x4A | Checksum de la trama API. |

Una vez que el Xbee remoto recibe inalámbricamente el paquete anterior, debería enviar al Xbee coordinador la información sobre el estado de sus entradas. Para lo cual emplea una trama API del tipo Respuesta de Comando Remoto (ver figura 3.11) que contiene dentro del campo de Datos de comando (Command Data) la cabecera mostrada en la figura 3.12 y el cuerpo mostrado en la figura 3.13. La información que contiene la cabecera es el número de muestras tomadas y una máscara que indica los canales que están activos, ya sean analógicos o digitales. En el cuerpo aparecen 2 bytes para la información sobre las salidas digitales y a continuación la información de todas las entradas analógicas activas en ese momento. Para mostrar la información de cada entrada analógica se dedican 2 bytes. Dichas entradas vienen ordenadas en la trama partiendo de la menor (AD0) hasta la mayor (AD6). Es de estos campos de donde se obtendrá la información deseada con el valor del sensor.

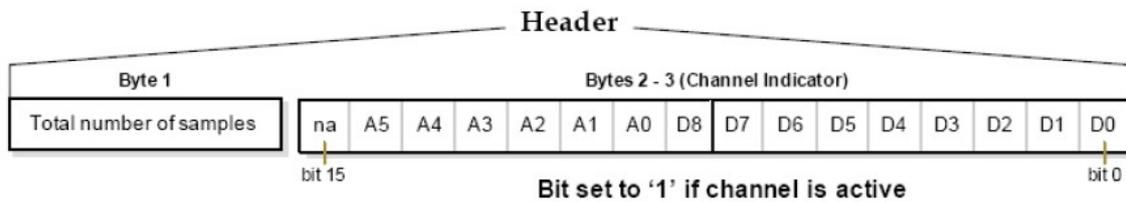


Figura 3.12: Cabecera. Número de muestras y máscara del canal

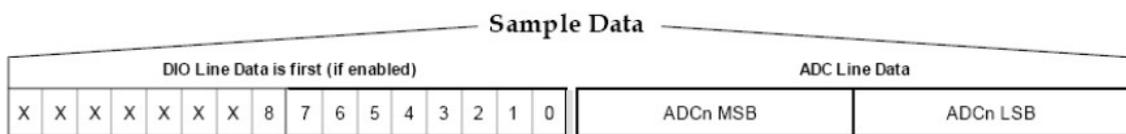


Figura 3.13: Cuerpo. Estados I/O digitales y ADC

Un ejemplo de una posible trama Respuesta de Comando Remoto que podría salir a través del puerto serie del Xbee Coordinador, sería:

7E 00 16 97 01 00 13 A2 00 40 AA 5E A0 00 01 49 53 00 01 04 00 00 00 01 68 BF

El significado de los bytes es:

- 0x7E Identificador del comienzo de trama.
- 0x0016 Longitud del Frame Data.
- 0x97 Identificador de Respuesta de Comando Remoto.
- 0x01 Identificador de trama (el mismo en la trama anterior).
- 0x0013A20040AA5EA0 Dirección de 64 bits/ de hardware del modulo que responde al Comando AT Remoto.
- 0x0001 Dirección de 16 bits/corta/de red del Xbee que responde al Comando AT Remoto. 0x4435 Comando ATD5.
- 0x00 Estado correcto en la ejecución del comando remoto.

| | |
|--------|--|
| 0x01 | Número de muestras |
| 0x0400 | Máscara de canales activos (en este caso indica que sólo está activo A1, es decir únicamente el pin 19 configurado como AD1) |
| 0x0000 | Estado de las salidas digitales (en este caso desactivadas, aunque los 0 binarios también podría indicar que están configuradas como salidas digitales de nivel bajo). |
| 0x0168 | Valor en Hexadecimal de la única entrada analógica activada (En este caso AD1=360). Este valor digital representa la lectura del sensor que esta conectado. |
| 0xBF | Checksum de la trama API. |

3.1.7. Configuración de los Módulos XBee

Los módulos Xbee que forman los nodos de la red deben ser adecuadamente configurados para que desempeñen la función que el usuario les desea asignar. Para configurarlos es necesario hacer uso de un conjunto de comandos, algunos ya mencionados y otros que se detallarán a lo largo de esta sección, los cuales modifican los parámetros del XBee.

La configuración de los módulos Xbee se puede realizar de diversas maneras, introduciendo directamente comandos AT o a partir de tramas API. Esta configuración se puede realizar a través de un microcontrolador, a partir de un ordenador, etc.

Aunque una de las posibilidades que ofrece el sistema, una vez esté operativo, es la de modificar alguno de los parámetros de configuración de los XBee End Devices enviando tramas API, antes de eso, los módulos deben haber sido configurados según la función que van a desempeñar en la red. En nuestro caso esta primera configuración básica de los XBee se ha realizado a través del ordenador con el programa X-CTU. Para comunicar los módulos Xbee con el ordenador se utilizan unas tarjetas de desarrollo que actúan de interfaces.

A continuación se analizarán estas tarjetas, se profundizará en el programa de configuración X-CTU y finalmente se detallarán los parámetros con los que han sido configurados los XBee del sistema.

3.1.7.1. Tarjetas de desarrollo

Existen gran cantidad de equipos fabricados para permitir la conexión entre el ordenador y los módulos XBee, desde dispositivos sencillos llamados USB Explorer hasta otros más complejos que incorporan periféricos como sensores, pulsadores, etc. En este caso se van a utilizar los dispositivos con los que cuenta el laboratorio, que son dos tarjetas de desarrollo una con interface RS-232 y la otra con interface USB (ver figura 3.14). Ambas han sido fabricadas por Digi y se denominan XBIB-R-DEV y XBIB-U-DEV respectivamente.

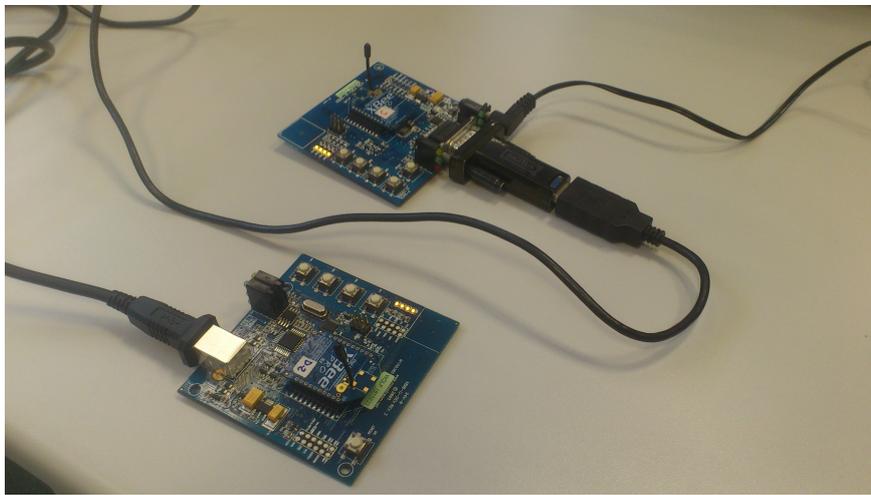


Figura 3.14: Tarjetas de desarrollo XBIB-U-DEV y XBIB-R-DEV

Estas tarjetas disponen numerosos componentes, entre ellos se puede destacar la presencia de LEDs que ofrecen la posibilidad de ver el estado de los pines que actúan como salidas digitales, otros que indican la actividad de los pines de transmisión y recepción de la UART del módulo Xbee y otros tres que indican la diferencia entre la RSSI (Received Signal Strength Indicator) la sensibilidad del receptor. También disponen de jumpers y de pulsadores para reiniciar el modulo o para cambiar el estado de los pines configurados como entradas digitales. Además ambas placas de desarrollo tienen 2 conectores o zócalos de 10 pines en los que se inserta cualquier tipo de módulo XBee (Serie1 o 2).

Las dos tarjetas disponen de reguladores de voltaje que adaptan la alimentación que reciben a la tensión de trabajo de los XBee. El modem con interface USB se alimenta de la propia conexión USB, mientras que el modem con interfaz RS-232

requiere de una alimentación externa a través de un DC-Jack, soportando un rango de alimentación desde 6Voltios a 20Voltios.

La placa XBIB-U-DEV dispone de un conector USB tipo B para la conexión con el ordenador. Su chip (FT232BM) convertirá señales TTL a un puerto Serie-USB simulado. La placa XBIB-R-DEV tiene un conector DB-9 para interactuar con el puerto Serie del PC y mediante el chip Max232 convierte señales TTL a señales RS232. Sin embargo ya que el ordenador no disponía de ningún puerto serie RS-232, se tuvo que utilizar un cable adaptador USB a SerieRS232(DB9). Por lo tanto para ambos casos se necesitó instalar los drivers VCP (Virtual COM Port) de FTDI, que permiten que los USB aparezcan como un puerto serie en el PC. Estos drivers pueden obtenerse de la página del fabricante (FTDI): <http://www.ftdichip.com/Drivers/VCP.htm>

Estas tarjetas de desarrollo junto con X-CTU permitirán configurar los módulos XBee, realizar pruebas del alcance inalámbrico de sus radios, y también probarlos y comunicarse con otros XBee. A menudo se las llama también tarjetas de entrenamiento o módems.

3.1.7.2. Software X-CTU

X-CTU es un programa de ordenador ofrecido por Digi International que permite configurar los módulos Xbee de manera fácil, visual y rápida. El software puede ser descargado gratuitamente de la página de Digi.

La pantalla de inicio del Software X-CTU será como la que se observa en la figura 3.15. En dicha ventana se mostrarán las conexiones USB-Serial disponibles, las cuales se establecen automáticamente siempre que los drivers hayan sido instalados.

En esta misma pantalla se podrán configurar los parámetros de conexión UART de la manera que se indicó en la sección 3.1.4.2.1. Finalmente hay un botón Test/Query que al presionarlo comprueba si se ha establecido la conexión correctamente. En caso afirmativo se muestra un mensaje de confirmación que devuelve la serie y el modelo del módulo.

En la pestaña Range Test se podrá probar el alcance de la radio. Para realizar estas pruebas se necesitará que, además del módulo Xbee conectado al ordenador, haya otro activo que vuelva a enviar todo lo que reciba. Ambos módulos deberán estar en modo transparente. El Xbee conectado al ordenador empezará a enviar paquetes

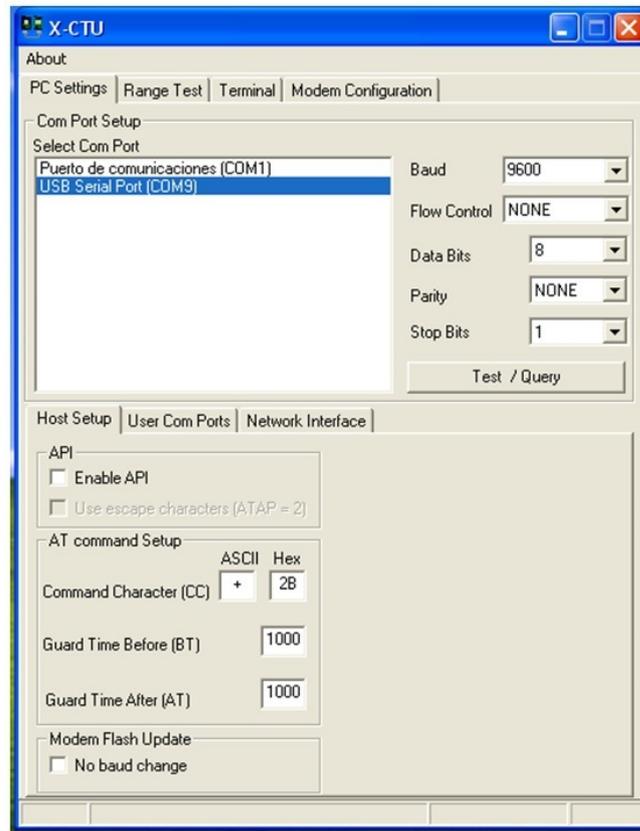


Figura 3.15: Pantalla de inicio de software X-CTU

automáticamente y el Xbee remoto, en caso de recibirlos, se los reenviará. En el X-CTU se podrá ver si se reciben todos los paquetes y con qué intensidad se reciben. De esta manera se podrá tener una aproximación rápida de cuál es el límite del alcance.

En la pestaña Modem Configuration se pueden configurar todos los parámetros del Xbee al mismo tiempo (ver figura 3.16). Pulsando el botón Read se mostrará por pantalla la configuración del módulo, es decir, aparecerán todos los parámetros del Xbee incluidos su modelo y su versión. Se mostrará la dirección corta del Xbee (MY), el identificador de la red (ID), el parámetro AP, etc. Los parámetros de configuración podrán ser almacenados en un fichero o cargados mediante los botones Save y Load respectivamente.

Los parámetros podrán ser cambiados según las necesidades de conexión o aplicación. Para grabarlos en la memoria del Xbee y que queden configurados con sus nuevos valores basta con pulsar el botón Write. En esta pestaña también se podrá actualizar

el firmware de los módulos, sin embargo hay que tener en cuenta que todos los dispositivos de la red deben operar con la misma versión de firmware. Durante este proyecto los Xbee con los que se trabaja usan la versión 0x10EC.

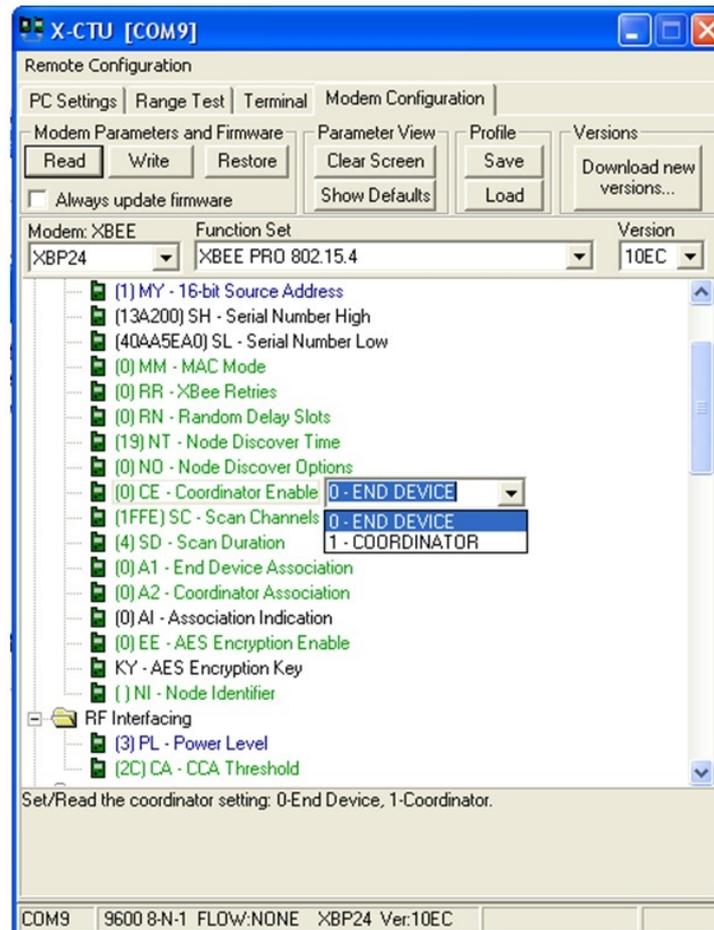


Figura 3.16: Configuración general del módulo

A través de la pestaña Terminal se realizaron las pruebas (entrenamiento) de comunicación entre módulos XBee¹. Inicialmente se realizaron pruebas y se ensayó con 2 XBee trabajando en modo transparente (ver figura 3.17). Tras esto, se pasó a realizar ensayos con dos módulos trabajando ambos en modo API (ver figura 3.18 y figura 3.19). Durante estos ensayos se probó con diferentes tipos de tramas API, y se eligió las que mejor se adaptaban a nuestras necesidades. Durante esta fase del proyecto se

¹En el X-CTU los datos en azul y en rojo representan los datos transmitidos y los datos recibidos respectivamente.

comprobó que las tramas se creaban con una estructura correcta, que eran enviadas inalámbricamente por los XBee y que realizaban la función que se esperaba.

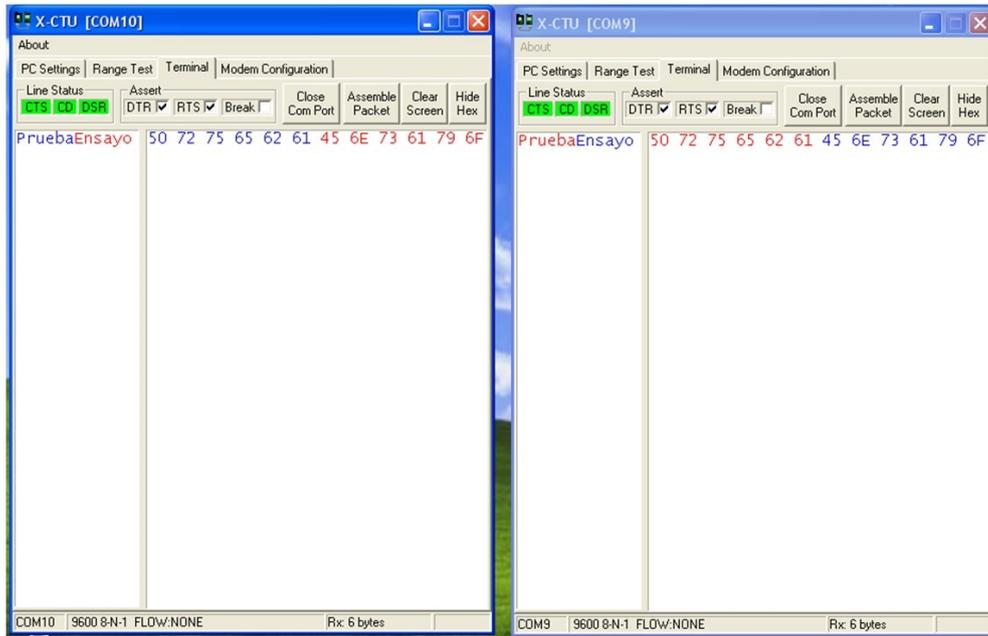


Figura 3.17: Pestaña Terminal. Ejemplo de comunicación en Modo Transparente

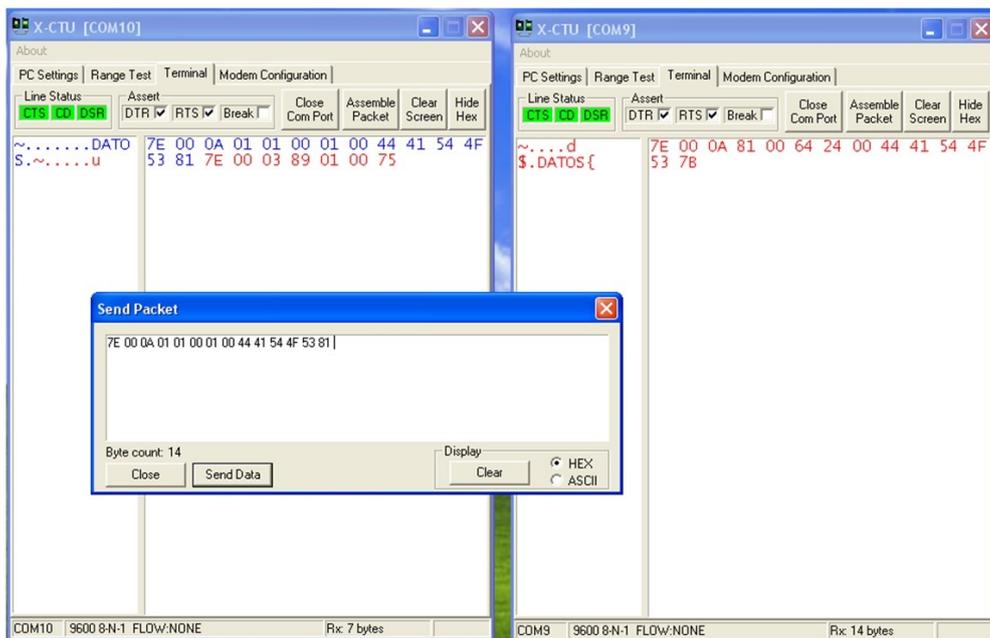


Figura 3.18: Pestaña Terminal. Ejemplo de comunicación API (Transmisión de datos)

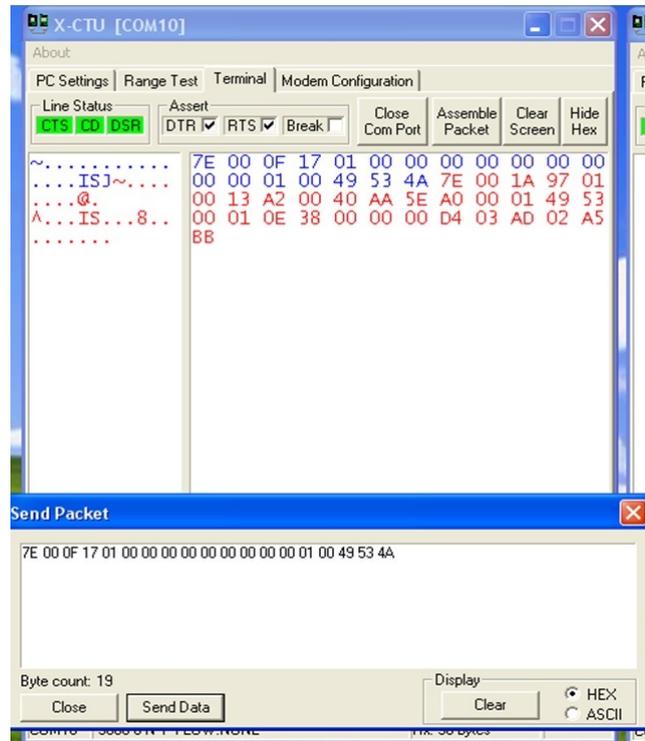


Figura 3.19: Pestaña Terminal. Ejemplo de comunicación API (Comandos Remotos)

3.1.7.3. Parámetros de Configuración

Los módulos XBee permiten direccionamiento de 16 bits o de 64 bits. La dirección hardware (64 bits) viene de fábrica con cada módulo XBee y es única y no modificable. Las direcciones cortas (16 bits), junto a otros parámetros, las fija el usuario al introducir el XBee en la red. El direccionamiento de 16 bits permite hasta 65536 posibles nodos en una red y ha sido el elegido para este proyecto.

La dirección de 16 bits del XBee dentro de la red la define el parámetro MY, que debe tener un rango entre 0x0000 y 0xFFFFD (0xFFFFE y 0xFFFFF son para habilitar direccionamiento de 64 bits). En el caso de usar este tipo de direccionamiento el parámetro DH debe configurarse a 0 y el comando DL debe configurarse con la dirección del módulo destino. Aunque en este caso este parámetro no afecta a la comunicación al usar modo el API y por tanto definir la dirección destino en cada trama, se recomienda fijarlo con la dirección del coordinador.

El objetivo de este proyecto es crear un sistema con topología de red en estrella y que

esté formado por un coordinador y el número de dispositivos finales que se desee. Todos los módulos XBee trabajarán en modo API comunicándose mediante un pequeño grupo de tramas de tipo Comando Remoto y Respuesta de Comando Remoto. La comunicación será Maestro Esclavo, es decir los nodos finales solo enviarán paquetes al coordinador cuando éste lo pida. Para este proyecto solo se ha implementado el sistema prototipo, constituido por el coordinador y un único nodo End Device. En la tabla 3.6 se representan alguno de los parámetros de configuración más relevantes de los XBee, en ella se puede observar como se han configurado tanto el XBee coordinador como el XBee Dispositivo Final. También se muestra como debería configurarse cualquier otro dispositivo final que se añada a la red, ya que ésta se ha pensado para ser flexible y ampliable a nuevos nodos.

| | XBee Coordinador | XBee Disp. Final Utilizado | XBee Disp. Final Posible | Descripción |
|-----------|-----------------------------|---------------------------------------|-------------------------------------|---|
| CH | 0xC | 0xC | 0xC | Todos deben de trabajar en el mismo canal |
| ID | 0x9032 | 0x9032 | 0x9032 | Mismo identificador de red |
| MY | 0x0064 | 0x0001 | A elegir | Una dirección distinta por Xbee |
| DH | 0x0000 | 0x0000 | 0x0000 | Direccionamiento de 16 bits |
| DL | 0x0001 | 0x0064 | 0x0064 | No afecta en API |
| CE | 1 | 0 | 0 | Un único coordinador de red |
| AP | 1 | 1 | 1 | Todos en API |
| D0 | 0 | 2 | A elegir | - |
| D1 | 0 | 2 | A elegir | - |
| D2 | 0 | 2 | A elegir | - |
| D3 | 0 | 4 | A elegir | - |
| D4 | 0 | 4 | A elegir | - |
| D5 | 0 | 4 | A elegir | - |
| IT | 1 | 1 | 1 | Nº de muestras ADC |
| IU | 1 | 1 | 1 | Habilitar salida de datos por UART |

Tabla 3.6: Configuración de los XBee de la red

Algunos parámetros como el Identificador de Red (ID) o el canal de comunicaciones

(CH) se han elegido arbitrariamente y lo importante es que el valor coincida en todo los nodos de la red. En este caso se decidió trabajar en la banda de 2.4GHz, concretamente se opera en el canal 12 (CH=0xC), lo que implica que la frecuencia central de trabajo es 2.41 GHz.

$$F_c[\text{MHz}] = 2405 + 5 * (\text{N}^\circ \text{canal} - 11)$$

En el anexo D se adjunta la información con la configuración completa (todos los parámetros) de los módulos XBee.

Capítulo 4

Unidades remotas

Las unidades remotas o periféricas constituyen los nodos de cualquier red de control y automatización. A lo largo de este capítulo, se detallarán cómo son las tarjetas o circuitos impresos (PCB) que se han construido para que formen la unidad remota de este sistema.

Como se ha comentado, el hecho de que el sistema disponga de una topología en estrella permite tener varias unidades remotas trabajando al mismo tiempo y en diferentes lugares de la vivienda. Estas unidades podrían tener muchos tipos de diseños y componentes diferentes [18][19], por lo que se ha optado por desarrollar uno específico de acuerdo a las aplicaciones que debía cubrir el sistema domótico: consulta del estado de sensores repartidos por la vivienda y la activación o desactivación de diferentes dispositivos en cualquier lugar de la casa.

La unidad remota ha sido desarrollada y construida para cubrir dos objetivos, en primer lugar disponer de un dispositivo que permita realizar pruebas y comprobar el funcionamiento del sistema domótico completo, y en segundo lugar para disponer de una unidad remota, que se podría definir como básica, pero que pueda ser desplegada por cualquier lugar de la casa siendo capaz de cubrir las necesidades habituales y dejando para el futuro otras más específicas.

Tras analizar diferentes configuraciones hardware para la unidad remota estándar, se concluyó que ésta debía incluir tanto actuadores como sensores.

Los actuadores son componentes imprescindibles en cualquier sistema domótico, al permitir realizar cambios físicos en el entorno. Existen muchos tipos de actuadores

como son los motores eléctricos, los dimmers, los relés, etc. Éstos últimos actúan a modo de interruptor permitiendo cerrar o abrir un circuito eléctrico y por tanto apagar o encender dispositivos como bombillas. Para la unidad remota se ha optado por incluir varios relés que permitan activar o desactivar equipos al mismo tiempo.

Los sensores también son elementos imprescindibles en cualquier instalación, ya que permiten medir magnitudes externas y en función de ello, tomar las decisiones oportunas. En el área de la domótica hay muchos tipos de sensores que son de gran utilidad, para este proyecto se ha optado por utilizar sensores que midan magnitudes de interés y que al mismo tiempo puedan ser obtenidas con sensores analógicos. Otros tipos de sensores como detectores de presencia o sensores de alarma (detectores de humo e incendios) también son de gran utilidad pero requieren de la presencia de un microcontrolador. La unidad periférica desarrollada incorpora los tres sensores siguientes:

- Sensor de temperatura. Conocer la temperatura interior o exterior permite una regulación óptima de la calefacción. Entre las muchas aplicaciones que ofrece este tipo de sensores, destacan: mantener una temperatura específica en un lugar concreto de la vivienda, decidir si abrir o cerrar las persianas, si es el momento adecuado para activar el riego en función de la temperatura exterior, etc.
- Sensor de luminosidad. Algunas de las posibilidades que ofrece un sensor de este tipo son: informar al usuario del estado de la iluminación, lo que permite conocer si todas las luces han sido apagadas; regular el encendido de las luces en función de la luminosidad ambiente, incrementando o reduciendo la potencia aplicada a las bombillas; etc.
- Sensor de humedad. Conocer el grado de concentración de agua en el ambiente permite optimizar la gestión del riego al saber si ha llovido en el exterior. Además un sensor de este tipo puede ser útil como alarma en caso de inundación.

Además de incorporar sensores y actuadores, la unidad remota incluye el módulo Xbee, el cual permite la interconexión por radiofrecuencia con la Unidad Central de Control (ver figura 4.1). Por otro lado, también incorpora otros elementos necesarios,

desde componentes electrónicos auxiliares como reguladores de tensión o condensadores, hasta conectores y cables. La tarjeta impresa se encargará de adaptar la tensión adecuada para los elementos integrados en ésta.

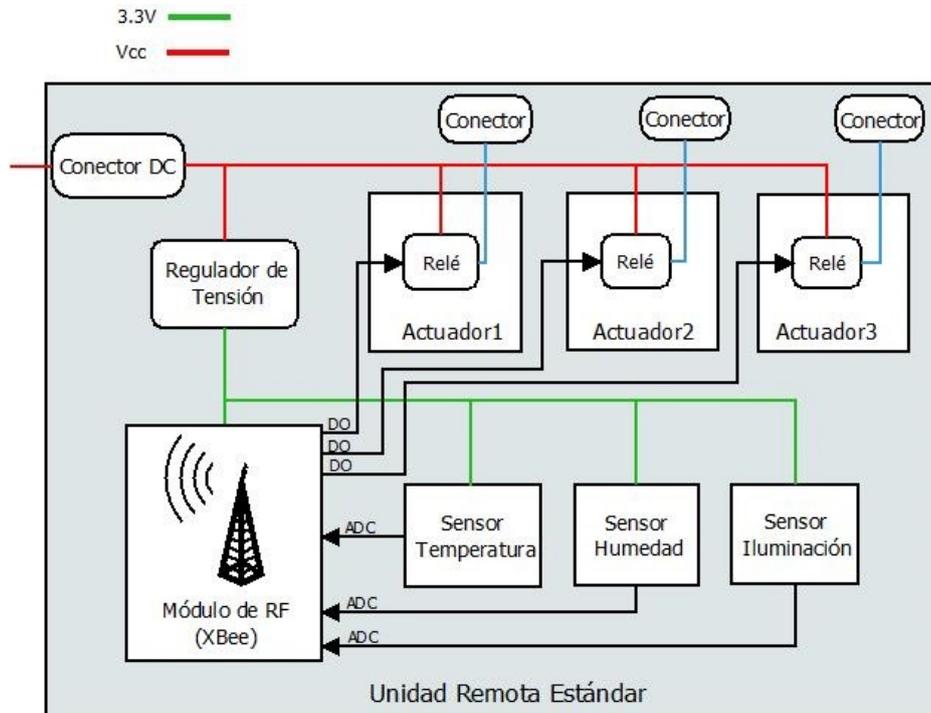


Figura 4.1: Esquema de la unidad remota desarrollada

4.1. Hardware

A lo largo de esta sección se presentará los componentes de hardware y el diseño y construcción de los circuitos que componen la unidad periférica.

4.1.1. Componentes

En esta sección se detallarán algunos de los principales componentes utilizados para realizar el diseño del hardware. Otros componentes presentes en el diseño, serán: resistencias, condensadores, DC-Jacks, leds, conectores, etc.

Debido a que el diseño de la unidad remota prototipo y sus circuitos impresos no presenta exigencias en cuanto a la superficie máxima que pueden tener, se ha optado

por seleccionar los encapsulados de los componentes que presentan menos dificultades a la hora de ser soldados con las herramientas e instrumentos disponibles en el laboratorio.

4.1.1.1. Sensor de temperatura

Hay varios tipos de sensores que obtienen medidas de la temperatura. Existen los Termopares y los sensores RTD, pero ambos han sido descartados ya que son más adecuados para situaciones que requieran rangos de temperatura muy grandes y, por lo tanto, no útiles en el área de la domótica donde el rango de medida se va a limitar a la temperatura ambiente en una vivienda o en el exterior. Los Termistores ofrecen una alta precisión a baja y media temperatura pero tienen el inconveniente de una calibración difícil y de ser exponenciales (no lineales). Finalmente, se optó por emplear sensores de temperatura de circuitos integrados, los cuales son muy económicos y se adaptan adecuadamente a las aplicaciones domóticas.

| | LM35 | TMP36 | AD22103 |
|-----------------------|------------------------|------------------------|----------------|
| Fabricante | Texas Instruments | Analog Devices | Analog Devices |
| Precio por unidad (€) | 3.90 | 1.46 | 3.87 |
| Encapsulados | TO-92, TO-220, SOIC | TO-92, SOT-23, SOIC | TO-92, SOIC |
| Alimentación (V) | 4-30 | 2.7-5.5 | 2.7-3.6 |
| Límites (°C) | -55-150 | -40-125 | 0-100 |
| Tensión de salida (V) | 0.02-1.5 | 0.1-2 | 0.25-3.05 |
| Precisión máxima (°C) | 0.5 | 1 | 0.5 |
| Tensión a 25°C (mV) | 250 | 750 | 950 |

Tabla 4.1: Sensores de temperatura candidatos a formar parte de la electrónica

Tras analizar varios sensores de temperatura analógicos, el circuito integrado escogido fue el modelo AD22103. Este componente es la mejor opción ya que su rango de medida es el más pequeño y por tanto el que más se ajusta a las variaciones de la temperatura en un clima templado. Además, este sensor es el que tiene mayor Salida a Fondo de Escala (3.05V-0.25V), lo que le permite aprovechar al máximo los 10 bits de resolución del convertidor ADC que tiene el módulo Xbee, el cual convierte muestras analógicas de entre 0 y 3.3 voltios.

En la figura 4.2 se puede observar el circuito de acondicionamiento que necesita el sensor según sus especificaciones.

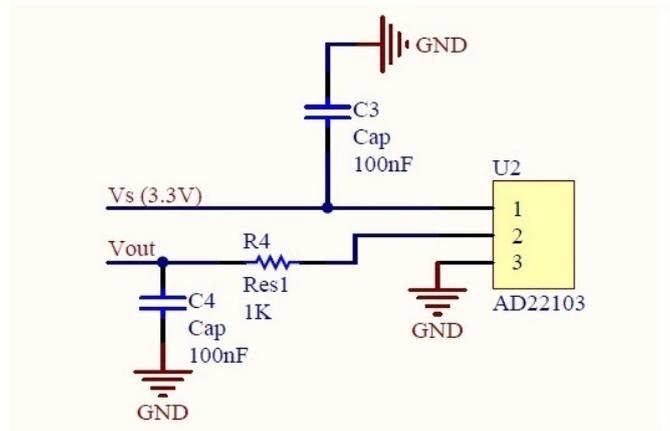


Figura 4.2: Esquemático del circuito de acondicionamiento del sensor de temperatura

4.1.1.2. Sensor de humedad

| | CC2A23 | SENS808H5V5 | HIH-5030 |
|-------------------------|---------------|-------------|-----------|
| Fabricante | Amphenol A.S. | Sencera Co | Honeywell |
| Precio por unidad (€) | 13.45 | 22.34 | 7.24 |
| Límites (HR) | 0 %-100 % | 0 %-100 % | 0 %-100 % |
| Tiempo de respuesta (s) | 7 | 15 | 5 |
| Precisión (HR) | 2 % | 4 % | 3 % |
| Alimentación (V) | 3.3 | 5 | 3.3 |
| Temp. de trabajo (°C) | -40-125 | -40-85 | -40-85 |

Tabla 4.2: Sensores de humedad candidatos a formar parte de la electrónica

Tras analizar varios tipos de sensores de humedad analógicos se ha optado por el modelo HIH-5030-001. Este componente ha sido seleccionado por ser el de menor coste y por presentar unos valores de salida comprendidos entre 0.5 y 2.6 voltios, que están dentro del rango de conversión del ADC del Xbee. En la figura 4.3 se puede observar el circuito de acondicionamiento que requiere.

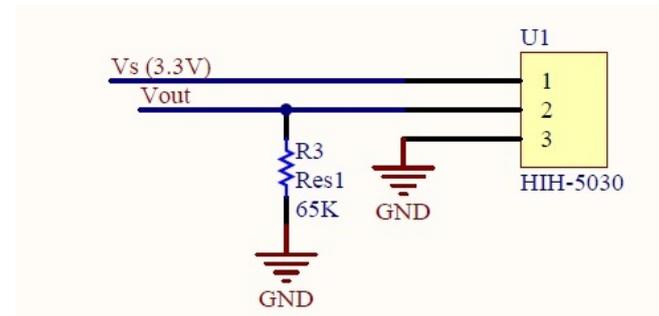


Figura 4.3: Esquemático del sensor de humedad

4.1.1.3. Sensor de luminosidad

Como sensor de luminosidad se ha decidido utilizar una fotorresistencia (LDR), que es un sensor resistivo basado en semiconductores. Las resistencias de este tipo de componentes varían según la luz que incide en su superficie. Así, cuando están en oscuridad su resistencia es alta y cuando recibe luz su resistencia disminuye considerablemente. El componente seleccionado ha sido una LDR modelo C-2795 con diámetro de 3.4 mm.

Según el Sistema Internacional de Unidades, el nivel de iluminación se mide en la unidad Lux. En la tabla 4.3 se puede observar una aproximación del nivel de iluminación en algunas circunstancias habituales.

| Situación | Nivel de iluminación aproximado (LUX) |
|------------------------------|---------------------------------------|
| Día de verano despejado | 100000 |
| Día de verano nublado | 20000 |
| Oficina o Laboratorio HCTLab | 1000 |
| Salón de una vivienda | 200 |
| Claro de luna | 1 |

Tabla 4.3: Niveles de iluminación

En la hoja de características de la fotorresistencia se encuentra la gráfica con la curva característica de la resistencia LDR en función de la iluminación, de donde se puede obtener que la expresión matemática de esta curva logarítmica es:

$$y = -0,699x + 2,99$$

Por lo tanto, en valores reales la curva de la fotorresistencia viene definida por la siguiente ecuación:

$$R_{LDR} = 10^{2,966} L^{-0,699}$$

De tal modo, la luminosidad (medida en LUX) viene dada por la siguiente expresión:

$$L = 10^{-\frac{\lg\left(\frac{R_{LDR}(k)}{10^{2,966}}\right)}{0,699}} \quad (4.1)$$

Para conocer el nivel de iluminación se necesita conocer el valor de la Resistencia R_{LDR} , para lo cual se procede como con cualquier sensor resistivo, es decir, se incluye otra resistencia auxiliar de tal manera que se consigue tener un divisor de tensión que actúa como circuito de acondicionamiento (ver figura 4.4). Estos tipos de circuitos permiten calcular el valor de la resistencia a través de un valor de tensión (ver ecuación 5.2)

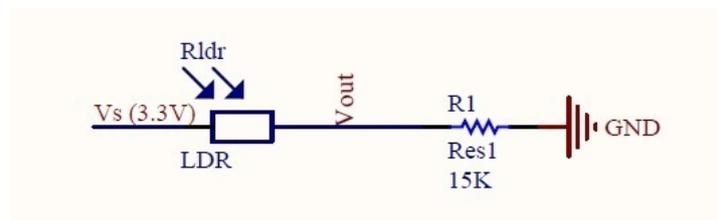


Figura 4.4: Circuito de acondicionamiento de la fotorresistencia

$$V_{OUT} = V_S \left(\frac{R_1}{R_1 + R_{LDR}} \right) \quad (4.2)$$

El valor de tensión de salida analógico (V_{OUT}) se convertirá a digital en el ADC. En la unidad central de control se usan las ecuaciones 5.1 y 5.2 para obtener el valor de luminosidad.

El rango de luminosidad que el sistema puede manejar depende del valor de la resistencia R_1 . Como se desprende de la tabla 4.3 los niveles de iluminación varían mucho

del exterior a los interiores. En este caso se ha optado por emplear una resistencia R_1 (de valor 15K) que aporte información de interés en el interior de una vivienda. El nivel de iluminación será fiable entre valores comprendidos entre 1 y 2500 lux.

4.1.1.4. Relés

Los relés funcionan como un interruptor controlado por un circuito eléctrico en el que, a través de un electroimán, se acciona un contacto que permite abrir o cerrar circuitos eléctricos independientes. El electroimán está formado por una barra de hierro y por una bobina de hilo de cobre alrededor. Al pasar la corriente eléctrica por la bobina, la barra de hierro se magnetiza convirtiéndose en un imán que provoca que los dos contactos del relé se toquen y permitan el paso de la corriente. Al dejar de circular la corriente por la bobina desaparece el campo magnético y los dos contactos del relé dejan de tocarse. La corriente que circula por estos dos contactos puede ser mucho mayor que la corriente que necesita pasar por la bobina para activar el relé.

En la unidad remota desarrollada se van a utilizar relés del modelo OMRON G5V-2-5 VDC. Según su datasheet, se trata de un relé de DPDT (Doble Polo Doble Tiro) que tiene una tensión nominal de 5 Voltios y una corriente de excitación de 100mA. De tal manera que para activar el relé, se necesita una tensión de 5 voltios o superior y que circule por la bobina una intensidad de alrededor de 100 mA.

A la hora de analizar qué equipos se pueden conectar a estos relés, hay que tener en cuenta que los relés admiten una corriente de conmutación máxima de 0.6 amperios en alterna y de 2 amperios en continua. La tensión máxima de conmutación es de 125 VAC y de 30 VDC.

Para conectar el relé a un circuito electrónico digital (es decir, a las salidas digitales de los módulos Xbee), se necesita crear un circuito de acondicionamiento como el que se observa en la figura 4.5, con componentes auxiliares como resistencias, diodos o transistores. En este caso, además de las resistencias necesarias, se emplearán diodos rectificadores del tipo 1N4148W y los MOSFET 2N7002.

Los diodos 1N4148 empleados tienen un voltaje inverso de 75 Voltios y una capacidad máxima de corriente continua de 150mA, por lo que funcionarán correctamente como diodos de protección del relé anteriormente explicado. Se ha optado por usar el encapsulado SOD-123 para estos diodos.

Un MOSFET es un transistor de efecto de campo metal-óxido-semiconductor que permite activar, de manera efectiva y sencilla, un relé a partir de un circuito de control. El usar un MOSFET da la opción de emplear relés de voltajes diferentes a la tensión del circuito de control. Además, estos componentes requieren intensidades de corriente insignificantes en su terminal Gate por lo que cualquier circuito de control puede proveer esa corriente. El MOSFET 2N7002 tiene una corriente de drenaje máxima de 115mA por lo que también es válido para trabajar con los relés G5V-2-5 VDC (corriente de excitación de 100mA). Se usa el encapsulado SOT-23.

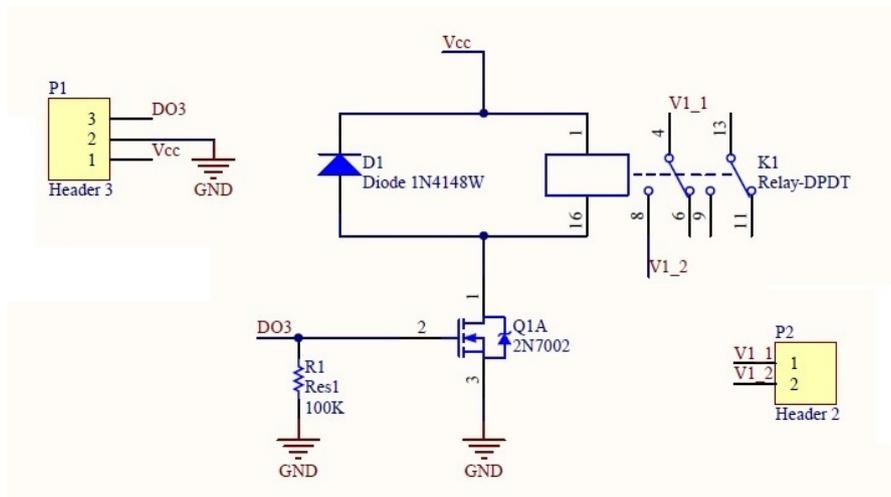


Figura 4.5: Esquemático del circuito de acondicionamiento para los relés

4.1.1.5. Regulador de tensión

Los componentes de la unidad remota tienen diferentes tensiones de alimentación. Los módulos Xbee o los sensores analógicos utilizan un voltaje de alimentación de 3.3 voltios, mientras que los relés necesitan una tensión nominal de 5 o más voltios. Por lo tanto, se ha tenido que hacer uso de un regulador de tensión con objeto de conseguir una tensión de salida regulada de 3.3 voltios a partir de un voltaje de entrada superior a 5 voltios, de modo que también sirva para alimentar los relés. De esta manera se consigue el objetivo de alimentar todos los componentes de la unidad periférica sin tener que usar dos reguladores de tensión.

| | LM117H | TC1262-3.3 | uA78M33C |
|--------------------------|------------------|--------------------|-------------------|
| Fabricante | Texas Intr. | Microchip | Texas Intr. |
| Precio por unidad (€) | 4.99 | 0.52 | 0.77 |
| Encapsulados | TO-39 | TO-220, SOT-223 | TO-220, TO-252 |
| Tensión de salida (V) | 1.2-37 | 3.3 | 3.3 |
| Tipo | Lineal ajustable | Lineal fijo | Lineal fijo |
| Low dropout (LDO) | No | Si | No |
| Temp. de trabajo (°C) | -55-150 | -40-125 | 0-125 |
| Corriente de salida (mA) | 500 | 500 | 500 |
| Tensión de entrada (V) | 4.2-40 | 3.95-6 | 5.3-25 |

Tabla 4.4: Reguladores de tensión candidatos a formar parte de la electrónica

En función de estos requisitos, se optó por emplear el modelo uA78M33C debido a las características eléctricas y mecánicas que presenta. Este regulador ofrece una corriente máxima de salida bastante aceptable (0.5mA), lo cual es más que válido para las cargas que se van a conectar, ya que prácticamente todo el consumo de corriente entre los componentes que se alimentan a 3.3 voltios proviene del modulo Xbee, el cual tendrá un consumo de corriente máximo de unos 215 mA. Este consumo máximo se dará en los momentos de transmisión de mensajes por radiofrecuencia.

Por otro lado, el regulador uA78M33C admite un gran abanico de niveles de tensión superiores a 5 voltios, lo que permite que la unidad remota sea alimentada mediante varias posibles fuentes electrónicas cableadas: 5, 6, 7, 9 VDC e incluso 12 VDC.

La ecuación de la potencia máxima de disipación a temperatura ambiente viene dada por la siguiente expresión:

$$P_{D_{MAX}} = \frac{(Tj_{MAX} - T_{AMB})}{\theta_{JA}}$$

Este regulador, haciendo uso de un encapsulado TO-252, presenta unos valores Tj_{MAX} de 150°C y θ_{JA} de 30.3 °C/W, por lo que disipa una potencia máxima de 4.125 wattios a temperatura ambiente. Según la ecuación 5.3, para el hipotético caso de que la corriente de salida fuese la máxima de 0.5 amperios se obtiene que el máximo voltaje de entrada sería 11.55 voltios.

$$P_D = (V_{IN_{MAX}} - V_{OUT}) I_{LOAD_{MAX}} \quad (4.3)$$

Cuanto mayores son los voltajes de entrada menos eficiencia energética, ya que el regulador disipa más potencia en forma de calor, llegando al punto de que si se desea alimentar la unidad remota con tensiones superiores a 12 voltios habría que añadir un disipador de calor externo. Se recomienda alimentar la unidad con fuentes de alimentación de 5 voltios.

En la figura 4.6 se puede observar el circuito de acondicionamiento que necesita este modelo de regulador según sus especificaciones.

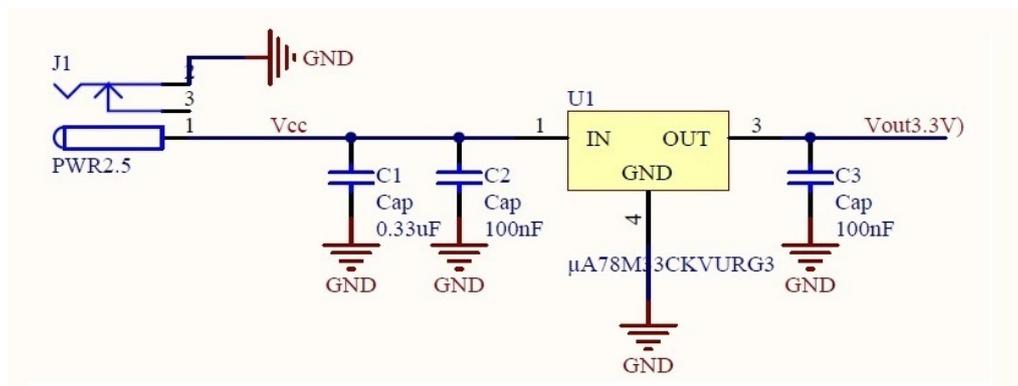


Figura 4.6: Esquemático del circuito de alimentación y regulación

4.1.2. Construcción del hardware

Todos los circuitos impresos realizados durante este proyecto fueron diseñados en el programa Altium Designer y fueron construidos en el taller de circuitos impresos de la Escuela Politécnica Superior (UAM) a partir de los archivos de diseño generados en Altium Designer.

En la figura 4.7 se puede apreciar una imagen de la unidad remota fabricada, una vez soldados todos sus componentes. Como se puede observar, en lugar de un único circuito impreso que englobe todos los componentes, existen tres PCBs diferentes y separados, los cuales están montados sobre una misma base de plástico. Con esta solución se ha pretendido aportar versatilidad a la unidad remota, de tal modo que al dividirla en tres áreas funcionales sea posible, por ejemplo, cambiar los tipos de sensores y seguir aprovechando los otros dos PCBs. A continuación se muestra el resultado del diseño y construcción de los tres circuitos impresos: PCB del módulo Xbee, PCB de los sensores y PCB de los actuadores.

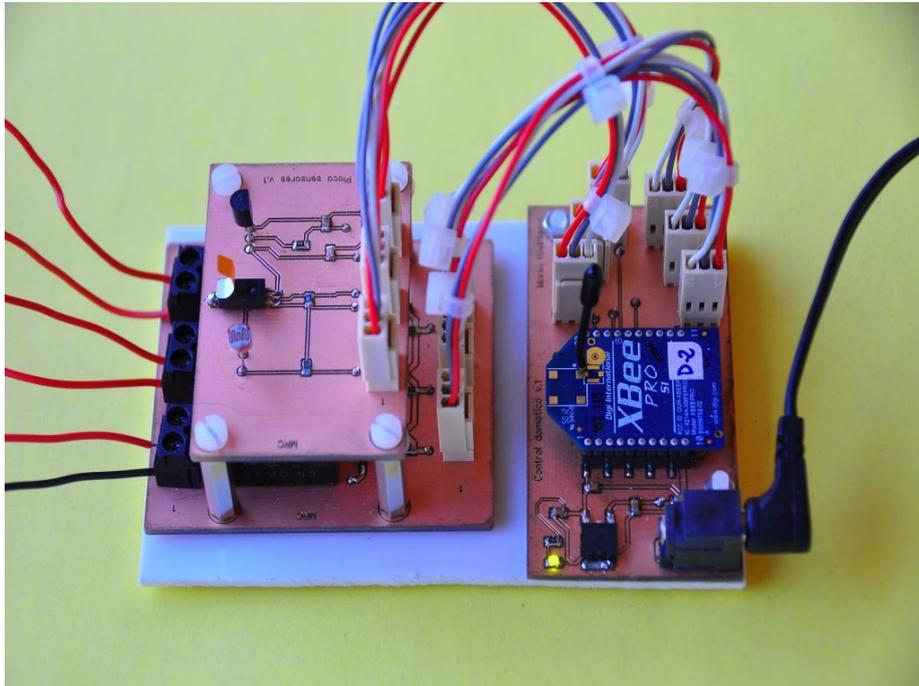


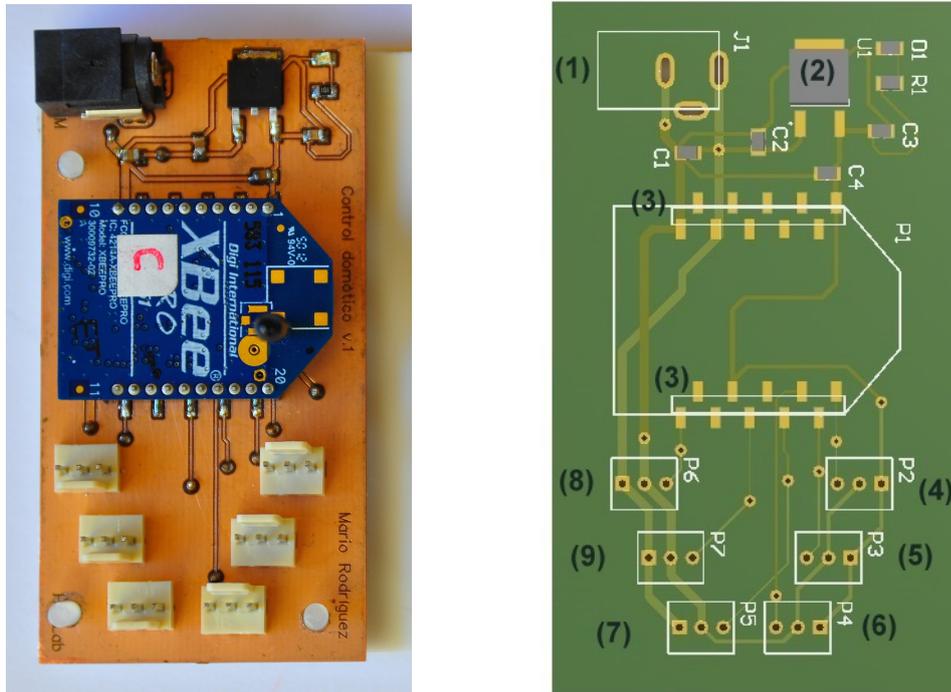
Figura 4.7: Unidad remota construida

4.1.2.1. PCB del módulo Xbee

Este circuito impreso servirá de base para conectar el modulo Xbee y se encargará de extraer la alimentación a través del conector Jack, convertirla a la tensión adecuada y conducirla a los diferentes integrados y conectores.

El diseño para este PCB contiene: el conector Jack, los conectores header de 10 pines que permiten conectar directamente el módulo Xbee al PCB ¹, un diodo LED, el resto de conectores, el regulador lineal de tensión y el resto de componentes pasivos que acondicionan estos dispositivos. En la figura 4.8 se puede observar el diseño del circuito impreso y en el anexo F el esquemático completo.

¹Los módulos Xbee tienen una separación especial entre pines de 2 mm, por lo que los conectores y los footprints diseñados deben adaptarse a estas especificaciones.



(a) Imagen del circuito impreso construido (b) Esquema del PCB diseñado

Figura 4.8: PCB del Módulo Xbee

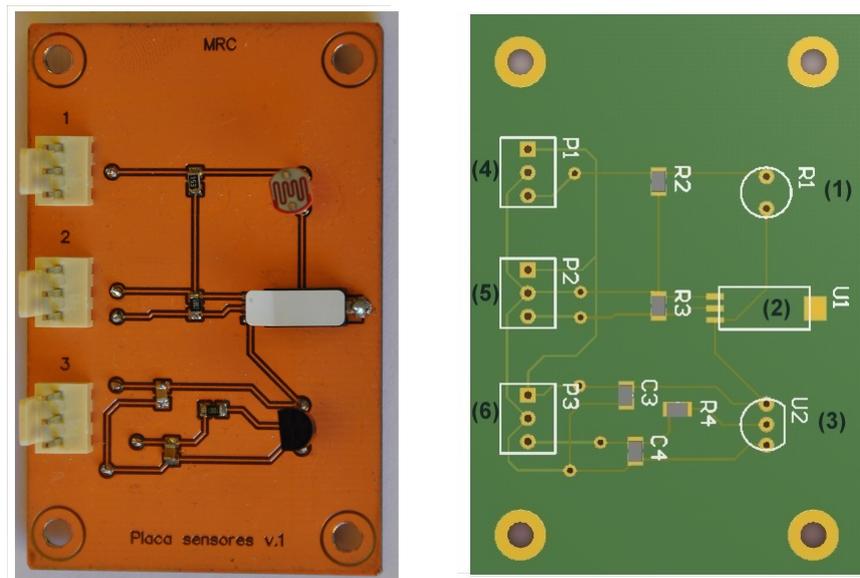
A continuación se muestra la localización de los componentes más destacables (ver figura 4.8):

- (1) Conector DC Jack hembra para circuito impreso: Las fuentes electrónicas cableadas alimentan la placa a través de este conector.
- (2) Regulador lineal de tensión.
- (3) Conectores para el módulo Xbee: Dos sockets SMD de 10 pines cada uno, que permiten que el módulo XBee proporcione sus salidas y entradas:
 - ✧ Conector superior. Pines utilizados, numerado en orden de derecha a izquierda:
 - Pin 1 / VCC: Este pin proporciona la alimentación de 3.3V al módulo XBee.
 - Pin 10 / GND: Tierra.

- ✧ Conector inferior. Pines utilizados, numerados de izquierda a derecha:
 - ❑ Pin 11 / DIO4: Pin que proporciona la salida digital para activar/-desactivar el actuador número 2.
 - ❑ Pin 14 / VREF: Pin de entrada, conectado a 3.3V, que establece la referencia máxima para realizar las conversiones analógicas-digitales.
 - ❑ Pin 15 / DIO5: Pin que proporciona la salida digital para activar/-desactivar el actuador número 3.
 - ❑ Pin 17 / DIO3: Pin que proporciona la salida digital para activar/-desactivar el actuador número 1.
 - ❑ Pin 18 / AD2: Pin por el que entra el valor de tensión analógico del sensor de temperatura.
 - ❑ Pin 19 / AD1: Pin por el que entra el valor de tensión analógico del sensor de humedad.
 - ❑ Pin 20 / AD0: Pin por el que entra el valor de tensión analógico del sensor de luminosidad.
- (4),(5) y (6) Conectores para los sensores de luminosidad, humedad y temperatura respectivamente. Pines de derecha a izquierda:
 - ✧ Pin 1: Este pin proporciona la alimentación de 3.3V a los sensores.
 - ✧ Pin 2: Tierra.
 - ✧ Pin 3: Este pin obtiene la señal con el nivel de tensión de salida de los sensores.
- (7),(8) y (9) Conectores para los actuadores 1,2 y 3 respectivamente. Pines de izquierda a derecha:
 - ✧ Pin 1: Este pin proporciona la tensión de alimentación a los relés. Voltaje del valor de la fuente de alimentación conectada al DC Jack.
 - ✧ Pin 2: Tierra.
 - ✧ Pin 3: Este pin proporciona la señal de control digital para activar/desactivar los actuadores.

4.1.2.2. PCB de los sensores

Este PCB contiene los tres sensores analógicos que se han mencionado y sus respectivos circuitos de acondicionamiento. Además incorpora 3 conectores que permiten alimentar los sensores y conectarlos con los ADC del PCB del Modulo Xbee. En la figura 4.9 se puede observar el diseño del circuito impreso y en el anexo F el esquemático completo.



(a) Imagen del PCB construido (b) Esquema del PCB diseñado

Figura 4.9: PCB de los sensores

A continuación se muestra la localización de los componentes más destacables (ver figura 4.9):

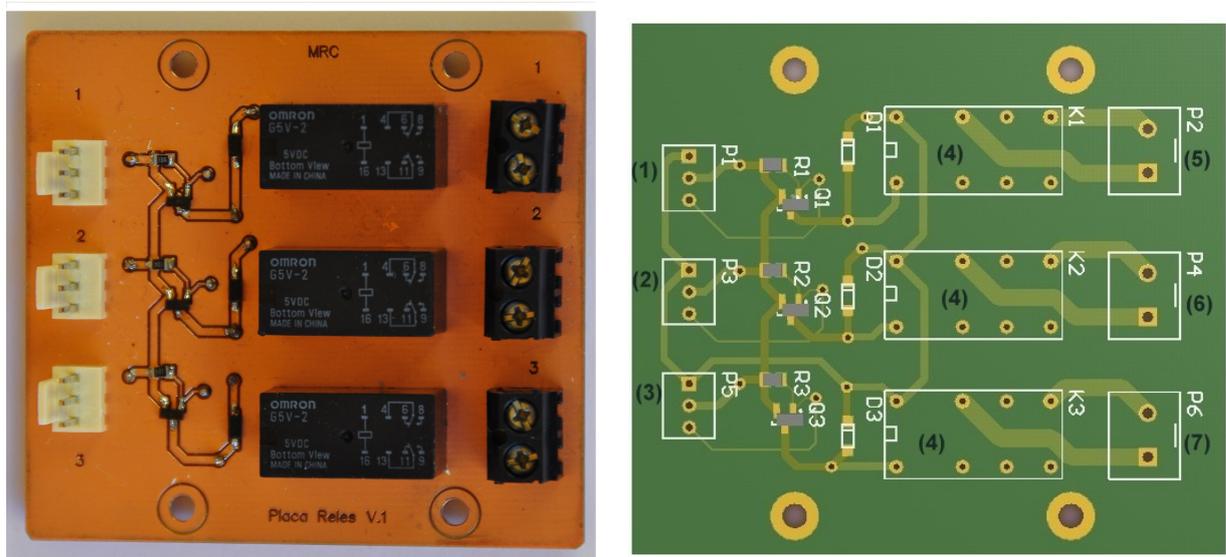
- (1) Sensor de luminosidad: Fotorresistencia LDR.
- (2) Sensor de humedad: HIH-5030-001
- (3) Sensor de temperatura: AD22103
- (4),(5) y (6) Conectores para los sensores de luminosidad, humedad y temperatura respectivamente. Pines numerados en orden descendiente desde el superior:
 - ◇ Pin 1: Este pin proporciona la alimentación de 3.3V a los sensores.

- ✧ Pin 2: Tierra.
- ✧ Pin 3: Este pin obtiene la señal con el nivel de tensión de salida de cada sensor.

4.1.2.3. PCB de los actuadores

El diseño de este circuito impreso se compone de los tres relés y del resto de componentes que forman los circuitos de acondicionamiento. También incluye seis conectores, tres para alimentarlos y conectarlos a los pines digitales del PCB del Módulo Xbee y los otros tres para permitir la conexión de los equipos que se desea que sean activados o desactivados. En la figura 4.10 se puede observar el diseño del circuito impreso y en el anexo F el esquemático completo.

A la hora de realizar el diseño de este PCB, al igual que en los otros dos circuitos impresos, se diseñaron pistas de diferentes anchuras en función la intensidad de corriente máxima que puede circular por ellas. En este PCB en concreto, las pistas que forman el circuito de potencia (al que se conectan los equipos a activar o desactivar) tienen un ancho especial de 1mm de manera que no limitan la corriente máxima de conmutación que soporta el relé.



(a) Imagen del PCB construido

(b) Esquema del PCB diseñado

Figura 4.10: PCB de los actuadores

A continuación se muestra la localización de los componentes más destacables (ver figura 4.10):

- (1),(2) y (3) Conectores para los actuadores 1,2 y 3 respectivamente. Pines arriba a abajo:
 - ✧ Pin 1: Este pin proporciona la tensión de alimentación a los relés. Voltajes a partir de 5 voltios.
 - ✧ Pin 2: Tierra.
 - ✧ Pin 3: Este pin proporciona la señal de control digital para activar/desactivar cada actuador.
- (4) Relés: OMRON G5V-2-5 VDC.
- (5),(6) y (7) Conectores de tornillo de circuitos impresos y de 2 pines que enlazan la electrónica diseñada con los equipos a activar o desactivar.

4.2. Software

En esta unidad periférica la lógica de control reside en el módulo XBee. Gracias a la capacidad que tienen estos módulos (disponen de conversores, salidas digitales, etc.) y al hecho de utilizarlos bajo el modo API, se ha conseguido realizar el control de la unidad remota sin la necesidad de haber añadido microcontroladores ni haber tenido que desarrollar nuestro propio software.

En la figura 4.11 se puede observar el diagrama de flujo de la unidad remota, concretamente del módulo XBee. Una vez que dicho módulo recibe un paquete de radiofrecuencia, lo procesa y actúa en consecuencia en función de su contenido. Dependiendo del tipo de paquete la actuación puede consistir en leer los valores de los pines de entrada (conversiones analógico-digitales), en cambiar un parámetro de configuración o en actualizar los cambios en la memoria. Finalmente, genera el paquete de respuesta correspondiente que es transmitido al módulo que le envió el mensaje inicial, es decir, a la unidad central.

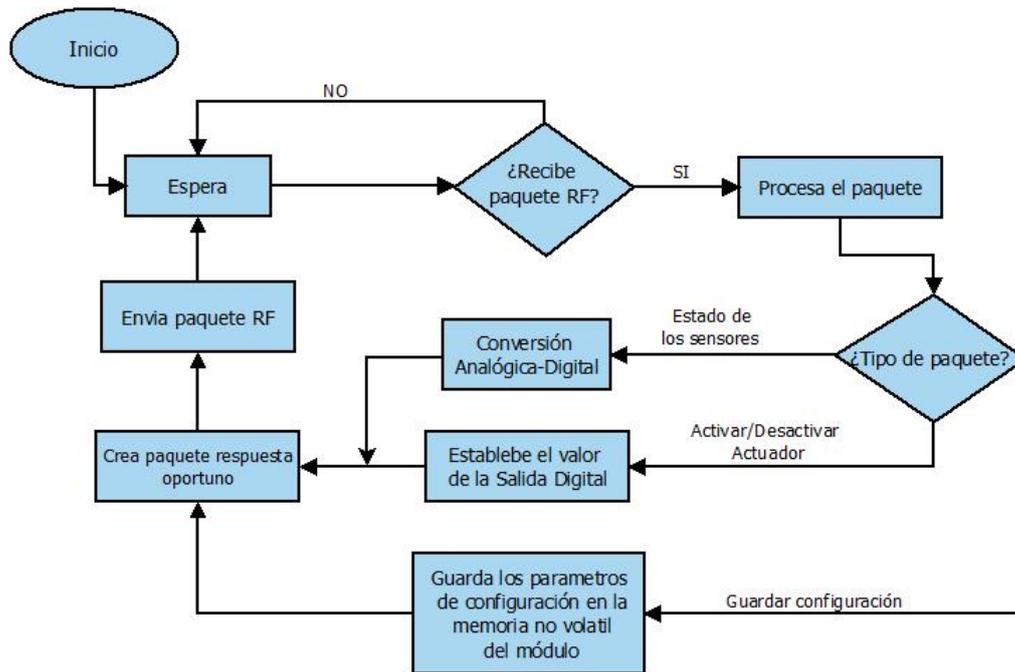


Figura 4.11: Diagrama de flujo de la unidad remota

Capítulo 5

Unidad central de control

A la hora de elegir un tipo arquitectura para un sistema domótico, no hay ninguna opción que sea la mejor para todos los casos, sino que dependerá de aspectos como las prestaciones que va a cubrir el sistema, la topología de red que se va a implementar, el medio de transmisión que se va a utilizar, etc. Como ya se ha mencionado con anterioridad el sistema va a tener una arquitectura centralizada, es decir, desde una única unidad central se realizará el control de todos los dispositivos remotos de la red.

La arquitectura centralizada es la que tiene unos menores costes económicos, ya que no se requiere de dispositivos inteligentes repartidos por la vivienda, los cuales son más costosos. Además es más sencilla a la hora de diseñar e implementar la red, por ejemplo la programación se realiza en único dispositivo. Emplear otro tipo de arquitectura supondría aumentar la flexibilidad del sistema pero a costa de aumentar su complejidad y los costes de manera innecesaria para las prestaciones que ofrece. El inconveniente de tener una arquitectura centralizada es que se dependerá completamente de la unidad central, lo cual implica que cualquier fallo en dicha unidad provocará la caída de todo el sistema.

Otros motivos que justifican este tipo de arquitectura son: el empleo de la radiofrecuencia como medio de transmisión y la elección de una topología de red en estrella, la cual se adapta perfectamente a una arquitectura de tipo centralizada.

Como también se ha comentado, el sistema domótico está abierto a que puedan ser introducidas unidades remotas con microcontrolador, en cuyo caso el sistema pasaría

a tener una arquitectura mixta entre un sistema centralizado y descentralizado, en el que la unidad central controlaría al resto de microcontroladores repartidos por la red. Además, tendría a su cargo los sensores y actuadores que no dependieran de ninguna unidad remota con microcontrolador.

La unidad central de control va a estar formada por un módulo de radiofrecuencia y por un dispositivo con capacidad computacional que pueda desempeñar el control de la red (ver figura 5.1). El módulo de radiofrecuencia empleado va a ser el mismo que en el resto del sistema, es decir, el módulo XBee que se usa en las unidades remotas. Durante este capítulo no se profundizará demasiado en el hardware y software del módulo ya que éste ha sido detallado en la sección 3.1. Únicamente se verá la manera en la que se conecta con el dispositivo central de control.

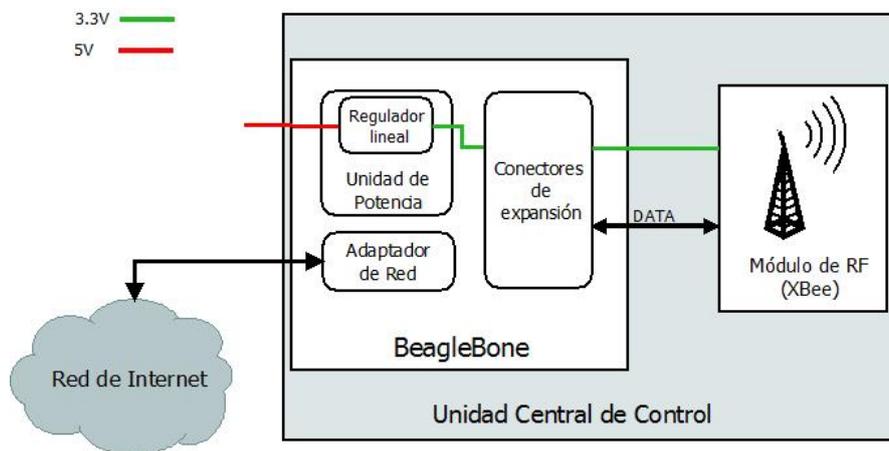


Figura 5.1: Esquema de la unidad central de control

El dispositivo central de control tiene como objetivos crear y enviar las instrucciones oportunas a las unidades remotas y procesar e interpretar las respuestas de éstas. El dispositivo deberá ser programado para llevar a cabo esta labor, lo que requerirá de una alta capacidad computacional. Además, deberá incluir una aplicación que sirva de interfaz con el usuario y que será accesible mediante un servidor web. Finalmente tendrá que disponer de las herramientas necesarias para poder conectarse a Internet, al módulo de radiofrecuencia, a los sensores externos que se puedan añadir, etc.

Hay varios tipos de equipos que podrían servir como dispositivo de control de una red domótica como la implementada, desde chips microcontroladores a ordenadores pasando por plataformas embebidas. En este caso se optó por utilizar la última de

éstas, ya que a diferencia de los microcontroladores, vienen con el hardware auxiliar necesario, y a diferencia de los ordenadores, aportan mayor flexibilidad al poder ser localizadas en cualquier lugar de la casa. Además presentan un consumo menor que los ordenadores y unas capacidades más que suficientes para las prestaciones que deben ofrecer.

Por sistema embebido o empotrado se denomina a un sistema de computación diseñado y empleado para cubrir una serie de necesidades específicas. La parte fundamental de estos sistemas es la placa base (plataforma o tarjeta embebida) que contiene el microprocesador, las memorias, los puertos de comunicación (conectores físicos), otros chips, etc.

Existen varias alternativas de plataformas embebidas, desde FOX Board G20 a IGEP-Board, aunque las tres más populares son: Arduino, Raspberry Pi y BeagleBone (BeagleBone Black es una versión posterior muy similar a la anterior).

Arduino es más adecuada para proyectos simples (aunque hay versiones con diferentes capacidades computacionales). Raspberry Pi es idónea para proyectos multimedia o en general proyectos complejos basados en Linux, pero está limitada para interactuar con sensores u otros dispositivos externos en general. BeagleBone es una opción adecuada para proyectos que son complicados para Arduino pero que no necesitan de complejos gráficos como Raspberry Pi. La plataforma de desarrollo BeagleBone permite una fácil conexión a internet y es idónea para emplearla en proyectos con sensores externos y redes.

Finalmente se ha decidido usar la tarjeta embebida BeagleBone ya que dispone de una gran capacidad computacional y presenta muchas posibilidades de comunicación con el exterior (incluido puerto de Ethernet). Además de su bajo coste, el laboratorio HCTLab ya disponía de varias tarjetas BeagleBone.

5.1. Hardware

BeagleBone es una tarjeta embebida, de proporciones similares a las de una tarjeta de crédito (9 cm x 5 cm), manufacturada por CircuitCo a partir de componentes fabricados por Texas Instruments. La plataforma BeagleBone se podría definir, por sus funcionalidades, como un ordenador en pequeño. Como se detalla más adelante,

dispone de un sistema operativo Linux embebido (aunque puede utilizar otros) que permite tener la funcionalidad de un ordenador. Cualquier tarjeta BeagleBone se basa en un microprocesador AM335X 720MHz ARM Cortex-A8.

Algunas de sus características fundamentales son:

- 256MBytes de memoria RAM DDR2.
- Acelerador Gráfico 3D.
- Procesador ARM Cortex-M3 para la administración de la energía.
- Arquitectura computacional RISC de 32 bits en la CPU.
- Tarjeta microSD de 4GB que incorpora el sistema operativo.
- Conector Ethernet, conectividad USB, 92 pines en dos header para completar el resto de comunicaciones, etc.

El microprocesador de bajo coste AM3359, basado en el procesador ARM Cortex-A8, es usado en un rango muy amplio de equipos industriales. Cuenta con una unidad programable en tiempo real que fue diseñado para ser capaz de implementar tecnologías de comunicación en tiempo real. En definitiva, es un procesador eficiente que permite una amplia gama de aplicaciones.

En la figura 5.2 se puede observar el diagrama de bloques de la plataforma de desarrollo Beaglebone¹. A continuación se describen las características y los principales elementos del diseño de la tarjeta BeagleBone (ver figura 5.3):

- Procesador: Utiliza un microprocesador AM3359, cuya velocidad real vendrá determinada por como sea la conexión y la manera de su alimentación.
- Memoria: Usa una memoria DDR2 RAM (Double Data Rate type two RAM) de 16 bits, el modelo de tarjeta utilizado trae la configuración de 256MB a 400MHz. Además contiene una EPROM de 32 KBytes con información de la placa BeagleBone.

¹Las imágenes, las tablas y el resto de la información relacionada con el hardware han sido obtenidas del manual técnico de la BeagleBone [20].

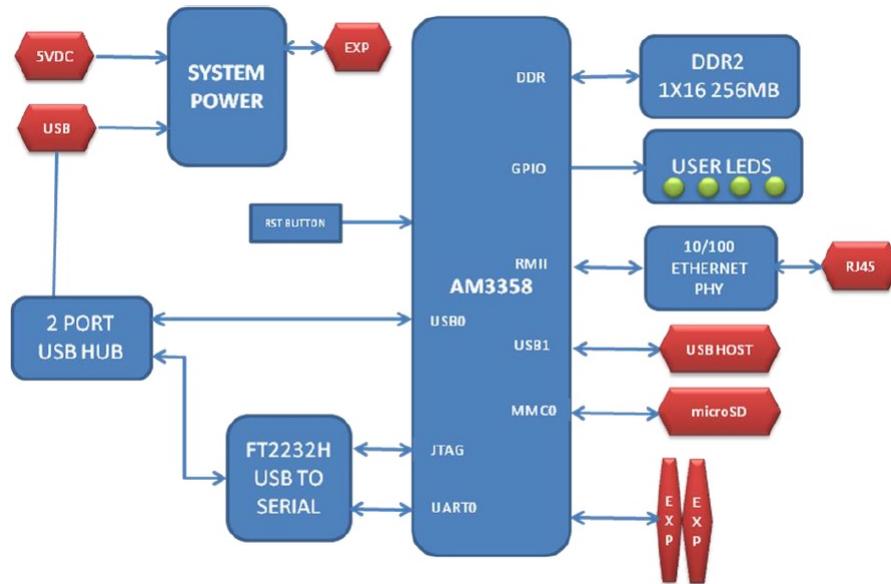


Figura 5.2: Diagrama de bloques del sistema

- Control de la alimentación: El circuito integrado TPS65217 está diseñado para dar soporte a los procesadores AM335X. Se usa junto con un regulador LDO externo para alimentar todo el sistema.
- Interfaz USB: La tarjeta BeagleBone tiene un USB HUB, que permite concentrar los dos puertos USB para operar como:
 - ✧ Comunicación serie por puerto USB.
 - ✧ JTAG por puerto USB.
 - ✧ Acceso directo al procesador por el puerto USB (puerto USB0).
- Conector MicroSD.
- Puerto USB1: La tarjeta dispone de un conector USB de tipo A que conecta con USB1 en el procesador. El puerto puede proporcionar alimentación de encendido (5 voltios) y hasta 500 mA de corriente dependiendo de cómo este alimentada la tarjeta BeagleBone.
- Puerto USB Cliente: Proporciona acceso a USB0 a través del USB HUB. Se mostrará en un PC como un dispositivo estándar.

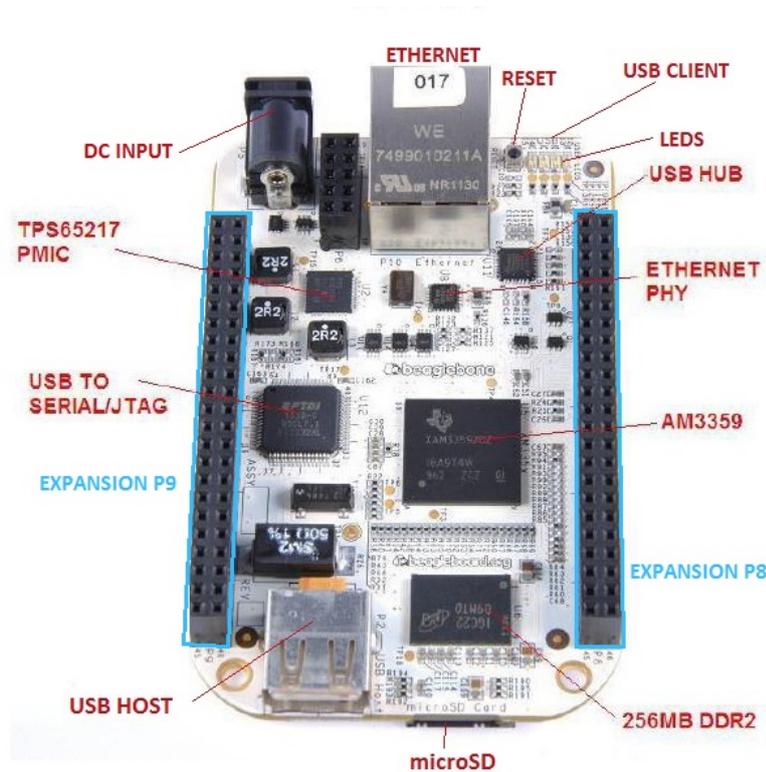


Figura 5.3: Componentes y conectores de la tarjeta BeagleBone

- Alimentación: La plataforma BeagleBone puede ser alimentada a través de un puerto USB de un ordenador o desde una fuente de alimentación externa de 5 VDC. Si se alimenta desde el USB la velocidad del procesador está limitada a 500 MHz, en cambio si es a partir de una fuente de alimentación el procesador opera a 720 MHz.

En el proyecto se ha trabajado siempre alimentando la tarjeta BeagleBone con una fuente electrónica cableada de alimentación positiva con valor nominal a 5 VDC y 1 amperio.

- Botón Reset.
- Indicadores: La tarjeta dispone de cinco LEDs, cuatro de ellos pueden ser controlados por el usuario, mientras que el otro indica si está siendo alimentada.
- Interfaces de Expansión: BeagleBone dispone de 2 conectores, de 46 pines cada uno, para el acceso a las señales de expansión. A continuación se enumeran

las principales funcionalidades que ofrecen estos conectores de expansión y los periféricos adicionales que se pueden conectar.

- ✧ LCD. Existe la posibilidad de conectar a la BeagleBone una pantalla LCD de 24 bits o de 16 bits (la cual utiliza menos pines de los conectores de expansión y por tanto deja libres más pines para ser utilizados por otras tarjetas de expansión). La pantalla tiene luz de fondo y funciones de pantalla táctil. La potencia máxima para la luz de fondo es de 25mA, por lo que puede no ser suficiente para paneles mayores.
- ✧ GPMC. Los puertos de expansión disponen de líneas GPMC.
- ✧ MMC1. Permite acceder a una tarjeta MMC (MultiMediaCard) localizada en el conector microSD.
- ✧ SPI. En los conectores de expansión se dispone de 2 puertos SPI (Serial Peripheral Interface) disponibles (SPI0 y SPI1).
- ✧ Puerto Serial. Hay 4 puertos serie en los conectores de expansión (UART1, UART2, UART4 y UART5). Los tres primeros tienen señales Tx,Rx,RTS y CTS mientras que el puerto UART5 sólo dispone de señales Tx y Rx.
- ✧ Conversores Analógico-Digital. Existen siete ADC con un voltaje de conversión máximo de 1.8 voltios.
- ✧ GPIO. Los puertos de expansión disponen de hasta 66 pines genéricos GPIO (General Purpose Input/Output). Estos pines son de 3.3 voltios. La gran cantidad de pines de este tipo es una de las características de la tarjeta BeagleBone.
- ✧ CAN Bus. Existen dos interfaces CAN bus.
- ✧ TIMERS. Los conectores de expansión tienen cuatro temporizadores de salida.
- ✧ PWM. Se disponen de ocho salidas PWM (Pulse-Width Modulation). Seis de ellas de alta resolución y dos eCAP.

5.1.1. Puertos y conectores de expansión

Como ha quedado de manifiesto en las características anteriores, la tarjeta BeagleBone posee varias formas de comunicarse con el exterior. Todos estos protocolos de

comunicación que soporta, la permiten comunicarse con equipos muy variados y de diferentes ámbitos.

En la figura 5.3 se pueden observar los conectores de los que dispone para comunicarse: un conector RJ45 para Ethernet, un conector USB tipo A, un conector USB tipo B y un conector de expansión de diez pines. El resto de las opciones de comunicación se encuentran en los dos conectores de expansión (P8 y P9), los cuales disponen de 46 pines cada uno ². Sin embargo, estos pines no están dedicados a una tarea exclusiva (ni siquiera está fijado que sean de entrada o salida), sino que cada uno de ellos soporta siete modos de operación distintos y deberán ser configurados por software para que actúen en el modo deseado.

| SIGNAL NAME | PROC | CONN | CONN | PROC | SIGNAL NAME |
|-------------|------|------|------|------|-------------|
| | GND | 1 | 2 | GND | |
| GPIO1_6 | R9 | 3 | 4 | T9 | GPIO1_7 |
| GPIO1_2 | R8 | 5 | 6 | T8 | GPIO1_3 |
| TIMER4 | R7 | 7 | 8 | T7 | TIMER7 |
| TIMER5 | T6 | 9 | 10 | U6 | TIMER6 |
| GPIO1_13 | R12 | 11 | 12 | T12 | GPIO1_12 |
| EHRPWM2B | T10 | 13 | 14 | T11 | GPIO0_26 |
| GPIO1_15 | U13 | 15 | 16 | V13 | GPIO1_14 |
| GPIO0_27 | U12 | 17 | 18 | V12 | GPIO2_1 |
| EHRPWM2A | U10 | 19 | 20 | V9 | GPIO1_31 |
| GPIO1_30 | U9 | 21 | 22 | V8 | GPIO1_5 |
| GPIO1_4 | U8 | 23 | 24 | V7 | GPIO1_1 |
| GPIO1_0 | U7 | 25 | 26 | V6 | GPIO1_29 |
| GPIO2_22 | U5 | 27 | 28 | V5 | GPIO2_24 |
| GPIO2_23 | R5 | 29 | 30 | R6 | GPIO2_25 |
| UART5_CTSN | V4 | 31 | 32 | T5 | UART5_RTSN |
| UART4_RTSN | V3 | 33 | 34 | U4 | UART3_RTSN |
| UART4_CTSN | V2 | 35 | 36 | U3 | UART3_CTSN |
| UART5_TXD | U1 | 37 | 38 | U2 | UART5_RXD |
| GPIO2_12 | T3 | 39 | 40 | T4 | GPIO2_13 |
| GPIO2_10 | T1 | 41 | 42 | T2 | GPIO2_11 |
| GPIO2_8 | R3 | 43 | 44 | R4 | GPIO2_9 |
| GPIO2_6 | R1 | 45 | 46 | R2 | GPIO2_7 |

Tabla 5.1: Pines del conector P8

²Los pines de los conectores P8 y P9 están numerados en orden creciente de izquierda a derecha y de arriba a abajo, según la figura 5.3.

Las tablas con la descripción de los diferentes modos de configuración de cada pin se mostrarán en la sección de software. En las tablas 5.1 y 5.2 únicamente se puede observar los pines que han sido utilizados, así como los nombres que tienen asignados por el fabricante.

| SIGNAL NAME | PIN | CONN | PIN | SIGNAL NAME |
|-------------|------------|------|-----|---------------|
| | GND | 1 | 2 | GND |
| | VDD_3V3EXP | 3 | 4 | VDD_3V3EXP |
| | VDD_5V | 5 | 6 | VDD_5V |
| | SYS_5V | 7 | 8 | SYS_5V |
| PWR_BUTTON* | | 9 | 10 | A10 |
| UART4_RXD | T17 | 11 | 12 | U18 |
| UART4_TXD | U17 | 13 | 14 | U14 |
| GPIO1_16 | R13 | 15 | 16 | T14 |
| I2C1_SCL | A16 | 17 | 18 | B16 |
| I2C2_SCL | D17 | 19 | 20 | D18 |
| UART2_TXD | B17 | 21 | 22 | A17 |
| GPIO1_17 | V14 | 23 | 24 | D15 |
| GPIO3_21 | A14 | 25 | 26 | D16 |
| GPIO3_19 | C13 | 27 | 28 | C12 |
| SPI1_D0 | B13 | 29 | 30 | D12 |
| SPI1_SCLK | A13 | 31 | 32 | VDD_ADC(1.8V) |
| AIN4 | C8 | 33 | 34 | GND_ADC |
| AIN6 | A5 | 35 | 36 | A5 |
| AIN2 | B7 | 37 | 38 | A7 |
| AIN0 | B6 | 39 | 40 | C7 |
| CLKOUT2 | D14 | 41 | 42 | C18 |
| | GND | 43 | 44 | GND |
| | GND | 45 | 46 | GND |

Tabla 5.2: Pines del conector P9

5.1.2. Placa adaptadora

En esta sección se detalla la manera en la que se conectan los dos dispositivos de la unidad central de control (BeagleBone y módulo XBee). Como se vio en la sección 3.1.4.2.1, los módulos Xbee solo disponen de un puerto UART para comunicarse con otros dispositivos, por lo que se tendrá que hacer uso de éste para comunicar la plataforma BeagleBone con el módulo Xbee.

Para llevar a cabo esta conexión se ha diseñado y fabricado un pequeño circuito impreso que conecta, a través de pistas y vías, los pines de los conectores de expansión de la BeagleBone con los pines correspondientes del módulo Xbee.

La idea es que la tarjeta BeagleBone vaya conectada en la parte inferior de este PCB y que el módulo Xbee se conecte en la parte superior del circuito impreso (ver figura 5.4). Para realizar el diseño se tuvo que tener en cuenta las dimensiones exactas de la BeagleBone y sus componentes, de tal modo que las dos tiras de pines verticales macho (46 pines, 2x23 y de paso 2.54mm) incorporadas al circuito impreso, encajasen perfectamente en los conectores de expansión P8 y P9 de la plataforma BeagleBone. Al mismo tiempo (y al igual que para diseñar el PCB del módulo Xbee de la unidad remota), se tuvo que tener en cuenta las dimensiones específicas de los Xbee. Se utilizaron 2 sockets (de 10 pines con paso de 2mm) especialmente colocados donde poder encajar el módulo Xbee.

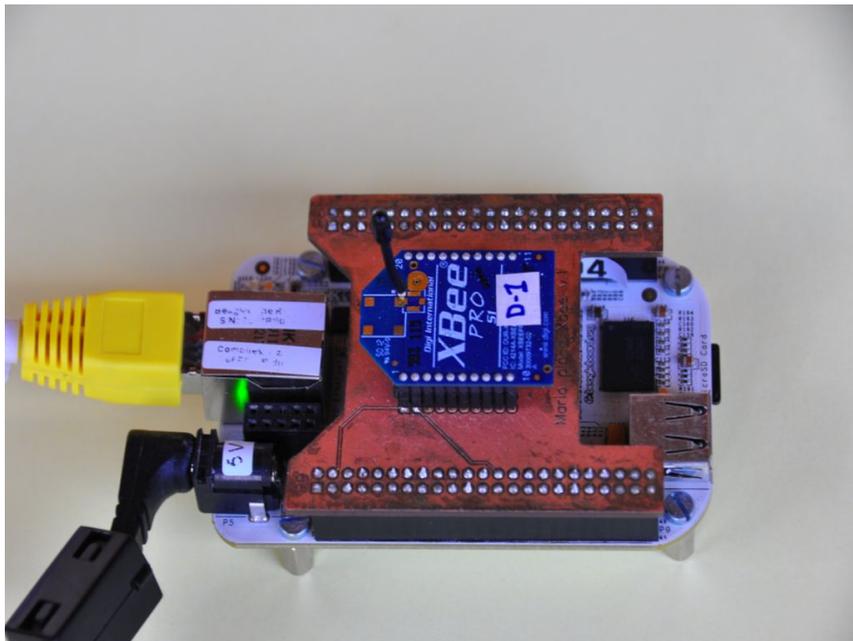


Figura 5.4: Placa adaptadora BeagleBone-XBee

En la tabla 5.3 se pueden observar qué pines de la BeagleBone y del Xbee han sido conectados entre sí. En primer lugar para poder alimentar el módulo Xbee, el cual necesitaba un voltaje de 3.3 voltios, se utiliza la propia plataforma BeagleBone, ya que uno de sus pines tiene la capacidad de suministrar dicha tensión. Este voltaje es generado gracias al circuito regulador de tensión que contiene, a partir de los 5Vcc

que la alimentan. Para realizar la comunicación serie se utilizan los pines 24 y 26 pertenecientes al puerto UART1, los cuales serán la entrada y salida de los datos que son transmitidos por la antena de radiofrecuencia. Finalmente se establecieron conexiones entre varios pines DIO del modulo Xbee y señales GPIO de la BeagleBone.

| Conexión | Pin BeagleBone | Pin Xbee |
|--------------|--------------------------|----------------|
| Alimentación | Pin4 (VDD_3V3EXP con.P9) | ⇔ Pin1 (VCC) |
| Tierra | Pin1 (GND con.P9) | ⇔ Pin10 (GND) |
| Serie | Pin24 (UART1_TXD) | ⇔ Pin3 (DIN) |
| Serie | Pin26 (UART1_RXD) | ⇔ Pin2 (DOUT) |
| Digital | Pin3 (GPIO1_6 con.P8) | ⇔ Pin20 (DIO0) |
| Digital | Pin5 (GPIO1_2 con.P8) | ⇔ Pin19 (DIO1) |
| Digital | Pin11 (GPIO1_13 con.P8) | ⇔ Pin16 (DIO6) |
| Digital | Pin15 (GPIO1_15 con.P8) | ⇔ Pin15 (DIO5) |
| Digital | Pin17 (GPIO1_27 con.P8) | ⇔ Pin12 (DIO7) |
| Digital | Pin21 (GPIO1_30 con.P8) | ⇔ Pin11 (DIO4) |

Tabla 5.3: Conexiones BeagleBone-XBee

5.2. Software

BeagleBone, al igual que otras plataformas embebidas, utiliza un sistema operativo que sirve de apoyo para la ejecución de sus programas. La mayoría de las veces se trata de sistemas operativos de tiempo real (RTOS) y que no usan mucha memoria.

Una de las características más valoradas de la plataforma BeagleBone es que puede trabajar con diferentes sistemas operativos. Por ejemplo, se podría instalar Android, que permitiría a los diseñadores de aplicaciones embebidas la capacidad de disponer de un SOP de alto nivel. Finalmente, se ha optado por emplear un sistema operativo Linux debido a las muchas posibilidades que dispone de elaborar programas de usuario que aprovechan los recursos de la tarjeta BeagleBone. Además, en internet había mayor cantidad de documentación sobre la plataforma de desarrollo BeagleBone trabajando con Linux que con otros sistemas operativos.

En la tarjeta microSD se encuentra la imagen con el sistema operativo. Otras imágenes con diferentes distribuciones (Ubuntu, Gentoo, Debian, Ångström Linux, etc) y los drivers necesarios para el ordenador pueden ser descargados desde la página

oficial (<http://beagleboard.org/bone>). Aunque la placa BeagleBone ya viene con una distribución Ångström Linux en la tarjeta microSD, se optó por usar una distribución Debian como sistema operativo ³.

A la hora de trabajar en la plataforma BeagleBone, ya sea para escribir programas y crear aplicaciones o para manejar sus recursos hardware, se realiza desde un ordenador que se conecta a ella. Esta conexión se puede realizar de dos maneras:

- En primer lugar, mediante un cable USB que conecta el puerto USB del ordenador y el puerto microUSB de la tarjeta BeagleBone. Esta opción hace que no sea necesario alimentar la tarjeta con una fuente externa, ya que ésta se alimenta a través del USB del ordenador. Únicamente en los primeros momentos del proyecto, mientras la plataforma de desarrollo BeagleBone utilizaba el sistema operativo Ångström que trae por defecto, se trabajó de esta manera.
- La segunda opción consiste en conectar, mediante un cable Ethernet, los conectores RJ45 del ordenador portátil y de la BeagleBone. Esta opción (conexión de red) hace que se tenga que alimentar la tarjeta BeagleBone con una fuente externa de 5 voltios, pero permite que el procesador trabaje al máximo de sus posibilidades. Además de esta manera se consigue tener la suficiente corriente de salida como para poder alimentar el módulo Xbee. Desde que se instaló la distribución Debian en la plataforma de desarrollo se trabajó siempre de esta forma.

Una vez conectados físicamente el ordenador y la plataforma BeagleBone, se deben fijar las condiciones para realizar la comunicación entre ellos, de tal manera que finalmente podamos acceder a los recursos de ésta y manejarla completamente desde el ordenador.

Para comunicar el ordenador con la BeagleBone se emplea el protocolo de comunicaciones cifrado SSH (Secure SHell) que permite controlar máquinas remotas a través de una red, en este caso acceder de forma remota a un sistema Linux. Para que se produzca una de estas conexiones SSH se necesita de un servidor SSH (BeagleBone) y de un cliente SSH (el ordenador). Si el ordenador trabaja con un sistema operativo Windows, para disponer de un cliente SSH se necesita descargar PuTTY.

³Para escribir la imagen en la microSD se empleó el programa Win32DiskImager.

Desde la terminal PuTTY (ver figura 5.5) se accede al servidor SSH a partir de la dirección IP (192.168.1.2) y a través del puerto por defecto para comunicaciones SSH (Puerto 22)⁴. Tras esto, ya se puede trabajar sobre la terminal Linux (PuTTY) de la plataforma de desarrollo BeagleBone desde el ordenador.

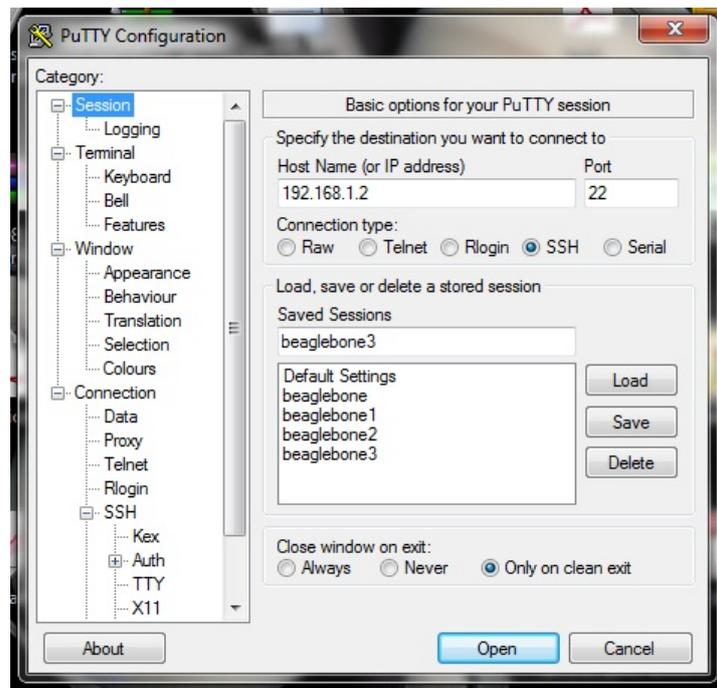


Figura 5.5: PuTTY

Para este proyecto se ha tenido que desarrollar el software necesario para que la plataforma BeagleBone sea capaz de realizar la misión que tiene encomendada y para que el usuario pueda interactuar con el sistema doméstico.

El software desarrollado se puede dividir en dos bloques. Por un lado, se tiene lo que se podría denominar como ‘software esclavo’, que viene siendo la programación que obligatoriamente tiene que ir embarcada en la BeagleBone, ya que se encarga de manejar los recursos hardware de ésta. Recibe las órdenes del software de control, las procesa y actúa en base a ello. Por otro lado, se tiene el denominado ‘software de control’ que consiste en la interfaz que permite al usuario interactuar con el sistema. Recoge las órdenes del usuario y se las pasa al software esclavo.

⁴En la subred creada entre el ordenador y la BeagleBone, el ordenador debe tener una IP 192.168.1.1

La puerta de enlace debe ser 192.168.1.1 y DNS: 8.8.8.8

Uno de los objetivos que se ha buscado y que influye en el desarrollo del software, es que la unidad central de control opere automáticamente (Plug-and-play). Esto significa que una vez que esté completamente programada, al activar la tarjeta BeagleBone, ésta sea capaz de realizar directamente sus funciones sin que el usuario final tenga que acceder a ella para realizar ajuste alguno. Esto implica que ha sido necesario programarla para que ejecute automáticamente alguna instrucción auxiliar (algún script) durante el proceso de arranque del sistema Linux, justo después del arranque del Kernel. A lo largo de las siguientes secciones se irán viendo qué instrucciones ha sido necesario incluir en la fase de arranque y cómo se ha hecho.

5.2.1. Software esclavo

BeagleBone permite escribir programas en diferentes lenguajes y entornos de programación (IDE), desde los lenguajes Python o Ruby hasta utilizar el IDE Cloud9 que accede, compila y ejecuta desde el navegador web. En este caso para realizar la programación del software “esclavo” se optó por usar el lenguaje C/C++, ya que se había manejado con anterioridad durante la carrera.

Los programas en C/C++ se pueden escribir, compilar y ejecutar directamente sobre el Shell de Linux Debian (a través de PuTTY). Sin embargo, esa opción no es muy visual para programas con bastantes líneas de código ni efectiva para depurar errores de compilación, por lo que se decidió trabajar a través del entorno de programación NetBeans. Este programa se ejecuta en el ordenador pero utiliza el compilador de la plataforma de desarrollo BeagleBone, es decir, el código fuente y el código ejecutable se crean en dicha plataforma⁵.

Se ha desarrollado un programa en C/C++, que a la hora de ser ejecutado, recibe las ordenes del software de control. El programa realiza básicamente dos tareas.

Por un lado, se encarga de interpretar esas instrucciones (en forma de argumentos que se pasan al código C) y de crear los paquetes API que se necesita que sean radiados a las unidades remotas. Al mismo tiempo, se encarga de procesar los paquetes API recibidos y obtener los datos de interés en función de las órdenes recibidas.

⁵Es necesario que la plataforma BeagleBone disponga del programa Samba. Gracias a este programa también se pueden meter archivos en ella.

Por otro lado, se encarga de los aspectos relacionados con la comunicación serie entre la tarjeta BeagleBone y el Xbee, es decir, de activar el puerto UART, de fijar el puerto UART con los parámetros variables del protocolo y de escribir los datos a transmitir, así como de leer los datos recibidos.

5.2.1.1. Comunicación serie

Como se comentó anteriormente para realizar la comunicación se utiliza el puerto UART1, concretamente el pin 24 (del conector P9) actuando de señal de salida UART1_TXD y el pin 26 (del conector P9) actuando de señal de entrada UART1_RXD. Por la línea UART1_TXD la BeagleBone envía los paquetes API al Xbee, el cual los transmite por radiofrecuencia. Por la línea UART1_RXD la BeagleBone recibe los paquetes API que llegan al Xbee por radiofrecuencia. Como se observa en la tabla 5.4 para que ambos pines operen de esta manera deben ser configurados al modo 0⁶.

Según el manual técnico del microprocesador AM335X ARM® Cortex™-A8, para fijar la salida del multiplexador encargado de la configuración de los pines se tienen 7 bits, ya que los otros 25 bits más significativos están reservados. En la tabla 5.5 se puede observar que los 3 bits menos significativos (LSB) fijan el modo de operación y que el 5º LSB se tiene que configurar a '0' en UART_TXD y a '1' en UART_RXD. El resto de los bits se fijan siempre a cero.

A continuación se necesitó realizar el paso previo de montar el fichero especial debugfs en el directorio `/sys/kernel/debug`. Esto se realizaba mediante el comando:

```
mount -t debugfs none /sys/kernel/debug/
```

Sn embargo era necesario realizar esto de manera automática en el arranque del sistema, por lo que en el fichero de configuración `/etc/fstab`, donde se encuentran la lista de discos y particiones disponibles (así como las instrucciones sobre cómo montar cada dispositivo), se introdujo la siguiente instrucción:

```
debugfs /sys/kernel/debug debugfs rw 0 0
```

⁶Las tablas completas, con todos los modos de operación de cada uno de los pines de los conectores de expansión, se pueden obtener del manual de la BeagleBone [20]

| PIN | PROC | SIGNAL NAME | MODE0 | MODE1 | MODE2 | MODE3 |
|-----|------|-------------|---------------|--------------|-------------|------------------|
| 1 | | GND | | | | |
| 2 | | GND | | | | |
| 3 | | DC_3.3V | | | | |
| 4 | | DC_3.3V | | | | |
| 5 | | VDD_5V | | | | |
| 6 | | VDD_5V | | | | |
| 7 | | SYS_5V | | | | |
| 8 | | SYS_5V | | | | |
| 9 | | PWR_BUTTON | | | | |
| 10 | A10 | SYS_RESETn | RESET_OUT | | | |
| 11 | T17 | UART4_RXD | gpmc_wait0 | mii2_crs | gpmc_csn4 | rmii2_crs_dv |
| 12 | U18 | GPIO1_28 | gpmc_be1n | mii2_col | gpmc_csn6 | mmc2_dat3 |
| 13 | U17 | UART4_TXD | gpmc_wpn | mii2_rxerr | gpmc_csn5 | rmii2_rxerr |
| 14 | U14 | EHRPWM1A | gpmc_a2 | mii2_txd3 | rgmii2_td3 | mmc2_dat1 |
| 15 | R13 | GPIO1_16 | gpmc_a0 | gmii2_txen | rmii2_tctl | mii2_txen |
| 16 | T14 | EHRPWM1B | gpmc_a3 | mii2_txd2 | rgmii2_td2 | mmc2_dat2 |
| 17 | A16 | I2C1_SCL | spi0_cs0 | mmc2_sdwp | I2C1_SCL | ehrpwm0_synci |
| 18 | B16 | I2C1_SDA | spi0_d1 | mmc1_sdwp | I2C1_SDA | ehrpwm0_tripzone |
| 19 | D17 | I2C2_SCL | uart1_rtsn | timer5 | dcan0_rx | I2C2_SCL |
| 20 | D18 | I2C2_SDA | uart1_ctsn | timer6 | dcan0_tx | I2C2_SDA |
| 21 | B17 | UART2_TXD | spi0_d0 | uart2_txd | I2C2_SCL | ehrpwm0B |
| 22 | A17 | UART2_RXD | spi0_sclk | uart2_rxd | I2C2_SDA | ehrpwm0A |
| 23 | V14 | GPIO1_17 | gpmc_a1 | gmii2_rxdv | rgmii2_rxdv | mmc2_dat0 |
| 24 | D15 | UART1_TXD | uart1_txd | mmc2_sdwp | dcan1_rx | I2C1_SCL |
| 25 | A14 | GPIO3_21 | mcaspo_ahclkx | eQEP0_strobe | mcaspo_axr3 | mcaspo_axr1 |
| 26 | D16 | UART1_RXD | uart1_rxd | mmc1_sdwp | dcan1_tx | I2C1_SDA |
| 27 | C13 | GPIO3_19 | mcaspo_fsr | eQEP0B_in | mcaspo_axr3 | mcaspo_fsx |

Tabla 5.4: Modos de operación del conector P9

Para modificar el modo de operación de estos pines se accede hasta el directorio `/sys/kernel/debug/omap_mux`, desde el cual se tiene acceso a controlar el hardware de los pines 24 y 26. Éstos vienen configurados por defecto de la siguiente manera (ejemplo del pin 24):

```
name: uart1_txd.gpio0_15 (0x44e10984/0x984 = 0x0037), b NA, t NA
```

```
mode: OMAP_PIN_INPUT_PULLUP | OMAP_MUX_MODE7
```

Ambos pines operan por defecto en el modo 7 como pines de propósito general (GPIO), por lo que no valdría con modificarlos desde el shell de linux (comando: `echo 0 > uart1_txd`), toda vez que al encender nuevamente la BeagleBone los pines

| Bit | Field | Type | Reset | Description |
|-------|-------------------------------|------|-------|--|
| 31-20 | Reserved | R | 0h | |
| 19-7 | Reserved | R | 0h | |
| 6 | conf_<module>_<pin>_slewctrl | R/W | X | Select between faster or slower slew rate 0: Fast 1: Slow Reset value is pad-dependent. |
| 5 | conf_<module>_<pin>_rxactive | R/W | 1h | Input enable value for the PAD 0: Receiver disabled 1: Receiver enabled |
| 4 | conf_<module>_<pin>_pullupsel | R/W | X | Pad pullup/pulldown type selection 0: Pulldown selected 1: Pullup selected Reset value is pad-dependent. |
| 3 | conf_<module>_<pin>_pullupden | R/W | X | Pad pullup/pulldown enable 0: Pullup/pulldown enabled 1: Pullup/pulldown disabled Reset value is pad-dependent. |
| 2-0 | conf_<module>_<pin>_muxmode | R/W | X | Pad functional signal mux select. Reset value is pad-dependent. |

Tabla 5.5: Descripción de los bits de configuración

volverían a tener su configuración inicial. Finalmente se optó por configurarlos en el propio código del programa de tal manera que cada vez que se usa la comunicación serie se procede a configurar los pines de nuevo. La configuración es la siguiente:

- UART1_TXD -> 0x00 (0000 0000)
- UART1_RXD -> 0x20 (0010 0000)

En un sistema UNIX los puertos serie son accesibles a través de ficheros especiales de dispositivo (“device files”). Una vez que el puerto UART1 está activado, para transmitir y recibir los datos se usa el fichero de dispositivo */ttyO1*, que está localizado en el directorio */dev*.

Los parámetros de la comunicación serie deben ser los mismos que los del módulo Xbee, los cuales fueron detallados en la sección 3.1.4.2.1. Es decir, es necesario fijar una tasa de 9600 baudios. El resto de parámetros (longitud de palabra de 8 bits, sin bit de parada y sin paridad) vienen así por defecto.

Para lograr el objetivo de comunicarse con el dispositivo hardware Xbee, se requieren librerías adicionales a las habituales de los programas C, por lo que se necesitó incluir el fichero de cabecera *termios.h* en el programa. *Termios* es la interfaz estándar especificada por POSIX (‘Application Programming Interface’ de sistemas operativos UNIX). Las funciones *termios* definen una interfaz general para los terminales

entrada/salida. El fichero de cabecera `termios.h`, específico para la programación en C, define las macros, las funciones y las estructuras necesarias para realizar la configuración del puerto serie. Para trabajar con un puerto serie haciendo uso de las funciones y estructuras de `termios.h`, se siguen los siguientes pasos:

- Abrir el fichero especial asociado al puerto. [función: `open()`]
- Configurar los parámetros de comunicación y las propiedades de la interfaz. [funciones: `cfsetospeed()`, `tcgetattr()` y `tscsetattr()`. Estructura: `termios`]
- Leer y escribir los datos. [funciones: `read()` y `write()`]
- Cerrar el fichero asociado al puerto. [función: `close()`]

Para leer los datos de entrada de dispositivos serie se procede, dependiendo de la aplicación, básicamente de dos maneras diferentes [21]:

- Entrada canónica: La entrada se organiza en líneas. Los datos se van almacenado en el buffer hasta que haya una línea de entrada completa, lo cual está establecido por un carácter fin de fichero o por un carácter fin de línea. La llamada al sistema `read()` solo podrá devolver líneas de entradas completas.
- Entrada No canónica: La entrada se organiza por un número fijo de caracteres o por un parámetro temporal.

En este proyecto se ha optado por un procedimiento de entrada no canónico, ya que con el procedimiento canónico se tendrían problemas debido a que las tramas API pueden tener bytes que correspondan a caracteres especiales (como CR o NL), lo que generaría problemas a la hora de leer las entradas.

5.2.1.2. Paquetes API

El programa construye los paquetes API necesarios mediante Arrays de caracteres ASCII, que equivalen a los diferentes campos que componen los paquetes. Cada carácter corresponde a un byte de la trama API. Además, como los campos de las tramas están formados por un número entero de bytes, la construcción de los paquetes API resulta sencilla. Por otro lado, el programa extrae la información de interés

contenida en los paquetes API recibidos y la almacena en archivos para que esté a disposición del software de control.

En esta sección no se va a entrar en los tipos de paquetes API que se crean y se reciben o en cómo es su composición interna, ya que esto fue detallado en la sección 3.1.6 y lo único que se ha hecho ha sido plasmarlo en código C.

5.2.2. Software de control

Para que el usuario pueda interactuar con el sistema domótico, se ha desarrollado un software de control que sirve de interfaz de usuario, permitiendo a éste, seleccionar una unidad remota y a continuación consultar el valor de los sensores, así como activar o desactivar los actuadores. Al mismo tiempo, la interfaz debe servir para probar el funcionamiento general del sistema y para realizar pruebas de funcionamiento.

Llegados a este punto, conviene recordar de nuevo que uno de los objetivos básicos del proyecto es que se pudiese realizar el control del sistema domótico remotamente a través de internet.

Para la realización de esta interfaz se barajó la posibilidad de desarrollar una aplicación de escritorio que se instalase en un ordenador (bajo un entorno Windows). Esta opción, sin embargo, tenía la desventaja de que limitaba los dispositivos a través de los cuales se podría controlar el sistema, ya que de querer emplear otros equipos con distintos sistemas operativos, como por ejemplo tablets Mac o Android, habría que desarrollar aplicaciones diferentes. Se decidió que la opción más idónea era desarrollar una aplicación web, la cual no necesita ser instalada, ya que se ejecutaría a través de un navegador. Esta opción, permite al usuario poder interactuar cómodamente con el sistema desde cualquier dispositivo.

En definitiva, la principal ventaja de esta elección es que el sistema será accesible desde cualquier dispositivo que tenga un navegador web, independientemente de su sistema operativo, no siendo necesario disponer de software específico instalado en los ordenadores, smartphones o tablets.

El sistema se basa en un modelo cliente-servidor, donde el usuario (cliente), requiere del servicio de un servidor. Para el funcionamiento del sistema se requiere de una aplicación web alojada en un servidor y en el lado del cliente de un navegador web que permita acceder a la aplicación.

El funcionamiento se basará en que el cliente enviará peticiones http que serán atendidas por el servidor y éste enviará un código HTML para que el cliente pueda visualizar las páginas web en el navegador

5.2.2.1. Servidor web

Por servidor web habitualmente se hace referencia a cualquier dispositivo de una red que brinda un servicio a otros equipos, aunque técnicamente el servidor web es el software (programa) que se está ejecutando continuamente sobre ese dispositivo a la espera de peticiones de ejecución por parte de un cliente.

Durante la realización del proyecto se consideró que la mejor manera de poder realizar el control del sistema domótico a través de internet era convertir la plataforma BeagleBone, que desempeña el papel de dispositivo central de control de la red, en un servidor web al mismo tiempo. Para ello, se investigó sobre las alternativas existentes, sobre qué se necesitaba instalar o sobre los ajustes necesarios. Había diferentes opciones de servidores web que podían ser instalados en la BeagleBone, por ejemplo, Lighttpd o Tomcat, pero finalmente se optó por hacer uso de uno de los más populares y sencillos a la hora de instalarse, Apache [22].

Apache es un servidor web HTTP de código abierto, ejecutable bajo diferentes sistemas operativos, como GNU Linux (Debian), Windows o Mac, que implementa el protocolo HTTP/1.1. Una de las grandes ventajas de Apache es que es un servidor modular, es decir, permite ir ampliando sus capacidades con el paso del tiempo (existen muchísimos módulos para ser instalados cuando se necesiten). Además, es altamente configurable, permitiendo personalizar la respuesta ante posibles errores, trabajar con bases de datos, etc.

Para instalar un servidor Apache en la plataforma BeagleBone se puede descargar el paquete `apache2` disponible para distribuciones Debian (instrucción: `sudo apt-get install apache2`).

Como ya se ha mencionado el objetivo del servidor es alojar una aplicación web que permita interactuar con el usuario. Para este propósito no es suficiente con tener una página web estática que se limite a mostrar información fija y predefinida, sino que se necesitará de una página web dinámica (aplicación web). Este tipo de páginas web requieren, además de HTML (HyperText Markup Language-language), de algún len-

guaje de programación del lado del servidor. Dicho lenguaje permitirá que el servidor ejecute un programa antes de enviar la información oportuna al cliente.

El servidor Apache se puede usar tanto para páginas estáticas como para páginas dinámicas, ya que soporta varios lenguajes de programación del lado del servidor como PHP, Perl, etc. Se optó por el primero, por lo que se necesitó instalar PHP (versión 5) y el módulo PHP para Apache (instrucción: `sudo apt-get install php5 libapache2_mod_php5`).

La carpeta raíz del servidor apache es `/www/var`, por lo que todos los documentos que se desea que sean accesibles vía web deben estar dentro de esta carpeta. En su interior se encuentran las páginas creadas y todas las imágenes contenidas en ellas.

Aunque para este proyecto no ha sido necesario, existe la posibilidad de instalar también MySQL, que es un sistema de base de datos (utiliza SQL). En este caso, se dispondría de un servidor LAMP (Linux, Apache, MySQL, PHP). Los servidores LAMP son un conjunto de soluciones que, al combinar estas tecnologías, son capaces de ofrecer una infraestructura de servidor robusta, de gran funcionalidad y con capacidad de adecuación a muchas necesidades.

5.2.2.2. Aplicación web

Como software de control se ha desarrollado una aplicación web que sirve de interfaz para el usuario. Esta interfaz debe ofrecer los servicios ya conocidos que están al alcance del sistema domótico implementado, y debe hacerlo de una manera sencilla, visual e intuitiva, de tal modo, que pueda ser utilizada por usuarios sin conocimientos sobre el funcionamiento del sistema.

La aplicación web no se limita a presentar información al usuario, sino que requiere interactuar con él, por lo que se necesita de lenguajes de programación. Éstos pueden ser básicamente de dos tipos: los lenguajes de programación del lado servidor, que son ejecutados e interpretados por el propio servidor y los lenguajes del lado cliente que son aquellos que se envían al cliente junto al lenguaje HTML. Se optó por emplear PHP al ser un lenguaje del lado del servidor y por ser uno de los soportados por Apache.

En definitiva, la aplicación web consiste en un programa escrito en PHP, alojado en un servidor y que incorpora código HTML, el cual será enviado al navegador web

del cliente para que lo interprete. PHP aporta la funcionalidad a la aplicación web, mientras HTML se encarga de los aspectos visuales de la interfaz⁷.

En la figura 5.6 se puede observar el diagrama de flujo del funcionamiento general de la aplicación⁸.

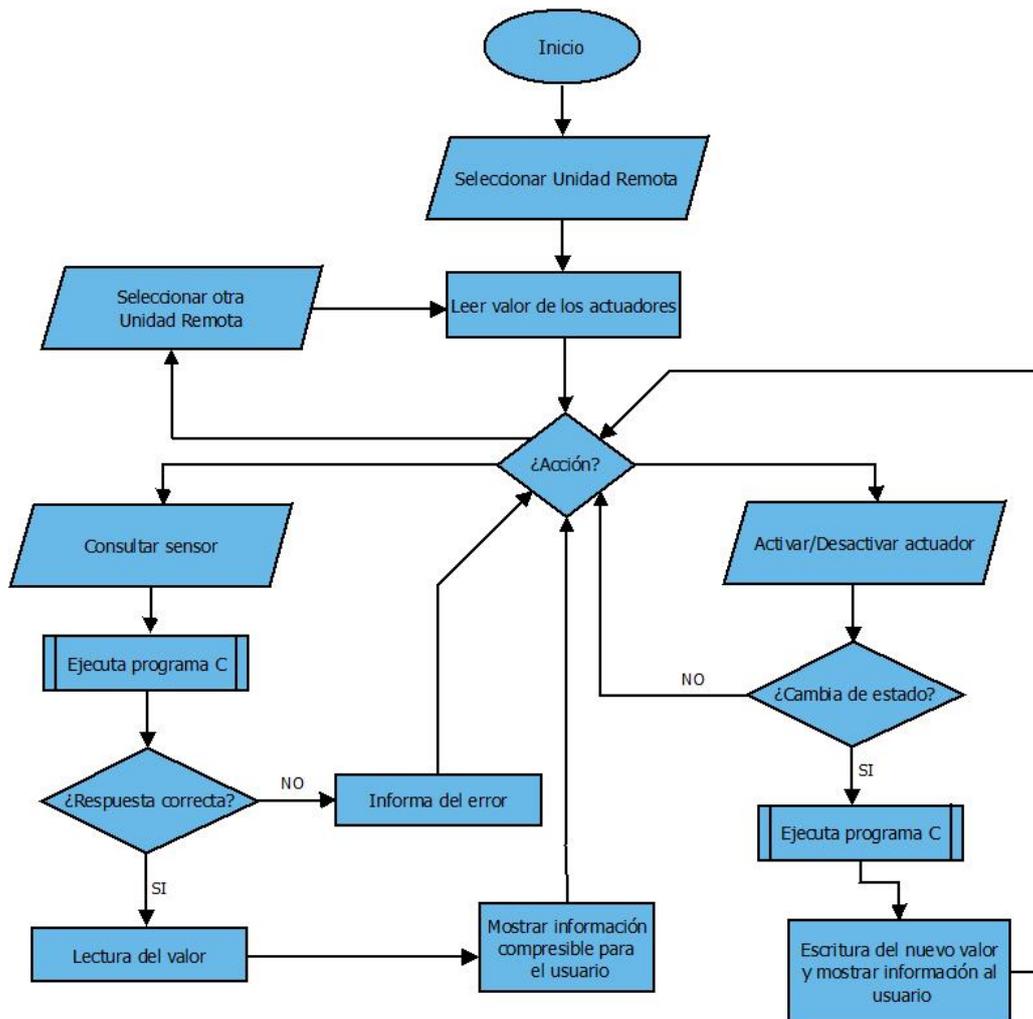


Figura 5.6: Diagrama de flujo del funcionamiento general

La aplicación se encarga de ejecutar el programa externo detallado en la sección anterior (software esclavo), pasándole las ordenes en forma de argumentos mediante

⁷También se incluyó una pequeña sección de código en el lenguaje del lado del cliente Javascript. Su única función es mostrar mensajes de alerta en una ventana que despliega el navegador en caso de que el usuario intente realizar una opción no válida.

⁸El navegador envía los datos al servidor, a través del método GET y del método POST.

el formato Dispositivo/Valor/Lugar⁹. El campo “Dispositivo” representa el sensor que se desea consultar o el actuador que se quiere modificar. El “Valor” indica si se debe activar o desactivar el actuador, siendo su contenido para el resto de casos irrelevante. El “Lugar” representa al identificador de la unidad remota, que coincide con el parámetro DH de sus respectivos módulos Xbee.

Como se desprende del diagrama anterior, se necesita de una serie de archivos donde se almacenan los valores de los actuadores de cada unidad remota, es decir, si están activados o desactivados. De esta manera al iniciar la aplicación web se carga el estado de la unidad remota seleccionada. Por otro lado, se pensó inicialmente en la posibilidad de aprovechar el código de retorno (Exit Status), que devuelve el programa externo, para pasarle al software de control el valor del sensor solicitado. Sin embargo, en los sistemas Linux este valor de retorno está restringido a un entero de 8 bits, por lo que es insuficiente para representar la precisión de nuestros conversores ADC, que como se recuerda, era de 10 bits. Por tanto, se necesita de otro archivo auxiliar, donde el software esclavo escribe el valor del sensor oportuno y la aplicación web lo lee.

Es también en el software de control donde se realizan las operaciones necesarias para obtener las medidas de los sensores en las unidades adecuadas. Se tienen en cuenta los límites del conversor analógico-digital, los circuitos de acondicionamientos y las curvas características de los sensores.

Finalmente, durante el desarrollo del software de control se tuvo que tener en cuenta que la aplicación web no podía ejecutar algunos programas o acceder a ciertos recursos, ya que el servidor web corre con unos privilegios inferiores a los de superusuario. Para solucionar este inconveniente se ajustaron los permisos de los archivos que eran leídos y escritos por el programa PHP, así como del ejecutable del programa C, lo cual implica que también se necesitó modificar los permisos de los recursos de la BeagleBone a los que accedía éste. Concretamente, se necesitó modificar los permisos de los pines `uart1_tx` y `uart1_rx` del conector de expansión P8 y los permisos del fichero de dispositivo `/ttyO1` para que se pudiera leer y escribir en él.

Sin embargo surgió otro problema, ya que estos pines y ficheros tenían permisos de superusuario, y que cada vez que se reiniciaba la tarjeta BeagleBone volvían a tener

⁹Para ejecutar el programa externo se utiliza la función `exec()`. La ejecución es en modo secuencial, es decir, hasta que no finaliza el proceso la aplicación no continua con la siguiente instrucción.

sus permisos por defecto. Para solucionarlo, se tuvo que modificar los permisos en el arranque del sistema a través del archivo */etc/rc.local*, que ejecuta sus comandos y sus scripts al final del arranque, es decir, una vez que todos los scripts del nivel de ejecución correspondiente han sido ya ejecutados y, por lo tanto, tras haberse montado el fichero especial *debugfs* (explicado en la sección anterior). Fue necesario introducir los comandos:

```
chmod 777 /sys/kernel/debug/omap_mux/uart1_txd  
chmod 777 /sys/kernel/debug/omap_mux/uart1_rxd  
chmod o+r /dev/ttyO1  
chmod o+w /dev/ttyO1
```

En la figura 5.7 se puede ver la página web creada para controlar el tipo de unidades remotas diseñadas.

- (1) Selección de la unidad periférica sobre la que interactuar. En la pestaña desplegable aparecen las unidades remotas que hay disponibles en la instalación.
- (2) Selección de Actuadores. En cada actuador hay dos botones para activación y desactivación y un icono representativo del estado del actuador.
 - ✧ (a) Actuador1.
 - ✧ (b) Actuador2.
 - ✧ (c) Actuador3.
- (3) Selección de Sensores.
 - ✧ (a) Botón para obtener la medida del sensor de luminosidad.
 - ✧ (b) Botón del sensor de humedad.
 - ✧ (c) Botón del sensor de temperatura.
 - ✧ (d) Recuadro con la información del sensor correspondiente.

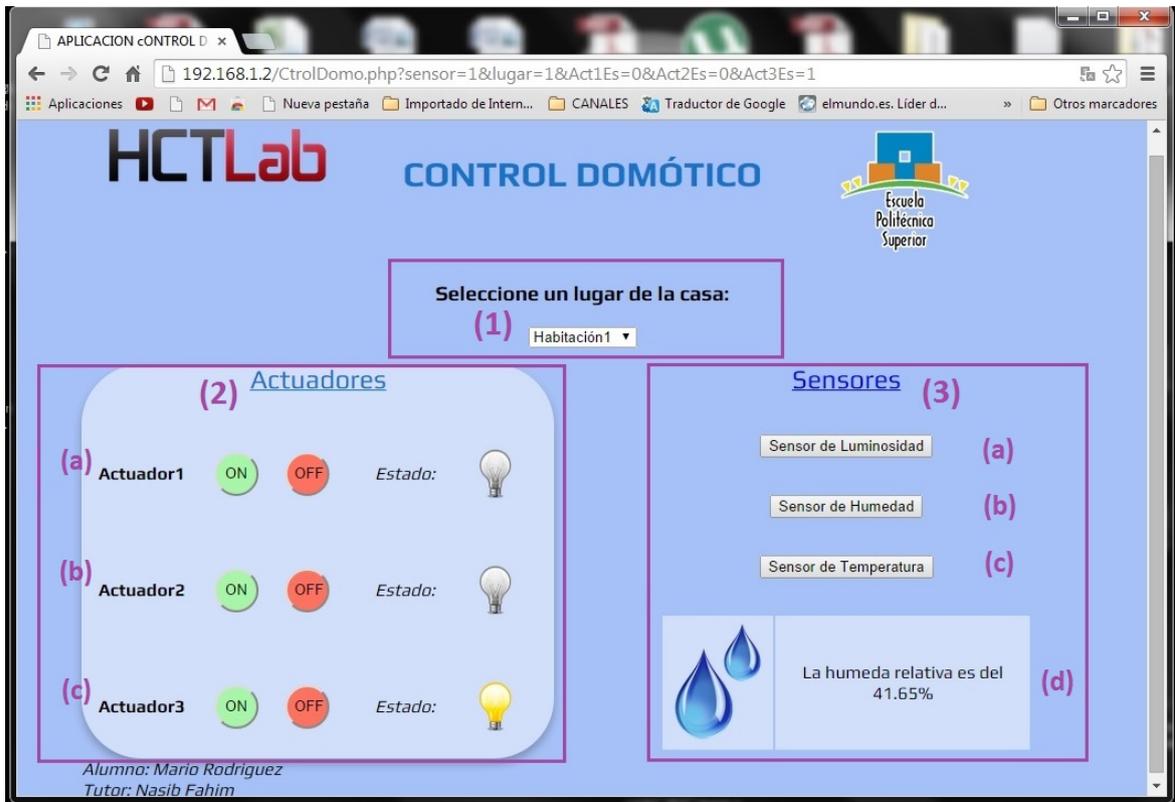


Figura 5.7: Aplicación web

Capítulo 6

Pruebas y Resultados

En este capítulo se presentan los resultados de las diferentes pruebas realizadas durante la elaboración del proyecto. Su objetivo es comprobar el funcionamiento del sistema implementado. La realización de estas pruebas permitió la detección de algunos errores y problemas de funcionamiento que trataron de ser corregidos al menos en parte.

En primer lugar se analizarán una serie de pruebas realizadas sobre partes del sistema por separado y sobre tareas concretas del sistema y por último, se presentará una prueba de la aplicación web controlando el sistema completo en su totalidad.

6.1. Comunicaciones UART

Inicialmente en la comunicación serie de la unidad central de control, se trataba al fichero de dispositivo `/ttyO1` como un fichero normal, lo que generaba errores. Tras realizar pruebas y analizarlo con un osciloscopio, se comprobó que había problemas ya que durante la lectura de los datos aparecían también los datos que habían sido enviados. Esto se pudo solucionar manejando el puerto UART a través del fichero de cabecera `termios.h`, mencionado con anterioridad.

Para desarrollos futuros se considera que es bastante más sencillo emplear dos puertos UART diferentes de la BeagleBone para realizar la comunicación serie (uno para transmisión de datos y otro para recepción de datos). De esta manera se trabajaría con

dos ficheros de dispositivo diferentes y habría menos complicaciones para coordinar las escrituras y las lecturas.

6.2. Sistema de comunicaciones

Como se comentó en la sección 3.1.7.2, se hizo uso del software X-CTU para comprobar que los módulos Xbee funcionasen correctamente y para realizar pruebas que simulasen las comunicaciones que tienen lugar en la red, cuando los XBee están en sus posiciones finales. Las pruebas, que consistieron en el envío de los paquetes API que se utilizan en el sistema, arrojaron las siguientes conclusiones:

1. EL módulo que corresponde al de las unidades remotas, recibe en la totalidad de las ocasiones los paquetes API de una manera correcta¹. Esto supone que siempre que se deba, las unidades remotas activarán o desactivarán los equipos que tengan conectados.
2. El módulo que corresponde a la unidad central, en ocasiones no recibe correctamente los paquetes API de respuesta. Durante los ensayos se ha observado que hay dos modalidades de errores a la hora de recibir los paquetes. En la figura 6.1 se puede observar una situación con todos los escenarios de comunicación posibles.
 - a) En el caso 1 se puede apreciar una transmisión correcta de paquetes; el módulo XBee envía un paquete API tipo comando remoto solicitando el valor de las entradas del Xbee remoto (bytes azules) y recibe un paquete API tipo respuesta de comando (bytes rojos) con dichos valores.
 - b) En el caso 2 se puede observar el primer tipo de anomalía que se ha encontrado. Una vez que se envía el paquete API, en lugar de recibir la respuesta correcta, se recibe:

7E 00 0F 97 01 00 00 00 00 00 00 00 00 01 49 53 04 C6

Este paquete indica que se ha producido fallo en la transmisión del comando remoto respuesta, pero como los módulos XBee vienen fijados por el

¹En estas pruebas se ha asegurado que los módulos estaban correctamente configurados, alimentados y dentro de un alcance válido.

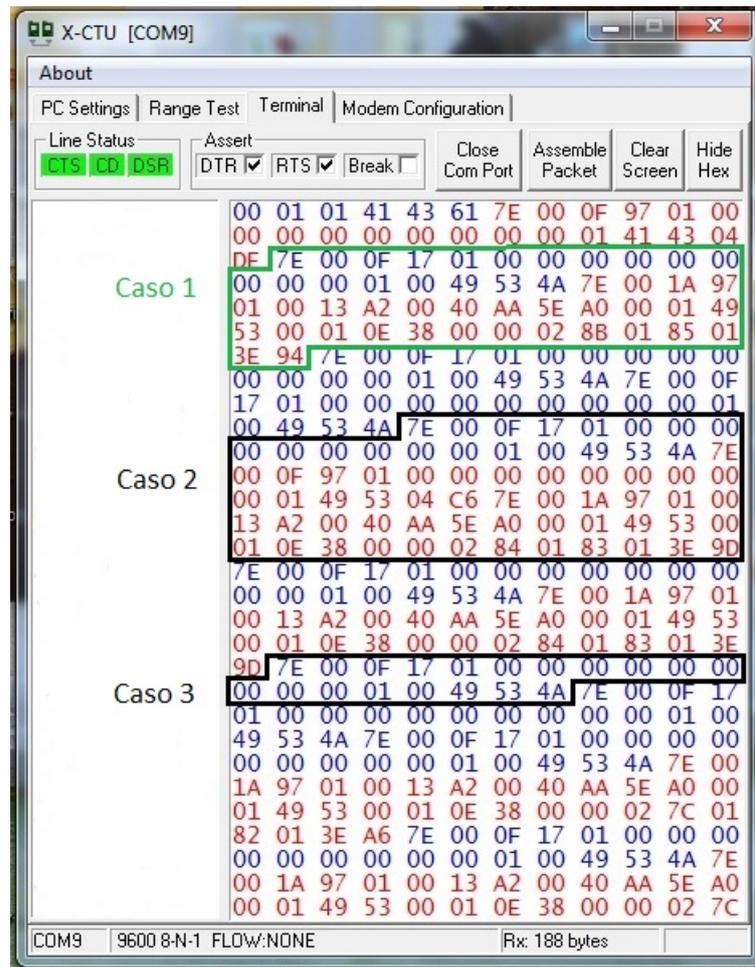


Figura 6.1: Paquetes API en la unidad central

estándar 802.15.4 (Capa MAC) para ejecutar hasta tres reenvíos si no han recibido el ACK (aparte de los que se le indiquen según el parámetro RR), automáticamente reenvían de nuevo el paquete. Como se puede observar, tras los datos de error se recibe el paquete correcto:

7E 00 0F 97 00 13 A2 00 40 AA 5E A0 00 01 49 53 00 01 0E 38 00 00 ...

Para intentar solucionar este problema, el código se adaptó para que se pudiesen leer entradas de más bytes de los que tendría un paquete correcto y obtener la información de los datos adecuados (**Ajuste 1**).

- c) Como se observa en el caso 3, en alguna ocasión no se reciben los paquetes API de respuesta. Aunque se ha investigado bastante sobre este asunto

no se ha podido averiguar el motivo por el que a veces se producen estos problemas². A efectos prácticos, esto supone que hay ocasiones en que los paquetes API con el estado de los sensores no llegan al control. Para intentar minimizar este problema, en el programa desarrollado en C se incluyó un mecanismo de control, de tal modo que si no se recibe el paquete API con los datos del sensor, se reenvía otro paquete API de solicitud (**mecanismo de pulling**). Si tras 10 intentos siguiese fallando, se deja de intentarlo y se informa al usuario de que hubo un error.

En la tabla 6.1 se pueden apreciar los resultados de las pruebas realizadas sin los mecanismos de corrección y con ellos. Cada prueba consta de cincuenta solicitudes para conocer el valor de los sensores de la unidad remota. Las pruebas del tipo A son envíos únicamente de paquetes API para conocer el estado de los sensores. En las pruebas del tipo B dichos paquetes se intercalan con paquetes API para la activación o desactivación de actuadores.

| Pruebas | % de errores Sin ajustes | % de errores Ajuste 1 | % de errores Ajuste 1 y Pulling |
|------------|--------------------------|-----------------------|---------------------------------|
| Prueba A-1 | 6 | 6 | 2 |
| Prueba A-2 | 0 | 0 | 0 |
| Prueba A-3 | 2 | 2 | 0 |
| Prueba A-4 | 10 | 10 | 0 |
| Prueba A-5 | 2 | 0 | 0 |
| Prueba B-1 | 22 | 8 | 2 |
| Prueba B-2 | 18 | 2 | 0 |
| Prueba B-3 | 14 | 6 | 0 |
| Prueba B-4 | 26 | 10 | 2 |
| Prueba B-5 | 24 | 2 | 0 |

Tabla 6.1: Pruebas recepción de paquetes

Al analizar los resultados se puede observar que gracias a las soluciones incorporadas se consigue un buen resultado, mejorando el funcionamiento del sistema.

²Se cree que el problema puede deberse a que el software que viene en los módulos Xbee tiene algunas limitaciones bajo el modo API. Concretamente, se piensa que el problema puede ser que el Xbee de la unidad central recibe el paquete, pero no lo saca por su puerto UART.

6.3. Circuitos impresos

Antes de realizar la construcción y montaje de los circuitos impresos que forman la unidad remota, se probaron los futuros diseños y sus componentes desde una proto-board. Una vez que los circuitos impresos fueron construidos también se comprobó que funcionasen correctamente, sin cortocircuitos ni otros problemas.

| | Fte. Alimentación 5V | F.A. 6V | F.A. 7V |
|-------------------------------|----------------------|---------|---------|
| Tensión entrada regulador (V) | 4.976 | 5.980 | 6.982 |
| Tensión salida regulador (V) | 3.285 | 3.282 | 3.275 |
| Tensión salidas digitales (V) | 3.267 | 3.272 | 3.265 |
| Consumo condiciones 'A' (mA) | 71 | 71 | 71 |
| Consumo condiciones 'B' (mA) | 156 | 167 | 181 |
| Consumo condiciones 'C' (mA) | 240 | 262 | 291 |
| Consumo condiciones 'D' (mA) | 323 | 357 | 398 |
| Consumo condiciones 'E' (mA) | 335 | 370 | 375 |

Tabla 6.2: Mediciones del consumo y las tensiones

En la tabla 6.2 se puede observar las medidas de las tensiones y del consumo de corriente de la unidad remota, en función de las pruebas realizadas con diferentes fuentes de alimentación. Como se puede observar la unidad remota tiene diferentes consumos dependiendo de las situaciones:

- Condiciones 'A'. La unidad remota se encuentra en reposo, es decir, los 3 relés desactivados y el módulo XBee en reposo (modo Idle). En esta situación es cuando la unidad remota tiene un menor consumo.
- Condiciones 'B'. El módulo XBee está en reposo y un relé activado.
- Condiciones 'C'. Dos relés activados y el XBee en reposo.
- Condiciones 'D'. El módulo Xbee en reposo y los tres relés activados a la vez.
- Condiciones 'E'. Los 3 relés están desactivados y el módulo XBee está transmitiendo paquetes por radiofrecuencia. Es en estas situaciones cuando el XBee presenta mayor consumo con gran diferencia respecto al modo reposo o a cuando está recibiendo datos por radiofrecuencia.

Los consumos relacionados con el módulo XBee son aproximados y pueden variar respecto a los indicados en las especificaciones; esto es normal ya que medir el consumo de un Xbee es una tarea compleja en la que intervienen factores como el tipo de topología, si el módulo es coordinador o dispositivo final, el firmware, las condiciones ambientales, etc [23].

6.4. Pruebas de Alcance

Para realizar las pruebas de cobertura y alcance del sistema se procedió a ir colocando la unidad remota en varios lugares, separada de la unidad central por diferentes distancias. Estos ensayos se realizaron en dos ambientes, en el exterior y en el interior de nuestra vivienda (ver figura 6.2). Para las pruebas se realizaron envíos de órdenes de encendido y apagado de los actuadores. En cada situación se enviaron 20 paquetes de activación y desactivación. En la tabla 6.3 se pueden observar los resultados obtenidos en estos tests.

| | Situaciones | Paquetes recibidos |
|--|----------------------------------|--------------------|
| Entorno Controlado (Aire libre) | Distancia 10 metros | 100 % |
| | Distancia 50 metros | 100 % |
| | Distancia 100 metros | 100 % |
| | | |
| Entorno Ruidoso (Interior vivienda) | Misma habitación ('A') | 100 % |
| | Habitación contigua ('B') | 100 % |
| | Habitación intermedia ('C') | 100 % |
| | Extremo opuesto de la casa ('D') | 100 % |
| | | |

Tabla 6.3: Pruebas de Alcance

Tras analizar los resultados se puede concluir que la distancia a la que se coloquen las unidades remotas y la cobertura no ocasionaran ningún problema, de hecho, este resultado, con todos los paquetes recibidos independientemente de la separación entre la unidad remota y la unidad de control, se debe a la elección de módulos Xbee del tipo PRO que tienen un alcance excepcional. Sin embargo estos resultados, junto con los de la sección anterior que mostraban el gran consumo que tiene la unidad remota,

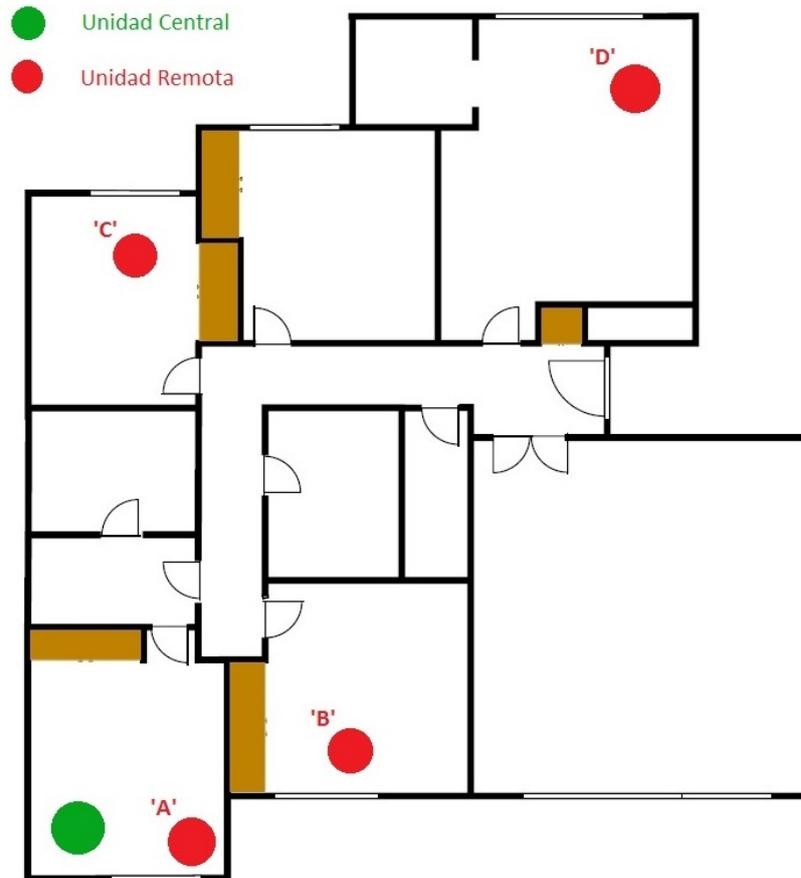


Figura 6.2: Plano de la vivienda donde se realizaron las pruebas

ponen de manifiesto que tal vez hubiese sido más adecuado utilizar módulos XBee del tipo normal, que serían igual de válidos para una vivienda media y que presenta un consumo bastante menor.

Como se menciona en el capítulo 3, los módulos XBee del tipo PRO de la Serie 1 pueden ser sustituidos por módulos XBee normales en cualquier momento sin que haya que modificar nada en el sistema o del hardware construido y con los mismos parámetros de configuración.

6.5. Aplicación y sistema completo

A continuación se muestra una prueba del funcionamiento de la aplicación a la hora de controlar, en su totalidad, el sistema domótico desarrollado. Las pruebas se realizan

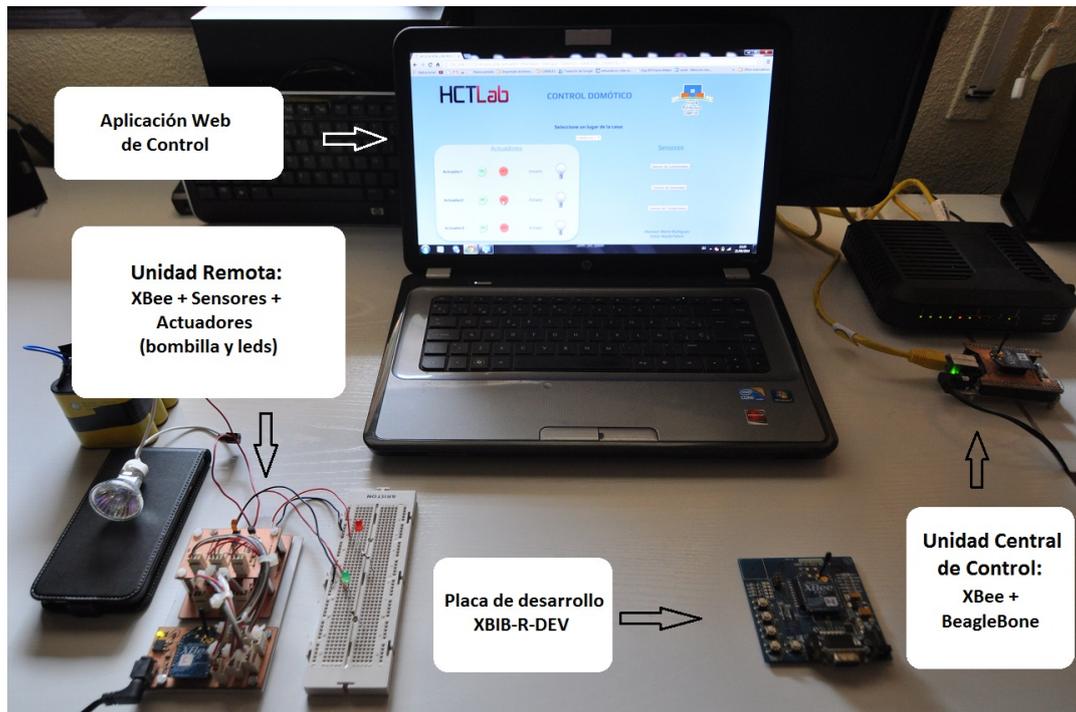


Figura 6.3: Implementación del sistema

con el sistema prototipo, formado por una unidad de control y una única unidad remota. Ambas son alimentadas por una fuente de alimentación cableada de 5 voltios. Para poder simular los equipos que se podrían conectar a los actuadores de la unidad remota, se utilizaron LEDs y bombillas alógenas (ver figura 6.3). En el anexo G se puede observar el diagrama de secuencia de la prueba.

1. Acceso del usuario y selección de la unidad remota: Desde la pestaña desplegable situada en mitad de la aplicación, se elige la unidad remota deseada (*Habitación 1*. Figura 6.4).

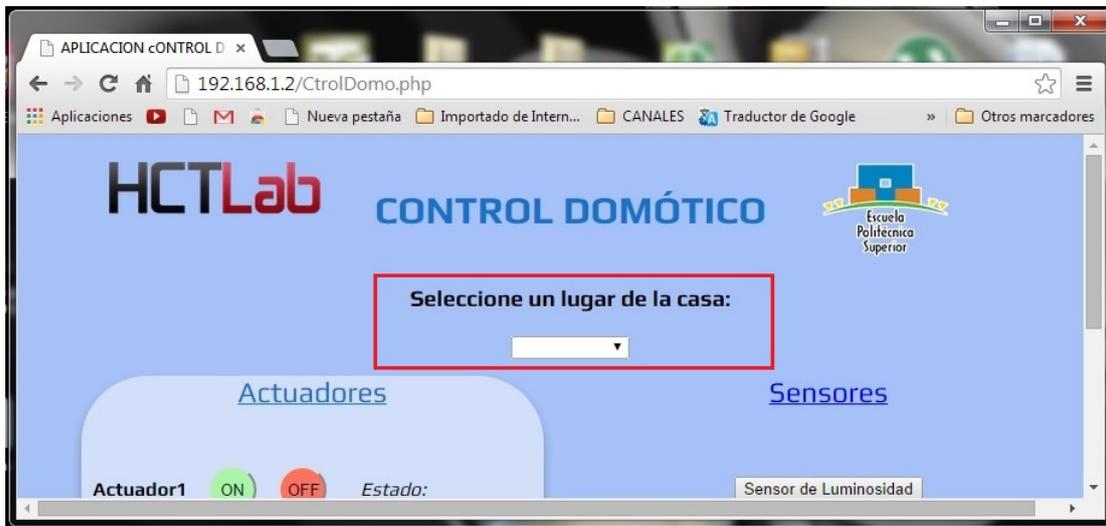


Figura 6.4: Aplicación Web I

2. Activación de los actuadores: A partir de los botones situados a la izquierda de la interfaz, se puede activar o desactivar el actuador deseado (*Activar actuador 2*. Figura 6.5).



Figura 6.5: Aplicación Web II

3. Lectura de los sensores: A partir de los botones situados a la derecha de la interfaz se puede consultar el valor de los sensores (*Sensor de temperatura*. Figura 6.6).

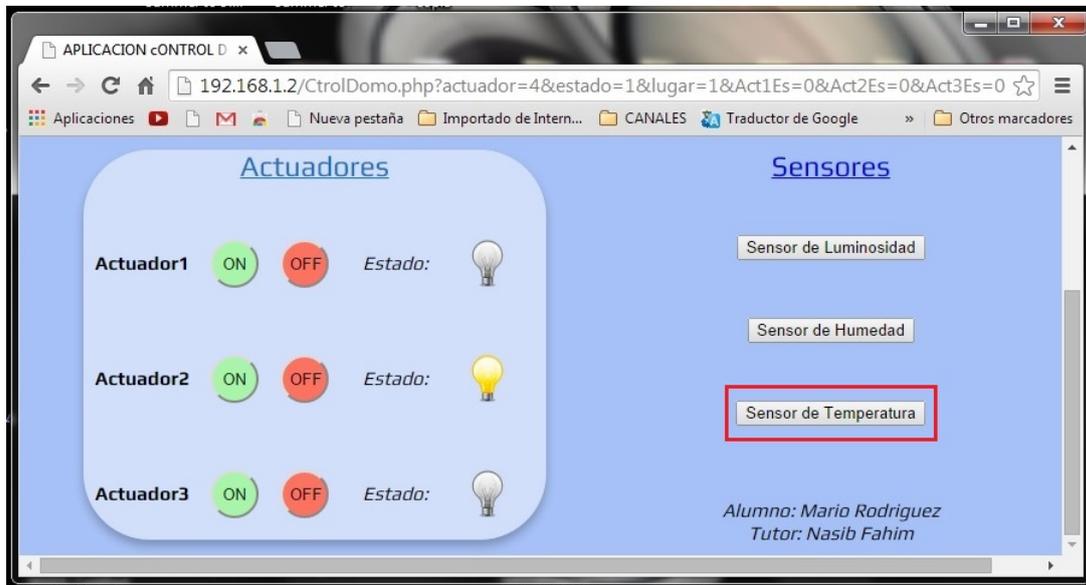


Figura 6.6: Aplicación Web III

4. Se muestra el valor del sensor: Aparece un recuadro con la información del valor solicitado. (*Sensor de temperatura*. Figura 6.7).



Figura 6.7: Aplicación Web IV

Esta aplicación ha sido probada en el ordenador bajo los navegadores Google Chrome e Internet Explorer y sobre una tablet Android bajo su navegador nativo.

Las pruebas se han realizado conectando la tarjeta BeagleBone al Router de nuestra casa. De esta manera se tiene el sistema listo para acceder desde cualquier dispositivo conectado a la red local. Para acceder desde el exterior (fuera de la red local), cada usuario deberá abrir los puertos y ajustar los filtros necesarios de su router. La forma de hacer esto dependerá del modelo de router del que disponga.

Para poder acceder al servidor a través de un navegador en la URL se introduce la dirección IP del servidor (La BeagleBone tiene la IP fija 192.168.1.2)³, seguida de CtrolDomo.php. El puerto no es necesario incluirlo, ya que se utiliza el puerto por defecto para http (puerto 80).

³En el caso de que se accediese desde el exterior, se pondría la IP pública que tiene asignada el router

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

La realización de este proyecto me ha dado la oportunidad de adentrarme en el mundo de la domótica y de sus posibilidades. También me ha permitido conocer más acerca de los protocolos de comunicación y las aplicaciones web. Lo más positivo ha sido la oportunidad de aprender sobre el mundo de la electrónica y tener que enfrentarme a situaciones reales.

En la fase inicial del proyecto, se investigó sobre los distintos tipos de instalaciones domóticas y sus posibilidades de desarrollo. Tras esto, se definieron las características que debía poseer el sistema y se establecieron las tareas que era necesario realizar, es decir, qué software y hardware había que implementar.

Un aspecto importante supuso la selección y construcción del hardware para poder llevar a cabo la implementación física del sistema domótico, el cual se compone de una unidad central y de una unidad remota. Las decisiones se tomaron en función de aspectos como: el sistema de comunicaciones, el tipo de control, las aplicaciones que debía cubrir, etc. En este proceso se seleccionaron los módulos de radiofrecuencia, la plataforma embebida que realiza el control y se construyó el hardware electrónico de una unidad remota de funcionalidad estándar.

Una de las partes fundamentales del proyecto fue la relacionada con la selección y configuración del sistema de comunicaciones inalámbrico que permite la comunicación entre los distintos dispositivos de la red. Según avanzaba el proyecto y ante las

dificultades que se iban encontrando, se fueron realizando variaciones en los modos de transmisión y en los tipos de paquetes a transmitir.

Se ha desarrollado un sistema con red en estrella, que permite ampliar la cantidad de unidades remotas en tantas como se quiera. Una de las ventajas que se ha conseguido, es hacer uso de un protocolo de comunicaciones que permite realizar el control de unidades remotas de funcionalidad sencilla, que carecen de inteligencia y que aprovechan las posibilidades de la tecnología de comunicaciones y de los módulos de radiofrecuencia. Al mismo tiempo, este protocolo permite controlar cualquier otra unidad remota con nuevas funcionalidades.

La última fase del proyecto fue el desarrollo del software que controla el sistema, con lo que ahora se dispone de una aplicación web que permite al usuario interactuar y controlar la instalación a través de Internet, cumpliendo con uno de los requisitos que se fijó al inicio del proyecto. La aplicación web tiene la ventaja de ser accesible desde cualquier dispositivo con navegador y permite con facilidad añadir nuevas opciones de control y supervisión, así como nuevas funcionalidades.

Pienso que este proyecto ha logrado el objetivo de desarrollar el prototipo de un sistema domótico completo que pueda ser controlado remotamente a través de Internet.

7.2. Trabajo futuro

Dado los numerosos ámbitos que componen la domótica, este proyecto está sujeto a una profunda evolución. En futuros trabajos se podrán ir incorporando nuevas aplicaciones y posibilidades. Se cree que las principales líneas de trabajo que deja abierto el proyecto son:

- Añadir otras unidades remotas. En este proyecto se ha desarrollado un sistema que será la base de futuras ampliaciones, por lo que se considera que la opción más idónea sería el diseño de otros tipos de unidades periféricas que se puedan acoplar al sistema. Resultaría interesante diseñar unidades con inteligencia propia que incorporasen microcontroladores lo que permitiría realizar funciones más avanzadas. De hecho, en el HCTLab ya se está trabajando en esta línea, concretamente se está realizando un proyecto denominado “*Sistema para realización de auditorías de consumo de energía eléctrica a través de internet*”, que

permitirá controlar el consumo eléctrico de equipos en una vivienda. De la misma manera, al disponer de microcontroladores, las unidades remotas podrían incluir sensores digitales complejos y emplear otros tipos de actuadores

- Aprovechar los recursos que ofrece la tarjeta BeagleBone, para que desempeñe nuevas tareas añadidas al control de la red. Es decir, se podría hacer uso de los sistemas de comunicación que dispone para manejar sensores y actuadores incorporados en la propia unidad de control. También se podría incorporar una pantalla táctil que, mediante una aplicación, ofreciese una alternativa al control del sistema a través de Internet.
- Hacer uso de otras tecnologías domóticas que complementen a la tecnología ZigBee utilizada en el sistema, de tal manera que la unidad central de control pudiera coordinar diferentes redes. Por ejemplo, se podría hacer un uso conjunto de ZigBee y Bluetooth, dedicando ésta última para aspectos multimedia que requieran más tráfico de datos y dejando ZigBee para la red de control (sensores y actuadores).

Bibliografía

- [1] Angel Ayala Bernal. Las empresas domóticas en España 1990-2010: evolución, estado actual y perspectivas de futuro en el sector residencial y de pequeño y mediano terciario. Master's thesis, Trabajo Fin de Master, Universitat Politècnica de Catalunya, 2013.
- [2] Cristóbal Romero Morales, Francisco Vázquez Serrano, and Carlos de Castro Lozano. *Domótica e Inmótica Viviendas y Edificios Inteligentes*. Segunda edición. Madrid, España: RA-MA, 2007.
- [3] Manuel Jiménez Buendía. *Desarrollo de sistemas domóticos utilizando un enfoque dirigido por modelos*. PhD thesis, Tesis Doctoral, Universidad Politècnica de Cartagena, 2009.
- [4] José Manuel Huidobro and Ramón Jesús Millán Tejedor. *Manual de domótica*. Creaciones Copyright SL, 2010.
- [5] Ana Durán Anca. Instalación domótica de una vivienda unifamiliar. Master's thesis, Proyecto Fin de Carrera, Universidad Pontificia Comillas, 2009.
- [6] Stefan Junestrand, Xavier Passaret, and Daniel Vázquez. *Domótica y hogar digital*. Editorial Paraninfo, 2005.
- [7] KNX Association. url: <http://www.knx.org/es>.
- [8] R Piyare and M Tazil. Bluetooth based home automation system using cell phone. In *Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on*, pages 192–195. IEEE, 2011.
- [9] Melisa Barrera Durango, Nelson Londoño Ospina, Jorge Carvajal, and Alejandro Fonseca. Analysis and design of a low cost home automation prototype system. *Revista Facultad de Ingeniería Universidad de Antioquia*, (63):117–128, 2012.
- [10] Vongsagon Boonsawat, Jurarat Ekchamanonta, Kulwadee Bumrunghet, and Somsak Kittipiyakul. Xbee wireless sensor networks for temperature monitoring. In *the Second Conference on Application Research and Development (ECTI-CARD 2010)*, Chon Buri, Thailand, 2010.

- [11] Khusvinder Gill, Shuang-Hua Yang, Fang Yao, and Xin Lu. A zigbee-based home automation system. *Consumer Electronics, IEEE Transactions on*, 55(2):422–430, 2009.
- [12] Miguel Eduardo Carpio Miranda, Tania Alejandra Cárdenas Sanchez, and Patricia Ximena Chavez Burbano. Desarrollo e implementación de un sistema de seguridad y confort para hogares monitoreado y administrado a través de una aplicación web. 2014.
- [13] Ankit Patel, Himanshu Prajapati, Hardik Sheth, and Ekata Mehul. Home automation system using zigbee and pandaboard as a gateway (has-zp). *International Journal on Recent Trends in Engineering & Technology*, 10(2), 2014.
- [14] Andrés Oyarce, Paul Aguayo, and Eduard Martin. Guía del usuario xbee series 1. *Ingeniería MCI Ltda*, 2010.
- [15] *XBee-PRO OEM RF Modules. Product Manual v1. xAx-802.15*.
- [16] José Navarro Muñoz. Control inalámbrico basado en redes inalámbricas de sensores mediante módulos xbee. Master's thesis, Proyecto Fin de Carrera, Universidad Politècnica de Cartagena, 2013.
- [17] Robert Faludi. *Building wireless sensor networks: with ZigBee, XBee, arduino, and processing*. "Reilly Media, Inc.", 2010.
- [18] B Ivanov, O Zhelondz, L Borodulkin, and H Ruser. Distributed smart sensor system for indoor climate monitoring. In *KONNEX Scientific Conf., Mnchen*, pages 10–11, 2002.
- [19] Guangming Song, Fei Ding, Weijuan Zhang, and Aiguo Song. A wireless power outlet system for smart homes. *Consumer Electronics, IEEE Transactions on*, 54(4):1688–1691, 2008.
- [20] Gerald Coley. *BeagleBone Rev A6 System Reference Manual*, 2012.
- [21] Adolfo Antonio Fernández Trabanco. *Manuales del programador*, 2005.
- [22] Instituto Nacional de Tecnologías Educativas y Formación del profesorado. url: <http://www.ite.educacion.es/formacion/materiales/85/cd/linux/indice.htm>.
- [23] Goran Horvat, Damir Sostaric, and Drago Zagar. Power consumption and rf propagation analysis on zigbee xbee modules for atpc. In *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on*, pages 222–226. IEEE, 2012.
- [24] Jorge Dignani. *Análisis del protocolo ZigBee*. PhD thesis, Facultad de Informática, 2012.

Parte II

Apéndices

Apéndice A

IEEE 802.15.4 y Protocolo ZigBee

El estándar IEEE 802.15.4 y por ende ZigBee, operan habitualmente en la banda de frecuencia de 2.4 GHz, aunque también permiten trabajar en las bandas de 868 y 915 MHz. La primera es usada en todo el mundo, mientras que las otras se utilizan en Europa y Norteamérica de forma ocasional. En todas ellas se utiliza DSSS (Direct Sequence Spread Spectrum) como método de codificación de canal, lo que permite, a diferencia de Bluetooth, realizar la comunicación a través de una única frecuencia (canal). Las bandas 868MHz y 915MHz soportan uno o diez canales respectivamente, mientras que en la banda de 2.4 GHz se dispone de 16 canales con valores asignados del 11 hasta el 26 (ver figura A.1).

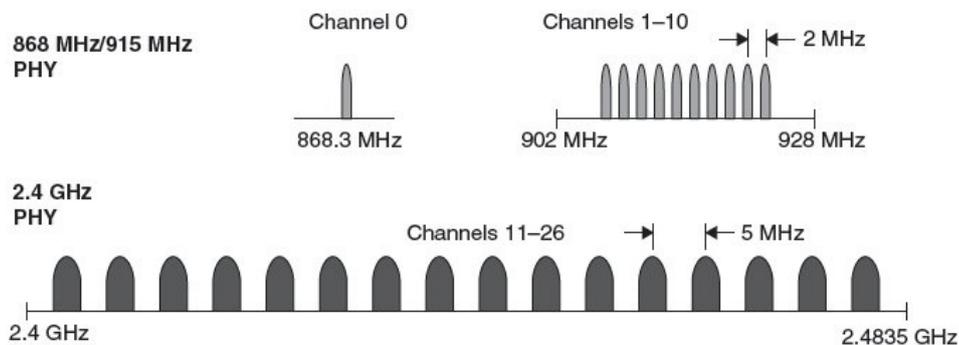


Figura A.1: Canales del estándar IEEE 802.15.4

Trabajando a 2.4GHz se consigue un ancho de banda más grande y más canales, pero tiene el inconveniente de encontrar más interferencias al solaparse con las frecuencias utilizadas por otras tecnologías. Una ventaja añadida es que en esta banda de frecuencia las antenas utilizadas son de menor tamaño. En la banda de 2.4GHz se emplea O-QPSK (Offset Quadrature Phase Shift) como técnica de modulación, mientras que las otras bandas usan BPSK (Binary Phase Shift Keying).

En este proyecto se trabajará en la banda de 2.4GHz, en la cual, la frecuencia central de cada canal viene dada por:

$$F_c[\text{MHz}] = 2405 + 5 * (N^{\circ}\text{canal} - 11)$$

A.1. Capas

ZigBee es un protocolo de comunicaciones dividido en capas. Se basa en el modelo de referencia OSI y sus 7 capas, aunque en este caso sólo utiliza 4 para simplificar la red y su arquitectura (ver figura A.2) [24].

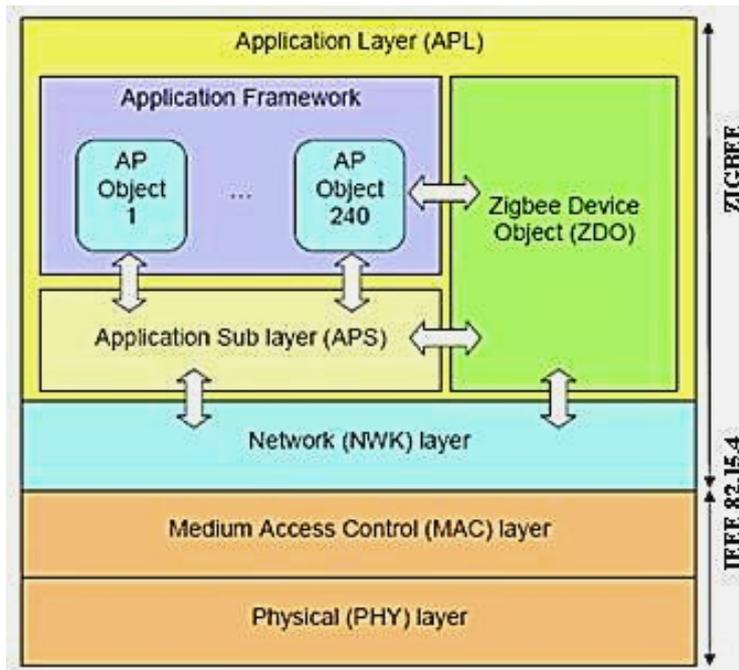


Figura A.2: Capas de IEEE 802.15.4 y ZigBee

Las 2 capas inferiores, la capa física (PHY) y la capa de acceso al medio (MAC), están definidas por el estándar IEEE 802.15.4.

ZigBee añade las capas de nivel superior que el IEEE 802.15.4 no cubre, y adopta de dicho estándar las dos capas inferiores, la capa física y la capa de nivel de enlace (MAC). Este es el motivo por el cual se dice que la tecnología ZigBee se basa en el estándar IEEE 802.15.4 o que es una ampliación de él. Las dos capas propiamente definidas por ZigBee son la capa de red y la capa de aplicación, las cuales son más específicas y se encargan de aspectos como la forma de la red, consideraciones de seguridad, tipos de dispositivos, etc. Todas las capas se comunican con sus vecinas a través de SAPs (Service Access Point). Existen SAPs para la comunicación de los datos y SAPs para las órdenes de control.

Así pues, ZigBee y el estándar IEEE 802.15.4 se complementan entre sí para pro-

porcionar una pila completa de protocolos que permita establecer una comunicación entre varios nodos. Ambos están tan relacionados, que en ocasiones se habla de tecnología ZigBee aunque se debería hablar únicamente de estándar IEEE 802.15.4.

En las 3 capas inferiores se distinguen dos zonas diferentes. Por un lado existe el área de datos, que se encarga de la entrega de los datos (PDUs) a las capas vecinas, ya sea en su recepción o transmisión y, por otro lado, se encuentra el área de control o manejo, que es la encargada de comunicarse con sus capas vecinas para transmitir o recibir comandos, indicaciones, confirmaciones, etc.

En la figura A.3 se puede observar el camino que siguen los datos desde la capa de aplicación del dispositivo fuente a la de destino. Los datos son modificados al atravesar las diferentes capas hasta que finalmente son transmitidos. Durante la recepción, los datos recorren las capas en orden inverso.

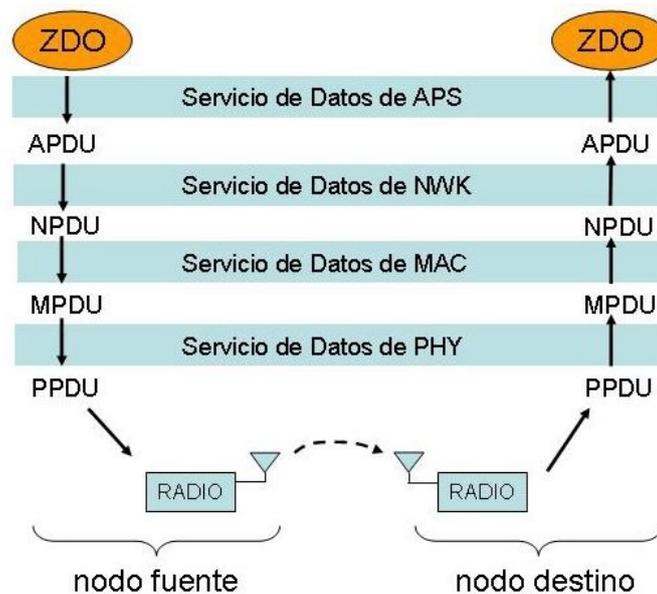


Figura A.3: Recorrido de los datos a través de las capas

La capa Física (PHY)

Al ser la capa más cercana al hardware, es la responsable de la recepción y transmisión de los paquetes. Opera bajo el estándar IEEE 802.15.4 y proporciona una interfaz a la capa superior de la pila de protocolos (MAC).

La capa física tiene como funciones principales: definir aspectos como la sensibilidad del receptor o la potencia del transmisor, indicar la calidad del enlace (Link Quality Indicator), detectar la energía del canal de comunicaciones y evaluar el canal libre (Clear Channel Assesement).

El estándar IEEE 802.15.4 y ZigBee emplean “primitivas” para indicar los servicios que una capa le proporciona a la superior. Las unidades de control y manejo de cada capa son las encargadas de desempeñar estas funciones y utilizan un SAP para requerir o facilitar dicho servicio.

La capa de Enlace (MAC)

Es la responsable de garantizar la comunicación entre un nodo y los que están conectados a él. Proporciona una interfaz entre la capa física y la capa de red. Al igual que la capa física, trabaja bajo el protocolo IEEE 802.15.4.

Esta capa está compuesta por la unidad de datos (MCPS) y la entidad que se encarga del control (MLME). La MCPS interactúa con las unidades de datos de la capa física y de red mediante el PD-SAP y el MCPS-SAP respectivamente, mientras que la MLME lo hace a través de PLME-SAP y de MLME-SAP (ver figura A.4).

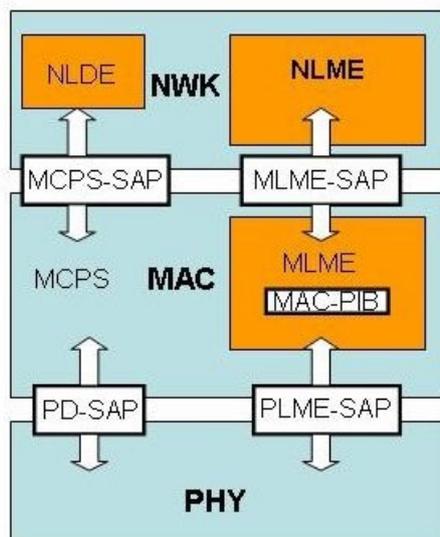


Figura A.4: Relación de la capa MAC con sus vecinas

En escenarios de baja latencia, permite la posibilidad de trabajar usando tramas

especiales denominadas tramas de baliza, que permiten disponer de ranuras de tiempo garantizadas (GTS), es decir, se garantiza a cada dispositivo un período de tiempo libre de colisiones, en el que no se requiere el uso de CSMA-CA.

Puede haber tramas de tipo dato, de baliza, de comando y de confirmación (ACK).

La capa MAC tiene entre sus funciones:

- Gestionar la conexión y desconexión de los nodos asociados a un coordinador.
- Gestionar el mecanismo de GTS y la sincronización mediante las balizas.
- El uso del algoritmo CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance) como método de acceso al canal de comunicaciones.

La capa de Red (NWK)

Esta capa, definida por el estándar ZigBee, es la encargada de realizar el control y gestión de las redes. Proporciona una interfaz entre la capa de red y la capa de aplicación.

Tiene como funciones:

- El enrutamiento (routing), que establece el camino que seguirá el mensaje hasta el dispositivo final.
- Asociar un dispositivo a una red, para lo cual usa de los servicios de la capa MAC.
- Inicializar la red en el caso de ser el coordinador y asignar las direcciones a los dispositivos que forman parte de ella.
- Garantizar la comunicación más allá de los nodos vecinos, si la topología de red así lo requiere. Este proceso se realiza mediante saltos en los que el paquete es retransmitido.

La unidad de datos es la NLDE y la de control o manejo la NLME, que se comunican con los homólogos de las capas adyacentes por SAPs (ver figura A.5).

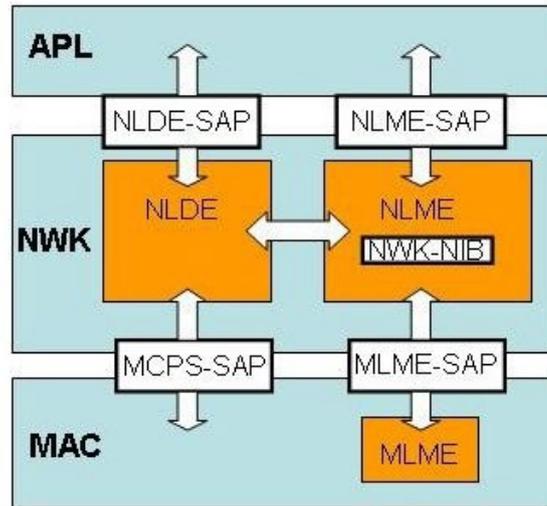


Figura A.5: Relación de la capa de Red con sus vecinas

La capa de Aplicación (APL)

Esta capa tiene como misión soportar las diferentes aplicaciones que debe desempeñar cada dispositivo.

Dentro de esta capa existen ‘perfiles’ que caracterizan el formato de los mensajes, tipos de dispositivos y acciones que se emplearán en determinadas aplicaciones. Estos perfiles pueden ser públicos si ya están especificados por ZigBee Alliance o privados, si son creados por un usuario para una aplicación específica no cubierta mediante perfiles públicos.

APL está compuesta por la subcapa de soporte de aplicación (APS), por un grupo de objetos de aplicación (APOs incluidos en la Framework de Aplicación) y por un objeto especial, denominado ZDO (ZigBee Device Object), que ofrece servicios al resto de aplicaciones (ver figura A.2).

Estos objetos tienen como función simplificar el manejo de la red a las aplicaciones de los usuarios. La APS se encarga de la transferencia de datos entre la capa NWK y los objetos de aplicación.

A.2. Seguridad

Como otras redes inalámbricas, ZigBee es vulnerable en aspectos de seguridad, ya que cualquier persona tiene acceso al medio de transmisión. La vulnerabilidad puede no ser un problema en función de las aplicaciones utilizadas.

La tecnología ZigBee permite la posibilidad de implementar mecanismos de seguridad para proteger las comunicaciones. Sin embargo, hay que tener en cuenta que ZigBee es una tecnología orientada a bajo consumo, por lo que la capacidad de cómputo de sus dispositivos es bastante limitada y, por lo tanto, los mecanismos de seguridad no podrán ser tan avanzados como con en otros protocolos de comunicaciones. Además, los paquetes de datos tienen unas dimensiones reducidas, haciendo que no tenga mucho sentido dedicar gran cantidad de bits a la seguridad.

La seguridad de las comunicaciones se puede dividir en dos ámbitos, la confidencialidad y la autenticidad de los datos.

Para resolver el problema de la confidencialidad, ZigBee soporta el algoritmo AES (Advance Encryption Standard) para la encriptación de los datos, los cuales son cifrados a partir de una clave (cadena de bits) para que sólo el destinatario sea capaz de obtener la información transmitida.

Para garantizar la autenticidad de los datos, el transmisor incluye en el mensaje un código especial denominado MIC (Message Integrity Code). El receptor calcula dicho código y si éste coincide con el que viene en el mensaje, se considera auténtico. El MIC se genera mediante el protocolo CCM* (enhanced Counter with CBC-MAC), que se usa conjuntamente con AES.

Apéndice B

Modos de operación de un Xbee

Los módulos XBee funcionan en cinco modos de operación según la tarea que estén realizando en cada momento. Los cinco estados posibles son Transmitiendo, Recibiendo, Modo Sleep, entrada de comandos y modo de reposo (Idle Mode). Éste último es el modo principal al que siempre se vuelve después de acabar las tareas propias de los otros estados (ver figura B.1).

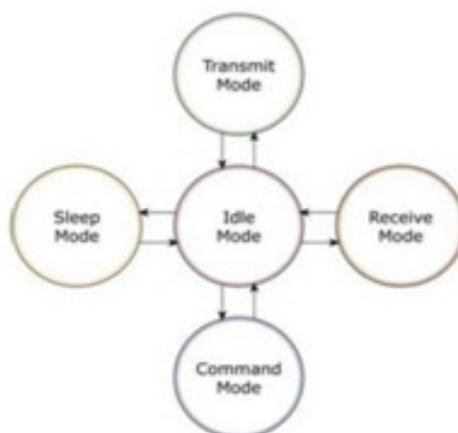


Figura B.1: Modos de operación del módulo XBee

A continuación se mencionaran las principales tareas de cada de estado de funcionamiento.

Modo Recibir (Receive Mode)

Se encuentra en este estado cuando le llega algún paquete de Radiofrecuencia a través de la antena. El consumo en los módulos Serie1 PRO, como los seleccionados, es de alrededor de 55mA.

Modo Transmitir (Transmit Mode)

Se encuentra en este modo cuando se manda información serial al buffer del pin 3 (DIN) para que sea transmitida. El consumo en los módulos PRO es de unos 215 mA (según especificaciones).

La información se puede transmitir de manera Directa o Indirecta. Con el primer método la información es enviada inmediatamente a la dirección destino. En la manera Indirecta la información es retenida durante un intervalo de tiempo y no es enviada

hasta que el dispositivo destino la solicita. Esta manera de transmisión se emplea habitualmente cuando se necesita asegurar la recepción de los datos por parte de dispositivos finales que pasan gran parte de su tiempo en modo sleep.

Modo de Comando (Command Mode)

Se encuentra en este estado cuando se está procediendo a leer o escribir los parámetros de configuración del módulo. Todos los Xbee soportan dos tipos de modo comando diferente, el modo AT y el modo API. Aunque ambos son posibles, AT es exclusivo de este estado por lo que se hablará de él a continuación. EL modo API permite no sólo la configuración de parámetros, sino también la transmisión o recepción por lo que es explicado con mayor detalle en la sección 3.1.5.2.

En el modo AT es necesario grabar (o leer) los comandos de configuración mediante algún dispositivo conectado al puerto serie UART del Xbee, es decir, se puede usar cualquier microcontrolador que maneje UART o bien, interfaces seriales USB que conecten al ordenador. En este modo AT, cuando los caracteres sean recibidos a través del puerto serial serán interpretados como comandos (AT) y no serán transmitidos por la antena.

Para acceder a este modo se deben seguir los siguientes pasos:

1. Esperar el tiempo dado por el parámetro GT (Guard Time), que por defecto es de un segundo.
2. Seguidamente escribir tres caracteres (+++)
3. Esperar otro intervalo de tiempo fijado por GT, hasta recibir un OK.

Una vez en el modo AT, se tiene que seguir una sintaxis determinada para poder enviar comandos. Se debe enviar el prefijo AT, seguido del comando en ASCII y del valor que se desea asignar a ese comando. Finalmente, se enviará el carácter Carrier Return si se maneja a través de un microcontrolador o se pulsará ENTER si se maneja a través de una interfaz gráfica (X-CTU o Hyperterminal). En el caso de la lectura de un parámetro, se sigue el mismo formato pero sin introducir el valor.

Para que los comandos modificados conserven su valor tras un reinicio del dispositivo, se debe introducir el comando WR que permite grabar los nuevos parámetros en la memoria no volátil del módulo XBee.

Para salir del modo AT se puede esperar el tiempo establecido en el parámetro CT (Command mode Timeout) sin introducir ningún carácter o ingresar el comando CN.

Modo de Bajo Consumo (Sleep Mode)

El Modo Sleep permite que el XBee se encuentre en un estado de muy bajo consumo cuando no está siendo utilizado. Mientras permanecen en este estado, los módulos seleccionados de la Serie 1 tienen un consumo de corriente inferior a 310 uA.

El módulo puede entrar en este estado al recibir una señal en el pin 9 (SLEEP_RQ) o porque ha sido configurado así a través del software. Dispone de tres modos de suspensión diferentes que se configuran mediante el parámetro SM.

- Hibernación (SM=1). Es el modo que tiene un menor consumo y que requiere de más tiempo para entrar o salir del estado de Sueño. Se controla mediante el pin 9; al subir su estado lógico, el módulo Xbee entrará en el Modo Sueño, mientras que al bajar su estado lógico saldrá del Modo Sueño y estará listo para recibir o enviar datos.
- PinDoze (SM=2). Funciona de la misma manera que el anterior, pero presenta un tiempo de respuesta menor y un consumo mayor.
- Sueño cíclico (SM=4,5). El módulo está configurado para efectuar ciclos de sueño, despertando una vez por ciclo para revisar si existen datos destinados a él en el coordinador. Los parámetros ST (Time before Sleep) y SP (Cyclic Sleep Period) definen cuándo y cómo entrar en este modo.

La posibilidad de poner los módulos XBee en Modo Sleep permite una larga vida (incluso de años) a las baterías que alimentan a los dispositivos finales de la red.

Modo Reposo (Idle Mode)

Se encuentra en este estado siempre que el módulo esté activo, pero no se encuentre en ninguno de los otros modos. Es decir, si no está ni transmitiendo, ni recibiendo, ni en modo comando. El consumo de corriente es de unos 55 mA en los módulos Serie1 PRO.

Apéndice C

Comandos API

C.1. Comando AT

Los comandos AT son los mismos que se utilizan en el modo comando y permiten cambiar los parámetros de configuración u obtener el valor de dichos parámetros. Durante este proyecto este tipo de tramas apenas se han empleado ya que se ha utilizado el software de configuración X-CTU (ver sección 3.1.7.2) que es más cómodo y visual.

De la misma manera que otros tipos de tramas, los comandos AT comienzan con el byte 0x7E al que siguen los dos bytes que determinan la longitud. Los datos se encuentran en la trama de datos, justo a continuación de estos bytes.

Este tipo de comandos se identifica mediante 0x08. A continuación se encuentra la trama ID, que es, simplemente, un número correlativo que se adjunta al comando (ver figura C.1). La respuesta, identificada mediante 0x88, será etiquetada con el mismo ID, lo que nos permitirá saber si el comando se recibió correctamente o si hubo algún error.

Los dos bytes que siguen al identificador de frame contienen (en hexadecimal) el comando AT propiamente dicho. Finalmente puede haber más bytes en los que figura el valor. Como en todas las tramas API, checksum es el último byte.

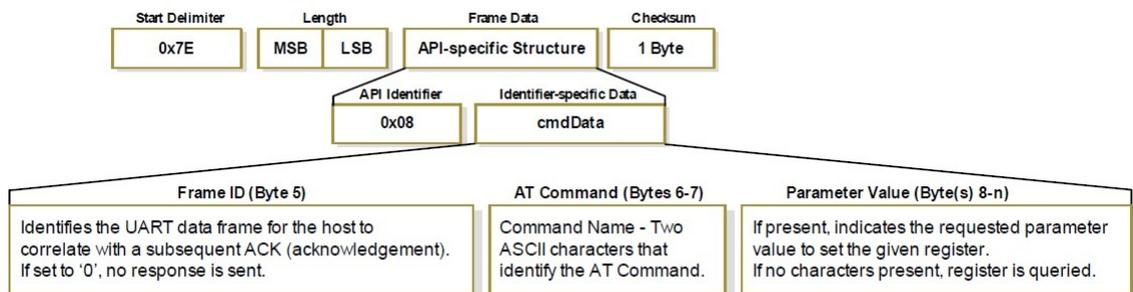


Figura C.1: Estructura trama comando AT

C.2. Transmisión de datos y recepción de datos

Estos tipos de comandos API no han sido empleados, pero pueden ser de gran utilidad para trabajos futuros ya que permiten enviar una serie de datos a un módulo Xbee

que esté conectado a un microcontrolador. A partir de estos datos el MCU podrá realizar la misión específica que tenga asignada.

Habrán dos tipos de comandos API orientados a esta función, con la única diferencia de que uno utiliza la dirección de 64 bits que tiene asignado cada Xbee de fábrica (API ID=0x00) y el otro empleará la dirección de 16 bits que se le asigna dentro una red (API ID=0x01). De la misma manera, las tramas API con los datos recibidos pueden usar direcciones de 64 bits (API_ID=0x80) o de 16 bits (API_ID=0x81).

En la figura C.2 se define la estructura que debe tener un frame que entra por la UART para transmitir datos mediante direcciones de 16 bits. Dentro de la estructura específica de la trama API se puede observar que el primer byte, como siempre, contiene el identificador del tipo de comando API, a continuación aparece un byte para identificar la trama, seguido de dos bytes que contienen la dirección destino de 16 bits y de otro byte para opciones. Por último se añaden los bytes con los datos deseados para ser transmitidos al MCU remoto.

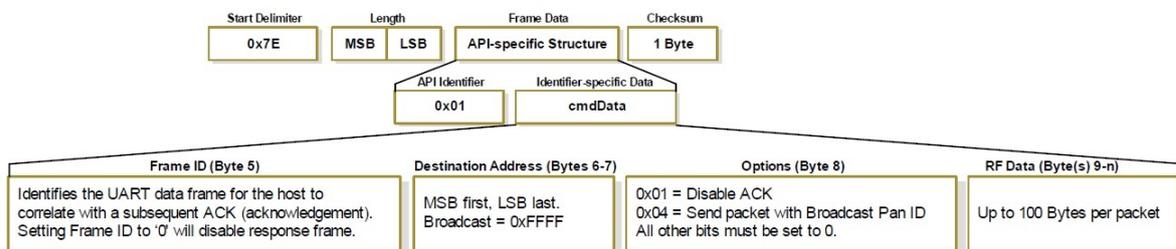


Figura C.2: Estructura API de Transmisión de Datos

En la figura C.3 se define la estructura que tiene la trama que sale por la UART del Xbee una vez los datos han sido recibidos inalámbricamente. Dentro de la estructura específica de la trama API, a parte del identificador de comandos, los primeros bytes corresponden con la dirección de origen de 16 bits del Xbee que envió el paquete. El siguiente byte (RSSI) indica la potencia de la señal recibida en dBm. Finalmente hay un byte de opciones y el resto son los datos.

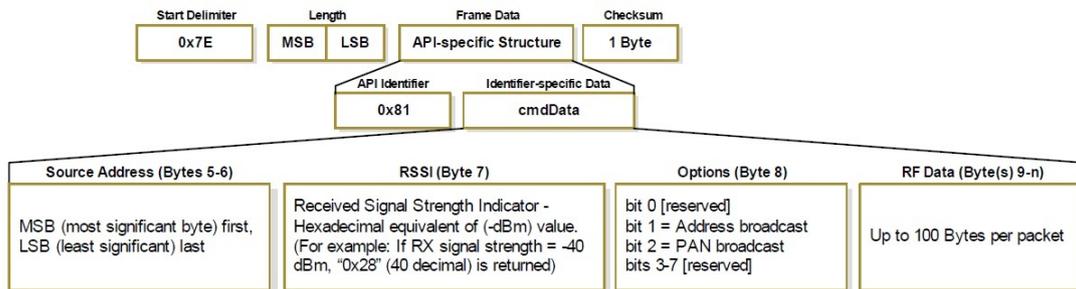


Figura C.3: Estructura API de Recepción de Datos

Las tramas de los comandos API que utilizan direcciones de 64 bits están estructuradas con los mismos campos.

Existe otro tipo de comando API (API_ID=0x89) que saldría por la UART del módulo que ha enviado inalámbricamente los datos, esta trama tiene como finalidad informar al usuario de resultado de la transmisión de datos (ver figura C.4).

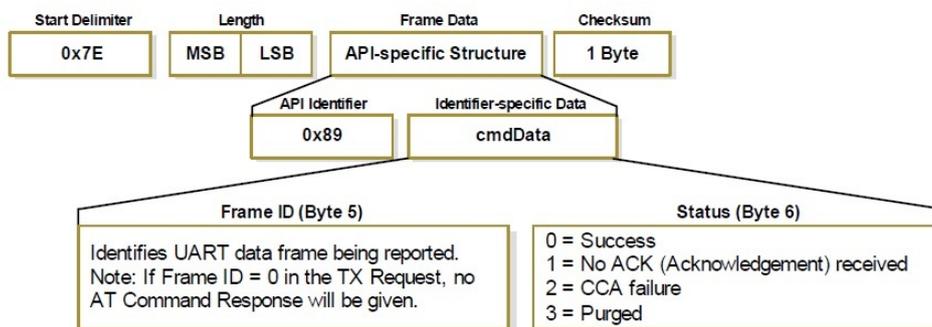


Figura C.4: Estructura API de Respuesta de Resultado de la Transmisión

Apéndice D

Configuración de los módulos de comunicación XBee

D.1. XBee coordinador

| | |
|---|--|
| xbp24_15_4_10ec.mxi FE 0 241 10EC 0 | |
| [A]CH=C [A]ID=9032 [A]DH=0 [A]DL=1 [A]MY=64 [A]MM=0 [A]RR=0 [A]RN=0 [A]NT=19 [A]NO=0 [A]CE=1 [A]SC=1FFE [A]SD=4 [A]A1=0 [A]A2=0 [A]EE=0 [A]NI= [A]PL=4 [A]CA=2C [A]SM=0 [A]ST=1388 [A]SP=0 [A]DP=3E8 [A]SO=0 [A]BD=3 [A]NB=0 [A]RO=3 [A]AP=1 [A]D8=0 [A]D7=1 | [A]D6=0 [A]D5=0 [A]D4=0 [A]D3=0 [A]D2=0 [A]D1=0 [A]D0=0 [A]PR=FF [A]IU=1 [A]IT=1 [A]IC=0 [A]IR=0 [A]P0=1 [A]P1=0 [A]PT=FF [A]RP=28 [A]IA=FFFFFFFFFFFFFFFF [A]T0=FF [A]T1=FF [A]T2=FF [A]T3=FF [A]T4=FF [A]T5=FF [A]T6=FF [A]T7=FF [A]DD=10000 [A]CT=64 [A]GT=3E8 [A]CC=2B - |

Tabla D.1: Parámetros de configuración XBee-unidad central

D.2. XBee dispositivo final

| | |
|---|--|
| xbp24_15_4_10ec.mxi FE 0 241 10EC 0 | |
| [A]CH=C [A]ID=9032 [A]DH=0 [A]DL=64 [A]MY=1 [A]MM=0 [A]RR=0 [A]RN=0 [A]NT=19 [A]NO=0 [A]CE=0 [A]SC=1FFE [A]SD=4 [A]A1=0 [A]A2=0 [A]EE=0 [A]NI= [A]PL=4 [A]CA=2C [A]SM=0 [A]ST=1388 [A]SP=0 [A]DP=3E8 [A]SO=0 [A]BD=3 [A]NB=0 [A]RO=3 [A]AP=1 [A]D8=0 [A]D7=1 | [A]D6=0 [A]D5=4 [A]D4=4 [A]D3=4 [A]D2=2 [A]D1=2 [A]D0=2 [A]PR=FF [A]IU=1 [A]IT=1 [A]IC=0 [A]IR=0 [A]P0=1 [A]P1=0 [A]PT=FF [A]RP=28 [A]IA=FFFFFFFFFFFFFFFF [A]T0=FF [A]T1=FF [A]T2=FF [A]T3=FF [A]T4=FF [A]T5=FF [A]T6=FF [A]T7=FF [A]DD=4 [A]CT=64 [A]GT=3E8 [A]CC=2B - |

Tabla D.2: Parámetros de configuración XBee-unidad remota

Apéndice E

Especificaciones módulo XBee

| Specification | XBee | XBee-PRO |
|--|---|---|
| Performance | | |
| Indoor/Urban Range | up to 100 ft. (30 m) | Up to 300' (100 m) |
| Outdoor RF line-of-sight Range | up to 300 ft. (100 m) | Up to 1 mile (1500 m) |
| Transmit Power Output (software selectable) | 1mW (0 dBm) | 60 mW (18 dBm) conducted, 100 mW (20 dBm) EIRP* |
| RF Data Rate | 250,000 bps | 250,000 bps |
| Serial Interface Data Rate (software selectable) | 1200 - 115200 bps (non-standard baud rates also supported) | 1200 - 115200 bps (non-standard baud rates also supported) |
| Receiver Sensitivity | -92 dBm (1% packet error rate) | -100 dBm (1% packet error rate) |
| Power Requirements | | |
| Supply Voltage | 2.8 – 3.4 V | 2.8 – 3.4 V |
| Transmit Current (typical) | 45mA (@ 3.3 V) | If PL=0 (10dBm): 137mA(@3.3V), 139mA(@3.0V) PL=1 (12dBm): 155mA (@3.3V), 153mA(@3.0V) PL=2 (14dBm): 170mA (@3.3V), 171mA(@3.0V) PL=3 (16dBm): 188mA (@3.3V), 195mA(@3.0V) PL=4 (18dBm): 215mA (@3.3V), 227mA(@3.0V) |
| Idle / Receive Current (typical) | 50mA (@ 3.3 V) | 55mA (@ 3.3 V) |
| Power-down Current | < 10 μ A | < 10 μ A |
| General | | |
| Operating Frequency | ISM 2.4 GHz | ISM 2.4 GHz |
| Dimensions | 0.960" x 1.087" (2.438cm x 2.761cm) | 0.960" x 1.297" (2.438cm x 3.294cm) |
| Operating Temperature | -40 to 85° C (industrial) | -40 to 85° C (industrial) |
| Antenna Options | Integrated Whip, Chip or U.FL Connector | Integrated Whip, Chip or U.FL Connector |
| Networking & Security | | |
| Supported Network Topologies | Point-to-point, Point-to-multipoint & Peer-to-peer | |
| Number of Channels (software selectable) | 16 Direct Sequence Channels | 12 Direct Sequence Channels |
| Addressing Options | PAN ID, Channel and Addresses | PAN ID, Channel and Addresses |
| Agency Approvals | | |
| United States (FCC Part 15.247) | OUR-XBEE | OUR-XBEEPRO |
| Industry Canada (IC) | 4214A XBEE | 4214A XBEEPRO |
| Europe (CE) | ETSI | ETSI (Max. 10 dBm transmit power output)* |
| Japan | n/a | 005NYCA0378 (Max. 10 dBm transmit power output)** |

Tabla E.1: Tabla comparativa XBee vs XBee PRO

Apéndice F

Esquemáticos

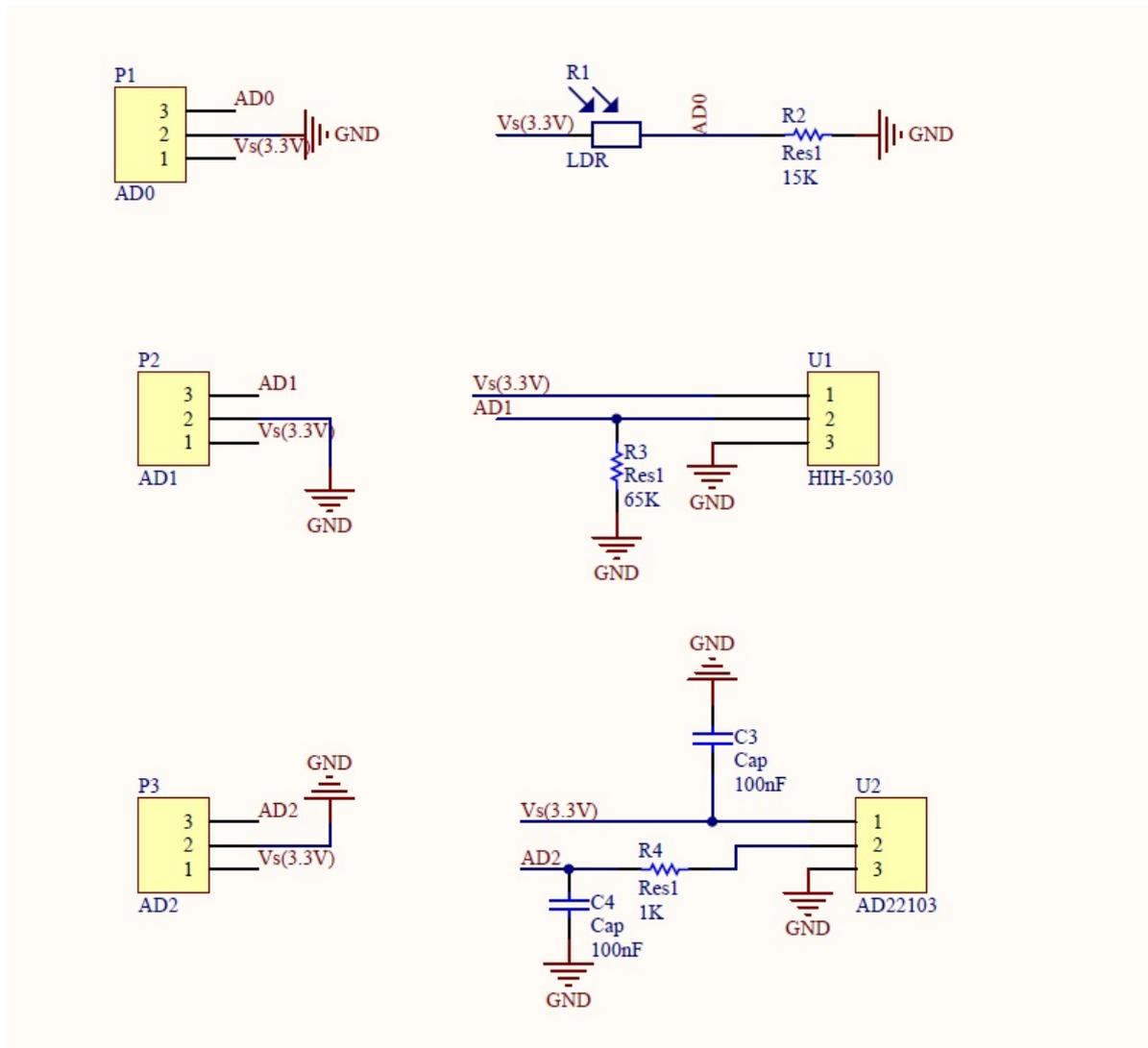


Figura F.1: Esquemático del PCB de los sensores

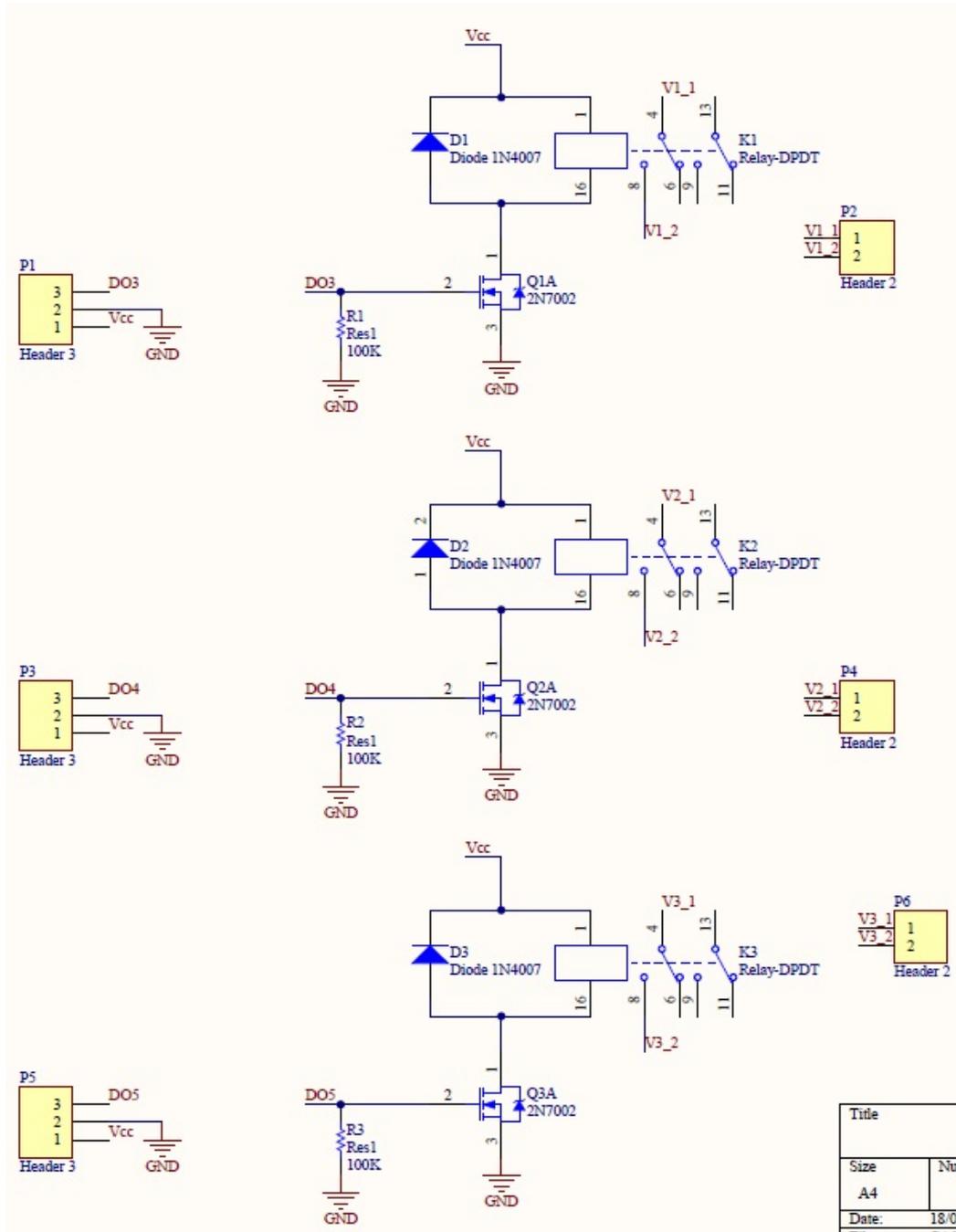


Figura F.2: Esquemático del PCB de los actuadores

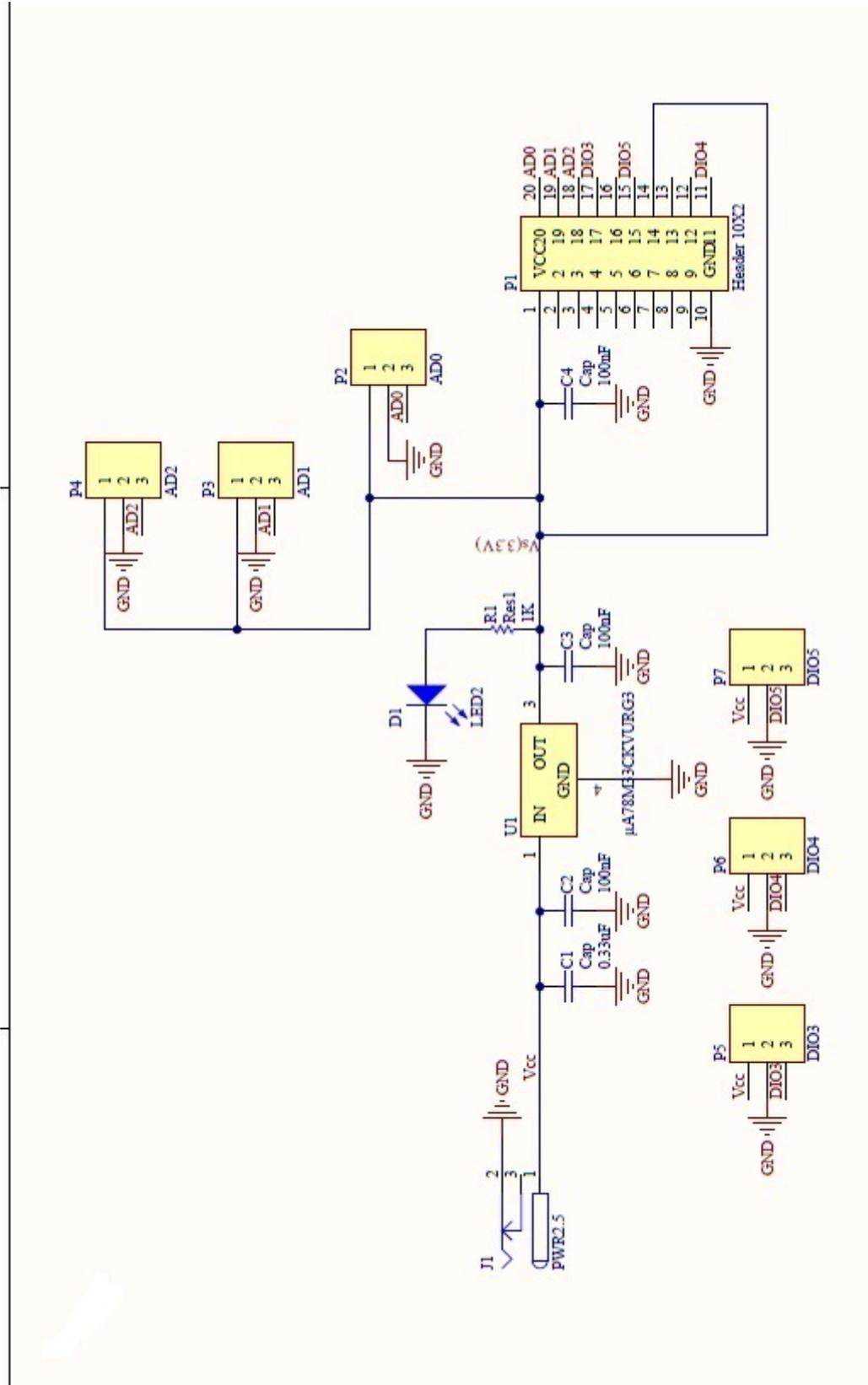


Figura F.3: Esquemático del PCB que incorpora el módulo XBee

Apéndice G

Diagrama de secuencia

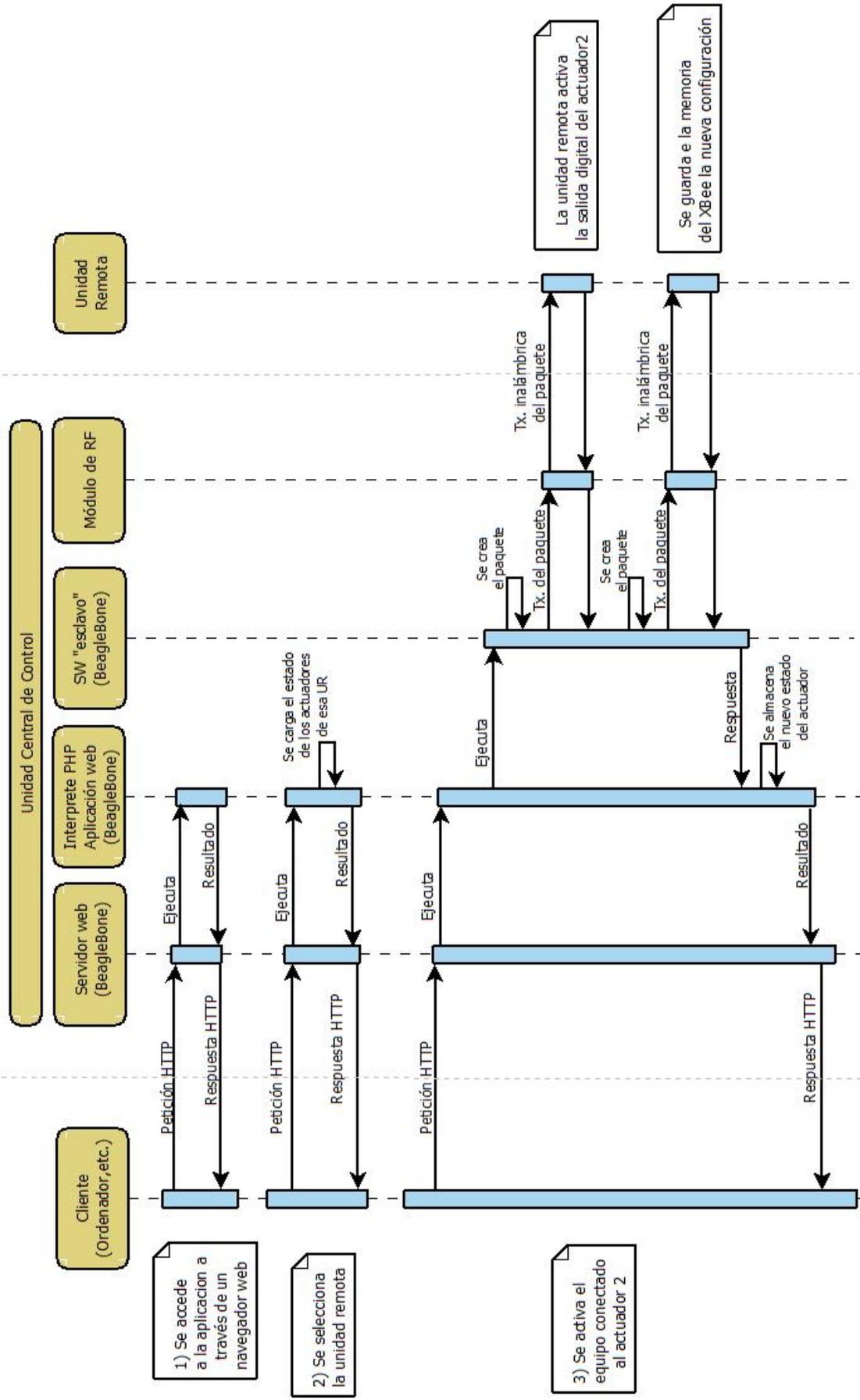


Figura G.1: Diagrama de secuencia (1ª Parte)

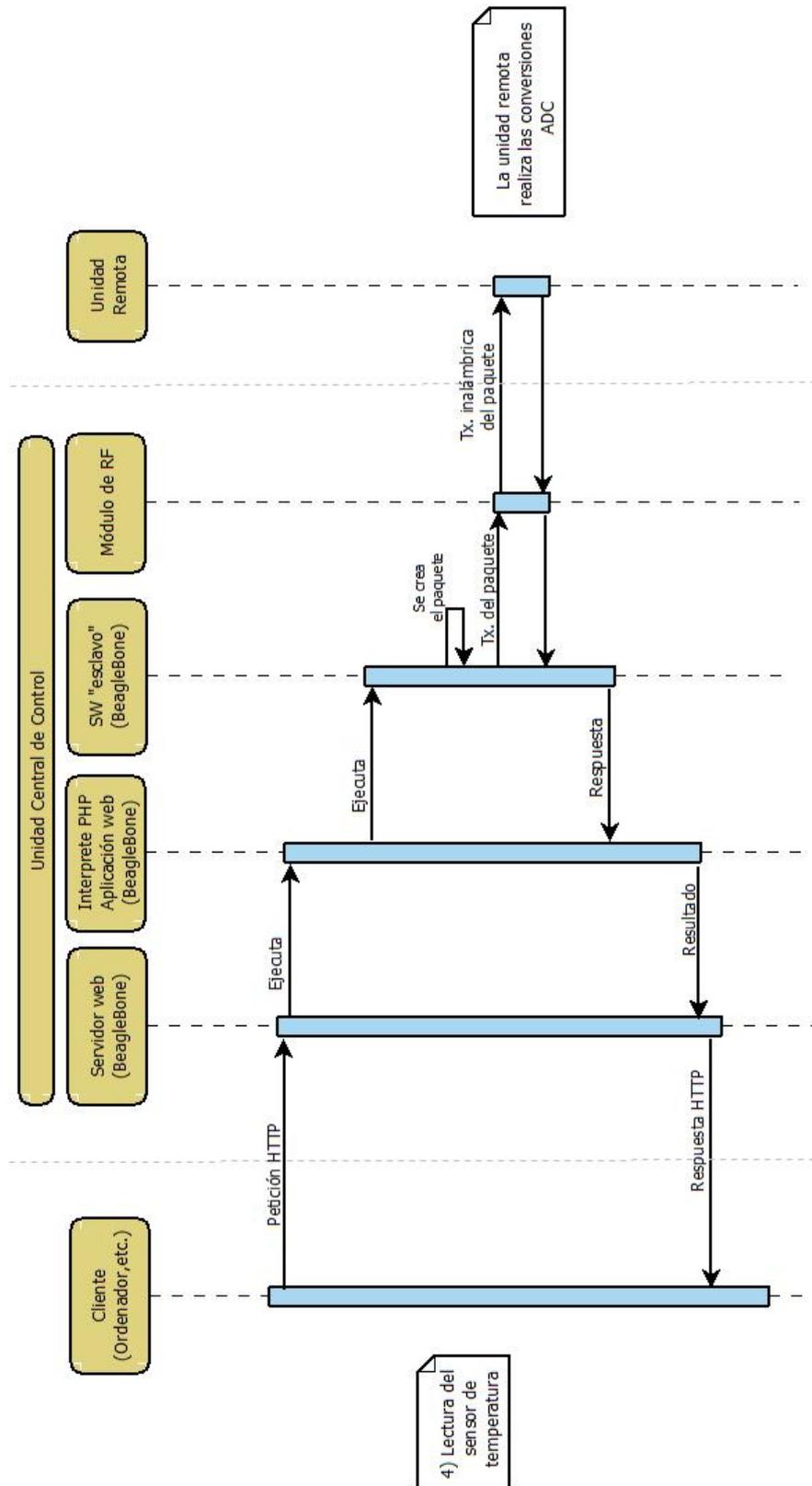


Figura G.2: Diagrama de secuencia (2ª Parte)

Apéndice H

Presupuesto

1. Ejecución Material

- Compra de ordenador personal 600€
- Compra de licencia del software de diseño 1200 €
- Tarjeta BeagleBone 80 €
- Kit de desarrollo ZigBee y módulos XBee 209 €
- Fabricación electrónica Unidad Remota 42 €
- Material de oficina 150 €
- Total de ejecución material 2281 €

2. Gastos generales

- 16 % sobre Ejecución Material 364.96 €

3. Beneficio Industrial

- 6 % sobre Ejecución Material 136.86 €

4. Honorarios Proyecto

- 640 horas a 15 € / hora 9600 €

5. Material fungible

- Gastos de impresión 60 €
- Encuadernación 200 €

6. Subtotal del presupuesto

- Subtotal Presupuesto 12642.82 €

7. I.V.A. aplicable

- 21 % Subtotal Presupuesto 2654.99 €

8. Total presupuesto

-
- **Total Presupuesto 15297.81 €**

Madrid, Octubre 2014

El Ingeniero Jefe de Proyecto

Fdo.: Mario Rodríguez Cerezo

Ingeniero de Telecomunicación

Apéndice I

Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un <<Sistema de control remoto para aplicaciones domóticas a través de Internet>>. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego. Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.
2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.
3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.
4. La obra se realizará bajo la dirección técnica de un Ingeniero de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.
5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.
6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.
8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.
9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.
10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.
11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo,

sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partidaalzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.
13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.
14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.
15. La garantía definitiva será del 4 % del presupuesto y la provisional del 2 %.
16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.
17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.
18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.
19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.
20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean

oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.
22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.
23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.
3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.
4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.
5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.
6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.
7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.
8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.
9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.
10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.
11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.