Escuela de Ingeniería y Arquitectura

Universidad Zaragoza

Ingeniería de Telecomunicación

Proyecto Fin de Carrera

Exploiting Wireless Sensors

School of Information and Communication Technology
KTH Royal Institute of Technology
Stockholm, Sweden

Autor: Francisco Javier Sánchez Galisteo
Director: Gerald Q. Maguire Jr.
Ponente: Pilar Molina Gaudó

30 de Marzo, 2012

Agradecimientos

A Gerald Q. Maguire Jr. y a Pilar Molina por su colaboración en el desarrollo de este proyecto.

A mis compañeros y amigos por hacerme disfrutar todos estos años. Mención especial a Cristina por el apoyo que siempre me ha dado.

A mis padres, a mi hermano y en especial a mi abuelo, por su ayuda y su confianza y por enseñarme, entre otras muchas cosas, el valor del esfuerzo y la constancia.

Resumen

El gran interés del ser humano en conocer y controlar todo lo que le rodea ha motivado la gran proliferación de sistemas con elementos sensores. Asimismo, la evolución de las tecnologías inalámbricas ha facilitado su implantación. Este proyecto surge de la idea de explotar el gran potencial que ofrecen los sensores inalámbricos y así ampliar y extender su uso.

El objetivo de este trabajo es diseñar el hardware de un *gateway* (o pasarela) que permita captar los datos provenientes de distintos tipos de sensores inalámbricos y transmitirlos a través de una red LAN para su posterior tratamiento. Se comienza por un análisis del estado del arte de las tecnologías relativas a redes de sensores, para después estudiar y descifrar el protocolo propietario que utiliza un determinado sensor comercial. A continuación, se diseña el hardware del *gateway*, que debe decodificar la información útil del sensor estudiado y proporcionársela a otro dispositivo conectado en una red. Este gateway consta básicamente de un microcontrolador, una interfaz Ethernet y un módulo de RF. Finalmente, se estudia cómo adaptar este diseño para aprovechar otro tipo de sensores, ampliando su utilidad.

La finalidad última de este desarrollo es la de dar soporte a futuros trabajos que puedan sacar provecho de este proyecto. Este *gateway* pretende ser un puente entre la información que nos rodea y una red o dispositivo inteligente dónde puedan desarrollarse diversas aplicaciones y servicios.

Índice general

Índice general vii
Lista de Figuras ix
Lista de Tablas xi
Lista de Acrónimos y Abreviaciones xiii
1 Introducción 1
1.1 Motivación 1
1.2 Descripción del problema 1
1.3 Objetivos del trabajo 2
1.4 Contexto 3
1.5 Estructura 3
2 Estado del arte 5
2.1 Redes inalámbricas de sensores 5
2.2 Tecnologías inalámbricas 6
2.3 Banda ISM 10
2.3.1 Dispositivos de corto alcance operando en 868MHz 11
2.4 Power over Ethernet 12
2.5 Artículos relacionados 14
3 Herramientas 17
3.1 Estación meteorológica por radio control y transmisor de temperatura 17
3.2 USRP: Universal Software Radio Peripheral 18
3.3 GNU Radio 18
4 Decodificación del protocolo propietario 21
4.1 Captura de datos 21
4.2 Decodificación de la señal 22
4.3 Análisis de los datos 25
4.3.1 Campo de temperatura 26
4.3.2 Campo de identificador 28
4.3.3 Código CRC-8 28
4.3.4 Resto de la trama 28
5 Gateway 31
5.1 Componentes principales 31
5.1.1 Microcontrolador MSP430 31

5.1.2 Receptor de RF 32
5.1.3 Controlador Ethernet 33
5.1.4 Alimentación mediante PoE y Regulador DC 33
5.2 Diseño del Hardware 33
5.2.1 Diseño del módulo receptor 34
5.2.2 Diseño del módulo principal 35
5.3 Montaje y test 38
6 Introducción de nuevos sensores 41
7 Conclusiones y líneas futuras 45
Bibliografía 47
Anexos 51

Lista de Figuras

Figura 1.1: Posición del *gateway* en una red 2
Figura 1.2: Configuración inicial para decodificar los mensajes transmitidos por el sensor, utilizando USRP 3
Figura 2.1: Estructura de una red inalámbrica de sensores 5
Figura 2.2: Estructura interna de un nodo sensor 5
Figura 2.3: Comparación de tecnologías inalámbricas, según su alcance y tasa de datos máxima 7
Figura 2.4: Estructura general de una red MiWi 8
Figura 2.5: Banda 868-870 MHz. Áreas azules están reservadas para uso particular. 12
Figura 2.6: Ejemplo de uso de PoE 13
Figura 3.1: Estación meteorológica por radio control (izquierda) con transmisor de temperatura (derecha) 17
Figura 3.2: Circuito impreso del transmisor de temperatura 17
Figura 3.3: Placa principal de USRP 18
Figura 4.1: Espectro de la señal, utilizando el script "*usrp_fft.py*" 22
Figura 4.2: Dos transmisiones diferentes, separadas 4 segundos 23
Figura 4.3: Fragmento de una trama 23
Figura 4.4: Espectro frecuencial de una trama 24
Figura 4.5: Fragmento de una trama. Datos extraídos 25
Figura 4.6: Hoja de cálculo con algunos de los datos recogidos de temperaturas positivas. El campo de temperatura esta marcado en naranja 26
Figura 4.7: Fragmento de la hoja de cálculo con algunos datos de temperaturas negativas 27
Figura 4.8: Contenido de una trama 29
Figura 5.1: Diagrama de bloques del diseño 31
Figura 5.2: Esquemático receptor RF 34
Figura 5.3: Aspecto del layout del receptor 35
Figura 5.4: Configuración del conector JTAG 37
Figura 5.5: Aspecto del layout del módulo principal 37
Figura 5.6: Imagen final del gateway tras su montaje 39
Figura 5.7: Imagen final del gateway con antena 39
Figura 6.1: Conexión en la red del *gateway* y proceso de descarga 42

Lista de Tablas

Tabla 2.1: Comparación de diversas tecnologías inalámbricas 7
Tabla 2.2: Frecuencias de la banda ISM 11
Tabla 2.3: Clasificación de PD según potencia en PoE 13
Tabla 2.4: Comparativa entre artículos relacionados y el presente trabajo 15
Tabla 5.1: Comparación de la familia de Microcontroladores MPS430F54xx 32

Lista de Acrónimos y Abreviaciones

AC	Alternating Current
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
ASK	Amplitude Shift Keying
CCS	Code Composer Studio
CEPT	European Conference of Postal and Telecommunications
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
DAC	Digital-to-Analog Converter
DC	Direct Current
DCF	Deutschland Long Wave Frankfurt
DCO	Digital Controlled Oscillator
DMA	Direct Memory Access
DSP	Digital Signal Processing
ERP	Equivalent Radiated Power
ETSI	European Telecommunications Standard Institute
FPGA	Field Programmable Gate Array
FSK	Frequency Shift Keying
GFSK	Gaussian Frequency Shift Keying
GNU	GNU's Not Unix
GSM	Global System for Mobile Communications
HART	Highway Addressable Remote Transducer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISA	International Society of Automation
ISM	Industrial, Scientist, Medical
ITU-R	International Telecommunication Union (Radiocommunication)
LCD	Liquid Crystal Display
LR-WPAN	Low-Rate Wireless Personal Area Network
MAC	Medium Access Control
MCU	Microcontroller Unit
MSK	Minimum Shift Keying
NFC	Near Field Communication

PC	Personal Computer
PCB	Printed Circuit Board
PD	Powered Device
PDA	Personal Digital Assistant
PHY	Physical Layer
PoE	Power over Ethernet
PSE	Power Sourcing Equipment
PSTN	Public Switched Telephone Network
RAM	Random Access Memory
RF	Radio Frequency
RFID	Radio Frequency Identification
RISC	Reduced Instruction Set Computing
RX	Receive
SDR	Software Defined Radio
SNMP	Simple Network Management Protocol
SoC	System on Chip
SPI	Serial Peripheral Interface
SRD	Short Range Device
TI	Texas Instruments
TV	Television
TX	Transmit
UART	Universal Asynchronous Receiver Transmitter
UDP	User Data Protocol
UHF	Ultra High Frequency
USB	Universal Serial Bus
USCI	Universal Serial Communication Interface
USRP	Universal Software Radio Peripheral
UWB	Ultra-wideband
Wi-Fi	Wireless Fidelity (a branding effort for IEEE 802.11 WLANs)
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

1 Introducción

1.1 Motivación

En los últimos años, las redes de comunicaciones han cambiado totalmente el punto de vista de la sociedad. La forma en la que las personas y las diferentes entidades intercambian información y coordinan sus actividades ha evolucionado. Y como se puede vislumbrar en la actualidad, es posible que seamos testigos de otra revolución en los próximos años, ya que las nuevas tecnologías, cada vez más, observan y controlan no solo el mundo virtual, sino también el mundo físico. Los últimos avances tecnológicos han permitido el desarrollo de sistemas distribuidos, usando diminutos procesadores de bajo coste y bajo consumo, capaces de procesar y transmitir la información sin necesidad de cables, esto es, de manera inalámbrica. La disponibilidad de microsensores, junto al progreso en el área de las comunicaciones inalámbricas, dará lugar al desarrollo de todo tipo de redes de sensores, utilizados para un gran abanico de aplicaciones, más allá incluso de lo que hoy en día se concibe.

Las personas quieren conocer qué es lo que está pasando alrededor de ellas, especialmente en algunos ámbitos como puede ser el hogar o el lugar de trabajo. Están surgiendo cada vez más dispositivos con capacidad de conectarse a algún tipo de red para adquirir o intercambiar información de todo tipo. La tecnología está haciendo posible los entornos inteligentes, por ejemplo, controlando y monitorizando el estado o las condiciones de lo que nos rodea, utilizando tan solo un pequeño dispositivo. Y es precisamente en este tipo de áreas donde las redes de sensores inalámbricas cobran especial importancia.

Un elemento fácil de encontrar en el hogar o en la oficina es un sensor de temperatura. Existen actualmente muchos dispositivos comerciales que muestran datos tales como temperatura, humedad, presión atmosférica… Muchos de éstos se basan en sensores que cuentan con algún tipo de conexión inalámbrica, y transmiten los datos desde el sensor a otro dispositivo equipado con un display para mostrar la información. El display normalmente está colocado en el interior de un edificio, mientras que los sensores pueden encontrarse en cualquier lugar (dentro o fuera del edificio). Por tanto, una persona puede leer los datos a través de la pantalla, pero generalmente no existe ningún otro medio de interacción con el sensor ni con la información. Éste es precisamente el ámbito de desarrollo del presente proyecto. Se trata de obtener mayor provecho de este tipo de sensores, de manera que la transmisión de la información no quede simplemente en la lectura de un valor en un display, sino que estos datos puedan ser utilizados en nuevos sistemas. Por tanto, una de las claves de este trabajo es explotar la información que ya se genera con los sensores existentes para permitir nuevas aplicaciones, sin que el desarrollador original del sensor tenga que preocuparse de ello en su diseño.

1.2 Descripción del problema

El propósito de este PFC es diseñar el hardware de un *gateway* o pasarela capaz de extraer los datos transmitidos por sensores en la banda ISM de 868 MHz. Este *gateway* deberá ser capaz de recibir y si es necesario desencriptar la información que varios sensores inalámbricos puedan estar transmitiendo, para suministrarla a un ordenador o dispositivo inteligente para su posterior uso. Existen ciertos sensores que utilizan un protocolo propietario para transmitir sus datos, por lo que debemos en primer lugar conocer y decodificar dicho protocolo para poder extraer la información relevante antes de proporcionársela a otro dispositivo mediante Ethernet, el cuál pueda utilizar dichos

datos. Las múltiples aplicaciones que se pueden crear, así como la implementación del protocolo decodificado o la programación software en el *gateway*, están fuera del alcance de este proyecto. En la Figura 1.1 se ilustra el contexto en el que se enmarca este *gateway* de sensores.

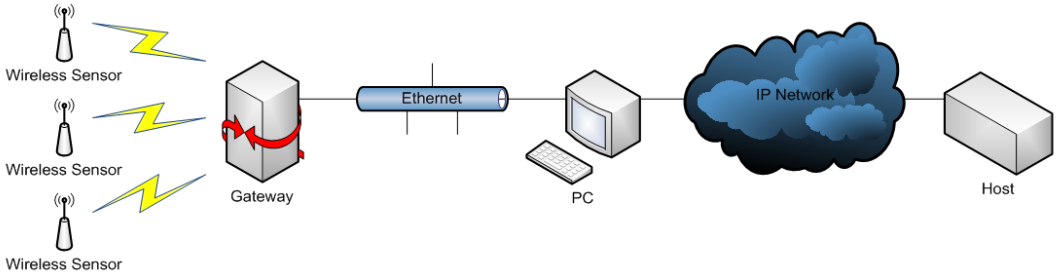

Figura 1.1: Posición del *gateway* en una red

Se comenzará este proyecto analizando un sensor inalámbrico concreto para averiguar qué y cómo transmite, por lo que el *gateway* será inicialmente adecuado para conectar tan sólo un tipo específico de sensor, en este caso de temperatura y cuyo rango de transmisión es de 100 metros. El primer problema a solventar será determinar la frecuencia exacta de operación de este sensor. Esto es relativamente fácil de resolver con un analizador de espectro. El siguiente paso será recibir la señal transmitida y decodificar el protocolo propietario.

No obstante, el objetivo final es estudiar la posibilidad de poder identificar todo tipo de sensores, creando así un dispositivo global o universal. Esto significa que se deberá considerar cómo reconocer nuevos sensores y cómo (y dónde) realizar la decodificación y extracción de la información.

1.3 Objetivos del trabajo

Para una mejor organización, se han establecido diferentes objetivos para estructurar el trabajo a llevar a cabo, descritos a continuación:

1. Realizar un análisis del estado del arte de las diversas tecnologías relativas a redes de sensores inalámbricas.
2. Examinar, mediante un dispositivo general (USRP, detallado en el apartado 3.2), el tráfico que genera un determinado sensor de temperatura comercial inalámbrico (apartado 3.1). La Figura 1.2 muestra la configuración a seguir para lograr este objetivo.
3. Analizando los datos obtenidos, averiguar las características propias de la transmisión de este sensor para poder decodificar el mensaje, concretamente los campos de información útil (dato de temperatura y campo de identificador).
4. Diseñar el hardware que conformará un *gateway* o pasarela capaz de recibir, decodificar y mandar esta información (proveniente de uno o más sensores del mismo tipo simultáneamente) a un ordenador o dispositivo inteligente (una IP dada) por cable, para que pueda ser aprovechada en futuras aplicaciones o servicios. Este *gateway* estará compuesto principalmente de un receptor, un microcontrolador y una interfaz Ethernet.

5. Estudiar la posibilidad de generalizar el diseño para poder aprovechar también otro tipo de sensores.

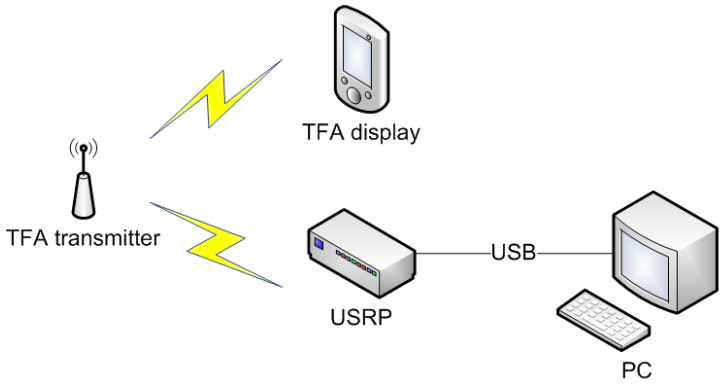

Figura 1.2: Configuración inicial para decodificar los mensajes transmitidos por el sensor, utilizando USRP

1.4 Contexto

Este trabajo se ha desarrollado a lo largo de 6 meses en el laboratorio Wireless@KTH del departamento Communication Systems (CoS) de la universidad KTH Kungliga Tekniska Högskolan de Estocolmo (Suecia), gracias al programa de intercambio de estudiantes ERASMUS. Se enmarca en el área de investigación, llevada a cabo en dicho laboratorio, denominada "*Networked M2M (Machine-to-Machine): Things that communicate*". Las comunicaciones entre máquinas u objetos móviles, su infraestructura y sus diversas aplicaciones son el objeto de estudio de esta línea de trabajo.

Destacar que el presente proyecto está fuertemente ligado a otro que se lleva a cabo paralelamente a éste en el mismo departamento. Mientras que este PFC analiza la comunicación de los sensores y diseña el hardware del *gateway*, el otro trabajo al que se alude desarrolla la parte software relativa al *gateway* e implementa una pequeña aplicación, haciendo uso del trabajo expuesto en este documento.

1.5 Estructura

A continuación se describe brevemente el contenido de los capítulos restantes que conforman esta memoria:

- El capítulo 2 presenta una visión del estado del arte en el que se encuentran las diferentes tecnologías relativas a redes de sensores inalámbricas.
- El capítulo 3 muestra las principales herramientas, tanto de hardware como de software, que se utilizan en el desarrollo de este trabajo.
- El capítulo 4 explica el proceso que se ha seguido para decodificar el protocolo propietario del sensor bajo estudio.
- El capítulo 5 describe el proceso de diseño del *gateway* de sensores.
- El capítulo 6 realiza un estudio sobre cómo aprovechar otro tipo de sensores inalámbricos.
- Y por último, el capítulo 7 expone las conclusiones extraídas de este proyecto, así como futuras líneas de trabajo.

2 Estado del arte

Este capítulo muestra un análisis del estado del arte en el que se encuentran las diferentes tecnologías que van a ser tratadas a lo largo de este documento. En una primera sección, se introducirán aspectos generales de las redes inalámbricas de sensores. Parte de los sensores a los que se hará referencia utilizan la banda ISM, por lo que se describirán sus características. Por otra parte, se presentará la tecnología Power Over Ethernet (PoE), ideada para suministrar la energía a los dispositivos compatibles a través del cable Ethernet, junto a los datos (se pretende implementar esta tecnología en el diseño del *gateway*). Y por último, se hará una revisión de algunos de los artículos más significativos relacionados con este PFC.

2.1 Redes inalámbricas de sensores

Una red inalámbrica de sensores (WSN, en inglés Wireless Sensor Network) es un sistema que consta de varios nodos autónomos distribuidos espacialmente que colaboran para realizar una tarea común [1]. Cada uno de estos nodos o sensores se encarga de la monitorización de una o varias características físicas o ambientales, como pueden ser temperatura, humedad, sonido, presión, vibración, luz, polución, movimiento… El conjunto de los nodos sensores envían la información recogida a un nodo central o *gateway*, que se encarga de conectar la red con un equipo terminal o con otras redes (ver Figura 2.1).

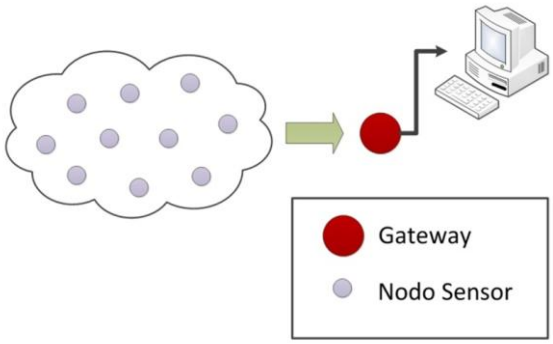

Figura 2.1: Estructura de una red inalámbrica de sensores

Cada nodo está compuesto básicamente, además del sensor, de un procesador, de una fuente de energía (normalmente una batería) y una interfaz radio para intercambiar información con los dispositivos de la misma red. La estructura interna típica de uno de estos nodos sensores, también denominados "motes", ya que suelen ser ligeros y diminutos [2], se puede ver en la Figura 2.2.

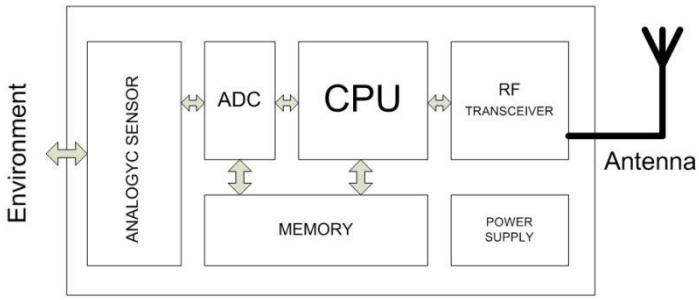

Figura 2.2: Estructura interna de un nodo sensor

Cada nodo es capaz de procesar una limitada cantidad de datos. Sin embargo, cuando se coordina el trabajo de un gran número de ellos, es posible conseguir medidas de gran precisión. Se habla entonces de "fusión de sensores". Un ejemplo de dicha fusión puede verse en la combinación de los datos de 2 cámaras diferentes, con las que es posible reconstruir una imagen en estéreo para determinar la localización de objetos en 3D. Cuanto más esfuerzo se ponga en la correcta colaboración de todos los motes y mayor sea el despliegue de la red, más precisa será su labor.

Las características físicas del área en el que va a trabajar la red deben tenerse en cuenta (para evitar posibles obstáculos que puedan bloquear la comunicación, por imposibilidad de colocar el nodo en la localización idónea…), ya que pueden determinar varios aspectos del diseño, por ejemplo la topología de la red (puede ser en anillo, en estrella, en malla, en árbol…).

Un aspecto importante es dónde realizar el procesado de la información. Se puede optar por realizar la mayor parte del procesado en el nodo central. Tiene la ventaja de que los nodos tan sólo tienen que transmitir los datos recogidos, por lo que su coste es bajo. Por otro lado, teniendo en cuenta que la mayor parte de la energía que se consume tiene lugar en la comunicación por radiofrecuencia, se puede procesar pequeñas partes de la información localmente, esto es, en el propio nodo. Esto supondrá un mayor coste en los nodos, ya que se necesita dotarles de mayor inteligencia, pero la información a transmitir puede ser menor, por lo que se consume menor energía. Generalmente, en sistemas alimentados por baterías, es recomendable escoger ésta última opción para alargar la vida de los nodos.

Nótese que la realización o el análisis de una red inalámbrica de sensores integra una amplia variedad de disciplinas. Es necesario aplicar conocimientos sobre diversos campos para determinar su jerarquía, routing, topología, tipo de sensor, protocolo de comunicación, acceso múltiple... Además, las posibles utilidades de estas redes son muy amplias [3]. Diferentes aplicaciones se pueden encontrar en campos como domótica, control de un paciente médico, control/seguridad pública, monitorización de diferentes estructuras físicas (puentes, edificios) o equipamientos (motores, maquinaria), eficiencia energética…

Puesto que el propósito de este proyecto no es el de desarrollar una red de sensores, ya que se parte del análisis de un sensor específico, se centrarán los siguientes apartados en las áreas relacionadas con el trabajo a realizar, especialmente en aquellas tecnologías que van a ser utilizadas en este desarrollo.

2.2 Tecnologías inalámbricas

Una parte trascendental en la evolución de las WSN está en el uso de las tecnologías inalámbricas. El estándar 802.11 del IEEE (del inglés, Institute of Electrical and Electronics Engineers) [4] fue el primero que reguló las redes inalámbricas de área local (WLAN, del inglés Wireless Local Area Network). Fue introducido por primera vez en 1997. Este estándar está basado en el protocolo de control de acceso al medio CSMA/CA, esto es, acceso múltiple por detección de portadora con evasión de colisiones [5]. Esta versión inicial fue revisada años más tarde para dar lugar al IEEE 802.11b, que permitía mayor velocidad de datos. Aunque esta familia de estándares fue diseñada para las citadas WLANs (por ejemplo, para conectar ordenadores con PDAs) ha sido también utilizada en redes de sensores. Sin embargo, el gran consumo y la alta tasa de bits que establece dicho estándar hace que sea inapropiado para las necesidades de muchas de las WSNs.

Tabla 2.1: Comparación de diversas tecnologías inalámbricas

Wireless Networking Technologies						
	ZigBee	Bluetooth	UWB	Wi-Fi	LonWorks	Proprietary
Standard	IEEE 802.15.4	IEEE 802.15.1	IEEE 802.15.3a (to be ratified)	IEE 802.11a, b, g (n to be ratified)	EIA 709.1, 2, 3	Proprietary
Industry organizations	ZigBee Alliance	Bluetooth SIG	UWB Forum and WiMedia Alliance	Wi-Fi Alliance	LonMark Interoperability Association	N/A
Topology	Mesh, star, tree	Star	Star	Star	Medium-dependent	P2P, star, mesh
RF frequency	868/915 MHz, 2.4 GHz	2.4 GHz	3.1 to 10.6 GHz (U.S.)	2.4 GHz, 5.8 GHz	N/A (wired technology)	433/868/900 MHz, 2/4 GHz
Data rate	250 kbits/s	723 kbits/s	110 Mbits/s to 1.6 Gbits/s	11 to 105 Mbits/s	15 kbits/s to 10 Mbits/s	10 to 250 kbits/s
Range	10 to 300 m	10 m	4 to 20 m	10 to 100 m	Medium-dependent	10 to 70 m
Power	Very low	Low	Low	High	Wired	Very low to low
Battery operation (life)	Alkaline (months to years)	Rechargeable (days to weeks)	Rechargeable (hours to days)	Rechargeable (hours)	N/A	Alkaline (months to years)
Nodes	65,000	8	128	32	32,000	100 to 1000

Ya que cada aplicación puede tener sus propias restricciones (de consumo, tasa de bits, número de nodos, alcance del enlace vía radio, fiabilidad, seguridad…) se han desarrollado diferentes tipos de tecnologías para satisfacer diferentes requerimientos (ver Tabla 2.1 [6]). Las características de cada tecnología afectan al diseño de los sistemas, dispositivos o aplicaciones a construir. La Figura 2.3 [7] muestra las diferencias entre estas distintas soluciones, en términos de alcance por radio máximo frente a tasa de bits.

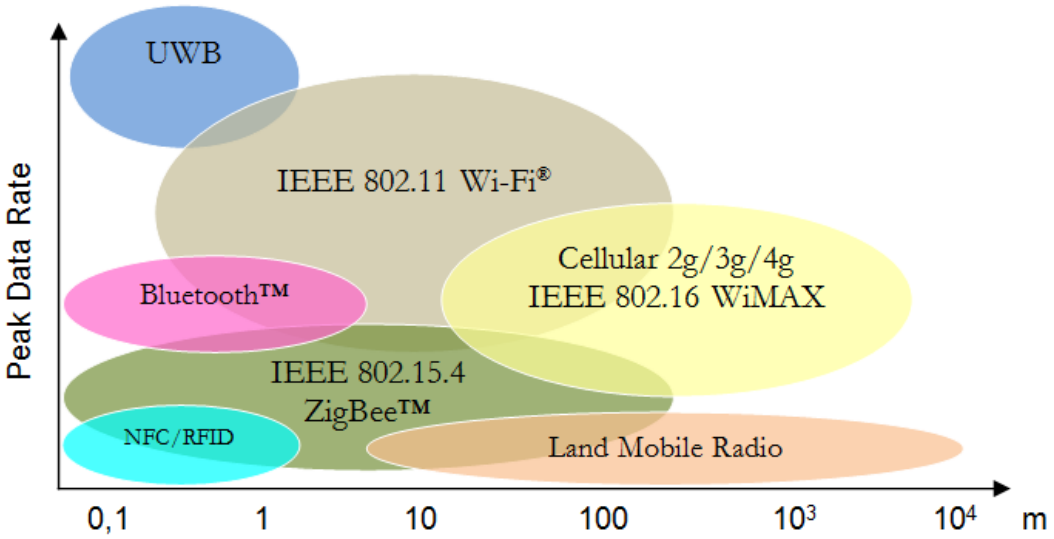

Figura 2.3: Comparación de tecnologías inalámbricas, según su alcance y tasa de datos máxima

Los sensores que conciernen a este proyecto tienen un corto alcance radio (alrededor de 100 m) y no requieren gran ancho de banda. Estas características son típicas de las

redes inalámbricas de área personal (WPAN, del inglés Wireless Personal Area Network). Las WPANs son concebidas para integrar e intercomunicar dispositivos u objetos muy cercanos a la persona. Hoy en día, las WPANs son capaces de comunicarse a su vez con otras redes, dando lugar al concepto generalizado de "Internet de los objetos" [8]. Estos "objetos" o elementos pueden estar situados en cualquier lugar de una habitación, de un edificio, en el propio cuerpo humano… Debido a la flexibilidad de la que es necesario dotar a este tipo de redes, el desarrollo de distintos tipos de comunicaciones inalámbricas es esencial. Y adicionalmente, a causa de los múltiples ejemplos de redes existentes, es necesario también el desarrollo de diversos tipos de *gateway*, que hagan de puente para interconectar todo este tipo de redes.

En relación a las distintas tecnologías para redes de sensores, la plataforma ZigBee, desarrollada por Zigbee Alliance [9], se ha convertido en una de las más populares. Zigbee hace uso del estándar IEEE 802.15.4 [10]. Este estándar especifica tanto el control de acceso al medio (MAC) como la capa física (PHY) de las redes WPAN. Zigbee define un conjunto de protocolos (de capa 3 o superior) para administrar la comunicación entre nodos. Además, es lo suficientemente flexible como para ser utilizada en numerosos campos: industria, sanidad, posicionamiento, sistemas de vigilancia… [11]. Esta tecnología está implementada en multitud de nodos comerciales, como pueden ser MICAz [12], TelosB [13] o IRIS [14] de la empresa MEMSIC, o el denominado Waspmote de la empresa Libelium (spin-off de la Universidad de Zaragoza) [15].

No obstante, Zigbee no es la única opción en cuanto se refiere a redes WSN. Actualmente existen muchas otras soluciones:

- SimpliciTI [16] es un protocolo de red de código libre, desarrollado por Texas Instruments. Está enfocado al uso de pequeñas y simples redes RF. Junto al software necesario para utilizar este protocolo, el fabricante proporciona el hardware necesario para conseguir una fácil implementación de una red WPAN. La Figura 2.4 muestra una estructura típica que usa esta tecnología. Como se puede observar, el alcance radio se puede extender a través de repetidores. Control de alarmas, detectores de humo y medidores automáticos son las principales aplicaciones que actualmente utilizan este protocolo.

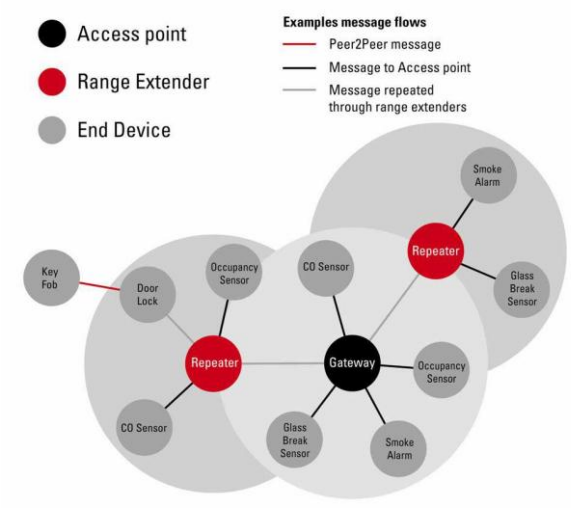

Figura 2.4: Estructura general de una red MiWi

- MiWi [17] fue diseñado por Microchip Technologies. Basado en el IEEE 802.15.4, está orientado a redes WPAN que requieren una baja tasa de bits (LR-WPAN, Low-Rate Wireless Personal Area Network). Ofrece una gran compatibilidad entre distintos transmisores/receptores de Microchip y diferentes protocolos (dependiendo de los requisitos de cada aplicación) sin tener que cambiar el firmware. Sin embargo, este protocolo no soporta redes del tamaño de Zigbee (el máximo número de nodos es de 1000, mientras que Zigbee soporta 65000). A pesar de que es de acceso libre, el hardware a usar debe ser del mismo fabricante.
- Synkro [18] es una pila de protocolos de red, que se apoya en la parte superior del IEEE 802.15.4. Está encaminado al uso en productos de "home entertainment", como televisores digitales, reproductores de DVD, audio/video receptores… Freescale es el propietario de esta solución, aunque desde 2008 es de acceso libre.
- PopNet™ [19] es un protocolo de red y un entorno operativo diseñado para sensores de bajo consumo y aplicaciones de control. Está desarrollado por San Juan Software. Es un protocolo muy flexible, por lo que se puede adaptar a casi cualquier aplicación. Utiliza cifrado AES para proteger la información y dotar de robustez al sistema.
- Z-Wave [20] es un protocolo orientado hacia la domótica o "casa inteligente", concretamente para aplicaciones de control remoto en hogares y control de iluminación en ciertos entornos. Desarrollada por el consorcio internacional de fabricantes Z-Wave Alliance, esta tecnología introduce un sistema embebido de RF en el interior de algunos aparatos electrónicos para controlar ciertos sistemas como la iluminación, el control de acceso al hogar, sistemas de entretenimiento o electrodomésticos. La frecuencia a la que opera Z-Wave está por debajo de 1 GHz (en Europa, en 868.42 MHz).
- ONE-NET [21] es un estándar de código abierto. Fue diseñado para aplicaciones de control de redes de bajo coste y bajo consumo como la domótica o sistemas de seguridad. ONE-NET no está atado a ningún propietario de hardware, por lo que puede ser implementado en gran variedad de dispositivos de bajo coste de diferentes fabricantes. Utiliza dispositivos radio compatibles con UHF, y opera en las bandas de 868 MHz y 915 MHz.
- DASH7 [22] es un estándar para redes de sensores, que opera en la banda ISM sin licencia de 433 MHz. Proporciona larga durabilidad de las baterías, alcance de hasta 2 km, una pequeña pila de protocolos de código abierto, encriptación de la información AES de 128 bits y transferencia de datos de hasta 200 KB/s. Está promovido por el consorcio sin fines lucrativos denominado DASH7 Alliance.
- WirelessHART [23] es una tecnología basada en el protocolo "Highway Addressable Remote Transducer" (HART). Éste puede operar en la banda ISM de 2.4 GHz utilizando interfaces radio estándar definidas en el 802.15.4.

Además de las diferentes tecnologías que podemos encontrar, también existen en el mercado sistemas operativos orientados al desarrollo de aplicaciones para redes de sensores. Algunos ejemplos de este tipo de sistemas son:

- TinyOS [24] es un sistema operativo de código abierto orientado hacia dispositivos inalámbricos de bajo consumo. Utiliza el lenguaje de programación nesC [25] (un dialecto de C), y proporciona interfaces, módulos y configuración específica. Trabaja de forma modular, esto es, permite desarrollar aplicaciones como una serie de módulos, cada uno de ellos con una tarea específica.
- Contiki [26] es un pequeño sistema operativo multitarea caracterizado por su gran portabilidad. Está orientado a sistemas con limitación de memoria, desde computadores de 8 bits a sistemas embebidos o microcontroladores. Es usualmente denominado "el sistema operativo del Internet de los objetos".

Como se puede observar, existen multitud de posibilidades. La elección de una de ellas dependerá muchas veces de los requisitos y requerimientos de cada aplicación en concreto.

2.3 Banda ISM

La banda de radiofrecuencia denominada ISM (Industrial, Scientific and Medical) está definida por el ITU-R en sus Regulaciones Radio 5138, 5150 y 5280 [27]. Los diferentes rangos que define esta organización se muestran en la Tabla 2.2. Los usos individuales designados por cada país pueden diferir debido a la variación de las regulaciones de cada nación en materia de radiofrecuencia. Nótese que existen usos en esta banda con necesidad de licencia y sin ella. Sin embargo, debido a la alta probabilidad de interferencia, los usos con licencia son muy bajos o utilizan mucha más potencia que los usos sin licencia.

En Europa, el rango de frecuencias de los 900 MHz forma parte de la asignación de GSM [28]. Esto implica que los equipos que trabajan en la banda ISM de 900 MHz, como puede ser equipamiento importado (ilegalmente) de EEUU, Asia o Suráfrica causan y sufren sustanciales interferencias. Es por ello que en Europa se usa el rango desde los 868 MHz a los 870 MHz (ver sección 2.3.1). De manera similar, el uso de la banda de 433-435 MHz en Estados Unidos es reemplazado en Europa por la banda 340-354 MHz.

Un inconveniente de esta banda ISM está en la falta de protección frente a interferencias. Para asegurar una cierta coexistencia entre nuevos equipos y los que ya ocupan esa parte del espectro, es necesario utilizar técnicas de espectro ensanchado [29], excepto para aplicaciones de potencia extremadamente baja. Estas tecnologías ofrecen protección tanto para los usuarios existentes (ya que así la densidad espectral de potencia media de los nuevos usuarios es baja) como para los usuarios nuevos (ya que la ganancia de procesamiento de estos sistemas reduce la interferencia causada por otras fuentes).

Tabla 2.2: Frecuencias de la banda ISM

Frequency range		Center frequency	Availability
6.765 MHz	6.795 MHz	6.780 MHz	Subject to local acceptance
13.553 MHz	13.567 MHz	13.560 MHz	
26.957 MHz	27.283 MHz	27.120 MHz	
40.660 MHz	40.700 MHz	40.680 MHz	
314.000 MHz	317.000 MHz	315.500 MHz	Japan
340.000 MHz	354.000 MHz	347.000 MHz	Region 2[1] only and subject to local acceptance
433.050 MHz	434.790 MHz	433.920 MHz	Region 1[2] only and subject to local acceptance
868.000 MHz	870.000 MHz	869.000 MHz	Region 1
902.000 MHz	928.000 MHz	915.000 MHz	Region 2
2.400 GHz	2.500 GHz	2.450 GHz	
5.725 GHz	5.875 GHz	5.800 GHz	
24.000 GHz	24.250 GHz	24.125 GHz	
61.000 GHz	61.500 GHz	61.250 GHz	Subject to local acceptance
122.000 GHz	123.000 GHz	122.500 GHz	Subject to local acceptance
244.000 GHz	246.000 GHz	245.000 GHz	Subject to local acceptance

2.3.1 Dispositivos de corto alcance operando en 868MHz

Por “dispositivos de corto alcance” (SRD, Short Range Devices) entendemos los transmisores radio que proporcionan tanto comunicaciones unidireccionales como bidireccionales a baja potencia en un rango acotado, ya que no se desea causar interferencia en otros equipos. Estos dispositivos pueden utilizar antenas integradas o externas.

El uso de la banda de espectro frecuencial de 868MHz está regulada por la CEPT (European Conference of Postal and Telecommunications Administrations). Una de sus recomendaciones [30] establece la asignación de las frecuencias que deben utilizar estos SRD´s en los países que conforman la CEPT.

La institución denominada ETSI (European Telecommunications Standards Institute) desarrolló el estándar para la explotación de estos dispositivos [31]. El radioespectro específico para el mercado europeo se sitúa entre los 868 MHz y los 870 MHz, y está separado en 4 secciones (G1 a G4). La Figura 2.5 muestra como se distribuyen estas 4 secciones, así como la potencia radiada aparente máxima. Nótese que esta potencia radiada aparente o equivalente (ERP, Equivalent Radiated Power) expresada en Watios es la energía radiada por un dispositivo después de tomar en consideración todas las fuentes de pérdidas y ganancias.

[1] North and South America and Pacific (East of the International Date Line)
[2] Europe, Middle East, Africa, the former Soviet Union, including Siberia; and Mongolia and China

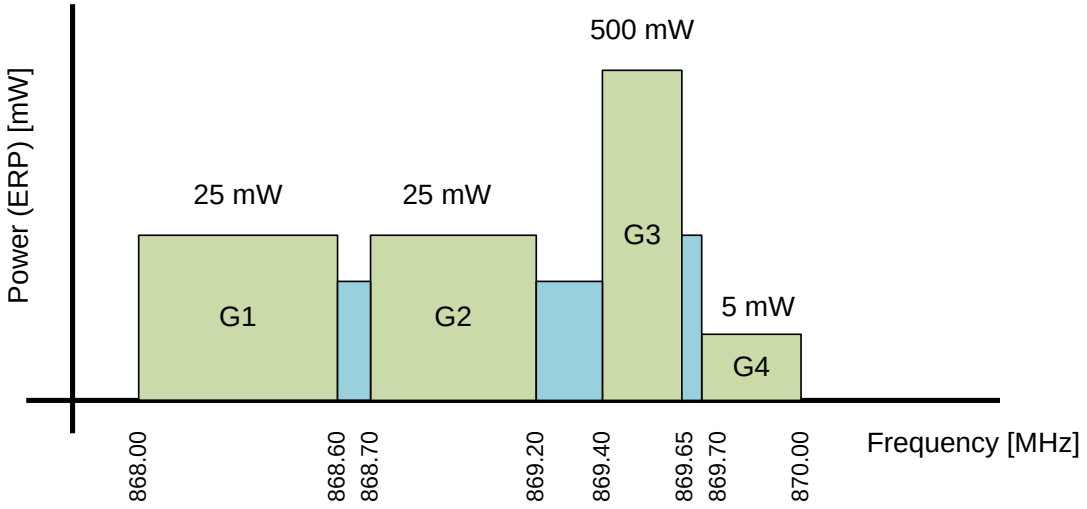

Figura 2.5: Banda 868-870 MHz. Áreas azules están reservadas para uso particular

2.4 Power over Ethernet

Power over Ethernet (PoE) es una tecnología que integra el suministro de energía en una red LAN estándar. Permite proporcionar energía a un dispositivo de la red, usando el mismo cable que se utiliza para la transmisión de datos. La energía se proporciona en modo común o en modo diferencial a través de 2 o más pares trenzados que se encuentran en los cables Ethernet.

PoE fue estandarizado en 2003 por el IEEE en la norma IEEE 802.3af, aunque actualmente este estándar está en desuso, ya que ha sido reemplazado por la norma IEEE 802.3at [32]. El mecanismo para transportar la energía es similar al que se utiliza para la Red Telefónica Conmutada (RTC). La potencia en continua máxima que se entrega en un cable es, teóricamente, de 15 W. Sin embargo, en la práctica, se consigue un valor máximo de 12.95 W, debido a las pérdidas en el cable. A partir de 2009, con la actualización de la norma, es posible alcanzar hasta 25.5 W de potencia.

El estándar describe dos tipos de dispositivos:

- Power Sourcing Equipment (PSE): Equipos que entregan potencia sobre un sistema Ethernet.
- Powered Devices (PD): Dispositivos que se alimentan con potencia recibida a través de un sistema Ethernet.

Para su correcto funcionamiento, esta tecnología requiere cable de categoría 5 o superior, aunque para potencias muy bajas, uno de categoría 3 puede ser suficiente. Este tipo de cables tienen 4 pares trenzados, pero solo 2 de ellos son utilizados para los datos. El estándar IEEE 802.3at permite transportar la energía tanto sobre los pares de datos como sobre los otros 2 pares restantes no utilizados.

Se especifican 4 fases o bloques que todo dispositivo PoE debe seguir para poder ser alimentado, que son:

- Detección: Se comprueba que el PD tiene una resistencia de entrada comprendida entre 15 KΩ y 33 KΩ. Si no es así, el proceso se detiene en este punto, y la fuente no suministra la energía en el cable Ethernet.

- Clasificación: Se determina a qué clase pertenece el dispositivo. Para ello, se alimenta el PD con tensiones entre 15-20 V, y dependiendo de la respuesta del dispositivo (la corriente resultante), se identifica dentro de una clase de potencia, de acuerdo a la Tabla 2.3.

Tabla 2.3: Clasificación de PD según potencia en PoE

CLASE	USO	POTENCIA PARA ALIMENTAR EL PD (WATIOS)
0	Por defecto	0.44 a 12.95
1	Opcional	0.44 a 3.84
2	Opcional	3.84 a 6.49
3	Opcional	6.49 a 12.95
4	Reservado	-

- Control: Este bloque se encarga de asegurar el correcto funcionamiento del dispositivo. No se debe suministrar potencia a un dispositivo que no lo espere. Por tanto, un PSE debe retirar la señal de energía cuando se desconecta el cable, y volver a aplicarla únicamente después que se haya pasado por las fases de detección y clasificación.
- Conversión DC/DC: La tensión nominal que se recibe del PSE es de 48V. Habitualmente se requiere un valor de alimentación menor, por lo que es necesario colocar un convertidor DC/DC que se adecue a las características de la aplicación.

La topología de los sistemas PoE siempre está configurada en forma de estrella, ya que cada uno de los dispositivos debe estar conectado directamente al equipo que suministra la energía. Actualmente existen gran variedad de dispositivos alimentados mediante PoE.

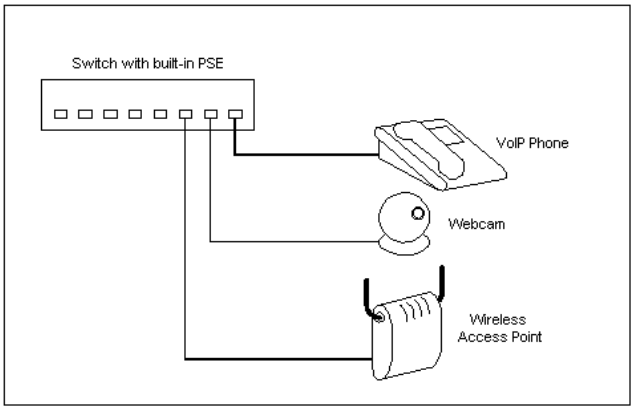

Figura 2.6: Ejemplo de uso de PoE

PoE soporta arquitecturas punto-multipunto, paralelamente a la red de datos. Gracias a esto, se pueden administrar los dispositivos conectados de forma remota, utilizando por ejemplo, el protocolo SNMP (Simple Network Message Protocol) [33].

2.5 Artículos relacionados

En la primera parte del desarrollo de este PFC, parte del tiempo se ha dedicado a la lectura de diferentes artículos y trabajos que pudieran ser de utilidad. En este apartado, se va a resumir algunos de los más importantes o relevantes.

Existen muchos artículos que describen como desencriptar diferentes protocolos que no se conocen, ya sean de sistemas inalámbricos o no. Señalar que algunos de ellos se concentran exclusivamente en protocolos de Internet [34][35], mientras que otros se centran en redes de sensores inalámbricas. Por ejemplo, el artículo [36] trata sobre captar tramas en una red de sensores Zigbee.

Otro ejemplo ilustrativo es el de este artículo [37]. En él, los autores definen un sistema, llamado Catcher, que es capaz de extraer la información de protocolos no conocidos. El proceso a seguir tiene 3 pasos. En primer lugar, simplemente se capturan los paquetes y el programa los lista. Después, se delimitan los distintos campos en las tramas y se halla la longitud de cada uno. En el último paso, Catcher identifica los campos dinámicos y los estáticos para extraer los datos que cambian en cada trama, esto es, la información o el mensaje en sí. A pesar de que no se trata específicamente del mismo caso que este proyecto, este artículo da una idea del procedimiento que se ha de seguir para este trabajo.

En lo referente al diseño del *gateway*, las investigaciones descritas en [38] y [39] usan el mismo chip RF y el mismo microcontrolador que los escogidos en este trabajo. Estos 2 artículos proporcionan datos acerca de la comunicación entre ambos módulos.

Zhou y Shen, los autores del artículo [40], describen la realización de un *gateway* inalámbrico ZigBee – Wi-Fi. La parte de hardware de la pasarela está compuesta por un chip RF a 2.4 GHz (éste incorpora el transmisor/receptor Zigbee y el microprocesador) y por un módulo de Wi – Fi. La comunicación entre ambos se logra mediante interfaz UART. La arquitectura del software descrito incluye control del sistema, configuración software del chip RF y del módulo Wi – Fi y la capa de aplicación del protocolo. También se explican los procedimientos de interacción entre la red Zigbee y la WLAN (en ambas dirección) usando este *gateway*. A pesar de que en este PFC no se va a usar Wi –Fi y que solo se va a implementar la comunicación desde el sensor hacia la red LAN, se encuentran similitudes entre ambos, por lo que se puede extraer información útil.

En conclusión, los trabajos aquí citados han servido de ayuda y de guía en diferentes momentos del desarrollo de este proyecto. A modo de resumen, en la Tabla 2.4 se puede ver una comparativa entre lo que se describe en estos artículos y lo que se ha realizado en el presente trabajo, según estén relacionados con el análisis y descodificación de la información, o con el diseño hardware del *gateway*.

Tabla 2.4: Comparativa entre artículos relacionados y el presente trabajo

	Decodificación de la información	**Diseño del *gateway***
[34]	Cómo identificar protocolos utilizados en una conexión TCP desconocida	
[35]	Método para extraer un determinado campo de un protocolo sin decodificarlo en su totalidad	
[36]	Captura y análisis de tramas Zigbee	
[37]	Sistema para extraer información de protocolos desconocidos	
[38]		Descripción de comunicaciones y test del integrado CC1101
[39]		Sistema que utiliza el chip integrado CC1101 y un microcontrolador MSP430
[40]		Diseño de *gateway* Zigbee – Wi-Fi
Este trabajo	Extracción de la información del protocolo propietario de un sensor inalámbrico	Diseño hardware de *gateway* de sensores utilizando integrado CC1101. Conexión Ethernet.

3 Herramientas

Para la realización de este trabajo, se han necesitado diferentes herramientas, tanto hardware como software. Todo el material requerido que no fuera de acceso libre fue facilitado por el Prof. Maguire, director del PFC en la universidad KTH de Estocolmo, Suecia. En esta sección se van a exponer las herramientas utilizadas, como el sensor a estudio junto con la estación receptora, el dispositivo USRP o el software GNU Radio, que permite capturar las señales del sensor.

3.1 Estación meteorológica por radio control y transmisor de temperatura

El dispositivo TFA Dostmann GmbH & Co. KG (TFA) 'Wave' (referencia 30.3016.54.IT en catálogo) consiste en un sensor de temperatura inalámbrico y una estación receptora [41]. Está pensado para ser utilizado en el hogar o en la oficina, ya que en su display muestra la temperatura interior (captada por la propia estación), la temperatura exterior, que es enviada a través del sensor, la fecha y la hora. Destacar además que para ajustar la fecha y la hora de manera automática puede utilizar la señal DCF77 [42], que es una señal horaria de onda larga que se emite desde Alemania.

Figura 3.1: Estación meteorológica por radio control (izquierda) con transmisor de temperatura (derecha)

El *gateway* a diseñar captará el tráfico generado por el sensor inalámbrico de temperatura para conocer la información que contiene. Pero en primer lugar, deberemos decodificar las transmisiones realizadas, para poder detectar las tramas y extraer de ahí la información útil. El display de la estación receptora servirá en el análisis del tráfico, ya que nos dará el valor de temperatura que corresponde con cada transmisión. Decodificar el mensaje no es tarea simple, ya que el protocolo utilizado es desconocido, y el circuito electrónico que se encuentra dentro del transmisor no se puede identificar, ya que, como se puede ver en la Figura 3.2, los integrados de la tarjeta (que podrían dar pistas acerca de cómo se está transmitiendo) están ocultos.

.

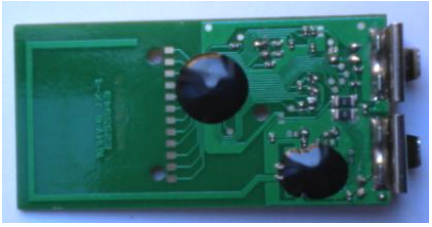

Figura 3.2: Circuito impreso del transmisor de temperatura

3.2 USRP: Universal Software Radio Peripheral

Para el desarrollo de diversos sistemas radio, la empresa Ettus Research LLC cuenta entre sus productos con el denominado USRP (Universal Software Radio Peripheral) [43]. Éste será utilizado para recibir la información del sensor, que será después analizada. El modelo concreto que se utiliza en este trabajo es el USRP1 [44]. Es un dispositivo con gran flexibilidad que permite al ingeniero un rápido diseño e implementación de sistemas radio. Requiere alimentación de 6 voltios DC, y se conecta a un ordenador a través de un puerto USB.

La placa base, mostrada en la Figura 3.3 [45], consiste básicamente de:

- Una FPGA de Altera Cyclone para conseguir procesamiento de señal a una alta velocidad
- 4 convertidores ADC de alta velocidad conectados a la FPGA
- 4 convertidores DAC de alta velocidad conectados a la FPGA
- Chip de interfaz USB 2.0

Sobre esta placa podemos encontrar 4 slots para conectar otras tarjetas: 2 receptoras (RXA y RXB) y 2 transmisoras (TXA y TXB). Cada una de estas tarjetas tiene acceso a 2 de los 4 convertidores (DAC en el caso de las tarjetas transmisoras, ADC en el caso de receptoras).

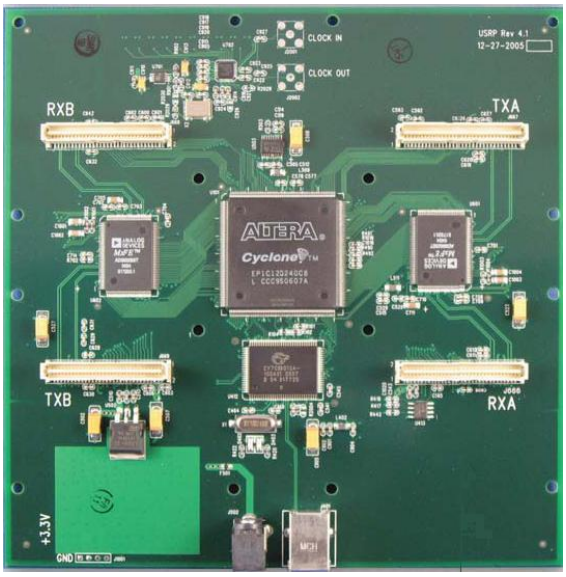

Figura 3.3: Placa principal de USRP

Existen diferentes tipos de tarjetas [46], pero en este caso, se utiliza la denominada "DBSRX". Esta tarjeta es un completo sistema receptor desde 800 MHz a 2 GHz, con una figura de ruido entre 3 y 5 dBs. Como el sensor a examinar trabaja a la frecuencia de 868 MHz, esta tarjeta DBSRX es la apropiada para este trabajo.

3.3 GNU Radio

GNU Radio es una herramienta software de desarrollo de código abierto que proporciona bloques de procesamiento de señal para la implementación de sistemas radio [45]. Funciona bajo el entorno Linux, y puede ser utilizado junto a un hardware RF externo para crear sistemas radio definidos por software, o sin hardware, como un entorno de simulación.

Un sistema radio definido por software (SDR por sus siglas en inglés, software-defined radio), es un sistema RF que realiza el procesamiento de señal requerido mediante software en lugar de utilizar circuitos integrados dedicados a este procesado. La ventaja de SDR es que se pueden crear diferentes tipos de sistemas utilizando el mismo hardware, con tan sólo modificar el código utilizado.

Las aplicaciones construidas con GNU Radio son desarrolladas principalmente en el lenguaje de programación Python. Python es un lenguaje multi-paradigma (también de código abierto) implementado a finales de los años 80. El uso de indentación para delimitar los bloques hace que este lenguaje de programación sea muy legible.

El software GNU Radio incluye un gran número de herramientas para facilitar el trabajo. Una de ellas es 'GNU Radio Companion', un entorno gráfico para crear grafos de flujo de señal y generar código a partir de ellos. Hay muchos bloques predefinidos (filtros, mezcladores, moduladores, demoduladores…) disponibles para el usuario a la hora de construir una aplicación. Trabajar con esta herramienta es verdaderamente intuitivo, por lo que no se requiere un gran esfuerzo para crear eficientes grafos y aplicaciones.

Combinando este software con el USRP (descrito en la sección anterior), se consigue un poderoso sistema SDR. Por ejemplo, se puede conseguir recibir señales de televisión y visualizar los canales, escuchar emisoras radio comerciales (incluso varias simultáneamente), recibir señales satelitales… Con la configuración adecuada, prácticamente cualquier sistema radio puede ser diseñado con esta suma de hardware y software.

4 Decodificación del protocolo propietario

Este capítulo trata de explicar el proceso seguido para decodificar el protocolo propietario del sensor a estudio y cómo extraer la información transmitida. Viene acompañado de diversas figuras, algunas de ellas extraídas de MATLAB‡, con el fin de facilitar la comprensión del procedimiento.

La primera sección describe como captar la señal transmitida por el sensor en la que viaja la información, gracias al USRP. Para ello, se utilizarán algunos scripts que proporciona la herramienta GNU Radio, escritos en lenguaje de programación Python, que permiten por ejemplo, visualizar el espectro de la señal o almacenar la señal digitalizada para poder tratarla después.

En la siguiente parte del capítulo, se analizará dicha señal almacenada. Una vez que se consiga recibir correctamente, el siguiente paso será decodificar la señal para poder distinguir en primer lugar las tramas, y después la información contenida en ellas, presumiblemente codificada digitalmente, así como conocer otros datos significativos como la velocidad de transmisión y la duración de las tramas.

Finalmente, una vez que se obtiene la información en bits, se analizarán las tramas para determinar qué significan, esto es, se identificaran los diferentes campos que puedan aparecer y la información que proporcionan.

4.1 Captura de datos

Antes de capturar los datos, se analiza el funcionamiento del sensor, para tratar de adivinar qué es lo que se va encontrar. Basándose en la documentación que acompaña al sensor, se sabe que la transmisión se realiza aproximadamente a 868 MHz, y que sólo se transmite cada 4 segundos para ahorrar batería. Por tanto, no debería encontrarse una transmisión continua en el tiempo, sino que se deberá ver una ráfaga de datos cada 4 segundos.

Para determinar con exactitud la frecuencia, se puede utilizar un analizador de espectro. Colocando una antena en el conector del analizador, se podrá observar las diferentes radiaciones que recibe dicha antena en el espectro frecuencial. Centrando el análisis en nuestra banda de interés (alrededor de los 868 MHz), lo que se visualiza es ruido en la mayor parte del tiempo, y algunos picos que aparecen y desaparecen rápidamente en torno a los 868.25 MHz. Esto concuerda con la suposición hecha en el párrafo anterior, en la que tan solo hay transmisión de corta duración cada 4 segundos. Pero la mayoría de estos picos no se consiguen ver en el analizador, debido a que por su corta duración, el analizador no los representa. Sin embargo, en el instante en el que el sensor se inicia, esto es, cuando se colocan las baterías, se muestra un pico de mayor duración (una transmisión continua), de unos 3 segundos aproximadamente. Gracias a esto, se puede determinar la frecuencia central, que corresponde con 868.265 MHz.

Con el mismo propósito de conocer los datos referentes al espectro de la señal, se puede utilizar el dispositivo USRP junto con el software GNU Radio para recibir la misma señal que visualizamos en el analizador de espectro, con el archivo ‘usrp_fft.py’. Este script trabaja exactamente como un analizador. La siguiente figura ilustra un ejemplo de lo que muestra este script. Como ya se ha dicho antes, no se obtiene una

‡ © 2011 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

transmisión continua de nuestro sensor, por lo que utilizando las opciones que este script nos ofrece se puede observar que la frecuencia central coincide con la estimada mediante en analizador. La línea azul muestra la componente frecuencial que se está recibiendo en cada instante, mientras que la línea verde muestra el valor máximo alcanzado en cada componente.

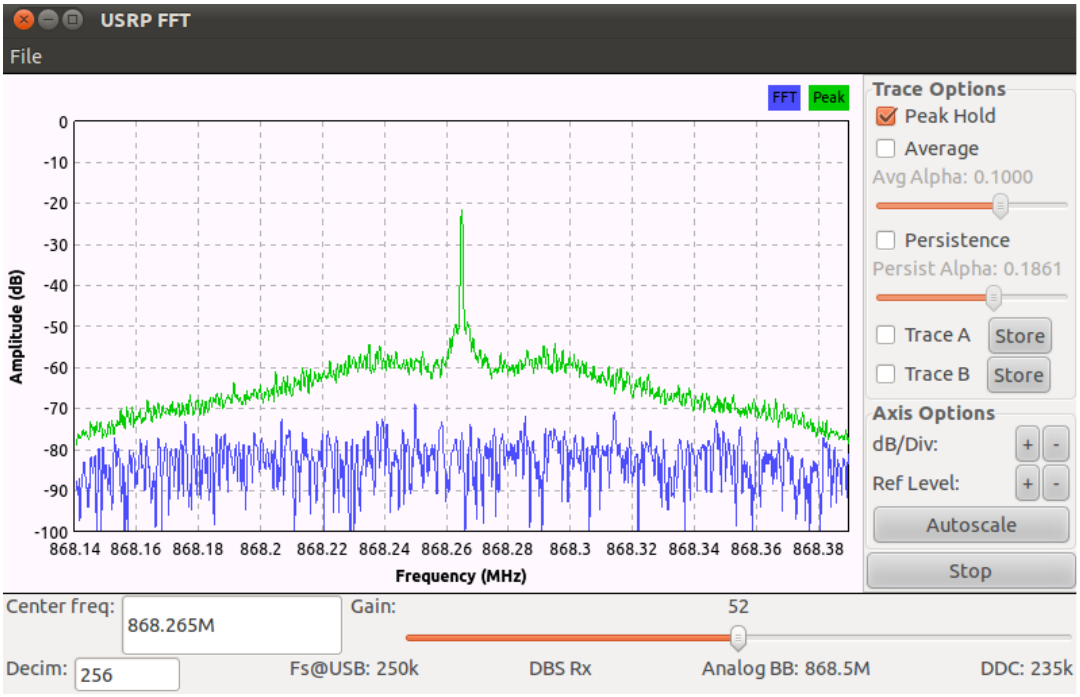

Figura 4.1: Espectro de la señal, utilizando el script "*usrp_fft.py*"

Una vez hecho esto, se puede pasar al siguiente paso, que no es otro que registrar la señal. Para este objetivo, se utiliza el archivo ejecutable escrito en Python 'usrp_rx_cfile.py' junto al USRP. Este script debe ir acompañado de diversos argumentos para especificar la frecuencia, el numero de muestras a registrar, la ganancia que queremos usar, el tipo de datos de las muestras, y la decimación (este valor fija el ancho de banda de la captura). Ejecutando este script en un terminal Linux se crea un archivo que contiene los datos recogidos en la frecuencia usada por el sensor de temperatura. Dichos datos están en formato binario.

4.2 Decodificación de la señal

En este punto, es necesaria la ayuda de alguna herramienta para tratar y analizar los datos registrados anteriormente. Aunque existen diferentes posibilidades y en primer momento se barajó la opción de utilizar el programa GNU Octave por estar trabajando en el entorno Linux, finalmente se escogió MATLAB, por ser una potente herramienta ya conocida que ofrece muchas posibilidades en cuanto a procesado de señal se refiere.

El primer paso a seguir es conseguir visualizar la señal. La función 'read_complex_binary.m' lee la información en formato binario de un archivo y los convierte en un vector de datos de tipo complejo. Una vez hecho esto, se representan los datos. La siguiente imagen muestra el valor real de los elementos del vector.

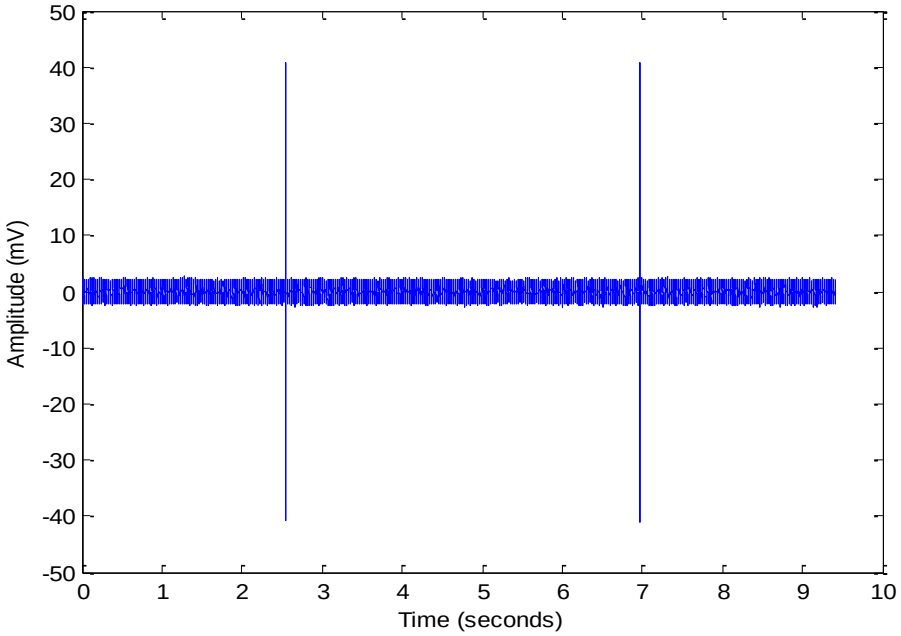

Figura 4.2: Dos transmisiones diferentes, separadas 4 segundos

Como se había predicho, la mayor parte del tiempo solo se recibe ruido, y se observa una transmisión de muy corta duración cada 4 segundos aproximadamente. Cada una de las transmisiones se puede suponer que equivale a una trama.

La Figura 4.3 muestra una parte de una de las 2 tramas representadas en la anterior ilustración. Solo se muestra una fracción de ellas para dar una mejor perspectiva y poder entender de mejor manera como la señal cambia con el tiempo.

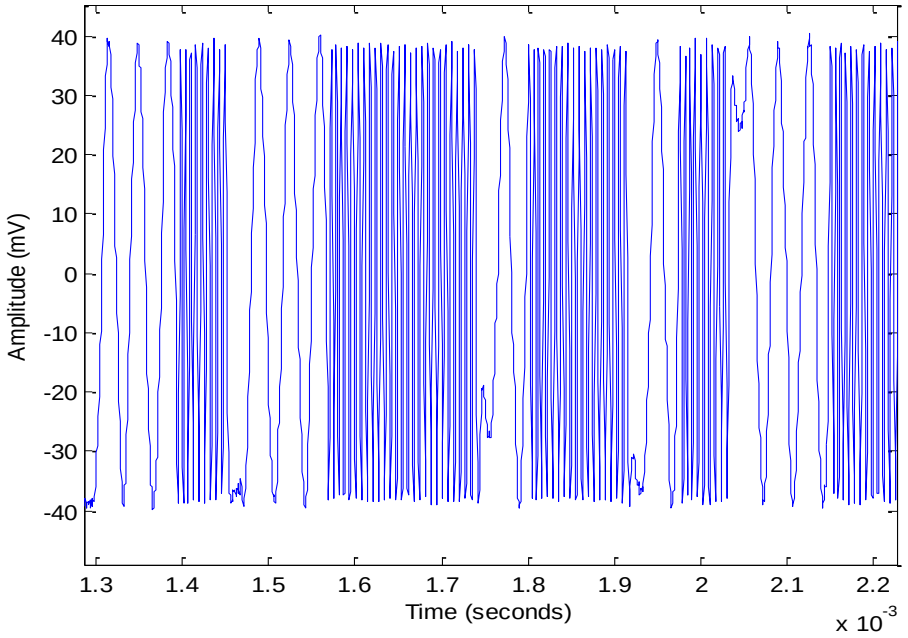

Figura 4.3: Fragmento de una trama

Como se observa, se trata de 2 oscilaciones con diferente frecuencia, y esto ocurre (la alternancia entre las 2 frecuencias) a lo largo de las 2 tramas capturadas. Esto lleva a

pensar que el transmisor está utilizando una modulación digital de frecuencia (FSK, Frequency Shift Keying). Si se está en lo cierto, una de las frecuencias corresponderá a '1' lógico, y la otra a '0' lógico. Como el transmisor tiene poca cantidad de información que comunicar al receptor, la modulación utilizada para codificar los datos debiera ser simple, ya que de otra manera se necesitaría mayor complejidad computacional y por tanto, mayor energía y mayor consumo en el sensor que haría agotar las baterías en menor tiempo. En este punto se puede además determinar el tiempo de transmisión de una trama, que corresponde aproximadamente a 3.71 milisegundos.

El teorema de Nyquist - Shannon establece que la frecuencia máxima que se puede representar sin cometer errores es la mitad de la frecuencia de muestreo, que en este caso es de 2 MHz. En la Figura 4.4 se representa, por tanto, el espectro frecuencial hasta 1 MHz. Destaca la presencia de 2 picos significativamente mayores que los demás, y pequeños picos a frecuencias equiespaciadas. Esto corresponde totalmente con una modulación 2-FSK, dónde los picos mayores se deben a las 2 frecuencias de oscilación de cada uno de los bits (aproximadamente 28.28 KHz y 148.96 KHz).

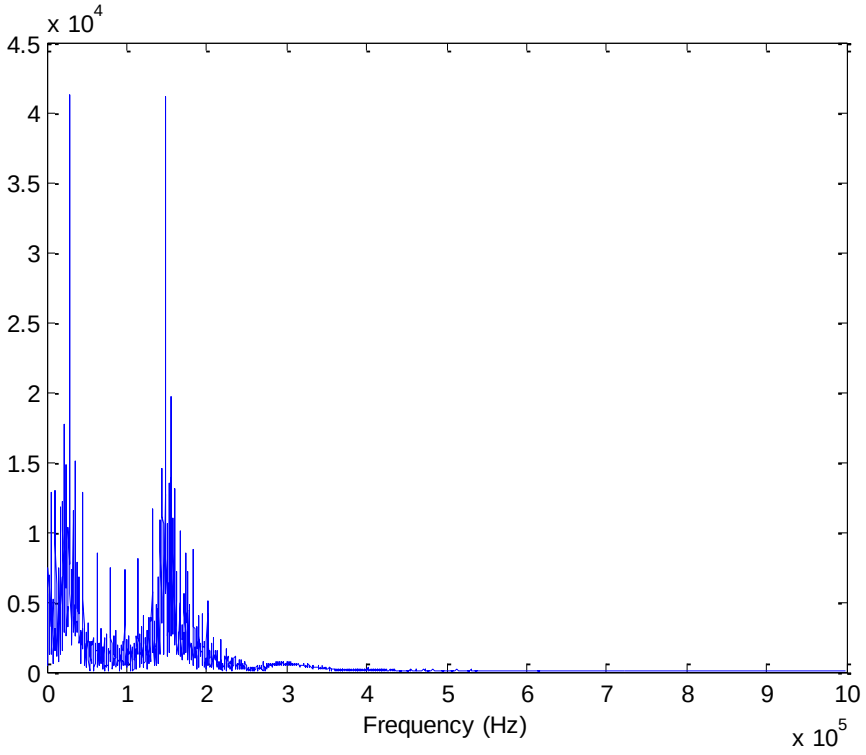

Figura 4.4: Espectro frecuencial de una trama

Una vez que se asume que la modulación es conocida y es FSK, y puesto que se va a necesitar un número considerable de tramas diferentes (que representen temperaturas diferentes), es necesario extraer los datos en bits de cada trama, sabiendo que una frecuencia corresponde con '1' y la otra con '0'. No es posible realizar este paso por inspección visual de las señales, ya que llevaría mucho tiempo y además es probable que se cometan bastantes errores. Por lo que es necesario automatizar este proceso. Para ello, se han desarrollado algunos scripts en MATLAB (ver anexo A). La idea para conseguir la información en bits es en primer lugar, determinar cuándo comienza cada trama para poder separarla de la demás parte de señal, que en su mayor parte es ruido. Y una vez hecho esto, procesar cada bit por separado y determinar si corresponde a '1' o '0' según su frecuencia de oscilación.

La Figura 4.5 ilustra la misma parte de señal mostrada en la figura anterior (en rojo) y los bits extraídos (círculos en azul). Como no se dispone de más información, se presupone que la frecuencia más alta corresponde a '1' y la baja a '0'. Más adelante se puede comprobar que esta suposición es cierta.

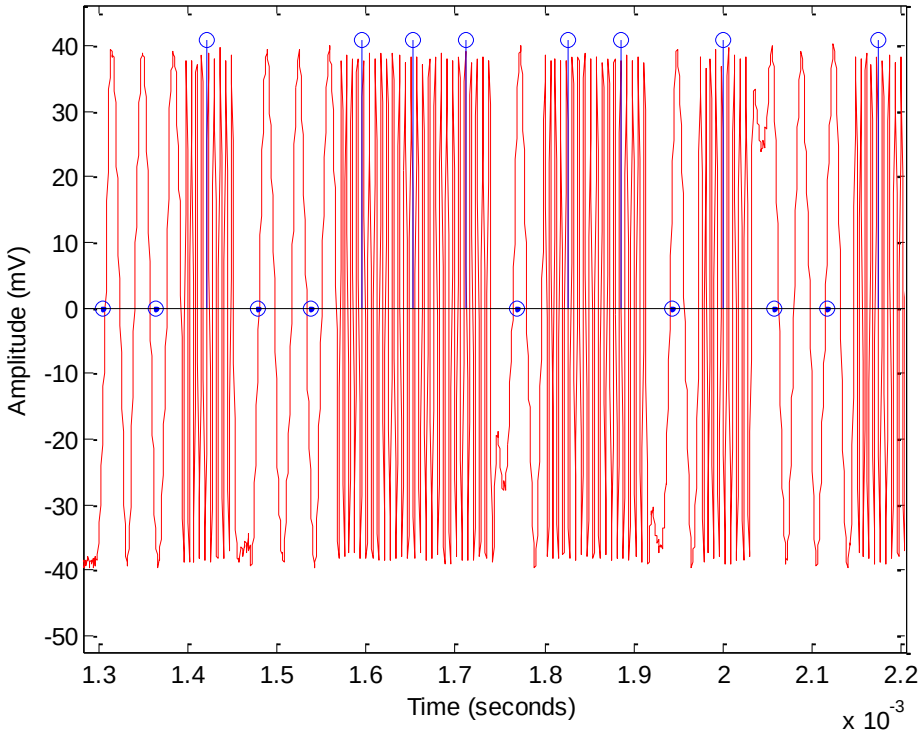

Figura 4.5: Fragmento de una trama. Datos extraídos

4.3 Análisis de los datos

En este punto, ya es posible capturar la señal y extraer los bits que conforman la información. Pero ahora surgen otras preguntas: ¿Que representan todos esos bits? ¿En qué campo se está representando la temperatura? ¿Y como está codificada esa información? Todas estas cuestiones son las que se van a analizar y responder en esta sección.

De igual manera que se hizo anteriormente, antes de entrar en el análisis, se hace una ligera reflexión acerca de lo que se puede encontrar en la trama. Al comienzo de ella, se podría hallar un preámbulo, empleado para que el receptor sea capaz de sincronizarse, esto es, que cuando el receptor vea este preámbulo, pueda sincronizar su reloj y sepa que lo que viene a continuación es información. La trama puede contener también un identificador del transmisor, para poder distinguir entre diferentes transmisores que estén enviando información simultáneamente en el mismo entorno. Finalmente, al término de la trama puede aparecer un campo con el objetivo de detectar errores que se hayan podido introducir durante la transmisión o la decodificación en el receptor. Por tanto, tal vez se encuentre una suma de verificación (checksum), o un código de comprobación de redundancia cíclica (CRC). Además de todo ello, por supuesto, deberá hallarse el campo que contiene la información de temperatura.

Para conseguir la decodificación la información, es preciso capturar un gran número de tramas. El rango de temperaturas que el sensor en cuestión puede transmitir está entre -39.9º y 59.9º Celsius, por lo que se intentará cubrir ese rango en la medida que

sea posible. Para alcanzar algunas temperaturas se calentará o se enfriará el sensor artificialmente (utilizando focos de luz o un congelador, respectivamente).

El método a seguir es el siguiente. Se capturarán diferentes transmisiones utilizando el USRP, mientras que simultáneamente, se irán anotando la temperatura correspondiente que indicará el display de la estación meteorológica. Después, se aplicarán a las señales registradas los scripts desarrollados en MATLAB y una vez obtenidos los bits, se almacenarán en una hoja de cálculo para facilitar su organización y su posterior análisis en detalle. La Figura 4.6 muestra una parte de dicha hoja de cálculo. El conjunto completo de todos los datos recogidos se encuentra en el anexo B. Los diferentes bytes y campos han sido coloreados para facilitar la tarea.

25,0	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 0 1 0 0 0 0	0 1 1 0 1 0 1 0	0 0 1 1 0 1 0 1
25,1	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 0 1 0 0 0 1	0 1 1 0 1 0 1 0	1 1 0 0 0 0 0 1
25,2	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 0 1 0 0 1 0	0 1 1 0 1 0 1 0	1 1 1 0 1 1 0 0
25,3	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 0 1 0 0 1 1	0 1 1 0 1 0 1 0	0 0 0 1 1 0 0 0
25,4	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 0 1 0 1 0 0	0 1 1 0 1 0 1 0	1 0 1 1 0 1 1 0
25,5	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 0 1 0 1 0 1	0 1 1 0 1 0 1 0	0 1 0 0 0 0 1 0
25,6	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 0 1 0 1 1 0	0 1 1 0 1 0 1 0	0 1 1 0 1 1 1 1
25,7	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 0 1 0 1 1 1	0 1 1 0 1 0 1 0	1 0 0 1 1 0 1 1
25,8	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 0 1 1 0 0 0	0 1 1 0 1 0 1 0	0 0 0 0 0 0 1 0
25,9	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 0 1 1 0 0 1	0 1 1 0 1 0 1 0	1 1 1 1 0 1 1 0
26,0	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 1 0 0 0 0 0	0 1 1 0 1 0 1 0	1 0 0 0 0 1 1 1
26,1	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 1 0 0 0 0 1	0 1 1 0 1 0 1 0	0 1 1 1 0 0 1 1
26,2	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 1 0 0 0 1 0	0 1 1 0 1 0 1 0	0 1 0 1 1 1 1 0
26,3	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 1 0 0 0 1 1	0 1 1 0 1 0 1 0	1 0 1 0 1 0 1 0
26,4	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 1 0 0 1 0 0	0 1 1 0 1 0 1 0	0 0 0 0 0 1 0 0
26,5	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 1 0 0 1 0 1	0 1 1 0 1 0 1 0	1 1 1 1 0 0 0 0
26,6	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 1 0 0 1 1 0	0 1 1 0 1 0 1 0	1 1 0 1 1 1 0 1
26,7	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 1 0 0 1 1 1	0 1 1 0 1 0 1 0	0 0 1 0 1 0 0 1
26,8	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 1 0 1 0 0 0	0 1 1 0 1 0 1 0	1 0 1 1 0 0 0 0
26,9	1 0 1 0 1 0 1 0	0 0 1 0 1 1 0 1	1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1	1 1 1 0	0 1 1 0	0 1 1 0 1 0 0 1	0 1 1 0 1 0 1 0	0 1 0 0 0 1 0 0

Figura 4.6: Hoja de cálculo con algunos de los datos recogidos de temperaturas positivas. El campo de temperatura esta marcado en naranja

Lo primero que se observa es que existen algunos bits que cambian entre diferentes medidas y otros que no. Cuando se toman diferentes medidas de temperaturas utilizando el mismo sensor, se comprueba que los únicos campos que cambian son los bits coloreados en naranja y el último byte. A primera vista, este último byte no parece seguir ninguna regla ni orden lógico, por lo que se asume que se trata de una suma de verificación o un código CRC. Por tanto, a continuación se van a inspeccionar los bits coloreados en naranja, ya que parecen ser los datos que codifican la temperatura.

4.3.1 Campo de temperatura

El análisis comienza con los valores positivos de temperatura. Se van a examinar, como ya se ha dicho, los 12 bits naranjas, (bits 37 a 48). Atendiendo a los valores de temperatura desde 25.0 a 25.9, se puede observar que cuando la temperatura asciende una décima, este campo se incrementa en una unidad. En concreto, si se miran los 4 últimos bits en decimal, estos van desde 0 hasta 9. Además, cuando la temperatura cambia de 25.9 a 26.0, estos bits (del 45 al 48) cambian de ‘1001’ (9 decimal) a ‘0000’ (0 decimal), Se puede deducir que en este caso, los últimos 4 bits de las celdas coloreadas en naranja representan la décima de la temperatura codificada en BCD (Binary Coded Decimal). Comprobando esta hipótesis a lo largo de todos los datos

registrados en valores positivos de temperatura, se cumple siempre, por lo que se presume que esta suposición es cierta.

Siguiendo este mismo patrón, se centra ahora la atención en los siguientes bits (del 41 al 44). Entre 25.0º y 25.9º, el valor de esos bits es '0101', en decimal se corresponde con 5. Para valores entre 26.0 y 26.9 ocurre lo mismo, estos 4 bits corresponden con 6 en decimal y así sucesivamente. Por tanto, parece que de nuevo el valor de unidad de temperatura está codificado en BCD. Igual que se procedió antes, se comprueba que esto se cumple para todos los valores positivos de temperatura.

En este punto podría pensarse que el valor de las decenas también estará codificado en BCD en los primeros 4 bits naranjas (37 a 41). Sin embargo, entre 10.0º y 19.9º este valor es '0101' (5 decimal), entre 20.0º y 29.9º es '0110' (6 decimal), entre 30.0º y 39.9º es '0111' (7 decimal) y así sucesivamente. Cuando la decena aumenta, este campo también aumenta en uno, pero no corresponde con el valor decimal esperado si fuera BCD. Para solventar este problema, se van a examinar los valores negativos de temperatura, algunos de ellos mostrados en la siguiente imagen.

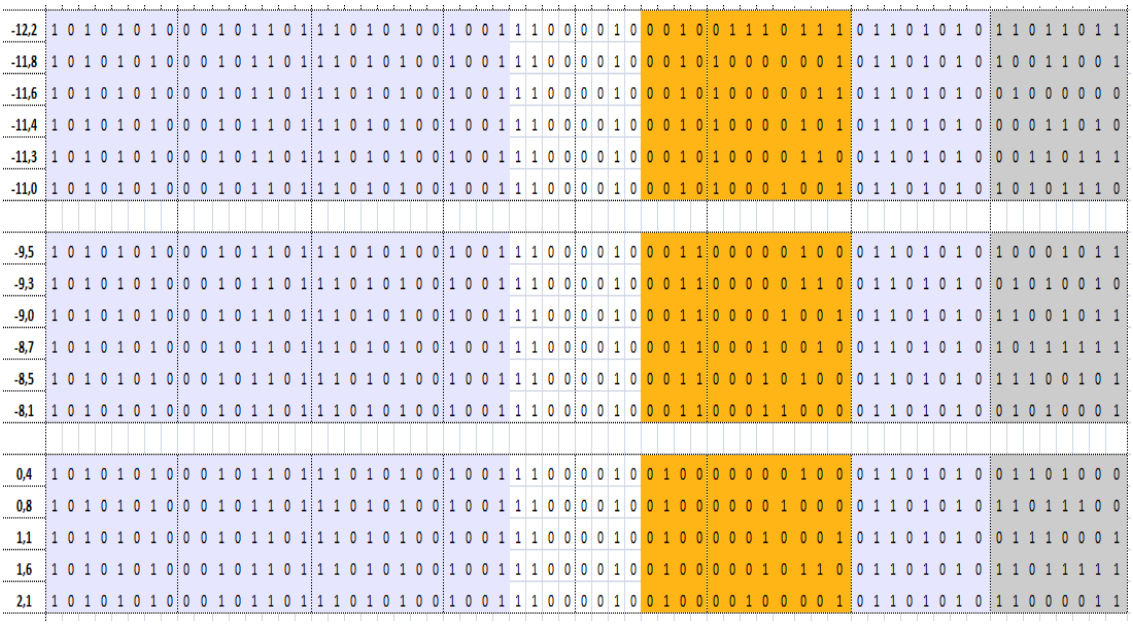

Figura 4.7: Fragmento de la hoja de cálculo con algunos datos de temperaturas negativas

La temperatura adquirida más baja corresponde con -12.2º. No obstante, es suficiente para deducir como están codificados estos bits. Si seguimos la pauta que se observa para temperaturas negativas y se piensa en el valor más bajo que se puede sensar (que es -39.9º), se comprueba cómo este valor corresponderá con '0000 0000 0000'. Mientras que la temperatura más alta que el transmisor permite, 59.9º, corresponderá con '1001 1001 1001'. Basándonos en esto, vemos que cada grupo de 4 bits se comporta como un contador de módulo 10, teniendo en cuenta el *carry* para el siguiente grupo de 4 bits. Para facilitar el trabajo a la hora de la decodificación, se busca una regla que permita calcular el valor de temperatura a partir de este campo de bits. El algoritmo a seguir es el siguiente:

- Decodificar el campo de temperatura en BCD, dividirlo entre 10 y después, restarle 40 al resultado. Ejemplo: '0100 0010 0001' (421 en BCD) corresponde con 2.1º; (421/10) – 40 = 2.1

- Si el resultado del paso anterior da un valor menor que cero, se le suma una décima. Ejemplo: '0010 0111 0111' (288 en BCD) corresponde, según la hoja de cálculo, con -12.2; (277/10) – 40= -12.3 <0 →-12.3 + 0.1= -12.2

Comprobando este algoritmo a lo largo de todos los datos registrados, se cumple en todos los casos, por lo que ya se ha descubierto la codificación del propietario del sensor para el campo de temperatura.

4.3.2 Campo de identificador

Puesto que se dispone de 2 sensores del mismo tipo, se dispone a examinar también las posibles diferencias entre transmisiones de uno y otro para el mismo valor de temperatura. Se observa como desde el bit 29 al 35, varían algunos de ellos según el transmisor. Además, cuando se reinicializa el sensor, esto es, cuando se quitan las baterías y se colocan de nuevo, este campo también varía, a pesar de que se trate del mismo sensor. Por tanto, la conclusión que se puede extraer es que en el momento en que se reinicia el sensor, se genera un código identificador aleatorio de 7 bits. De esta manera, no es fácil encontrar el caso en el que 2 sensores con el mismo identificador se hallen en un mismo entorno, lo que podría producir errores.

Este campo de identificador del sensor es importante. Puesto que el *gateway* o pasarela a diseñar debe ser capaz de tratar con distintos sensores a la vez, gracias a este campo se podrá distinguir entre ellos. Por tanto, teóricamente se podrá estar recibiendo la información de 127 sensores diferentes de este tipo, aunque en la práctica será poco probable que entre un número grande de sensores que se acerquen a este máximo teórico no haya repeticiones en el campo de identificador. Una posible solución a este problema sería dotar de cierta inteligencia a los sensores, de manera que a la hora de seleccionar el valor que lo identifica establezca conexión con el receptor para que éste le indique si el valor escogido está o no disponible.

4.3.3 Código CRC-8

El último byte parece ser siempre aleatorio, ya que no sigue un orden lógico. Como ya se indicó anteriormente, se sospecha que es algún tipo de CRC o suma de verificación, pero el algoritmo empleado no es a priori conocido.

A pesar de que no es absolutamente necesario conocer cómo se genera este byte, ya que los campos más importantes para el *gateway* son el de temperatura y el identificador, se piensa que sí puede resultar útil. Es por ello que tras varias pruebas con diferentes polinomios típicos de CRC8, y con las indicaciones precisas del profesor director del proyecto, se llega a la conclusión de que este byte es un código CRC8 que sigue el polinomio $x^8+x^5+x^4+1$, y que se aplica sobre 32 bits, del 25 al 56. En este proceso se hizo uso de calculadora CRC online que se encontró en Internet [47].

4.3.4 Resto de la trama

La demás parte de la trama no cambia nunca, por lo que no se puede conocer con certeza lo que se está representado. Se puede suponer que corresponde con algún código que indique la empresa o la familia de productos de este sensor, pero no es posible en este momento determinarlo con seguridad. Destacar que el primer byte, que siempre es '1010 1010', se puede suponer que corresponde con el preámbulo utilizado para que el receptor se sincronice correctamente.

En resumen, se ha analizado cómo transmite el sensor y se ha conseguido decodificar el protocolo propietario que utiliza. La Figura 4.8 muestra el contenido de una trama, según las conclusiones que se han extraído a lo largo de este capítulo. Para

mayor información acerca de los datos recogidos, es posible consultar el archivo ‘Temperaturas.xls’, ya que se incluye en el CD que acompaña a esta memoria.

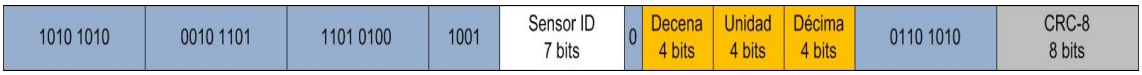

Figura 4.8: Contenido de una trama

5 Gateway

En este apartado se va a describir el diseño del *gateway*. En una primera sección, se definirán los componentes principales de hardware que van a formar parte de este dispositivo. Después se detallará el proceso seguido para diseñar el layout del circuito impreso con la herramienta EAGLE, así como el montaje de los componentes y la verificación del diseño.

5.1 Componentes principales

El *gateway* o pasarela que se va a diseñar consta básicamente de un microcontrolador, un receptor (también puede utilizarse como transmisor) y un controlador Ethernet. Adicionalmente, se debe contar con un módulo que haga al *gateway* compatible con la tecnología Power Over Ethernet (ver apartado 2.4) y obtenga la tensión de alimentación deseada. Los elementos específicos que se han escogido para cada una de estas tareas se explican en las siguientes secciones. En la siguiente figura se puede ver un diagrama de bloques del diseño del gateway.

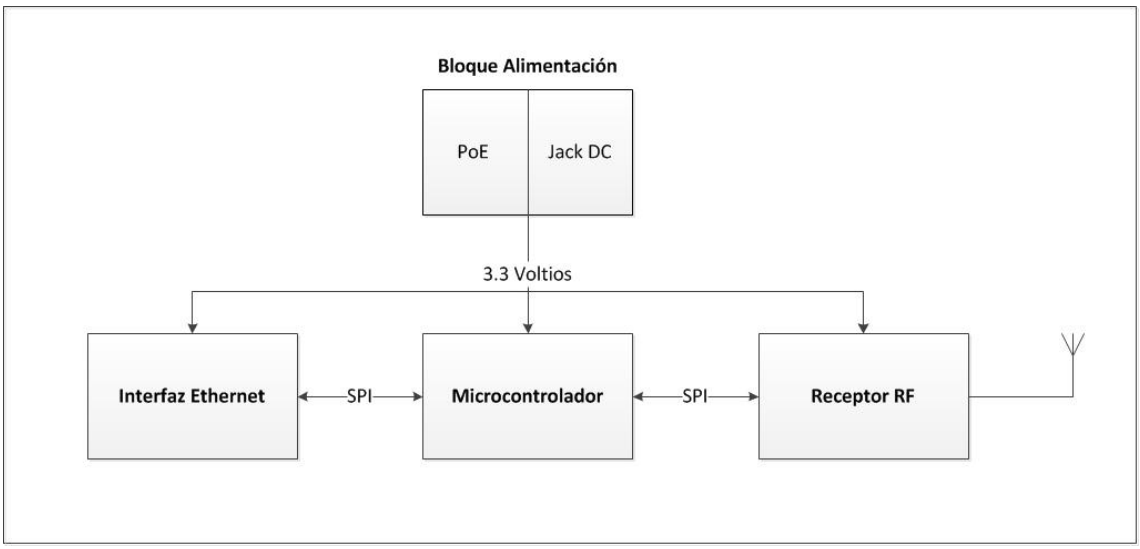

Figura 5.1: Diagrama de bloques del diseño

5.1.1 Microcontrolador MSP430

El núcleo del dispositivo a diseñar es una unidad microcontroladora (MCU por sus siglas en inglés, MicroController Unit). Se ha elegido un MCU de la familia MSP430 de Texas Instruments [48]. En la página web de este fabricante se pueden hallar las principales características de esta familia de microcontroladores:

- Muy bajo consumo
- Posibilidad de conexión con un gran número de periféricos
- Arquitectura con un set de instrucciones reducido de 16 bits (16-bit RISC)
- Cinco estados de bajo consumo
- Oscilador controlado digitalmente (DCO)
- Rápido "despertar" de estados de bajo consumo

Por cumplir las especificaciones necesarias y por el conocimiento del mismo que se tiene en el grupo de trabajo en el que se enmarca este proyecto, se escoge un microcontrolador de la familia MPS430F54xx. Estos controladores incluyen 3

contadores de 16 bits, convertidor ADC de 12 bits de alto rendimiento con 16 canales analógicos, multiplicador de 32 bits, acceso directo a memoria (DMA, Direct Memory Access), varios interfaces universales de comunicación serie (USCI, Universal Serial Communication Interfaces), y puede utilizar señales de reloj de hasta 25 MHz.

La Tabla 5.1 muestra las diferencias entre los diversos controladores de esta familia MPS430F54xx. Toda la información se ha obtenido de la página web del fabricante, Texas Instruments.

Tabla 5.1: Comparación de la familia de Microcontroladores MPS430F54xx

MSP430F54XX	Flash (KB)	SRAM (B)	GPIO	Pin/Package	USCI_A	USCI_B
					UART/LIN/ IrDA/SPI	I2C & SPI
MSP430F5418A	128	16384	67	80LQFP	2	2
MSP430F5419A	128	16384	87	100LQFP	4	4
MSP430F5435A	192	16384	67	80LQFP	2	2
MSP430F5436A	192	16384	87	100LQFP	4	4
MSP430F5437A	256	16384	67	80LQFP	2	2
MSP430F5438A	256	16384	87	100LQFP	4	4

Para determinar el microcontrolador específico a utilizar, se escoge el que cuenta con mayor memoria flash, y además, aquel cuyo encapsulado es más sencillo, entendiéndolo como el de menor número de pines. Por tanto, se decide trabajar con el microcontrolador MSP430F5437A [49].

Para el diseñ6o del *gateway*, un factor importante es el uso de las comunicaciones serie entre el MCU y el módulo RF y el chip Ethernet. Por tanto, se necesitan al menos 2 interfaces serie. El MCU escogido proporciona 2 puertos SPI, que serán usados simultáneamente para conectar por una parte el controlador Ethernet y por otra, el receptor a 868 MHz. Además, se pueden citar otras singularidades de este dispositivo que lo hacen particularmente adecuado para el diseño del *gateway*, como son su bajo consumo (haciendo posible el uso de la tecnología PoE) y el hecho de que para su programación se pueda utilizar una herramienta como Code Composer Studio [50], proporcionado por el fabricante Texas Instruments, que facilita la labor.

5.1.2 Receptor de RF

Existen multitud de soluciones comerciales posibles a la hora de diseñar e implementar un receptor RF. Para simplificar el proceso, se opta por escoger un módulo del mismo fabricante que el controlador. Texas Instruments oferta diferentes transmisores/receptores que operan por debajo de 1 GHz, que pueden adecuarse a las necesidades de este diseño.

Teniendo en cuenta el análisis del anterior capítulo, y analizando brevemente las diversas soluciones de las que se dispone, se ha escogido el integrado CC1101 [51]. Este receptor tiene gran sensibilidad, soporta altas tasas de bits y puede trabajar con distintos tipos de modulación. A pesar de que está fuera del objeto de este proyecto,

también puede operar como transmisor, por lo que se podría aprovechar esto para comunicarse bidireccionalmente con sensores que soporten esta opción.

Texas Instruments ofrece también otra solución para este tipo de dispositivos. Se basa en unos chips que incorporan tanto el controlador como el módulo receptor, denominados SoC (System-on-Chip). Por ejemplo, el CC430F6137 [52] integra un microcontrolador MSP430 junto a un receptor RF basado en el módulo CC1101. Es una opción interesante, ya que se evitan las pistas en el circuito impreso que conectan el MCU y el módulo de RF. Sin embargo, cómo se expondrá a continuación, para el diseño se ha optado por realizar el *gateway* en 2 partes: la placa principal (con conexión Ethernet y el controlador) y la placa de RF (que se conectará a la principal mediante un zócalo), por lo que se rechaza la posibilidad de utilizar un SoC.

5.1.3 Controlador Ethernet

El módulo controlador de Ethernet es responsable de conectar el dispositivo a diseñar con la red LAN. Además, el conector RJ45 se utilizará para suministrar la energía requerida por el *gateway*, haciendo uso de la tecnología PoE. Se ha escogido el integrado ENC28J60 de la empresa Microchip [53] por contar con una interfaz SPI para comunicarse con el procesador. Para controlar la comunicación Ethernet, se deberá leer y escribir en los registros de control que proporciona a través del puerto SPI. Este controlador integra también sendos buffers RAM para los paquetes de datos enviados y recibidos, evitando así la necesidad de memoria externa. Adicionalmente, este controlador incluye todo lo necesario para implementar las capas MAC y PHY.

5.1.4 Alimentación mediante PoE y Regulador DC

Para poder sacar provecho de la tecnología PoE, se necesita un controlador que cumpla las funciones que la norma 802.3at establece. En el apartado 2.4 se han especificado las fases que se deben cumplir para conseguir alimentación a través de Ethernet. Existen diversas soluciones en el mercado que realizan todo este proceso. Para facilitar la labor, se han escogido encapsulados del mismo fabricante del microcontrolador, Texas Instruments.

Para las fases de detección, clasificación y control, se ha escogido el chip TPS2375 [54], que es un controlador para PD´s, mientras que como regulador DC/DC se ha optado por el TL2575HV-33 [55], que transforma tensiones de entrada de entre 4.75 y 60 voltios en una tensión de salida de 3.3 voltios, que es la adecuada para alimentar todo el *gateway*. La configuración de estos 2 componentes se mostrará más adelante en este mismo capítulo.

5.2 Diseño del Hardware

Para el diseño del PCB se ha utilizado el software EAGLE (del acrónimo Easily Applicable Graphical Layout Editor). Es una herramienta que permite de manera simple crear circuitos impresos. Para ello, cuenta con un editor de esquemáticos y un editor de layout.

Se ha dividido el trabajo en 2 bloques. Por un lado, el diseño de la parte del receptor de RF, y por otro, el de la placa principal que contiene la interfaz Ethernet, el microcontrolador y un zócalo específico para conectar la placa RF. Se ha hecho de esta manera para dar mayor flexibilidad al diseño. Puesto que en este PFC se ha centrado el trabajo en un sensor cuya frecuencia de transmisión está en los 868 MHz, la elección de los componentes y de su valor está predeterminada por esta frecuencia. La idea es que en el futuro, si se quiere utilizar este *gateway* para sensores en otra banda se pueda

aprovechar este trabajo, con tan sólo conectar a la placa principal el nuevo receptor sintonizado a la frecuencia deseada.

5.2.1 Diseño del módulo receptor

En primer lugar, se ha creado el esquemático, siguiendo el circuito típico que proporciona la hoja de características del receptor CC1101 [51]. Los valores de los componentes que acompañan a este integrado vienen especificados, por lo que no es necesario realizar ningún cálculo adicional. Básicamente, consiste en un balun (para acoplar un elemento desbalanceado como es la antena, a una entrada diferencial, como la del CC1101), una red LC transformadora de impedancias, un filtro para reducir la emisión a 699 MHz, un oscilador de cristal y condensadores de desacoplo. Las señales de la interfaz SPI que se deben conectar al procesador, se llevan a un conector de 8 pines. Destacar que para la antena, se ha optado por colocar un conector hembra SMA en el circuito impreso. La siguiente figura muestra el esquemático diseñado.

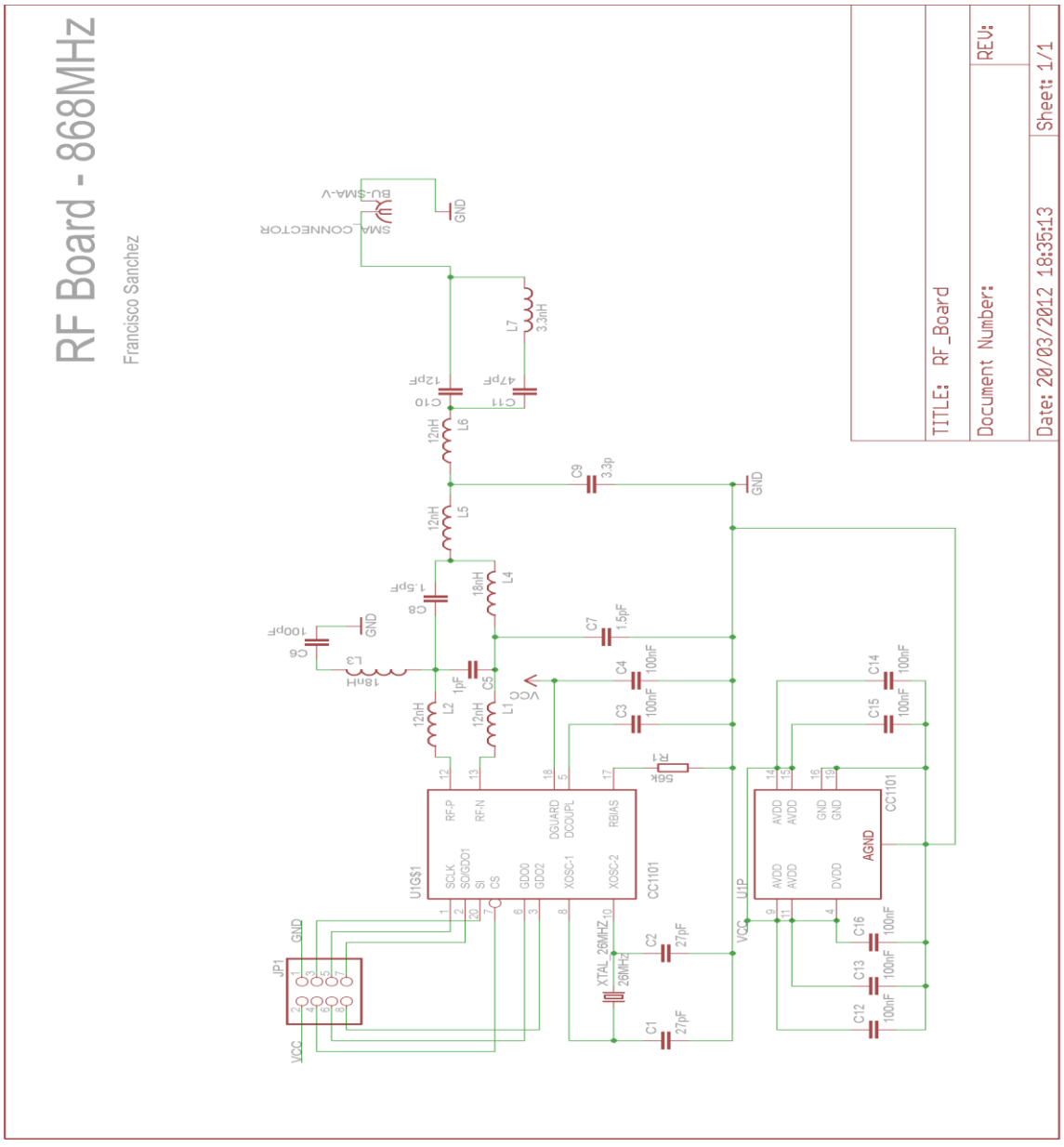

Figura 5.2: Esquemático receptor RF

A continuación, en el editor de layout, se han ubicado los componentes y se han creado las pistas necesarias para unirlos correctamente, siguiendo las recomendaciones básicas que el profesor director del proyecto ha proporcionado. El diseño en este caso ha sido sencillo, ya que el número de pistas necesarias no es elevado.

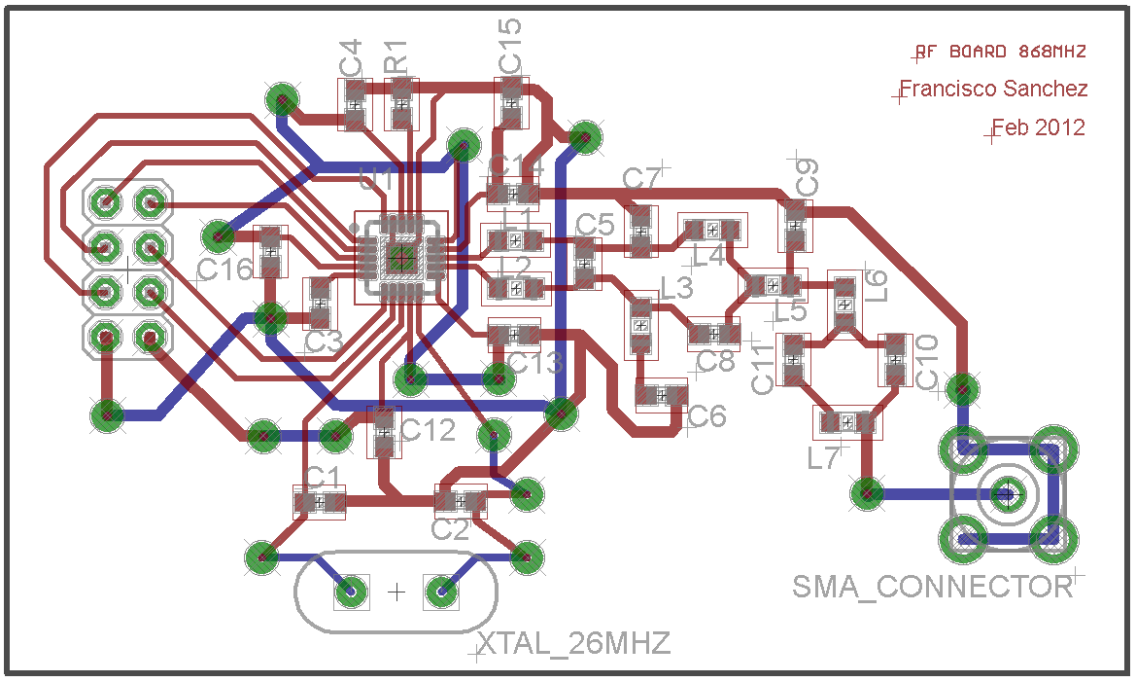

Figura 5.3: Aspecto del layout del receptor

En la Figura 5.3 se observa el diseño final. Nótese que las líneas en color rojo se corresponden con pistas en la capa superior de la placa, mientras que las de color azul, se emplazan en la capa inferior. Los componentes usados para los condensadores y bobinas son encapsulados SMD, por lo que el tamaño real de este circuito impreso es pequeño (63 x 37 mm). En el anexo C se encuentra la lista de componentes, imágenes de la placa y una descripción más detallada de este layout. Los archivos originados con el software EAGLE se pueden consultar en el CD que acompaña a esta memoria.

5.2.2 Diseño del módulo principal

El diseño de la placa principal se puede dividir en varios apartados. Por un lado, la interfaz Ethernet que se comunicará con un PC. Por otro, el bloque encargado de la alimentación del sistema, en el que se pretende aprovechar la tecnología PoE. Y por último, el microcontrolador con sus correspondientes conexiones SPI y el interfaz JTAG (Joint Test Action Group), definido en la norma IEEE 1149.1 [56], para permitir la programación y depuración del software. Además de ello, también se añadirá algún botón (como el de reset) y varios leds.

En primer lugar, se deben conocer los requerimientos referentes a la potencia que va a consumir el *gateway*. El valor de voltaje de entrada máximo que soporta el regulador escogido es de 60 v. Todos los encapsulados necesitan una alimentación de 3.3 voltios, por lo que esa será la tensión de alimentación que se obtendrá con el regulador DC. Por otra parte, según las correspondientes hojas de características, el microcontrolador demanda una corriente máxima de 10 mA, el controlador Ethernet, 180 mA y el módulo receptor RF, 33 mA. Además de ello, se debe tener en cuenta el consumo de la demás circuitería del *gateway*. Por tanto, se escoge un total de corriente máxima de 300 mA.

Estos 3 valores (voltaje de entrada máximo, voltaje de salida y corriente máxima) son necesarios a la hora de configurar adecuadamente el regulador DC.

Como se ha indicado anteriormente, se desea aprovechar en el *gateway* la tecnología PoE. Para que funcione correctamente, se necesitará un dispositivo externo específico (por ejemplo, un inyector PoE) que reúna tanto datos como alimentación en el mismo cable Ethernet que se conectará al *gateway*. Nótese que el voltaje máximo que se puede suministrar a través de esta tecnología es de 57 voltios, por lo que no supera el voltaje máximo de entrada en el regulador (60 voltios). Los 2 pares de pines del RJ45 que llevan las señales PoE se conectan a sendos puentes de diodos rectificadores de onda completa. Después, se llevan al controlador PoE. La configuración de los pines de este chip (extraída de su hoja de características) permite la correcta implementación de las fases PoE especificadas en el apartado 2.4. La salida de este encapsulado se deberá llevar a la entrada del regulador DC.

También se va a incluir una toma de alimentación de tensión continua (similar a la de un ordenador portátil), para facilitar la tarea a la hora de programar y utilizar el *gateway* en el caso de no contar con un inyector PoE. A través de un par de jumpers y un par de LEDs situados en la PCB principal se podrá seleccionar la alimentación a través de PoE o mediante el jack DC y comprobar que funciona de manera correcta.

Para regular y adecuar la tensión DC se utiliza el chip TL2575HV-33[55]. En primer lugar se ha colocado un diodo zéner, para limitar la tensión de entrada y asegurar que no se excede el máximo que el encapsulado soporta y así evitar posibles daños. De nuevo, para la configuración de los pines y los componentes que rodean este módulo, se ha atendido a su hoja de características. En ella se describe como calcular los valores de los componentes según los requerimientos (tensión de entrada máxima, tensión de salida y corriente máxima de salida). Se utiliza un condensador de entrada y otro de salida por cuestiones de estabilidad, un diodo Schottky para reducir el ruido, un inductor toroide y un filtro LC para minimizar el rizado de la tensión de salida. Esta tensión obtenida, debidamente regulada a 3.3 voltios, suministrará la energía necesaria a todos los componentes del *gateway*.

La pieza central de este *gateway* es el microcontrolador de Texas Instruments. Desde él se programará y configurará tanto el módulo de RF como la conexión Ethernet. A pesar de contar con 80 pines, aproximadamente la mitad de ellos no se van a utilizar (muchos corresponden con pines de entrada/salida digital de propósito general que no son utilizados en este diseño). El microcontrolador está conectado a los dos puertos SPI (para Ethernet y RF), el puerto JTAG, dos osciladores de cristal, un botón de reset, y un botón y 2 leds cuyas funciones se pueden determinar mediante software, además de varios condensadores de desacoplo.

En lo que respecta al puerto JTAG, se implementará una interfaz de 4 hilos (entrada de datos, salida de datos, señal de reloj y señal para controlar estado del micro), además de una señal para habilitar esta comunicación. A través de estas conexiones se consigue acceder a todos los módulos configurables del *gateway*, por lo que es de vital importancia para el programador. La norma JTAG no establece un conector físico estándar, por lo que, al trabajar con un micro de Texas Instruments, se utilizará el conector JTAG con la disposición que esta compañía establece. La Figura 5.4 muestra su configuración.

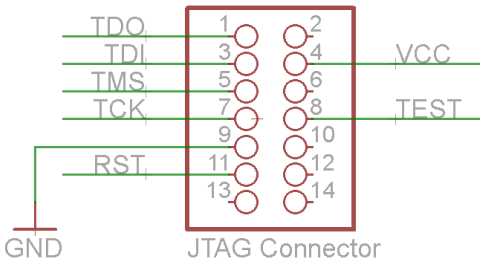

Figura 5.4: Configuración del conector JTAG

En lo referente a la conexión de red, las 4 líneas de datos del conector RJ45 (TX+, TX-, RX+, RX-) se deben llevar al controlador Ethernet. Este controlador requiere un oscilador de cristal de 25 MHz. Las señales correspondientes al interfaz digital SPI se conectan con el microcontrolador para permitir su comunicación. Los demás componentes necesarios vienen especificados en la correspondiente hoja de características.

De igual manera que con el bloque de RF, una vez que se obtienen los esquemáticos se procede al diseño del PCB con el editor de layout. En este caso el número de elementos, y por tanto, la cantidad de pistas necesarias es mucho mayor, por lo que el diseño es de mayor complejidad. Como norma básica, se desea que los componentes de mayor tamaño (como pueden ser los conectores RJ45 y el jack DC, osciladores de cristal o algunos condensadores), además de los botones y los leds, se encuentren todos en la capa superior del PCB. También se pretende separar al máximo el bloque correspondiente a la alimentación respecto de la conexión RF, para evitar posibles problemas de interferencia electromagnética. En la Figura 5.5 se puede observar el diseño final.

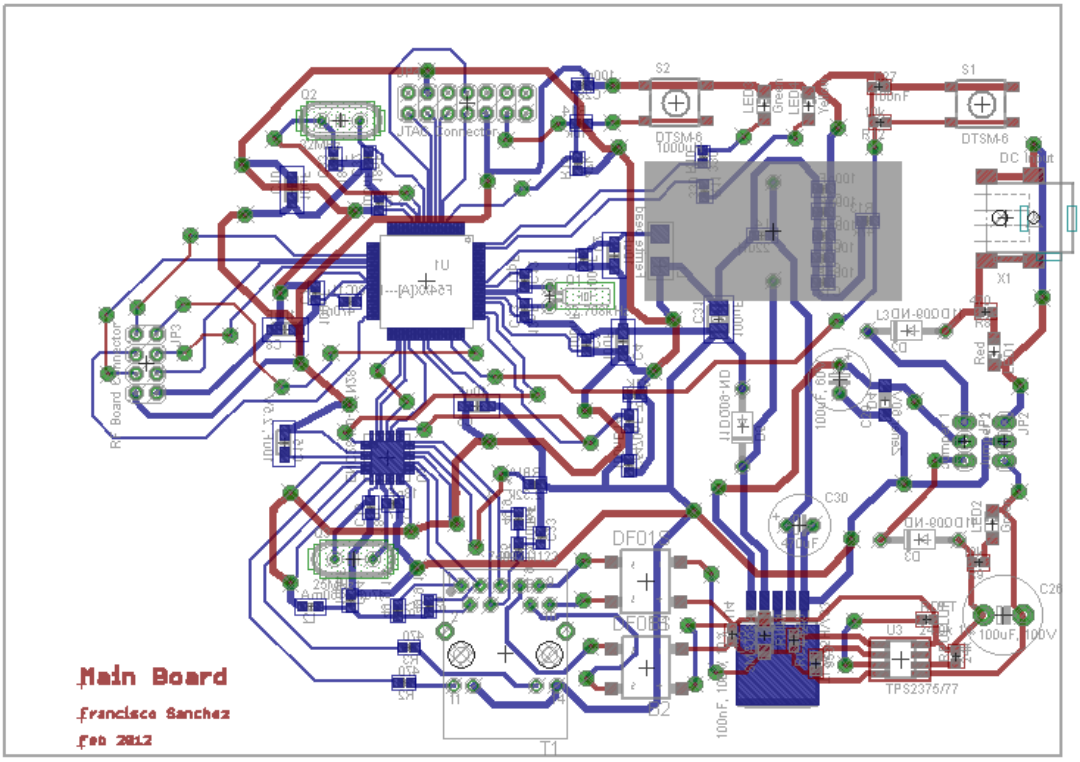

Figura 5.5: Aspecto del layout del módulo principal

De nuevo, las pistas en rojo corresponden con la capa superior, mientras que las de color azul, con la capa inferior. El tamaño de esta placa es de 136x167 mm. En el anexo D se encuentra una descripción más detallada con los esquemáticos e imágenes del resultado final, así como la lista de componentes. En el CD que acompaña a esta memoria se pueden consultar los archivos generados con el software EAGLE.

5.3 Montaje y test

Tras el diseño por CAD del gateway, el siguiente paso consiste en la construcción física del prototipo y su montaje. En primer lugar, con la colaboración del profesor director del proyecto, y gracias al robot con el que se cuenta en el laboratorio, se realizan las pistas y las perforaciones oportunas sobre ambas placas. A continuación, tras adquirir todos los componentes necesarios, se procede a su soldadura manual, con un soldador de tipo lápiz. Éste es un proceso bastante delicado, especialmente al soldar los componentes SMD o los integrados, ya que su tamaño es muy pequeño (aproximadamente, las patas de los integrados son de 0.2 mm de anchura). Por ello, se hace uso de un microscopio binocular que se encuentra en el laboratorio.

Para verificar el montaje, se examinan las uniones de todas las soldaduras a las pistas con ayuda de un voltímetro. Es importante asegurarse de que no existen cortocircuitos que puedan dañar algún componente cuando se le suministre la energía.

Una vez comprobado el montaje, se examina su funcionamiento. En primer lugar, se prueba el bloque de alimentación, mediante jack DC y a través de PoE, con resultados positivos, ya que de ambas maneras se consigue obtener los 3.3 voltios requeridos. También se comprueba la configuración del conector JTAG y el funcionamiento del microcontrolador mediante la realización un sencillo programa (ver anexo E). Con él se consigue controlar el uso de los leds y los botones. Y por último, se comprueban el correcto funcionamiento de las comunicaciones SPI a través de las cuales se deben configurar los demás integrados del gateway.

Por tanto, el test del prototipo resulta ser positivo. Tras las diferentes pruebas que se han llevado a cabo, se puede asegurar que el diseño descrito anteriormente está preparado para su programación software. Para finalizar, a continuación se muestran dos imágenes que ilustran el aspecto final de las dos placas tras su montaje. Además, en el CD que acompaña a esta memoria, se encuentra un vídeo con una demostración del funcionamiento de la placa principal.

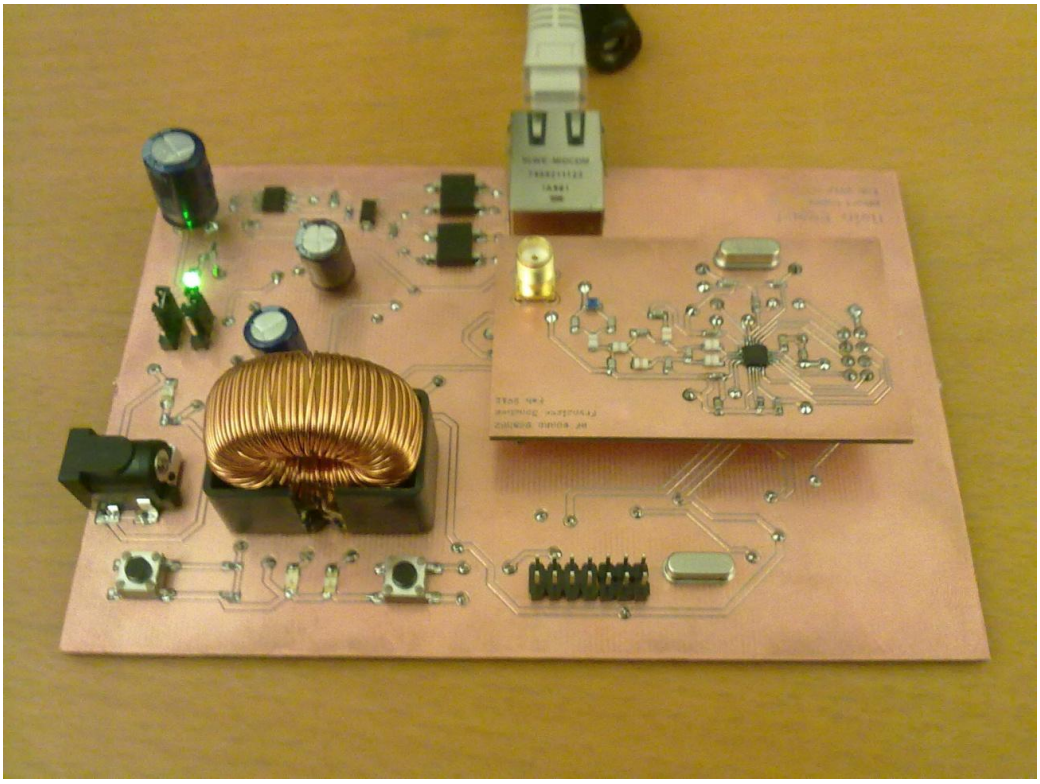

Figura 5.6: Imagen final del gateway tras su montaje

Figura 5.7: Imagen final del gateway con antena

6 Introducción de nuevos sensores

A pesar de que en este trabajo se ha analizado un sensor comercial concreto, se desea que el *gateway* diseñado pueda aprovechar la información proporcionada por otros sensores. Es por ello que se va estudiar cómo se podría alcanzar dicho propósito. Este apartado pretende ser un punto de partida para la consecución de un *gateway* universal que pueda compatibilizar varias tecnologías en un mismo dispositivo.

Nótese que es posible encontrar 2 escenarios diferentes, dependiendo de dónde se vaya a realizar el procesado y la decodificación de los datos:

- Realizar esta labor en el propio *gateway*, y suministrar al PC directamente los datos de información útil
- Utilizar el *gateway* como un simple receptor que registre la señal en la que se encuentra la información, y proporcionar dicha señal al PC para que éste sea el que procese y extraiga los datos útiles

En este PFC, lo que se pretende es decodificar y extraer los datos de interés directamente en el *gateway*, por lo que nos encontramos en el primero de estos dos casos. El análisis de este apartado se realiza bajo este supuesto.

En primer lugar, se debe adecuar el hardware para el tipo de sensor que se desea aprovechar. En este proyecto se ha diseñado una tarjeta receptora/transmisora sintonizada a 868/915 MHz, apta para modulaciones digitales 2-FSK, 4-FSK, GFSK y MSK, así como OOK y ASK. Por tanto, cuando los sensores no cumplan estos requisitos o condiciones, no se podrá utilizar dicha tarjeta. Será necesario construir una nueva, acorde con la estructura del *gateway* y con los requerimientos que exija el sensor en cuestión.

Reflexionando sobre cómo generalizar el diseño para poder sacar beneficio de otros sensores, se puede pensar en aprovechar la conexión Ethernet. A través de una red LAN, se podría descargar sobre el *gateway* un determinado programa, dependiendo del sensor que se quiere tratar. Por ejemplo, se puede implementar en el *gateway* un cliente TFTP (Trivial File Transfer Protocol) [33] para descargar archivos de un servidor. TFTP es un protocolo de transferencia de archivos muy simple, que fue precursor del protocolo FTP. Con esto, se podrían guardar en un servidor diferentes programas, cada uno de ellos adecuado a uno o varios tipos de sensores diferentes, y así descargarse uno u otro en función de los requerimientos de cada escenario. En el microcontrolador se debería implementar un programa bootloader (esto es, el primer programa que se ejecuta cuando se alimenta un dispositivo) que se encargue de descargar dicho fichero del servidor, y que seguidamente, lo ejecute.

Para dotar de esta adaptabilidad al sistema, se deben conocer los diferentes protocolos que el *gateway* necesita para poder conectarse a la red. TFTP utiliza UDP (User Datagram Protocol) como protocolo de transporte [33]. También se hará uso del protocolo IP. Y además, para conseguir una dirección válida en la red, se deberá implementar DHCP (Dynamic Host Configuration Protocol) [57] o BOOTP (Bootstrap Protocol) [33]. La Figura 6.1 muestra un ejemplo básico del proceso que seguiría el *gateway* en el momento que se conecta a una red. Los pasos a seguir son:

1. En primer lugar, el *gateway* manda un mensaje de broadcast en la red para alcanzar algún servidor DHCP.

2. El servidor DHCP contesta y le asigna su dirección IP, así como otros parámetros de red.
3. El *gateway* solicita al servidor TFTP leer o escribir un archivo (en este caso, leer).
4. El servidor TFTP envía el archivo solicitado. Éste será el programa que el *gateway* va a ejecutar y que estará asociado a uno o varios tipos de sensores.

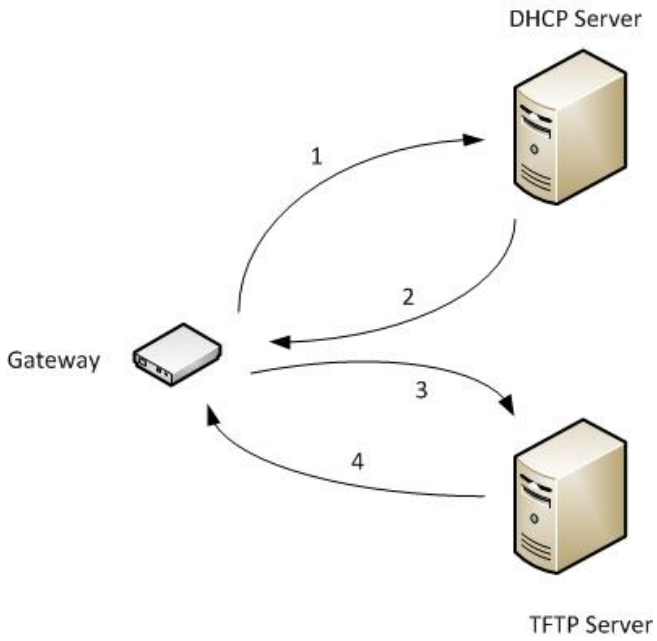

Figura 6.1: Conexión en la red del *gateway* y proceso de descarga

Es obvio que para poder llegar en este escenario, previamente se ha debido almacenar en el servidor los archivos o programas relacionados con los sensores. Para ello, es fundamental conocer el modo en el que cada sensor transmite los datos. Podemos estar en el caso de que se utilice una tecnología cuyos protocolos de comunicación sean conocidos (como puede ser Zigbee). En ese caso, bastaría con subir al servidor TFTP el archivo que implementa dicha tecnología. Si por el contrario, no se conoce el modo de comunicación, ya que se usa un protocolo propietario (como es el caso del sensor que se ha estudiado en este proyecto), se requiere un estudio previo (similar al realizado en el apartado 4) para sacar provecho del sensor en cuestión. De esta idea surge una de las líneas de trabajo futuro, basada en analizar e implementar el modo de comunicación de diferentes sensores, para que el *gateway* los pueda aprovechar.

Buscando información en la red, se ha encontrado un interesante trabajo relacionado con este proyecto [58]. En él, se analiza un sensor de temperatura y humedad, cuyo modo de comunicación es desconocido, y que utiliza la frecuencia de 868 MHz. El autor determina que la modulación utilizada es ASK. A pesar de que la manera en la que se examina la transmisión no es la misma que la que se ha descrito en el presente documento, las conclusiones a las que llega sí son muy afines. En primer lugar, la estructura de la trama es similar a la explicada en el apartado 4 (mismo número de bits, mismo código CRC, mismo preámbulo…). El autor determina que desde el bit 37 a 48 se codifica la temperatura del mismo modo que el sensor que se ha estudiado en este trabajo. Pero además, establece que la información de humedad relativa se codifica del

bit 50 al 56, y se indica que si el sensor no tiene capacidad para medir la humedad, el valor de esos bits es 106 en decimal. Esto concuerda de manera precisa con los datos registrados con el sensor analizado en este proyecto, como se puede comprobar en el anexo B. Por tanto, gracias a este trabajo se conoce cómo se comunica otro tipo de sensor. El *gateway* diseñado en este proyecto podría sacar provecho de él, y obtener otra información, como es la humedad relativa.

En conclusión, se ha descrito una posible solución para ampliar la cantidad de sensores reconocibles por el *gateway*, basada en sacar provecho de la conexión LAN. Previamente es indispensable desarrollar software para implementar cada tipo de sensor y almacenar los programas en el servidor. Destacar que el desarrollo de un programa para el *gateway* asociado al sensor estudiado en el capítulo 4, así como el proceso descrito en este capítulo acerca de la implementación del protocolo TFTP, son objeto de estudio de otro proyecto fin de carrera que se ha llevado a cabo de manera paralela a éste, como ya se ha indicado en el apartado 1.4.

7 Conclusiones y líneas futuras

En el presente proyecto se ha descrito el desarrollo hardware de un dispositivo cuya función consiste en adquirir datos de sensores inalámbricos para proporcionarlos a una red LAN, con la idea de que sean aprovechados en posibles servicios o aplicaciones. Para ello, en primer lugar, tras estudiar diferentes tecnologías relativas a redes de sensores inalámbricas, se ha capturado y analizado la transmisión procedente de un sensor de temperatura que utiliza un protocolo propietario para transferir la información a una estación base. Con las herramientas adecuadas, se ha conseguido capturar la señal para caracterizarla y averiguar sus características de transmisión: frecuencia, modulación, duración de trama y de bit y tasa de bits. Tras demodular la señal, se ha seguido un modelo lógico de análisis con el que se ha logrado identificar los campos de datos útiles (principalmente dato de temperatura e identificador de sensor) y el modo con el que se representa la información.

A continuación, se ha desarrollado el hardware correspondiente al dispositivo *gateway* que permita capturar los datos del sensor estudiado y transmitirlos en una red LAN. Tras el diseño por ordenador de los dos PCB´s descritos, se han realizado las pruebas y modificaciones oportunas para verificar el diseño. Se ha comprobado el correcto funcionamiento del bloque de alimentación obteniendo una tensión de 3.3 voltios a la salida del regulador, tanto alimentando a través del jack DC, como mediante la tecnología PoE. Además, a pesar de que el desarrollo de software no forma parte del alcance de este proyecto, se ha realizado un sencillo programa para constatar la actividad de todos los bloques que componen el diseño, con resultados satisfactorios.

Y por último, se ha aportado una posible solución acerca de cómo adaptar el diseño del *gateway* a otro tipo de sensores. Por tanto, se han cubierto los objetivos fijados al comienzo de este proyecto, especificados en el apartado 1.3. Como conclusiones principales que se han extraído del presente proyecto, señalar que se ha profundizado en el estudio de algunas de las principales tecnologías relativas a sensores inalámbricos, tratando y analizando ampliamente un sensor concreto. Y por otra parte, se ha trabajado en el ámbito del diseño electrónico de hardware, desarrollado de manera satisfactoria un dispositivo de red con 2 interfaces de comunicación (por radio y a través de Ethernet). Nótese que el diseño propuesto pretende ser un primer paso hacia un *gateway* general, capaz de adaptarse a todo tipo de sensores, sea cual sea su origen o su modo de transmisión.

A partir de aquí, surgen diversas líneas futuras de trabajo en las que sería posible profundizar:

- En primer lugar, el siguiente paso tras este trabajo consiste en la programación software del microcontrolador ubicado en el dispositivo diseñado. Se debe configurar, por un lado, el receptor acorde al sensor estudiado, y por otro, el controlador Ethernet para permitir la comunicación IP en una red LAN. Además también se puede implementar el sistema descrito en el apartado 1 para que el gateway conectado en una red pueda descargar los diferentes programas ejecutables (alojados en un servidor) asociados a diferentes sensores.
- Se propone examinar de manera similar a la descrita en este documento otros sensores, diferentes al estudiado en este trabajo, para determinar sus características y decodificar su mensaje.

- En lo referente al desarrollo de hardware, se pueden plantear diversas aportaciones. Existe gran variedad de sensores inalámbricos que trabajan a la frecuencia de 433 MHz, por lo que la elaboración de una tarjeta receptora sintonizada a esa frecuencia resultaría útil. Por otra parte, es posible realizar algunas modificaciones sobre el diseño construido, como puede ser la realización de una antena PCB integrada en la tarjeta receptora, o utilizar un módulo Wi – Fi para implementar la comunicación con la red, en lugar de utilizar una conexión cableada, para dotar de mayor flexibilidad al dispositivo.
- Por último, el cometido final de este proyecto es dar soporte para el desarrollo de aplicaciones o servicios que aprovechen diversos sensores. Se podrían pensar en aplicaciones de domótica (control de temperatura, luz o presencia en el hogar), industriales (automatización de maquinaria), etc. El punto distintivo que aportaría este proyecto es que no es preciso adquirir sensores realizados específicamente con ese cometido, sino que cualquier sensor comercial puede ser utilizado.

Bibliografía

[1] W. Dargie and C. Poellabauer, *Fundamentals of wireless sensor networks : theory and practice*. Chichester, West Sussex, U.K.: Wiley, 2010.

[2] D. J. Cook and S. K. Das, *Smart environments: technologies, protocols, and applications*, vol. 43. Wiley-Interscience, 2005.

[3] K. Sohraby, D. Minoli, and T. Znati, *Wireless sensor networks : technology, protocols, and applications*. Hoboken, N.J.: John Wiley & Sons, 2007.

[4] IEEE Computer Society. LAN/MAN Standards Committee. and Institute of Electrical and Electronics Engineers., *IEEE standard for information technology part 11, wireless LAN medium access control (MAC) and physical layer (PHY) specifications*. New York, N.Y.: Institute of Electrical and Electronics Engineers, 2007.

[5] A. S. Tanenbaum, *Computer Networks*. Upper Saddle River, NJ.: Pearson Education Benelux, 2003.

[6] Pattye Brown, 'Use ZigBee For Cost-Effective WPAN Sensing And Control Solutions', 2009. [Online]. Available: http://www.newark.com/pdfs/techarticles/freescale/FSL_Storefront_Jan20081-1.pdf. [Accessed: 28-November-2011].

[7] 'Graphic: Comparing Wireless Technology Range and Data Rates'. [Online]. Available: http://revolutionwifi.blogspot.se/2012/02/graphic-comparing-wireless-technology.html. [Accessed: 18-March-2012].

[8] J. M. Bohli, C. Sorge, and D. Westhoff, 'Initial observations on economics, pricing, and penetration of the internet of things market', *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 2, pp. 50–55, 2009.

[9] 'ZigBee Alliance'. [Online]. Available: http://www.zigbee.org/. [Accessed: 21-December-2011].

[10] IEEE Computer Society. LAN/MAN Standards Committee., Institute of Electrical and Electronics Engineers., and IEEE-SA Standards Board., *IEEE standard for information technology telecommunications and information exchange between systems--local and metropolitan area networks--specific requirements. Part 15.4, Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. New York, NY: Institute of Electrical and Electronics Engineers, 2006.

[11] A. Wheeler, 'Commercial Applications of Wireless Sensor Networks Using ZigBee', *IEEE Communications Magazine*, vol. 45, no. 4, pp. 70–77, April 2007.

[12] MEMSIC, 'MICAz Wireless Measurment System'. [Online]. Available: http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=147%3Amica2. [Accessed: 24-November-2011].

[13] MEMSIC, 'TelosB Mote Platform'. [Online]. Available: http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=152%3Atelosb. [Accessed: 29-November-2011].

[14] MEMSIC, 'IRIS Wireless Measurement System'. [Online]. Available: www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=135%3Airis. [Accessed: 29-November-2011].

[15] 'Libelium - Redes Sensoriales Inalámbricas - ZigBee - Smart Cities'. [Online]. Available: http://www.libelium.com/. [Accessed: 09-March-2012].

[16] 'SimpliciTI™ - RF software protocol'. [Online]. Available: http://www.ti.com/corp/docs/landing/simpliciTI/index.htm?DCMP=hpa_rf_general&HQS=NotApplicable+OT+simpliciti. [Accessed: 21-October-2011].
[17] 'Wireless Solutions'. [Online]. Available: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2664&param=en520414. [Accessed: 21-October-2011].
[18] 'SynkroRF'. [Online]. Available: http://www.freescale.com/webapp/sps/site/overview.jsp?code=PROTOCOL_SYNKRO. [Accessed: 03-October-2011].
[19] 'San Juan Software - PopNet™ The Easy, Economical Wireless Sensor and Control Network'. [Online]. Available: http://www.sanjuansw.com/?p=15. [Accessed: 21-October-2011].
[20] 'Z-Wave.com - ZwaveStart'. [Online]. Available: http://www.z-wave.com/modules/ZwaveStart/. [Accessed: 24-November-2011].
[21] 'Everything One-Net: one-net.info'. [Online]. Available: http://www.one-net.info/. [Accessed: 22-November-2011].
[22] 'Home | dash7.org'. [Online]. Available: http://www.dash7.org/. [Accessed: 24-November-2011].
[23] 'HART Communication Protocol - Wireless HART Technology'. [Online]. Available: http://www.hartcomm.org/protocol/wihart/wireless_technology.html. [Accessed: 22-November-2011].
[24] 'TinyOS Home Page'. [Online]. Available: http://www.tinyos.net/. [Accessed: 03-October-2011].
[25] 'nesC: A Programming Language for Deeply Networked Systems'. [Online]. Available: http://nescc.sourceforge.net/. [Accessed: 09-March-2012].
[26] 'The Contiki OS'. [Online]. Available: http://www.contiki-os.org/. [Accessed: 09-March-2012].
[27] International Telecommunication Union., *Radio regulations*. Geneva: International Telecommunications Union, 2008.
[28] M. Mouly, *The GSM system for mobile communications-- : a comprehensive overview of the European digital cellular systems*. Palaisu, France: Cell & Sys, 1992.
[29] R. C. Dixon, *Spread spectrum systems*. New York: Wiley, 1976.
[30] ERC Recommendation 70-03, 'Relating to the use of short range devices (SRD)'. Available at http://www.erodocdb.dk/docs/doc98/official/pdf/rec7003e.pdf, [accessed November 29, 2011].
[31] ETSI, 'ETSI EN 300 220-2 v2.3.1 Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD);' Available at http://www.rfm.com/company/etsi.pdf, [accessed November 1, 2011].
[32] Institute of Electrical and Electronics Engineers. and IEEE-SA Standards Board., *IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and Physical Layer Specifications : Amendment 3: Data Terminal Equipment (DTE) power via the Media Dependent Interface (MDI) enhancements*. New York: Institute of Electrical and Electronics Engineers, 2009.
[33] W. R. Stevens, *TCP/IP illustrated. 1, The protocols*. Reading, Massachusetts: Addison-Wesley, 2000.
[34] M. Gebski, A. Penev, and R. K. Wong, 'Protocol Identification of Encrypted

Network Traffic', Los Alamitos, CA, USA, 2006, vol. 0, pp. 957–960.
[35] K. Gopalratnam, S. Basu, J. Dunagan, and H. J.Wang, 'Automatically Extracting Fields from Unknown Network Protocols', November 2011, Available at http://research.microsoft.com/pubs/69364/sysml_114_cameraready.pdf.
[36] R. Martín Sánchez, 'Desarrollo de un sniffer para redes de sensores basadas en ZigBee', Thesis, Telecommunications Engineering, specializing in Telematics, Universitat Politècnica de Catalunya, UPC. Castelldefels School of Telecommunications and Aerospace Engineering (EETAC), 18 March2011, Available at http://upcommons.upc.edu/pfc/bitstream/2099.1/11946/1/memoria.pdf, [accessed November 24, 2011].
[37] Zhuanghui Yu, Yongzhong Huang, Shaozhong Guo, Bei Zhou, and Hua Ren, 'Extracting Information from Unknown Protocols On CampusNet', presented at the First IEEE International Symposium on Information Technologies and Applications in Education, 2007. ISITAE '07, 2007, pp. 535–539.
[38] Cheng Zhang, Shuran Song, Canxi Li, and Tiansheng Hong, 'Long-distance data communication based on wireless communication technology', presented at the Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011, pp. 4045–4048.
[39] Hao Haohao and Xiong Junqiao, 'Design of wireless sensor networks for density of natural gas', presented at the System Science, Engineering Design and Manufacturing Informatization (ICSEM), 2011, pp. 141–143.
[40] Chaoli Zhou and Jianhua Shen, 'The design and realization of ZigBee—Wi-Fi wireless gatway', presented at the 2011 International Conference on Electric Information and Control Engineering (ICEICE), 2011, pp. 1786–1790.
[41] 'Wave - Wireless Thermometer'. [Online]. Available: http://tfa-dostmann.de/index.php?id=61. [Accessed: 21-November-2011].
[42] 'Time and Standard Frequency Station DCF77'. [Online]. Available: http://www.eecis.udel.edu/~mills/ntp/dcf77.html. [Accessed: 09-March-2012].
[43] 'Ettus Research'. [Online]. Available: http://www.ettus.com/. [Accessed: 09-March-2012].
[44] 'Ettus_USRP1_datasheet'. Available at https://www.ettus.com/content/files/Ettus_USRP1_DS_FINAL_1.27.12.pdf, [accessed March 9, 2012].
[45] 'GNU Radio'. [Online]. Available: http://gnuradio.org. [Accessed: 24-November-2011].
[46] 'Ettus Research -Daughterboards'. [Online]. Available: https://www.ettus.com/product/category/Daughterboards. [Accessed: 09-March-2012].
[47] 'Online CRC Calculation'. [Online]. Available: http://ghsi.de/CRC/index.php. [Accessed: 09-March-2012].
[48] 'Texas Instruments'. [Online]. Available: http://www.ti.com/. [Accessed: 09-March-2012].
[49] Texas Instruments, 'MSP430f5437 Datasheet'. Available at http://www.ti.com/lit/ds/symlink/msp430f5437a.pdf, [accessed November 29, 2011].
[50] 'Code Composer Studio (CCStudio) Integrated Development Environment (IDE) v5 - CCSTUDIO - TI Tool Folder'. [Online]. Available: http://www.ti.com/tool/ccstudio. [Accessed: 24-November-2011].
[51] Texas Instruments, 'CC1101 Datasheet'. Available at http://www.ti.com/lit/ds/symlink/cc1101.pdf, [accessed November 24, 2011].

[52] Texas Instruments, ‘CC430F6137 Datasheet’. Available at http://www.ti.com/lit/ds/symlink/cc430f6137.pdf, [accessed November 29, 2011].
[53] Microchip, ‘ENC28J60 Datasheet’. Available at http://ww1.microchip.com/downloads/en/DeviceDoc/39662c.pdf, [accessed November 24, 2011].
[54] Texas Instruments, ‘TPS2375 Datasheet’. Available at http://www.ti.com/lit/ds/symlink/tps2375.pdf, [accessed February 20, 2012].
[55] Texas Instruments, ‘TL2575-HV Datasheet’. Available at http://www.ti.com/lit/ds/symlink/tl2575hv-adj.pdf, [accessed February 20, 2012].
[56] IEEE Computer Society. Test Technology Standards Committee., Institute of Electrical and Electronics Engineers., IEEE Standards Board., and IEEE Standards Association., *IEEE standard test access port and boundary-scan architecture*. New York: Institute of Electrical and Electronics Engineers, 2001.
[57] R. Droms and T. Lemon, *The DHCP handbook*. Indianapolis, Indiana: Sams, 2002.
[58] ‘Wireless Temperature Sensor’. [Online]. Available: http://fredboboss.free.fr/tx29/tx29_sw.php. [Accessed: 15-March-2012].