

UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN



TRABAJO FIN DE GRADO

Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación

DISEÑO, IMPLEMENTACIÓN Y PRUEBA DE
UNA PLACA DE EXPANSIÓN CON SENSORES Y
ACTUADORES PARA NODOS INALÁMBRICOS
CON CAPACIDADES COGNITIVAS

Enrique Javier Cobo Jiménez

Julio 2014

TRABAJO FIN DE GRADO

Título: DISEÑO, IMPLEMENTACIÓN Y PRUEBA DE UNA PLACA DE EXPANSIÓN CON SENORES Y ACTUADORES PARA NODOS INALÁMBRICOS CON CAPACIDADES COGNITIVAS

Autor: ENRIQUE JAVIER COBO JIMÉNEZ

Tutora: ELENA ROMERO PERALES

Ponente: ALVARO ARAUJO PINTO

Departamento: DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA

TRIBUNAL

Presidente: D. OCTAVIO NIETO-TALADRIZ GARCÍA

Vocal: D. MIGUEL ÁNGEL SÁNCHEZ GARCÍA

Secretario: D. ALVARO ARAUJO PINTO

Suplente: D. PEDRO JOSÉ MALAGÓN MARZO

CALIFICACIÓN:

Madrid, a _____ de _____ de 2014.

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN**



TRABAJO FIN DE GRADO

**Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación**

**DISEÑO, IMPLEMENTACIÓN Y PRUEBA DE
UNA PLACA DE EXPANSIÓN CON SENSORES Y
ACTUADORES PARA NODOS INALÁMBRICOS
CON CAPACIDADES COGNITIVAS**

Enrique Javier Cobo Jiménez

Julio 2014



Versión online. <http://bit.ly/enriquejcobo-tfg>

Documento creado a partir de TeXIS v.1.0+.

Este documento está preparado para su impresión a doble cara.

Todos los ficheros usados para hacer este documento son accesibles públicamente en el siguiente repositorio: <https://github.com/enriquejcobo/tfg-mem>.

Resumen

Una red de sensores es aquella formada por sensores distribuidos cuya misión es la de monitorizar diferentes parámetros del entorno, intercambiando información para conocer la parte del mismo que nos interese y actuar en consecuencia. Estas redes, por lo general, se conectan de forma inalámbrica, lo que les aporta mayor versatilidad y flexibilidad, si bien esto plantea retos de autonomía energética, al ser alimentadas por medio de baterías, y de fiabilidad en las comunicaciones.

Para mejorar tanto la fiabilidad de las comunicaciones como el consumo energético de las redes de sensores inalámbricas, introducimos propiedades cognitivas, de forma que los nodos sean capaces de apreciar el estado del espectro y, como resultado, modificar sus parámetros de comunicación de forma adaptativa. Estas redes se denominan CWSN (Cognitive Wireless Sensor Network).

En el Laboratorio de Sistemas Integrados del Departamento de Ingeniería Electrónica, estas redes son uno de los principales objetos de investigación, con proyectos como la puesta en marcha de un banco de pruebas basado en el nodo desarrollado en el laboratorio, cNGD (Cognitive Next Generation Device).

El trabajo se centrará en el diseño, implementación y prueba de una placa de expansión para el cNGD. En esta placa se incluirán sensores que permitan realizar medidas del medio, como su temperatura, luminosidad, presencia de personas o aceleración sobre la placa; y actuadores que intervendrán sobre él, modificando bien parámetros directos como la temperatura, o avisando para que se produzca una acción humana.

Asimismo, se realizará el desarrollo del software necesario para su funcionamiento dentro del nodo, mediante la creación de funciones que permitan gobernar dichos sensores y actuadores en combinación con el cNGD. Finalmente, se definirá un ejemplo de aplicación que muestre el funcionamiento del sistema.

PALABRAS CLAVE: *redes de sensores inalámbricas cognitivas, sensores y actuadores, placa de expansión, redes cognitivas, redes de sensores inalámbricas, diseño hardware.*

Abstract

A sensor network is formed by distributed sensors whose task is to monitor some parameters of their environment, exchanging information in order to obtain those that are relevant, and act consequently. These networks, in general, are connected wirelessly, which gives them more versatility and flexibility. However, this fact causes some challenges, such as energy autonomy, due to being powered by batteries, and communications reliability.

In order to improve both the communications reliability and the energy efficiency of WSNs (Wireless Sensor Networks), cognitive properties are introduced in these platforms. They give them the ability to sense the spectrum state and, as a result, modify their communication parameters in an adaptive way. These networks are called CWSNs (Cognitive Wireless Sensor Networks).

In the Laboratorio de Sistemas Integrados (LSI) of the Departamento de Ingeniería Electrónica (DIE), these networks are one of the most important research lines, with projects such as the deployment of a test-bed based on the cNGD (Cognitive Next Generation Device), a node developed in the laboratory.

This project is focused on the design, implementation and test of a board –known as shield– for the cNGD. This board will include both sensors that can measure the environment, such as its temperature, acceleration, presence or luminosity; and actuators that will act upon it, modifying these parameters in a direct way or making a call in order to request a human intervention.

In addition, software needs will be resolved by developing functions that allow the government of those sensors and actuators, used in combination with the cNGD platform. Finally, a demo application will be developed in order to show the system features.

KEY WORDS: *cognitive wireless sensor network,
sensors and actuators, shield,
cognitive network, wireless sensor network,
hardware design.*

*A mis padres,
y a mi familia,
por creer en mí
y haberme apoyado siempre
durante todos estos años.*

*A mis telecos,
y a mis compañeros del B105,
esperando que hayáis aprendido de mí
la mitad de lo que me habéis enseñado.*

*A toda mi gente,
jiénenses y mendelianos,
por todos esos ratos vividos
y los que nos quedan por vivir.*

Índice

Resumen	VII
Abstract	IX
1. Introducción	1
1.1. Antecedentes	1
1.2. Objetivos	3
1.3. Fases del proyecto	3
1.4. Estructura del documento	4
2. Estudio Previo	5
2.1. cNGD: Cognitive Next Generation Device	5
2.2. Sensores	8
2.2.1. Temperatura	9
2.2.2. Acelerómetro	11
2.2.3. Presencia	12
2.2.4. Luminosidad	13
2.3. Actuadores	14
2.3.1. Emisor Infrarrojo	14
2.3.2. Buzzer	14
2.3.3. Salidas digitales	14
3. Estudio del Diseño y de la Implementación	15
3.1. Elección de los sensores y actuadores	15
3.1.1. Sensores	15
3.1.1.1. Temperatura	15
3.1.1.2. Acelerómetro	17
3.1.1.3. Presencia	18
3.1.1.4. Luminosidad	18
3.1.2. Actuadores	19
3.1.2.1. Emisor Infrarrojo	19
3.1.2.2. Buzzer	20
3.1.2.3. LEDs	20
3.2. Diseño de la placa de expansión	21
3.2.1. Diagrama de bloques	21
3.2.2. Esquemático	22

3.3. Implementación en PCB	22
3.3.1. Trazado	22
3.3.2. Lista de Materiales	25
3.3.3. Resultado final	25
3.4. Costes hardware	26
3.5. Herramientas	27
3.5.1. Altium Designer	27
3.5.2. Puesto de Soldadura	28
4. Software	29
4.1. Interfaz Hardware-Software	29
4.1.1. Funciones generales y adaptación	29
4.1.2. Sensores	31
4.1.2.1. Temperatura	31
4.1.2.2. Acelerómetro	32
4.1.2.3. Presencia	33
4.1.2.4. Luminosidad	33
4.1.3. Actuadores	33
4.1.3.1. Emisor infrarrojo	33
4.1.3.2. Buzzer	34
4.1.3.3. LEDs	35
4.2. Aplicación de prueba	35
4.3. Herramientas	36
4.3.1. MPLAB X	36
4.3.2. Programador: ICD 3	36
5. Pruebas y Resultados	37
5.1. Hardware	37
5.1.1. Sensores y Actuadores	37
5.1.2. PCB	38
5.2. Software	39
5.2.1. HAL	39
5.2.2. Funciones para el manejo de los sensores y actuadores	40
5.3. Herramientas	42
5.3.1. Osciloscopio	42
5.3.2. Multímetro	42
6. Conclusiones y líneas futuras	43
6.1. Conclusiones	43
6.2. Líneas futuras	44
Referencias	45
Lista de Acrónimos	49

Listado de Figuras

1.1.	Aplicaciones y su ocupación en el espectro radio.	2
2.1.	Vista detallada del cNGD.	6
3.1.	Diagrama modular del diseño.	16
3.2.	Sensor de Temperatura Microchip MCP9800.	16
3.3.	Acelerómetro Freescale MMA8453Q.	17
3.4.	Sensor de Presencia Panasonic AMN33112J.	18
3.5.	Sensor de Luminosidad Avago APDS 9005.	19
3.6.	Emisor Infrarrojo Vishay TSUS5202.	20
3.7.	Buzzer TDK PS1720.	20
3.8.	LED de información y depuración.	21
3.9.	Diagrama de bloques de la placa de expansión.	21
3.10.	Esquemático de la placa de expansión.	23
3.11.	Trazado de la placa de expansión.	24
3.12.	Trazado de la placa de expansión previa.	25
3.13.	Resultado final: la placa de expansión sobre el cNGD.	26
3.14.	La placa de expansión previa sobre el cNGD.	27
4.1.	Repositorio público del <i>software</i>	30
4.2.	Tramas de comunicación infrarroja de distintos fabricantes.	34
4.3.	Temporalización de la comunicación infrarroja.	34
5.1.	Detector infrarrojo Vishay TSOP1138.	38
5.2.	Contenido de la trama del aire acondicionado.	41
5.3.	Detalle del comienzo de trama infrarroja.	41

Lista de Tablas

2.1.	Descripción del uso de los pines del <i>header</i> del cNGD.	7
2.2.	Comparación de sensores de temperatura comerciales.	10
2.3.	Comparación de acelerómetros comerciales.	12
2.4.	Comparación de sensores de presencia comerciales.	12
2.5.	Comparación de sensores de luminosidad comerciales.	13
3.1.	Entradas y salidas de los <i>headers</i> hacia la placa de expansión.	22
3.2.	Lista de Materiales.	26
3.3.	Costes <i>hardware</i> de la placa de expansión.	27

Capítulo 1

Introducción

La única fuente de conocimiento es la experiencia.

Albert Einstein

RESUMEN: En este capítulo se da una descripción general del proyecto, desde la introducción al contexto, pasando por la necesidad del trabajo y una breve descripción del mismo. Asimismo, se definen los objetivos, la organización del proyecto y la memoria.

1.1. Antecedentes

Una red de sensores inalámbrica, WSN (*Wireless Sensor Network*), consiste en un conjunto de nodos distribuidos que cubren un área geográfica sobre la que realizan mediciones. Cada uno de estos nodos tiene comunicación inalámbrica, así como cierta capacidad de procesamiento, lo cual lleva a que sean capaces de adaptarse rápidamente a nuevos entornos, pudiendo formar de forma autónoma, si fuera necesario, su propia red para comunicaciones.

Las WSN, por estas características, son aplicables a multitud de campos, como pueden ser la domótica, salud, control industrial, seguridad y vigilancia, entre otros. Como consecuencia directa de esta variedad de usos prácticos, estas redes han emergido a la primera línea, además de por su potencial económico y por su capacidad de transformar nuestras vidas a corto-medio plazo.

No obstante, algunas de las propiedades de las mismas plantean retos, tales como el modo de alimentación –ya que habitualmente es necesario el uso de baterías–, fiabilidad en las comunicaciones, o seguridad en las transmisiones. A esto se une el hecho de que la creciente demanda de comunicación inalámbrica, junto con una mala distribución del espectro radio¹, está llevando a que ciertas bandas del mismo se encuentren saturadas, como muestra la figura 1.1. Esta saturación se hace más patente en las bandas de acceso libre ISM (*Industrial, Scientific and Medical*).

Es por ello que las WSNs requieren, para un modo de operación eficiente, propiedades cognitivas, las cuales pasamos a presentar a continuación. Las redes cognitivas fueron introducidas por Mitola en los años 1999 [2] y 2000 [3], bajo el paraguas de la

¹El espectro radio refiere a la parte del espectro radioeléctrico en el que se encuentran las transmisiones radio, englobando frecuencias inferiores a los 300 GHz

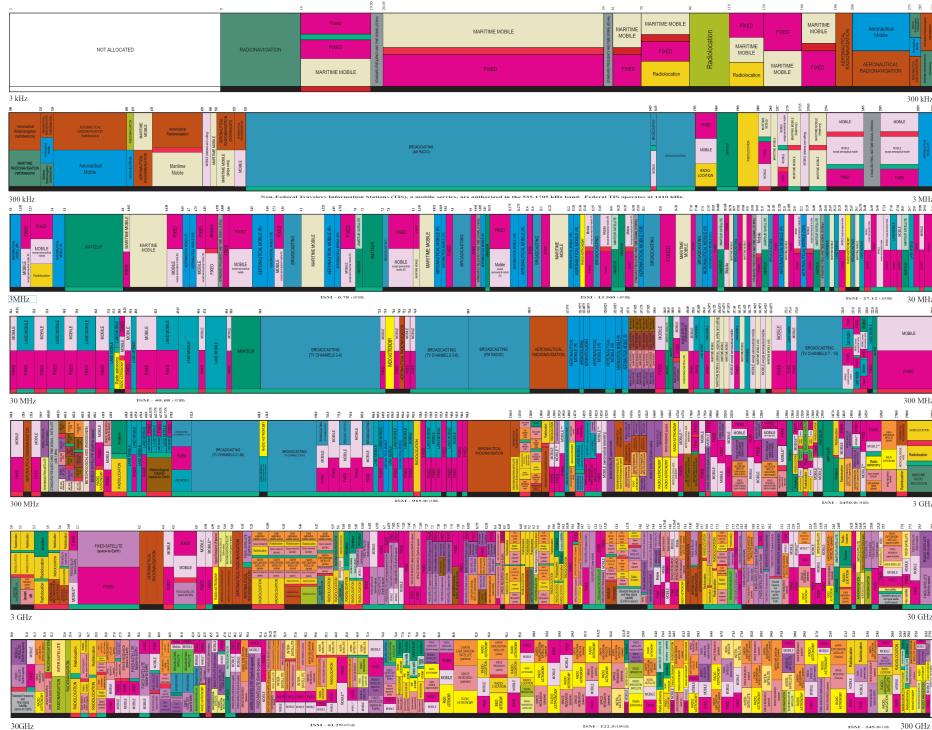


Figura 1.1: Aplicaciones y su ocupación en el espectro radio, obtenido de [1].

Radio Cognitiva, CR (*Cognitive Radio*). Fue entonces cuando comienzan a integrarse en las redes de comunicaciones conceptos como el sensado del espectro, el autoaprendizaje, la autogestión, o el análisis del contexto, pasando a ser redes cognitivas.

Integrando el concepto de CR en las WSN obtenemos las redes de sensores inalámbricas cognitivas, CWSN (*Cognitive Wireless Sensor Network*), las cuales son conscientes de su entorno y capaces de adaptar sus parámetros de comunicación radio, de forma que se alcancen comunicaciones más fiables y eficientes.

Como consecuencia del potencial de las mismas, se están desplegando bancos de pruebas para su estudio en muchos centros de investigación. Algunos ejemplos son el *TWIST* [4] de la Technische Universität Berlín, el *LOG-a-TEC* [5] en Eslovaquia, el *VT-CORNET* [6] en Virginia o el *FIT/CorteXlab* [7] en Lyon.

Para el despliegue de estas redes es necesario el uso de plataformas reales que contribuyan al estudio de las mismas. En el contexto del LSI (*Laboratorio de Sistemas Integrados*), perteneciente al DIE (*Departamento de Ingeniería Electrónica*) en la ETSIT (*Escuela Técnica Superior de Ingenieros en Telecomunicación*) de la UPM (*Universidad Politécnica de Madrid*) han sido diseñados dos nodos para CWSN: El FCD (*First Cognitive Device*) [8], desarrollado en 2011 como una primera implementación, y el cNGD (*Cognitive Next Generation Device*) [9], del año 2013, sobre el cual se están realizando el grueso de las investigaciones.

Unido al desarrollo de esta plataforma, se está llevando a cabo la implantación de un banco de pruebas [10]. En este marco se han realizado diversas investigaciones, como la creación de una arquitectura cognitiva [11] o un *firmware* para estos nodos [12].

No obstante, el cNGD, a pesar de haber sido diseñado con el propósito explícito

de ser un nodo para CWSN, no dispone de los sensores característicos de este tipo de redes, por lo que se hace necesario la creación de módulo auxiliar que los integre. En este módulo auxiliar, principal objetivo de este trabajo, se encontrarán presentes tanto los sensores, que midan parámetros ambientales, como los actuadores, que realicen modificaciones sobre el entorno. La placa de expansión –también denominada *shield*–, se integrará en la plataforma cNGD, completando de esta manera el estudio y desarrollo de las CWSN.

En este trabajo se detalla todo el proceso de creación de esta placa, desde el diseño, la implementación *hardware* y las pruebas de la misma hasta la implementación software desarrollada mediante una HAL (*Hardware Abstraction Layer*), así como una aplicación de demostración que integre el uso del nuevo *hardware*.

1.2. Objetivos

La principal meta del trabajo es **diseñar e implementar una placa de expansión que permita monitorizar el ambiente y actuar sobre él, para un nodo de CWSN**. En la implementación de la misma deberán considerarse parámetros como el bajo consumo, bajo coste y versatilidad, entre otros. Asimismo, se tendrá que desarrollar el *software* necesario en el cNGD para que la comunicación sea posible con esta nueva *shield*, así como una aplicación que muestre el funcionamiento del sistema en su conjunto.

En este sentido, podemos fragmentar esta meta en varios objetivos más específicos:

- *Diseño de la shield*. Se decidirán los sensores y los actuadores que formarán parte del sistema y se realizará su diseño.
- *Implementación y prueba de la placa de expansión*. Se diseñará la PCB (*Printed Circuit Board*) física en la que se implementará la placa de expansión, se soldarán los componentes y se probarán sus funciones.
- *Desarrollo del software*. Comprende la inclusión de las funciones necesarias para que el microcontrolador del nodo pueda hacer uso de la placa de expansión.
- *Aplicación*, que ilustre utilidades prácticas del sistema en su conjunto: la placa sobre un nodo cNGD en un entorno CWSN.

1.3. Fases del proyecto

La organización del proyecto se describe a continuación:

- *Estudio de la documentación*.

En esta primera etapa, se recopilará información acerca de las redes CWSN, los sensores y actuadores, y de la plataforma cNGD. Además, se adquirirán las capacidades adicionales necesarias para la realización del trabajo. Objetivos:

- Revisión de clases de sensores y actuadores de utilidad en un AmI (*Ambient Intelligence*).
- Estudio de las principales herramientas, tanto *software* como *hardware*, necesarias para la realización de este trabajo.

- Revisión de la plataforma cNGD, base para la *shield*.
- *Diseño e implementación hardware.*

Esta fase comprende desde la primera selección de elementos para la placa, hasta el montaje y la implementación. Objetivos:

- Decisión acerca de los sensores y actuadores: tipo, encapsulado, forma de comunicación, salidas, rangos, entre otros.
- Diseño del *hardware*. Selección de componentes auxiliares. Diseño del trazado.
- Implementación y soldadura. Pruebas funcionales.

- *Diseño software.*

En esta etapa se creará la interfaz que permita abstraer el *hardware* añadido en la anterior fase, de forma que pueda servir para cualquier aplicación futura. Objetivos:

- Implementación *software* de los módulos de la placa de expansión.
- Depuración del código. Pruebas funcionales.
- Generación de la documentación acerca del software desarrollado.

- *Aplicación de prueba.*

En último lugar, se llevará a cabo el desarrollo de una demostración del sistema funcional. Objetivos:

- Implementación *software* de la aplicación, incluyendo la comunicación de información entre nodos.

- *Generación de la documentación.*

Esta fase, transversal a todas ellas, consiste en la escritura de la memoria, para la cual se usará TEXIS[13], una plantilla de LATEX[14].

1.4. Estructura del documento

En el Capítulo 2 se realiza un resumen de la plataforma cNGD, así como de la investigación realizada en materia de sensores y actuadores, necesarios para la realización de la *shield*. El Capítulo 3 muestra una panorámica del proceso de diseño: esquemáticos, componentes usados, entre otros; y de la implementación de la PCB: trazado, montaje, etc. La capa de *software* necesaria para el control de la placa, así como la aplicación de prueba, se detallan en el Capítulo 4. Seguidamente, en el Capítulo 5 encontramos la batería de pruebas realizadas, tanto al *hardware* como al *software*. Finalmente, el Capítulo 6 ofrece las conclusiones y las investigaciones futuras en esta materia.

Capítulo 2

Estudio Previo

*El hombre no se mide por lo que conoce,
sino en cómo aplica lo que conoce.*

Anónimo

RESUMEN: En este capítulo encontramos, en primer lugar, una revisión de la plataforma cNGD, base de la placa de expansión. Seguidamente, se expone una primera aproximación de las necesidades y requisitos de la *shield*, en cuanto a dispositivos y funcionalidades se refiere, así como una revisión de los sensores y actuadores a incluir.

2.1. cNGD: Cognitive Next Generation Device

Comenzamos el trabajo describiendo brevemente el cNGD, plataforma para CWSN, puesto que será sobre esta donde se acoplará nuestra *shield*, lo que hace indispensable conocer sus características. La figura 2.1 muestra detalladamente este nodo.

El cNGD incluye un MCU (*Micro-controller Unit*) PIC32MX675F256L, fabricado por Microchip [15], cuyas características técnicas se resumen a continuación.

- MCU de 32 bits.
- Memoria Flash de 256 KB.
- Memoria RAM (*Random Access Memory*) de 64 KB.
- Velocidad máxima de funcionamiento de 80 MHz.
- Módulos para comunicación serie UART (*Universal Asynchronous Receiver-Transmitter*), SPI (*Serial Peripheral Interface*) e I²C (*Integrated Circuit*).
- Conectividad por medio de USB (*Universal Serial Bus*) 2.0 OTG (*On-The-Go*) y 10/100 Ethernet MAC.
- Interfaz para programación y depuración.
- Temporizadores digitales, ADC (*Analog-to-Digital Converter*) y GPIO (*General Purpose Input-Output*).
- Modos de bajo consumo.
- Encapsulado de 100 pines.

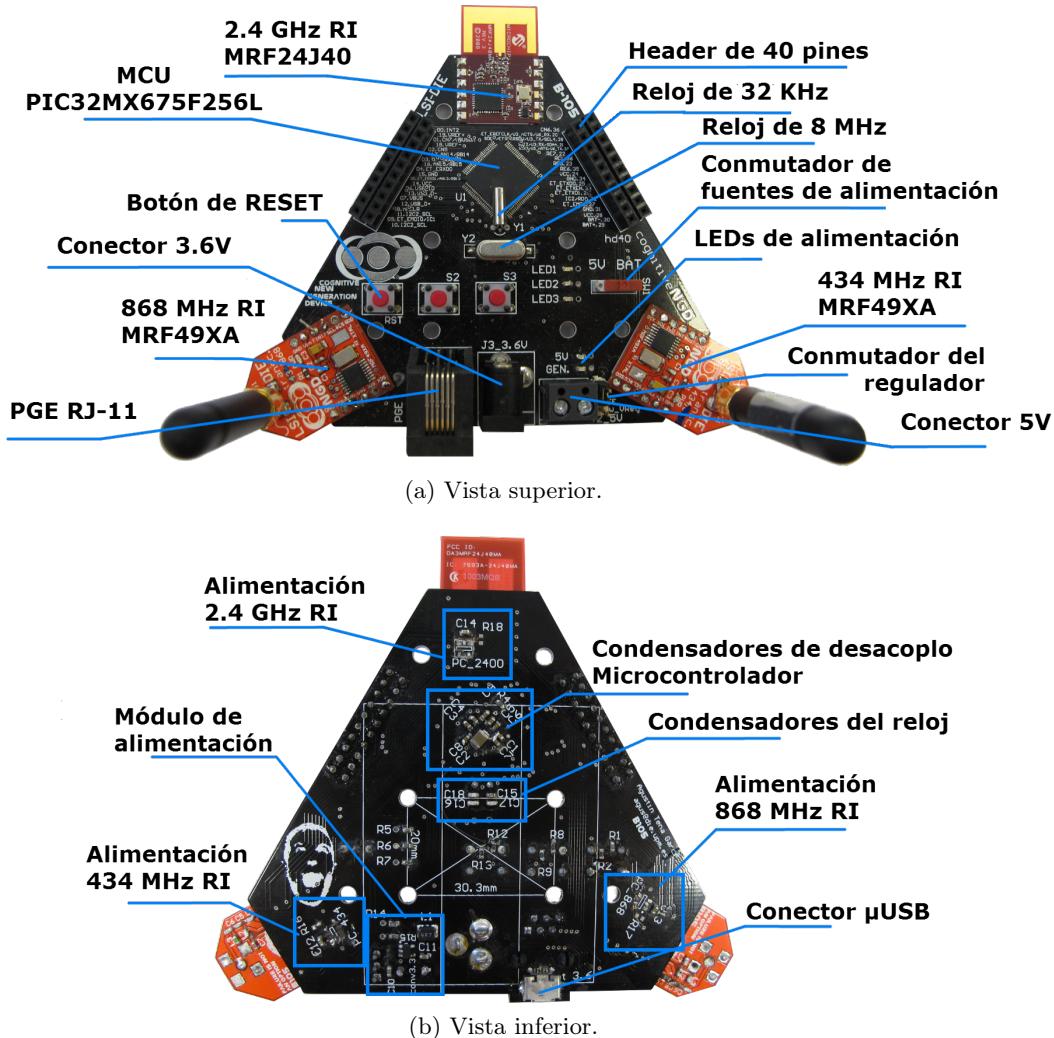


Figura 2.1: Vista detallada del cNGD, obtenidas de [9].

En cuanto a las Interfaces de comunicación Radio, RI (*Radio Interface*), se incluyen tres transceptores, con acceso a tres bandas del espectro: 434 MHz, 868 MHz y 2.4 GHz, todas ellas pertenecientes a bandas sin licencia, denominadas ISM. Esto es relevante, puesto que, al ser bandas de acceso libre, pueden estar ocupadas por otras aplicaciones. El cNGD, por medio de su capacidad cognitiva, es capaz de sensar el espectro en estas bandas y decidir dinámicamente cuál usar en función de la ocupación y del ruido en los canales. La comunicación se realiza bajo el protocolo MiWi [16], estándar inalámbrico propiedad de Microchip.

Las dos primeras (434 y 868 MHz) fueron diseñadas para esta plataforma a partir del integrado MRF49XA [17], soportando tasas binarias de hasta 115.2 Kbps. Para la comunicación en 2.4 GHz, se emplea el MRF24J40MA [18], el cual nos permite elevar la velocidad de transmisión hasta los 250 Kbps.

Una de las principales ventajas de este diseño es la cabecera de expansión, también denominada *header*, la cual permite ampliar el diseño de forma modular. Será en la misma donde insertaremos nuestra placa, por lo que le hacemos un estudio más exhaustivo. Se compone de 40 pines, en los que se permite el acceso a los periféricos del microcontrolador. En la tabla 2.1 encontramos una relación entre el pin del *header*,

el pin del microcontrolador y las funciones que puede realizar, pudiendo estar varias de estas funciones multiplexadas en el mismo pin.

PIN	MCU PIN	1a Func.	2a Func.	3a Func.	4a Func.	5a Func.
00	19	INT2	RE9	-	-	-
01	20	VBUSON	AN5	C1IN+	CN7	RB5
02	22	AN3	CN4	RB3	-	-
03	35	AN11	ERXERR	RB11	-	-
04	41	AN12	ERXD0	RB12	-	-
05	42	AN13	ERXD1	RB13	-	-
06	51	USBID	RF3	-	-	-
07	54	VBUS	-	-	-	-
08	13	nMCLR	-	-	-	-
09	68	EMDIO	IC1	RD8	-	-
10	59	SDA2	RA3	-	-	-
11	58	SCL2	RA2	-	-	-
12	57	D+	RG2	-	-	-
13	56	D-	RG3	-	-	-
14	-	VCC	-	-	-	-
15	-	GND	-	-	-	-
16	44	AN15	CN12	RB15	-	-
17	43	AN14	RB14	-	-	-
18	28	VREF-	CVREF-	RA9	-	-
19	29	VREF+	CVREF+	RA10	-	-
20	14	REFCLK	U3nCTS	U6RX	CN11	RG9
21	11	SDA4	SDI2	U3RX	CN9	RG7
22	5	RE7	-	-	-	-
23	3	RE5	-	-	-	-
24	-	VCC	-	-	-	-
25	88	ETXD0	RF1	-	-	-
26	87	ETXD1	RF0	-	-	-
27	71	EMDC	IC4	RD11	-	-
28	-	VCC	-	-	-	-
29	-	BAT+	-	-	-	-
30	-	BAT-	-	-	-	-
31	-	GND	-	-	-	-
32	69	IC2	RD9	-	-	-
33	83	ETXEN	CN15	RD6	-	-
34	-	GND	-	-	-	-
35	4	RE6	-	-	-	-
36	6	RC1	-	-	-	-
37	10	SCK2	U6TX	U3nRTS	CN8	RG6
38	12	ECRSDV	SCL4	SDO2	U3TX	CN10/RG8
39	21	AN4	C1IN-	CN6	RB4	-

Tabla 2.1: Descripción del uso de los pines, obtenido de [9].

Resaltamos las más relevantes para nuestro diseño:

- *Fuentes de alimentación.* Para la alimentación de los componentes son necesarios estos pines.
 - VCC. Conectado al sistema de alimentación del cNGD, accesible desde los pines 14, 24 y 28.

- GND. Conectado a la masa del cNGD, accesible desde los pines 15, 31 y 34.
- SPI. Para la conexión serie de dispositivos por medio de este estándar. Las tres señales que necesita, SDI, SDO y SCK son accesibles desde los pines 21, 38 y 37, respectivamente.
- I²C. Para la conexión serie de dispositivos por medio de este estándar. Encontramos disponibles dos canales del microcontrolador:
 - I²C 2. SCL en el pin 38 y SDA en el pin 21.
 - I²C 4. SCL en el pin 11 y SDA en el pin 10.
- ADC. Son accesibles desde los pines 9 de los 16 ADC del microcontrolador, referenciados en la tabla 2.1 con el prefijo ANx. Adicionalmente, los pines VREF+ y VREF-, situados en el 19 y 18, proporcionan funciones auxiliares a este periférico.
- CN (Change Notification). Se genera una interrupción en el sistema cuando se produce un cambio en la señal de entrada. Un total de 9 CN son accesibles desde los pines del *header*.
- GPIO. Representan casi todos los pines del *header*, multiplexados en su mayoría con otra función. Son identificados en la tabla 2.1 mediante el uso del prefijo Rxx.

Finalmente, el cNGD es programado por medio del módulo de programación PGE. Esta misma interfaz es usada para la depuración del sistema. Esto será útil para comprobar la ejecución correcta del código desarrollado, por lo que se ampliará en el capítulo 4.

2.2. Sensores

A continuación comenzamos el estudio de los sensores, con objeto de determinar cuáles incluir en la placa de expansión. Estos van a ser la entrada de información ambiental a nuestro sistema, ya que por medio de sus mediciones conoceremos los parámetros del entorno. Dada su relevancia debemos escogerlos adecuadamente, por lo que definimos el procedimiento que detallamos a continuación:

1. *Definir los requisitos y valorarlos.* En esta fase seleccionamos los parámetros más destacables que nuestro sensor debe cumplir, y un margen de tolerancia para los mismos. Además, como restricción adicional, tendremos que cumplir los requisitos impuestos por la plataforma, tales como la tensión de alimentación, comunicación por medio de puertos disponibles, entre otros.
2. *Buscar sensores comerciales.* Tomando como base los resultados de la fase anterior, y usando las páginas web de los principales proveedores (como Farnell [19], RS [20], Digikey [21] o SparkFun [22]), realizaremos la búsqueda.
3. *Estudiar las hojas de características.* Éstas son proporcionadas por los fabricantes y en ellas se detallan todas las propiedades que definen su producto. De esta evaluación obtenemos la verificación de que cumplen las características definidas.

Como requisitos genéricos de nuestro sistema, y que por tanto deberán ser motivo de evaluación, definimos:

- *Consumo.* No debemos olvidar que el sistema, al estar alimentado con baterías, requiere un estricto control sobre el gasto energético, así como permitir estados de bajo consumo. Este requisito fue considerado a la hora del diseño del cNGD, permitiendo la desconexión o la suspensión de los dispositivos que no se encuentran en uso, y por tanto será también objeto de evaluación.
Se buscará, de entre toda la gama de sensores, aquellos cuyas especificaciones de consumo sean menores, o bien que puedan ser sometidos a estados de bajo consumo.
- *Precio.* Al igual que en el caso anterior, el bajo precio es uno de los pilares fundamentales para el diseño. Esto es muy comprensible, puesto que a la hora de fabricar lotes de unidades una pequeña diferencia en coste puede llevar a grandes ahorros económicos.
Tendremos que encontrar una solución de compromiso entre especificaciones del sensor y el coste.
- *Salida.* La podemos clasificar en dos tipos:
 - Analógica, en el cual algún parámetro eléctrico, tal como la tensión, corriente o resistencia, es dependiente de la magnitud a medir y requiere, por tanto, del empleo de un ADC para su tratamiento en el microcontrolador.
 - Digital, en el cual la medición se representa por una palabra binaria, y que se conecta al microcontrolador, generalmente, por buses serie, como pueden ser SPI o I²C, o entradas genéricas (GPIO).

No imponemos ningún filtro *a priori*, puesto que el cNGD dispone de entradas tanto digitales como analógicas.

En el mercado podemos encontrar muchos tipos de sensores, en función del parámetro que se desee medir, por lo que centraremos nuestra búsqueda en base a la aplicación. Como nuestro propósito es el de crear una placa con un propósito genérico dentro de un AmI, escogeremos los sensores ambientales que puedan ser utilizados en un número alto de aplicaciones. Se presentan en los siguientes puntos.

2.2.1. Temperatura

Un sensor de temperatura se basa en la dependencia con la temperatura de alguno de los materiales que lo forman, y ofrecen este dato como resultado. Según [23], los sensores de temperatura se pueden clasificar en:

- *Metálicos resitivos*, en los que la dependencia con la temperatura viene relacionada por la presencia de un elemento metálico, como puede ser el platino, el cobre o el níquel; y la variación de la temperatura se transforma en una variación en su resistencia eléctrica.
- *Resistivos semiconductores*, conocidos como termistores, presentan en su seno materiales semiconductores. Encontramos dos subtipos, el NTC (*Negative Temperature Coefficient*), formados a partir de óxidos de materiales pesados; y el

PTC (*Positive Temperature Coefficient*), que incluye materiales como el $BaTiO_3$ o el $SrTiO_3$.

- *Semiconductores activos*, basados en la relación de la tensión base-emisor de un transistor con la temperatura.
- *Termopares*, basados en efectos termoeléctricos, como pueden ser el de Seebeck, el de Peltier y el de Thompson.

Un sensor que nos permita conocer la temperatura es muy habitual encontrarlo en WSN, ya que es un parámetro que es útil en muchas de las aplicaciones potenciales del sistema, ya sea conocer la temperatura de una estancia, con el objeto de actuar sobre ella, si nos encontramos en un entorno doméstico o de trabajo; o de conocer la temperatura del suelo, con el objeto de monitorizar su variación con el tiempo, si es usado en temas de investigación medioambiental.

Los principales parámetros que los definen son:

- *Rango de medida*. Define las temperaturas máximas y mínimas en las que se puede medir.
En nuestra aplicación puede ser tan amplio como se pueda, pero fijando como mínimo un rango de medida entre -20 °C y 100 °C.
- *Precisión*. Es una forma de expresar la calidad de la medida, entendiendo ésta como la fiabilidad entre el dato medido y el dato real. En ocasiones también aparece nombrado como *error*. Como es comprensible, es un parámetro que varía mucho el precio del mismo.

Puesto que lo usaremos en una aplicación genérica, debemos encontrar un compromiso entre precio y precisión, por lo que centraremos nuestra búsqueda en valores de entre ± 0.2 °C y ± 2 °C.

Los sensores de temperatura comerciales suelen ser, en su mayoría, termistores, a los que se les añade un circuito de acondicionamiento, de forma que en la salida únicamente encontramos variaciones en tensión o, directamente, una secuencia de bits. Los que han sido evaluados, siguiendo estos requisitos, aparecen resumidos en la tabla 2.2.

Fabricante	Modelo	Rango de Medida	Precisión ¹	Consumo ²	Precio	Sal. ³	Ref.
Analog Devices	TMP36	[-40 °C, 125 °C]	± 1 °C	50 μ A	1.61 €	An.	[24]
Microchip	MCP9701A	[-40 °C, 125 °C]	± 2 °C	6 μ A	0.24 €	An.	[25]
Texas Instr.	LM20B	[-55 °C, 130 °C]	± 2 °C	10 μ A	1.07 €	An.	[26]
Texas Instr.	TMP101	[-55 °C, 125 °C]	± 2 °C	45 μ A	2.56 €	Dig.	[27]
Analog Devices	AD7814	[-55 °C, 125 °C]	± 2 °C	250 μ A	2.71 €	Dig.	[28]
Microchip	MCP9800	[-55 °C, 125 °C]	± 0.5 °C	200 μ A	0.98 €	Dig.	[29]

Tabla 2.2: Comparación de sensores de temperatura comerciales.

Como comentario final, resaltar que una característica adicional de los modelos con salida digital es la disposición de modos de bajo consumo.

¹Típica en el margen que incluye a 25 °C.

²Típico durante el modo de funcionamiento

³Salida. “An.” denota salida analógica, y “Dig.” salida digital.

2.2.2. Acelerómetro

Definimos un acelerómetro como un sensor que nos permite medir las aceleraciones que sufre el elemento sobre el cual se inserta. Podemos encontrar varios tipos, a saber:

- *Mecánico.* Basado en la Segunda Ley de Newton, se fabrica mediante la unión de la masa de prueba a un dinamómetro cuyo eje se encuentra en la dirección en la que se quiere medir.
- *Piezoeléctrico.* Tiene su fundamento físico en el hecho de que, cuando se comprime un cristal con propiedades piezoeléctricas, se crea una carga eléctrica que es directamente proporcional a la fuerza que se aplica, la cual a su vez es proporcional a la aceleración.
- *Tecnología MEMS (Micro-Electro-Mechanical System).* Están basados en el traspaso por convección de calor, cuya variación es causada por la aceleración.

A priori, puede parecer que introducir un sensor de este tipo es extraño para la obtención de parámetros de un entorno. No obstante, el uso de este tipo de sensores está creciendo en multitud de aplicaciones, por la versatilidad que le caracteriza. Tanto es así que puede ser usado, por ejemplo, para detectar la posición de la placa en el plano tridimensional, o para detectar vibraciones, bien generadas por la acción humana (como pisadas en el suelo, vehículos pasando por un puente o la apertura de una puerta) o por la naturaleza (como los terremotos).

Las principales características que los definen se resumen a continuación:

- *Fondo de escala.* Es la máxima aceleración que se puede detectar. Se define como una proporción de la aceleración provocada por la gravedad terrestre, de forma que un valor de $2g$ implica una medida máxima de aceleración de 2 veces la de la gravedad.
Como consecuencia de estar desarrollando una placa para una aplicación genérica, pondremos como requisito que sea capaz de detectar, al menos, una aceleración de $8g$.
- *Sensibilidad.* Se define como el impacto que genera una variación de la variable de entrada a la salida, de forma que un valor alto de sensibilidad implica que una leve variación de la entrada ocasione una gran variación a la salida.
En los acelerómetros, este parámetro es dual al fondo de escala, de forma que cuanto más bajo sea el fondo de escala más alta será la sensibilidad.
- *Ejes de medida.* Número de ejes en los que se podrán detectar las variaciones en la aceleración. Estos se corresponden con los ejes de coordenadas: \hat{x} , \hat{y} y \hat{z} .
En este aspecto, no admitiremos considerar acelerómetros que no midan en los tres ejes, ya que entendemos que no podemos dejar de considerar ninguno.

En la actualidad, los acelerómetros comerciales para propósitos generales están basados en tecnología MEMS, construidos e integrados en un chip de silicio, y que permiten fabricar acelerómetros de tres ejes. La tabla 2.3 muestra los acelerómetros comerciales que han sido estudiados.

⁴Típico en estado de medición continuo.

Fabricante	Modelo	Fondo de escala	Ejes	Consumo ⁴	Precio	Ref.
Freescale	MMA8453Q	2g, 4g, 8g	$\hat{x}, \hat{y}, \hat{z}$	23 μA	1.05 €	[30]
Analog Devices	ADXL345	2g, 4g, 8g, 16g	$\hat{x}, \hat{y}, \hat{z}$	14 μA	5.09 €	[31]
ST	LIS3DH	2g, 4g, 8g, 16g	$\hat{x}, \hat{y}, \hat{z}$	11 μA	3.72 €	[32]

Tabla 2.3: Comparación de acelerómetros comerciales.

No aparece información acerca de la salida en dicha tabla puesto que todos los modelos estudiados tienen salida digital. Asimismo, todos disponen de modos de bajo consumo.

2.2.3. Presencia

Un sensor de presencia nos permite determinar el movimiento de personas bajo el área geográfica que cubre. Estos se denominan sensores PIR (*Passive Infrared*), debido a que se basan en el efecto de que la persona emite radiación infrarroja por consecuencia de tener una temperatura distinta del cero absoluto. El sensor es capaz de determinar la presencia por comparación entre las radiaciones recibidas, de forma que se activa si detecta un cambio.

Este tipo de sensores son muy comunes en AmI en los que es necesario el control de personas. Así, este sensor puede emplearse en aplicaciones de seguridad (si se detecta presencia en momentos en los que no se debería detectar puede generarse una alarma), eficiencia energética (regulando parámetros ambientales como la temperatura o la luminosidad en función de la ocupación del espacio), entre otros.

Las características más destacables que podemos encontrar en un sensor de presencia serían:

- *Lente de Fresnel*. Es la responsable de concentrar las radiaciones en el foco del sensor, para que puedan ser detectadas. Este parámetro es muy importante, puesto que un PIR sin este tipo de lente únicamente detectaría presencia en una zona muy limitada del espacio.

Como consecuencia, para nuestro diseño exigiremos la existencia de esta lente, bien sea integrada en el propio sensor o añadida a parte.

- *Área cubierta*. Guarda relación directa con la lente de Fresnel, como hemos visto, siendo ésta la que determine la máxima distancia a la que se podrá detectar la presencia.

En el contexto de una aplicación genérica, pondremos como requisito una distancia de detección mínima de 3 metros.

La tabla 2.4 muestra los sensores de presencia bajo estudio, en los que han sido evaluadas todas las características enunciadas anteriormente.

Fabricante	Modelo	Lente	Distancia	Consumo ⁵	Precio ⁶	Ref.
Murata	E710ST0	Externa	[3 m, 6 m]	N/D	5.61 €	[33]
Panasonic	EKMC160	Integrada	5 m	100 μA	15.49 €	[34]
Panasonic	AMN33112J	Integrada	3 m	270 μA	11.91 €	[35]

Tabla 2.4: Comparación de sensores de presencia comerciales.

La salida de todos los modelos estudiados es digital, de forma que obtenemos un nivel alto de tensión en caso de detectar presencia, y un nivel bajo en caso contrario.

2.2.4. Luminosidad

Un sensor de luminosidad es aquel que permite determinar la cantidad de luz que se encuentra en el espacio de medida del mismo. Podemos encontrar principalmente dos tipos: las fotorresistencias, conocidas por sus siglas inglesas LDR (*Light Dependent Resistor*), las cuales son unas resistencias cuyo valor depende de la luz, y los fotodiodos, que ofrecen a la salida una corriente que es proporcional al nivel de iluminación que encuentren. Serán estos últimos sobre los que centraremos nuestra búsqueda.

La elección de este sensor viene motivada por las múltiples aplicaciones que puede desempeñar en el contexto de los AmI, como puede ser la de eficiencia energética, regulando la luz artificial en función de la luz solar, o la de confort, adaptando los niveles de luminosidad en función de nuestra preferencia o de la situación en la que nos encontramos.

Un sensor de luminosidad se define principalmente en base a los siguientes parámetros:

- *Pico de longitud de onda.* Determina la longitud de onda del fotón cuya respuesta mejor se percibe. A partir de ésta, se dibuja la curva de respuesta del sensor frente a otras longitudes.
Impondremos que esta curva de respuesta sea lo más similar posible a la del ojo humano, puesto que esto hará que el sensor responda con la misma información que recibe una persona.
- *Corriente de oscuridad.* Parámetro que determina la corriente que ofrece a la salida en condiciones de iluminación nula. Supone un *offset* a la medida.
Al ser un dato parásito, exigiremos que sea tan pequeño como sea posible.
- *Salida.* Todos ellos ofrecen una salida analógica en corriente, al estar buscando fotodiodos. No obstante, algunos modelos presentan algún circuito que modifica su curva de salida, pasando de ser lineal con la luminosidad a hacerlo de forma logarítmica.

En la tabla 2.5 se encuentran los sensores de luminosidad bajo estudio, donde se evalúan todas las características enunciadas anteriormente. En la misma, el consumo coincide con las corrientes en condiciones de iluminación (a 100 luxes) y oscuridad, como consecuencia de ser un sensor analógico en corriente.

Fabricante	Modelo	Pico en λ	Oscuridad	Iluminación ⁷	Salida	Precio	Ref.
Everlight	ALS PT19	630 μm	100 nA	15 μA	Lineal	0.76 € ⁸	[36]
Avago	APDS 9005	500 μm	300 nA	40 μA	Lineal	0.74 €	[37]
Osram	SFH5711	550 μm	<10 nA	20 μA	Logarítmica	2.27 €	[38]

Tabla 2.5: Comparación de sensores de luminosidad comerciales.

⁵Típico en detección.

⁶Incluye el precio por sensor y lente, en caso de que la última sea externa.

⁷Típico bajo 100 luxes.

⁸De venta en SparkFun.

2.3. Actuadores

Los actuadores constituyen la forma de modificación del entorno como consecuencia de la medición realizada por parte de los sensores y de la inteligencia del sistema. Estos pueden ejercer la función de actuar directamente, como podría ser el caso de un relé, o bien de hacer de nexo entre el sistema y el controlador responsable de la actuación, como puede ser emular un mando a distancia.

En esta placa, dado su propósito general, se tenderá hacia la fabricación de este segundo tipo, de forma que se buscarán actuadores que se caractericen por su versatilidad y su generalidad, aunque teniendo siempre presente la funcionalidad. No se presentan tablas comparativas, como se ha realizado con los sensores, porque los modelos comerciales de los actuadores bajo estudio poseen características muy similares.

Los actuadores que desarrollan esta función los encontramos resumidos a continuación.

2.3.1. Emisor Infrarrojo

Un emisor Ir (*Infrarrojo*) no es más que un LED (*Light-Emitting Diode*) cuyo espectro de emisión de luz se centra en estas longitudes de ondas, típicamente en el entorno de los 950 nm. Esta señal es modulada para la transmisión digital de comandos, propios para cada dispositivo.

La inserción de este actuador en la placa está más que justificada, ya que son incontables los dispositivos que pueden controlarse hoy en día mediante un mando a distancia por Ir, como pueden ser la televisión, equipos de música, la climatización o las persianas, entre otros.

Las características que deberá reunir este actuador serán las de potencia media de transmisión, para que su uso no se vea limitado en exceso con la distancia, y un tiempo de respuesta que permita la modulación de las señales a transmitir.

2.3.2. Buzzer

Un buzzer es un dispositivo que, mediante la vibración de un elemento piezoelectrónico, produce una onda acústica a unas determinadas frecuencias, las cuales dependen de la forma de excitación del dispositivo y del propio material.

El propósito de este actuador no es otro que el de avisar a los usuarios acerca de las condiciones ambientales, generalmente cuando suponen la existencia de un peligro potencial, para que sean éstos los que, en último término, lleven a cabo la actuación.

Como consecuencia de este hecho, se exigirá que el buzzer emita, al menos, a 70 dB(A), para asegurar un nivel suficiente de volumen que permita su audición.

2.3.3. Salidas digitales

Estas salidas son muy útiles, puesto que, a partir de ellas, es posible controlar cualquier dispositivo externo cuya señal de control sea digital. En el microcontrolador, estas salidas se corresponden con los puertos GPIO, a varios de los cuales da nuestra plataforma acceso en sus *headers*, como vimos en la sección 2.1.

Capítulo 3

Estudio del Diseño y de la Implementación

*El conocimiento, si no se sabe aplicar,
es peor que la ignorancia.*

Charles Bukowski

RESUMEN: En este capítulo, se seleccionan los componentes y los dispositivos que formarán parte de la placa. Se expondrán las decisiones de diseño, las especificaciones, los esquemas y los cálculos necesarios. Además, se describirán las tareas relativas a la implementación física de la PCB, así como el resultado final.

3.1. Elección de los sensores y actuadores

Con los conocimientos adquiridos durante la realización del estudio previo, pasamos a seleccionar los componentes que formarán de forma definitiva nuestra placa de expansión. En la figura 3.1 se presenta un esquema del diseño modular de nuestra *shield*.

La estructura que se seguirá en los siguientes subapartados se presenta a continuación.

1. Elegimos, de manera justificada, el sensor o actuador comercial que se ajusta mejor a las necesidades de nuestro problema.
2. Se presentará un diagrama de pines del dispositivo, junto con las señales de entrada y salida.
3. Se detallarán y se justificarán los componentes adicionales que se han de incluir.
4. Finalmente, se decidirá el conexionado de las señales con el cNGD.

3.1.1. Sensores

3.1.1.1. Temperatura

De los modelos presentados en la tabla 2.2, hemos decidido seleccionar el **Microchip MCP9800** [29], principalmente por ser el de mayor precisión a un precio

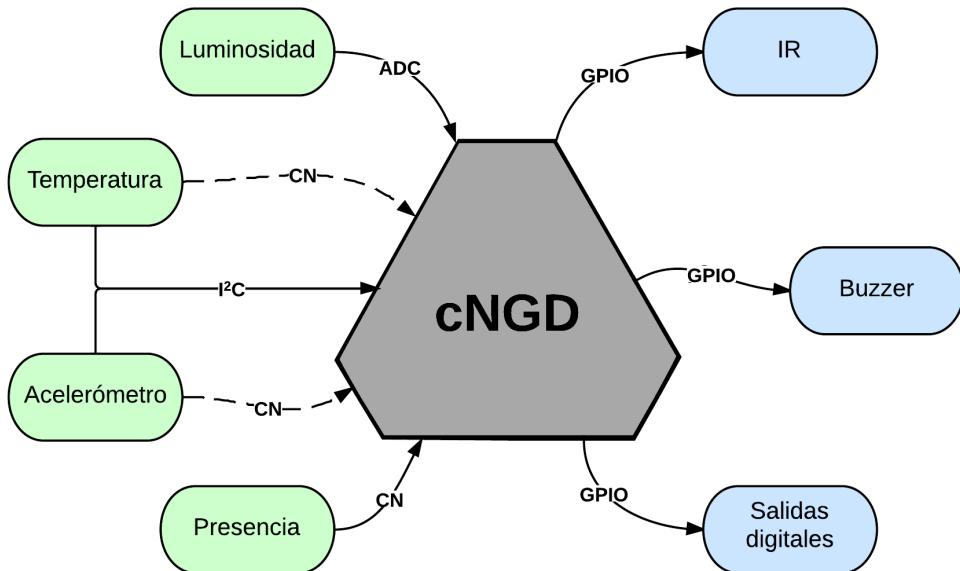


Figura 3.1: Diagrama modular del diseño.

asequible. Sus características detalladas se exponen a continuación.

- Precisión típica de ± 0.5 °C a 25 °C.
- Error máximo de ± 1 °C entre -10 °C y 85 °C.
- Resolución seleccionable de 9-12 bits.
- Conexión I²C.
- Corriente de operación de 200 μ A.
- Corriente de apagado de 1 μ A.
- Salida independiente configurable (ALERTA).

La figura 3.2 muestra las entradas y salidas del sensor por medio de su diagrama de pines, así como el circuito que precisa para funcionar.

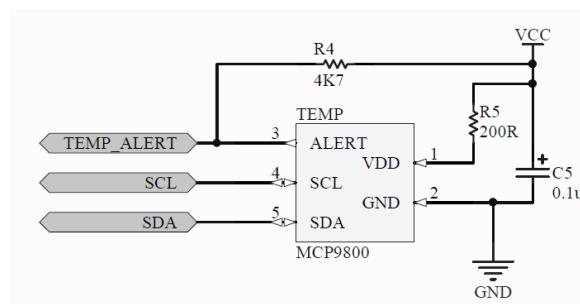


Figura 3.2: Sensor de Temperatura Microchip MCP9800.

Comentamos brevemente los componentes añadidos. En la etapa de entrada encontramos R5 y C5, cuya función es la de actuar a modo de filtro paso bajo ante el posible ruido procedente de la alimentación, influyente para la medida. R4 es una resistencia de *pull-up*, necesaria ya que la salida del pin 3 es de colector abierto. Además de esta resistencia, y con este mismo propósito, se incluirán dos resistencias de *pull-up* para la

conexión por I²C, que no vienen representadas en este esquema por estar compartida con todos los dispositivos que la usan.

3.1.1.2. Acelerómetro

En base al análisis realizado en la sección 2.2.2, en el que se presentaban los distintos sensores comerciales estudiados, seleccionamos el **Freescale MMA8453Q** [30] por ser el más económico que cumple las especificaciones. A continuación detallamos todas sus características:

- Fondos de escala seleccionables de $\pm 2g$, $\pm 4g$ y $\pm 8g$.
- Resolución seleccionable de 8-10 bits.
- Conexión I²C.
- Modos de bajo consumo.
- Consumo de corriente de entre $6 \mu\text{A}$ y $165 \mu\text{A}$.
- Dos salidas programables para seis fuentes de interrupción.
 - Dato disponible.
 - Caída libre.
 - Pulso (*tap*).
 - Orientación.
 - Movimiento.
 - *Auto-SLEEP*.

A continuación encontramos, en la figura 3.3, el diagrama de pines junto con el circuito requerido para su operación.

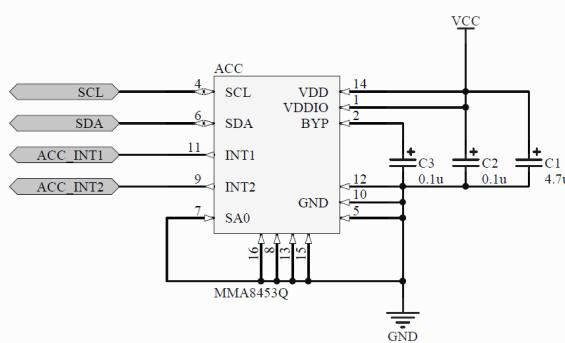


Figura 3.3: Acelerómetro Freescale MMA8453Q.

En primer lugar, resaltar que el pin 3 no se encuentra en la figura. Esto es así puesto que el fabricante nos indica que no debe ir conectado. Además, hemos conectado a masa el pin 7 (SA0), que hace que el bit menos significativo de la dirección I²C del dispositivo sea 0.

Comentamos a continuación los componentes del circuito. C1 y C2 son condensadores de desacoplo para las dos entradas de alimentación (principal e interfaz), mientras que el condensador C3 es denominado de *bypass*. Al igual que como hemos comentado en el sensor de temperatura, son necesarias dos resistencias de *pull-up* para la interfaz I²C y que no aparecen en este esquema.

3.1.1.3. Presencia

Siguiendo la misma línea que con los sensores anteriores, y en base a lo expuesto en la tabla 2.4, seleccionamos como sensor de presencia el **Panasonic AMN33112J** [35]. En un primer lugar seleccionamos el Murata E710ST0, pero tras una primera prueba vimos que no se adecuaba a nuestra aplicación, por lo que nos decantamos por el que a continuación exponemos. Sus características son:

- Salida binaria.
- Detección de personas en hasta 3 m.
- Detección de movimientos de 30 cm.
- Consumo en espera de $170 \mu\text{A}$.
- Consumo en detección de $270 \mu\text{A}$.

El diagrama de pines se presenta en la figura 3.4. Lo único que observamos es la resistencia R11, de *pull-down*, necesaria puesto que la salida del sensor en estado de espera es circuito abierto.

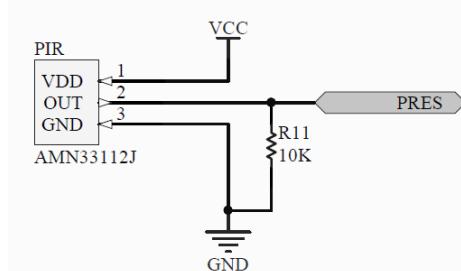


Figura 3.4: Sensor de Presencia Panasonic AMN33112J.

3.1.1.4. Luminosidad

Para finalizar, exponemos y justificamos la elección del sensor de luminosidad. Se trata, del conjunto de los estudiados en la sección 2.2.4, del **Avago APDS 9005** [37], seleccionado principalmente por su bajo precio y respuesta similar a la del ojo humano. Sus características ampliadas serían:

- Salida analógica en corriente.
- Fotocorriente con 10 lux de $19 \mu\text{A}$.
- Fotocorriente con 100 lux de $230 \mu\text{A}$.
- Corriente de oscuridad de 50nA .
- Tensión de saturación de 2,32 V.

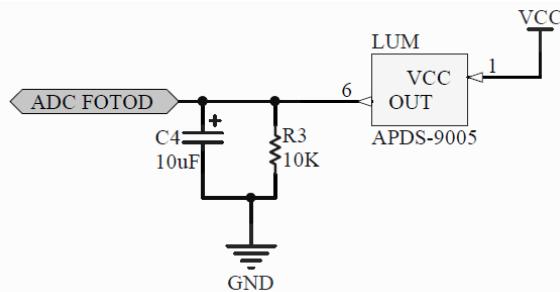


Figura 3.5: Sensor de Luminosidad Avago APDS 9005.

El diagrama de entradas y salidas del sensor lo encontramos en la figura 3.5. Al igual que en casos anteriores, los pines 2-5 no figuran al indicarse que no deben ser conectados a ningún punto.

De los componentes que encontramos a la salida del sensor, justificamos la presencia de la resistencia R3 con el objeto de convertir la photocorriente a tensión, de forma que puede ser introducida al ADC. Siguiendo la ley de Ohm, la tensión será tanto mayor como lo sea la resistencia. Esto nos crea un compromiso, ya que un valor bajo hará que se resienta la precisión de la medida (al ser el escalón del ADC constante), mientras que un valor alto hará que la salida llegue demasiado pronto a la saturación. Finalmente, se ha escogido un valor de 10 KΩ.

Finalmente, el condensador C4 es responsable de hacer un filtrado en la señal, ya que las luminarias basadas en fluorescentes introducen un rizado en la misma de 100 Hz.

3.1.2. Actuadores

3.1.2.1. Emisor Infrarrojo

El Ir escogido fue el **Vishay TSUS5202** [39], el cual se caracteriza por los siguientes parámetros técnicos, además de tener un precio de 0.17 €.

- Componente TH (*Through Hole*). ■ Tiempo de respuesta t_r de 800 ns.
- Longitud de onda λ_p de 950 nm. ■ Corriente máxima de 0.1 A.

De forma análoga a los casos anteriores, presentamos el circuito necesario correspondiente a este actuador en la figura 3.6.

El único componente que encontramos es una resistencia, cuya función es la de limitar la corriente que circula por el diodo, puesto que si excede de la máxima especificada podría dañarse. Para no disminuir la potencia de emisión, que es directamente proporcional a la corriente de excitación, impondremos que dicha corriente sea tan próxima a la máxima como sea posible. La calculamos como:

$$R_8 \geq \frac{V_{CC} - V_\gamma}{I_{Max}} = \frac{3,3V - 0,7V}{0,1A} = 26\Omega$$

Para asegurar, escogemos un valor de resistencia de 33Ω, de forma que se tendrá una corriente de 87 mA.

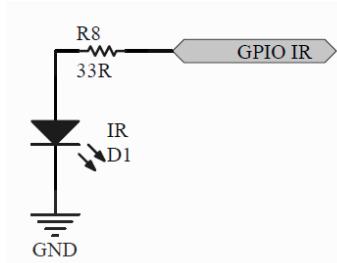


Figura 3.6: Emisor Infrarrojo Vishay TSUS5202.

3.1.2.2. Buzzer

Para el zumbador elegimos el modelo comercial **TDK PS1720** [40], idóneo al ser diseñado para aplicaciones de alarma, como es nuestro caso, y con un precio de 0.75 €. La figura 3.7 muestra el circuito para su correcto funcionamiento.

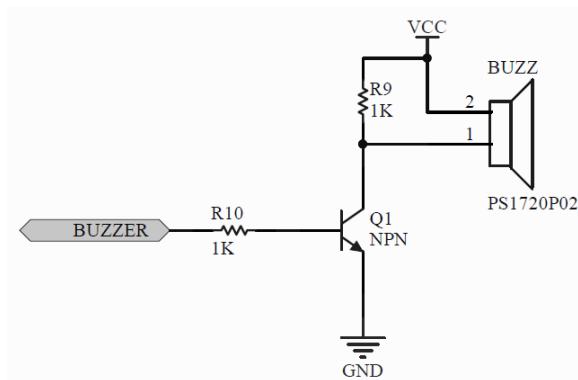


Figura 3.7: Buzzer TDK PS1720.

Podemos observar varios componentes en este diseño. En primer lugar remarcamos la presencia del transistor NPN Q1, cuya función es la de hacer de amplificador de corriente a la entrada del zumbador. Además, en paralelo con el buzzer se introduce la resistencia R9 con el propósito de realizar la carga y la descarga del elemento piezoelectrónico. Finalmente, la resistencia R10 se introduce por seguridad.

3.1.2.3. LEDs

Con el objeto de poder hacer depuración en etapas siguientes del proceso, y como salida visual de diferentes parámetros, añadimos dos LED, verde y rojo, al diseño de la placa. Su circuito se presenta en la figura 3.8.

En este caso, los valores de las resistencias han aumentado con respecto a lo visto en el caso del LED Ir, puesto que ahora nos basta con que las mismas tengan un nivel de iluminación tal que se permita discernir si están encendidas o apagadas, por lo que no es necesaria tanta corriente. En este sentido, hemos considerado que un valor de 330Ω es suficiente.

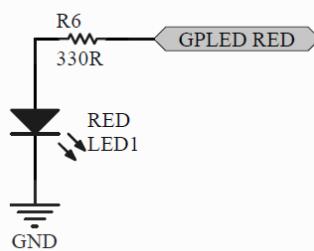


Figura 3.8: LED de información y depuración.

3.2. Diseño de la placa de expansión

En los siguientes subapartados se presenta la forma en la que los distintos componentes que han sido objeto de análisis en la sección 3.1 se combinan entre sí y cómo se integran dentro de nuestro nodo.

3.2.1. Diagrama de bloques

En el diagrama de bloques de la figura 3.9 presentamos las interfaces del MCU y su conexión con los sensores y actuadores de nuestra placa.

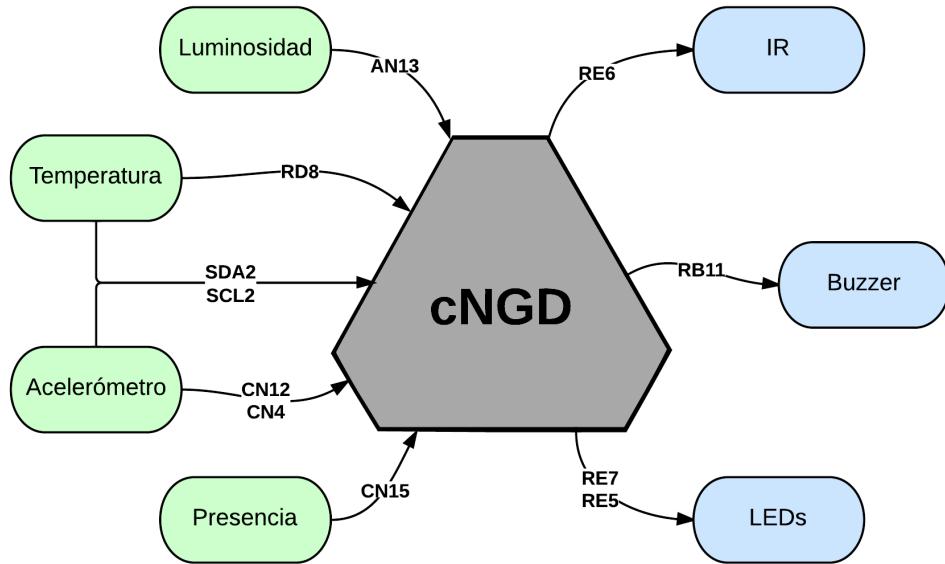


Figura 3.9: Diagrama de bloques de la placa de expansión.

En adición, y a modo de resumen, presentamos en la tabla 3.1 una relación de los dispositivos, tanto sensores como actuadores, la forma de comunicación y los pines del *header* que ocupa, así como el periférico del MCU del que hacen uso, en relación con lo visto en la tabla 2.1.

Justificamos la asignación de los puertos de salida de los sensores que deben generar interrupciones. Tales son los casos del acelerómetro –con INT1 e INT2– y del de presencia, que se encuentran rutados hacia CNs. No figura en este conjunto el sensor

Dispositivo	Comercial	Puerto	Com.	<i>Header</i>	Función
Temperatura	MCP9800	ALERT	GPIO	9	RD8
		I ² C	SDA SCK	10 11	SDA2 SCK2
Acelerómetro	MMA8453Q	INT1 INT2	CN	16 2	CN12 CN4
Presencia	AMN 33112J	CN	33	CN15	
Luminosidad	APDS 9005	ADC	5	AN13	
Infrarrojo	TSUS 5202	GPIO	35	RE6	
Buzzer	PS1720	GPIO	3	RB11	
LEDs	Rojo Verde	GPIO	22	RE7	
			23	RE5	

Tabla 3.1: Entradas y salidas de los *headers* hacia la placa de expansión.

de temperatura que, a pesar de tener una salida independiente de alerta, ésta solo se modifica cuando el sensor se encuentra en estado despierto, y, puesto que es presumible que sea el estado de bajo consumo el que se encuentre habilitado, no es necesario un puerto CN.

A la hora de considerar la asignación de los distintos pines a las entradas y salidas de los componentes, se tuvo en cuenta, aparte de que la función requerida estuviera disponible en el mismo, que no reutilizara ningún otro pin ya usado para hacer otras *shields*, con lo que conseguimos que todas sean compatibles entre sí. En este sentido, evaluamos el uso que hacían de los *headers* las placas de expansión correspondientes al módulo RS232 y al cargador [9] y a la del módulo de *wake-on radio* y OTAP (*Over The Air Programming*) –en desarrollo en la actualidad.

3.2.2. Esquemático

El esquemático, donde se aprecia todo el circuito que conforma la placa de expansión, se presenta en la figura 3.10.

Como peculiaridad, resaltar que las resistencias R1 y R2 son las responsables del *pull-up* de las dos señales del I²C que han sido referidas pero no reflejadas anteriormente.

3.3. Implementación en PCB

Una vez acabado el diseño, procedemos a realizar el circuito en PCB.

Detallamos a continuación los subapartados que encontraremos. En primer lugar, se expone el trazado de la placa, en el que se presenta el diseño de la PCB. Para el mismo, se detalla seguidamente la lista de materiales, que relaciona cada huella con su componente. Finalmente, la *shield* final es expuesta.

3.3.1. Trazado

A continuación, en la figura 3.11 se presenta el trazado de nuestro diseño.

Antes de entrar a analizar el diseño del trazado, comentamos brevemente el porqué de la forma física de la placa. Esta responde al hecho de que las *shields* ya realizadas

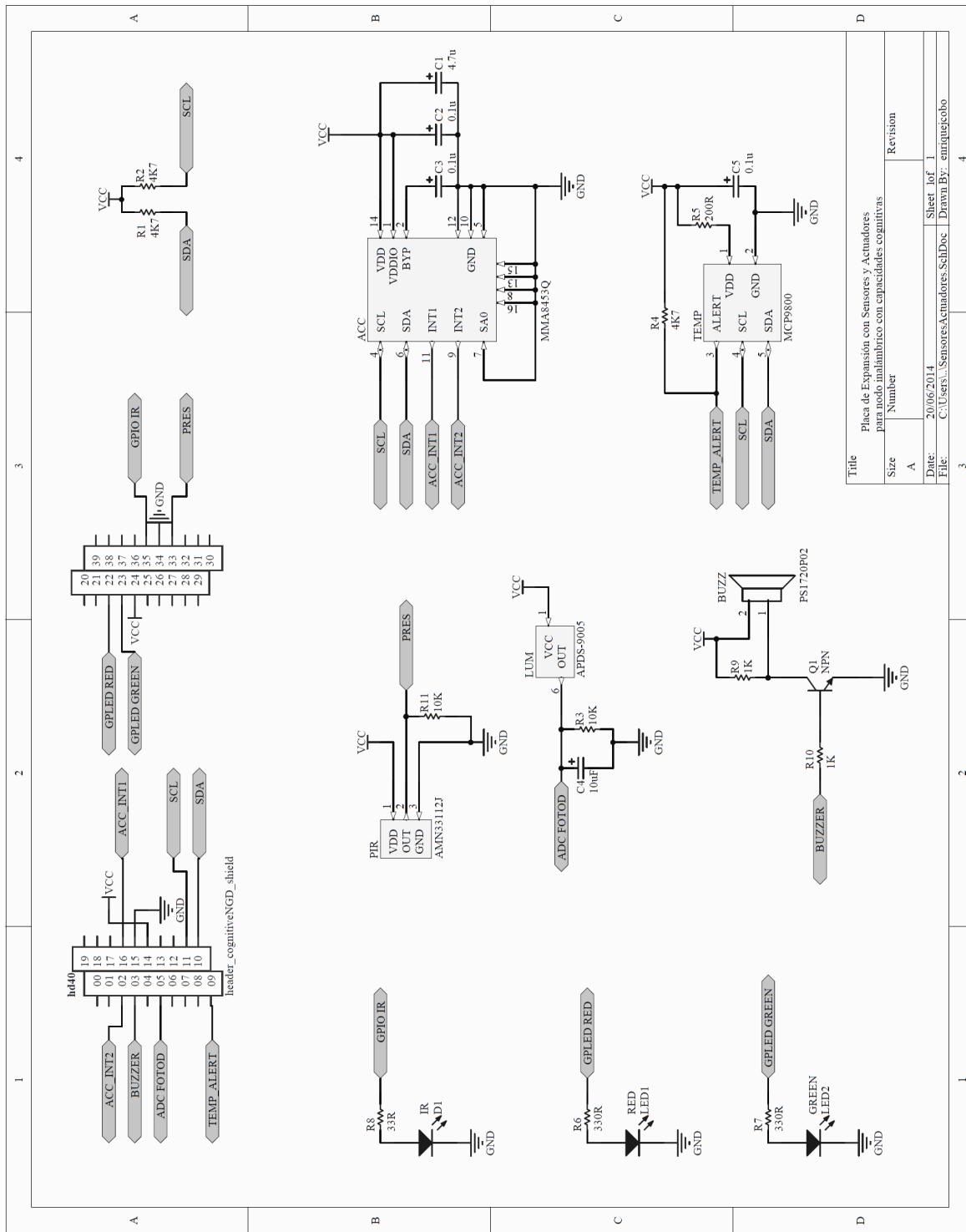
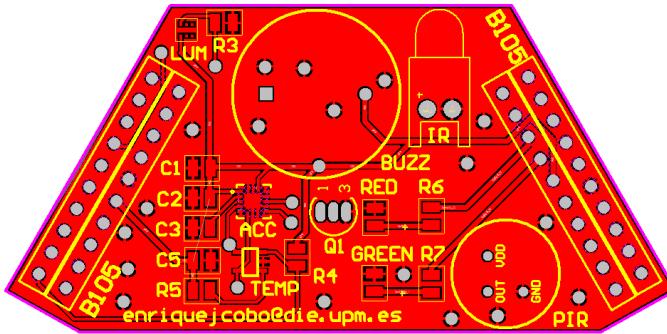


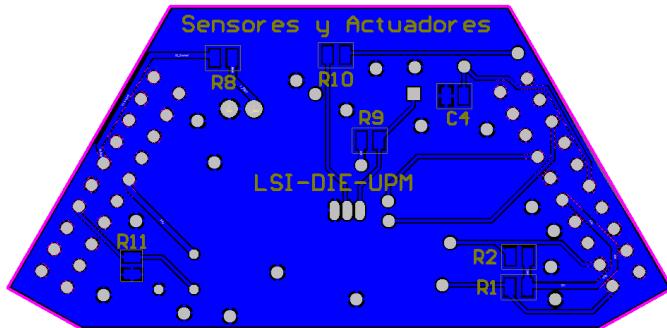
Figura 3.10: Esquemático de la placa de expansión.

tenían esta forma, por lo que se ha mantenido para que encajara física y estéticamente con el diseño del cNGD.

Para realizar el diseño de la PCB, se han tenido en cuenta diversas consideraciones. En primer lugar, y como se puede apreciar, se ha construido en dos capas, suficiente



(a) Trazado de la cara superior



(b) Trazado de la cara inferior

Figura 3.11: Trazado de la placa de expansión.

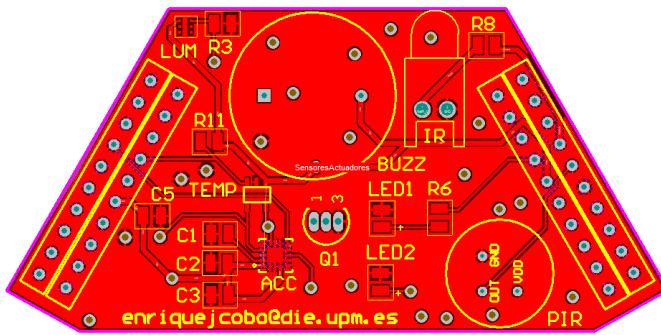
para la colocación de los componentes. Es por ello por lo que no ha sido necesario crear planos de alimentación, hecho que hubiera aumentado, además, el coste de producción.

En segundo lugar, durante el rutado, se buscó un compromiso entre anchura de pista (que disminuye las pérdidas por resistividad) y tamaño de la placa. En base a esto, las pistas correspondientes a alimentación, con mayor paso de corriente, se diseñaron más anchas que las usadas para transmitir información.

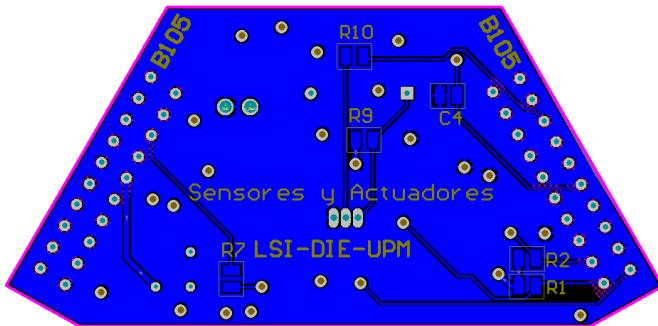
Adicionalmente, apuntamos la existencia de un plano de masa sobre ambas caras de la PCB. Con esto conseguimos una reducción del ruido y de la influencia de emisiones electromagnéticas parásitas, puesto que este plano ofrece un camino de baja impedancia para las mismas.

Como parte del trabajo asociado a diseñar el trazado de la *shield*, se implementaron las huellas de los componentes que no se encontraban en las librerías, como se detalla en la sección 3.5.

Remarcamos que el trazado que acaba de ser expuesto se corresponde con el de la segunda versión. El de la primera, que puede verse en la figura 3.12, tuvo que ser modificado, principalmente por el hecho de que la huella del acelerómetro aparecía reflejada, lo que hacía que los pines no coincidieran. No obstante, y aprovechando este segundo diseño, se modificaron ciertas características, como puede ser el rutado de las salidas de interrupción hacia CNs en lugar de hacia GPIOs, y la adición de la resistencia de *pull-down* y el reflejo de la huella del PIR, de forma similar al caso del acelerómetro. A esta versión la denominamos previa.



(a) Trazado de la cara superior



(b) Trazado de la cara inferior

Figura 3.12: Trazado de la placa de expansión previa.

3.3.2. Lista de Materiales

La tabla 3.2 muestra todos los componentes que son necesarios para la fabricación de la *shield*.

Como consideración de diseño, y siempre que ha sido posible, para los componentes adicionales hemos preferido usar encapsulado SMD (*Surface-Mount Devices*), y con tamaño 0805¹. Esto viene motivado debido al hecho de que ocupan un menor tamaño que sus homólogas en TH, ofreciendo características similares, y permiten igualmente diseños de bajo consumo.

3.3.3. Resultado final

La figura 3.13 muestra el resultado final de nuestra placa de sensores y actuadores. Se corresponde con el diseño de la segunda versión.

También se incluye en esta memoria el resultado final correspondiente a la versión previa de la placa de expansión, en la figura 3.14. Podemos observar que el sensor de presencia aparece conectado, ya que fue posible su inclusión girando los pines de inserción y soldando una resistencia de *pull-down* TH, y que la huella del acelerómetro aparece vacía. No obstante, esta versión nos ha sido muy útil puesto que nos ha permitido realizar sobre ella toda la batería de pruebas funcionales, con excepción del acelerómetro.

¹2 mm x 1.3 mm (0.08 in x 0.05 in).

Símbolo	Valor	Descripción	Encapsulado	Cantidad
TEMP	-	Microchip MCP9800	SOT 23-5	1
ACC	-	Freescale MMA8453Q	QFN 16	1
PIR	-	Panasonic AMN33112J	TH Radial	1
LUM	-	Avago APDS 9005	SMD	1
IR	-	Vishay TSUS 5202	TH Radial	1
BUZZ	-	TDK PS1720	TH Radial	1
RED	-	LED Rojo	0805	1
GREEN	-	LED Verde	0805	1
R1,2,4	4.7 KΩ	Resistencia	0805	3
R3,11	10 KΩ	Resistencia	0805	2
R5	200 Ω	Resistencia	0805	1
R6,7	330 Ω	Resistencia	0805	2
R8	33 Ω	Resistencia	0805	1
R9,10	1 KΩ	Resistencia	0805	2
C1	4.7 μF	Condensador cerámico	0805	1
C2,3,5	0.1 μF	Condensador cerámico	0805	3
C4	10 μF	Condensador cerámico	0805	1
Q1	-	Transistor NPN	TH Axial	1
hd40	-	Header apilable de 10 pines	TH Axial	4

Tabla 3.2: Lista de Materiales.

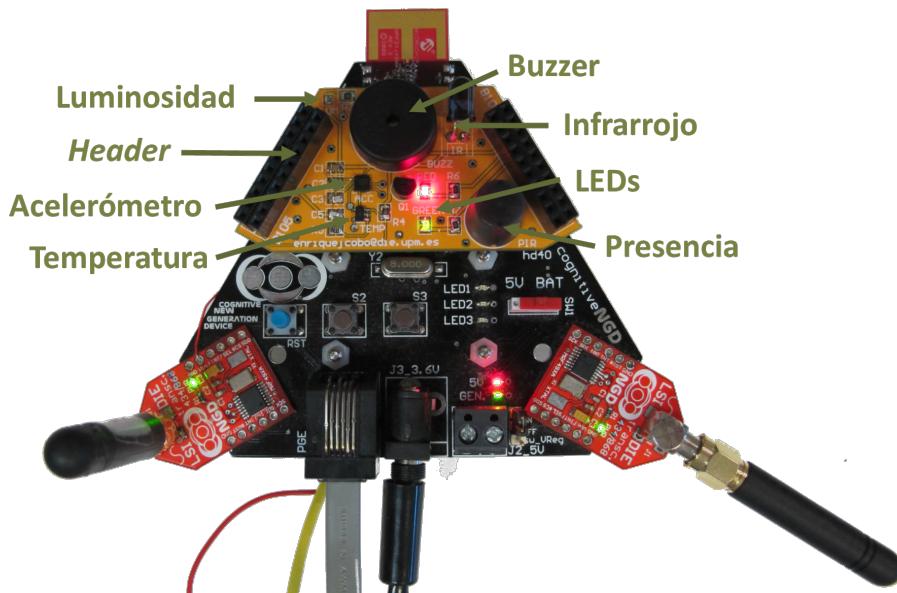


Figura 3.13: Resultado final: la placa de expansión sobre el cNGD.

3.4. Costes hardware

En la tabla 3.3 se encuentran desglosados los costes *hardware* de la realización completa de la placa, tanto de la PCB como de los componentes electrónicos necesarios.

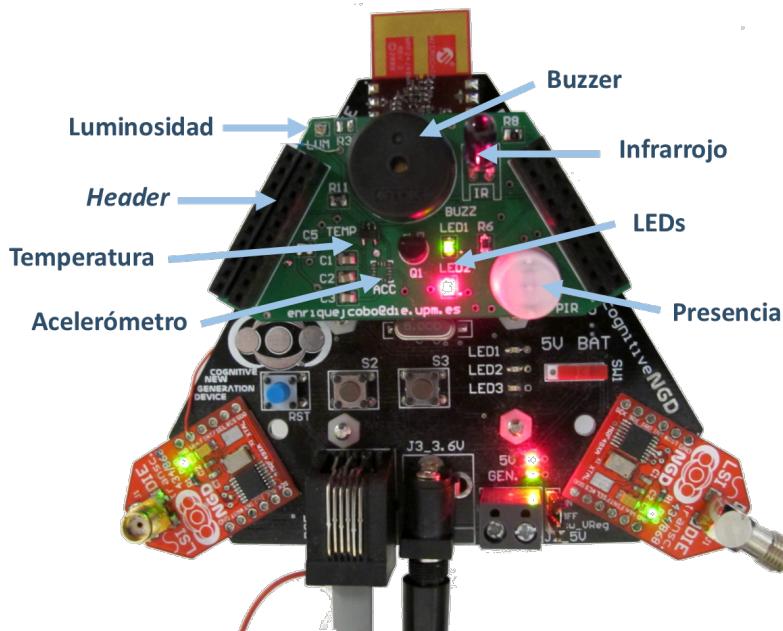


Figura 3.14: La placa de expansión previa sobre el cNGD.

Concepto	Precio unidad	Cantidad	Coste total
Fabricación de PCB			
Placa de circuito impreso	0.95 €	1	0.95 €
Componentes electrónicos			
Microchip MCP9800	0.98 €	1	0.98 €
Freescale MMA8453Q	1.05 €	1	1.05 €
Panasonic AMN33112J	11.91 €	1	11.91 €
Avago APDS 9005	0.74 €	1	0.74 €
Vishay TSUS5202	0.17 €	1	0.17 €
TDK PS1720	0.75 €	1	0.75 €
Transistor	0.12 €	1	0.12 €
LEDs	0.085 €	2	0.17 €
Resistencias	0.022 €	11	0.24 €
Condensadores	0.008 €	5	0.04 €
Header 10 pines	0.19 €	4	0.76 €
Total			17.88 €

Tabla 3.3: Costes *hardware* de la placa de expansión.

3.5. Herramientas

Para la realización del diseño y la implementación, ha sido necesario usar ciertas herramientas, tanto *software* como *hardware*. Las introducimos a continuación.

3.5.1. Altium Designer

Altium Designer ha sido el *software EDA* (*Electronic Design Automation*) utilizado para los aspectos eléctricos de diseño. Altium Designer es una aplicación orientada para

la creación de PCB, FPGA, entre otras; desarrollada por Altium Limited.

Del conjunto de funciones que integran este *software*, las principales usadas serán las concernientes al diseño del esquemático y de la PCB. En este sentido, destacamos las siguientes:

- *Gestión de librerías de componentes*, tanto de los esquemas como de sus huellas. En este sentido, se han usado:
 - LSI. Librería colaborativa desarrollada en el LSI en el que se encuentran los componentes ya utilizados por los investigadores.
 - SensoresActuadores. Librería propia en la que se diseñaron, tanto los esquemas como sus huellas, los componentes que no se encontraban en la anterior. Continuando con la filosofía, una vez testados se incorporaron a la misma.
- *Posicionamiento y rutado de componentes*, que nos permite diseñar libremente nuestro trazado.
- *Revisión de reglas de diseño*. Con los datos provenientes del fabricante de la PCB, introducimos una serie de requisitos, los cuales eran verificados automáticamente por Altium.
- *Generación de ficheros para fabricación*, fundamental para la realización física de la PCB.

3.5.2. Puesto de Soldadura

El montaje de la placa es una de las tareas de este trabajo. La estación de soldadura JBC AM6000 [41] fue usada para realizar las soldaduras, proporcionando puntas para este propósito de entre 1 y 4 mm de diámetro, además de otras herramientas prácticas, como pinzas calientes o soldadora por aire caliente.

Debido a la presencia de componentes de tamaño reducido, se hizo necesario el uso de un microscopio con el objeto de realizar soldaduras precisas. Es por ello que en el puesto de soldadura también contábamos con un microscopio Bausch & Lomb StereoZoom 4, así como de unas mordazas que permitían mantener fija la placa y unas pinzas adecuadas para poder colocar los componentes SMD en su posición a la hora de realizar la soldadura.

Capítulo 4

Software

*La gente crítica con el software
debería hacer su propio hardware.*

Alan Kay

RESUMEN: En este capítulo se presentan las funciones que permiten obtener medidas de los sensores y enviar órdenes a los actuadores, creando una capa *software* que permite su uso. Adicionalmente se detalla la aplicación de prueba que ilustra el funcionamiento del sistema en su conjunto.

4.1. Interfaz Hardware-Software

A continuación detallamos los trabajos realizados para el desarrollo de un conjunto de funciones *software* responsables de controlar los sensores y los actuadores de la placa de expansión. Puesto que esta interfaz realmente abstrae los recursos *hardware* implementados, se la denomina HAL.

El lenguaje usado para la realización de la misma ha sido C, por dos motivos. En primer lugar, porque se dispone de una amplia librería de funciones ya desarrolladas por el fabricante para el microcontrolador en este lenguaje, y en segundo lugar, porque el *software* que implementa el cNGD también fue escrito en C. El manual de referencia usado ha sido [42].

Nuestro código está integrado básicamente en dos ficheros: `SensorsHAL.c`, que es el fichero maestro en el que se implementan las funciones necesarias, y `SensorsHAL.h`, en el que se encuentran las cabeceras exportadas de las mismas.

Todo el código desarrollado, así como el necesario para que el sistema completo funcione, se encuentra accesible públicamente en GitHub, siguiendo el siguiente enlace o consultando el código QR de la figura 4.1: <http://github.com/enriquejcobo/tfg-soft>.

4.1.1. Funciones generales y adaptación

En los siguientes subapartados iremos presentando las funciones que se han desarrollado para el manejo de los sensores y actuadores. En primer lugar presentamos las funciones que son comunes al propio sistema, debido a que engloban varios dispositivos del mismo. Estas son:



Figura 4.1: Repositorio público del *software*.
<http://github.com/enriquejcobo/tfg-soft>

- **void InitSensors ()**. Su misión es la de preparar los periféricos dentro del MCU, de forma que se habiliten los necesarios y en los pines asignados.
 - Se habilita la interfaz I²C, usada por el sensor de temperatura y el acelerómetro.
 - Se configura el ADC que será requerido por el sensor de luminosidad.
 - Los puertos de entrada, tanto CN como GPIOs, usados por cada sensor, son inicializados.
 - Los GPIOs de salida para todos los actuadores se marcan como tal.
 - Por último, se configura el temporizador interno, mediante el uso de la librería **timer.h**, cuyo uso se justifica a continuación.
- **void IntTmp ()**. Rutina de atención a la interrupción causada por el temporizador interno TMR5, que se utiliza para:
 - Definir tiempos de espera, necesarios en la medida de temperatura bajo condiciones de bajo consumo.
 - Obtener señales periódicas, como la que necesita el Buzzer.
 - Generar modulaciones, imprescindible para la transmisión de datos vía infrarroja.
- **void IntCN ()**. Rutina de atención a la interrupción motivada por el cambio en alguna señal rutada hacia un CN, como son las interrupciones 1 y 2 del acelerómetro, o la salida del sensor de presencia.

No obstante, la necesidad y justificación del uso de las mismas será detallada en el apartado correspondiente a cada elemento de la *shield*.

A parte de estas funciones, ha sido necesario modificar en parte los siguientes archivos, pertenecientes al *firmware* del cNGD, para que la integración de la placa en el mismo sea posible. En este sentido, los archivos modificados han sido:

- **HardwareConfig.h**. En este fichero se encuentran las definiciones de los periféricos usados por el nodo, de forma que el diseño es totalmente modular, comentando y descomentando en función de las necesidades particulares de cada unidad. En este sentido, se ha definido la placa y, dentro de ella, cada uno de los sensores y actuadores que pueden formar parte, de tal manera que si se prescindiera de alguno las funciones y los periféricos del MCU asociados dejarían de ser usados con este propósito.
- **HardwareProfile.h**. Se producen las asignaciones de los pines a sus diferentes periféricos y a las funciones del microcontrolador, de forma que un cambio en los mismos se simplifica únicamente modificando su etiqueta.

4.1.2. Sensores

De forma paralela a como se viene estructurando este documento, presentamos a continuación todo el código desarrollado para los sensores que figuran en nuestra placa.

4.1.2.1. Temperatura

El sensor de temperatura dispone de una interfaz I²C, por el que se realiza la lectura y configuración de los registros, y una salida GPIO, denominada ALERT.

Sus funciones aparecen recogidas a continuación:

- **int getTemp ()**. Esta función devuelve la medida de la temperatura en forma de un entero, si bien éste representa un número real de coma fija, en el que el primer byte representa la parte entera y el segundo la parte decimal.
- **void setTempResolution (int res)**. Permite modificar el registro de configuración asignando un nuevo valor de resolución, en función del argumento, que es un número entre 0 y 3, ambos inclusive.
Este argumento muestra la cantidad de bits fraccionarios que se usarán, de forma que un 0 conlleva ningún bit fraccionario (con una precisión, por tanto, de ±0.5 °C), mientras que un 3 nos ofrece como resultado 3 bits (con una precisión de ±0.0625 °C).
- **void getTempAlert (int reg, INT8 alert)**. Modifica los registros correspondientes a los umbrales superior (**MAX_HIGH**) e inferior (**MAX_LOW**), poniendo el valor de temperatura **alert** en los mismos. **alert** debe redondearse al entero más próximo.
- **void setTempLowPower ()**. Modifica el registro de configuración poniendo el sensor de temperatura en un estado de bajo consumo, en el que se detiene la conversión de temperatura pero permanece abierta la interfaz I²C. La medida de temperatura, en este caso, necesita un tiempo de conversión, para lo que se hace uso del temporizador interno.
- **BOOL getTempAlert ()**. Lee, del GPIO correspondiente, el valor de si hay alarma detectada o no.

A continuación mostramos aquellas funciones que cumplen la misión de simplificar el código, evitando duplicidades si dentro de dos funciones distintas se requería hacer

la misma acción. En el caso del sensor de temperatura, serán principalmente las que llamen a las funciones que proporciona el fabricante en `i2c.h` para la comunicación por medio de este protocolo. Generalmente, estas funciones no están exportadas.

- `void setTempConf ()`. Realiza la escritura en el registro de configuración. Se encuentra exportada al ser requerida por todas las funciones que realizan cambios sobre el mismo, como pueden ser `setTempResolution ()` o `setTempLowPower ()`.
- `void setTempRegister (UINT8 reg)`. Abre la comunicación I²C haciendo las peticiones necesarias para leer el registro que se le pasa por parámetro. Es la función que la llama la responsable de obtener el dato y de cerrar la comunicación.
- `UINT8 getTempConf ()`. Función auxiliar que permite comprobar que el estado del registro interno de configuración coincide con la copia de él que el programa dispone. De esta manera garantizamos la concordancia entre ambos.

4.1.2.2. Acelerómetro

Nuestro acelerómetro se conecta al sistema por medio de la interfaz I²C y por medio de las señales INT1 e INT2, rutadas a dos puertos CN. Las funciones desarrolladas para manejarlo se describen a continuación.

- `void getAcc ()`. Función que consulta al dispositivo las aceleraciones correspondientes a los tres ejes (\hat{x} , \hat{y} , \hat{z}) y las guarda en variables locales. Puesto que son tres valores distintos, tomamos la decisión de diseño de que esta función únicamente trajera los datos, siendo necesario invocar otra para acceder a ellos.
- `INT8 getAccX (), INT8 getAccY (), INT8 getAccZ ()`. Devuelven el valor de las variables locales en las que previamente se han almacenado los valores de las aceleraciones en los ejes correspondientes.
- `void setAccInt (int interrupcion, int source)`. Permite modificar los parámetros correspondientes a la `interrupcion`, como puede ser la fuente (`source`) que la ocasiona. Hay definidas constantes tanto para la interrupción, `ACC_INT1` y `ACC_INT2`, como para las posibles fuentes: `DATA_READY`, `MOTION_FREEFALL`, `TAP`, `ORIENTATION`, `TRANSIENT` o `ASLEEP`. Asimismo, `NO_INT` borra la asociación de la fuente a la salida.
- `void setAccLowPower ()`. Induce al acelerómetro a un modo de bajo consumo.
- `BOOL getAccInt1 (), BOOL getAccInt2 ()`. Devuelve TRUE o FALSE en función de si hay interrupción pendiente o no. En principio estas funciones no serían necesarias, puesto que al estar rutadas hacia CNs, el vector de interrupción correspondiente atendería la petición cuando se genere. No obstante, se referencian por si fueran necesarias.

De forma similar a como sucedía en el caso del sensor de temperatura, existen funciones auxiliares que simplifican diversas tareas y son útiles a la hora de desarrollar nuevas funciones, puesto que parte del código ya se encuentra desarrollado. En este grupo también incluimos las que se ocupan de gestionar la conexión I²C haciendo uso de la librería `i2c.h`. Estas son:

- `void setAccRegister (UINT8 reg, UINT8 dato)`. Escribe en el registro el dato correspondientes, ambos pasados por parámetro. Realiza las peticiones I²C para que produzca esta transferencia de información.
- `UINT8 getAccRegister (UINT8 reg)`. Permite leer el contenido de un registro del acelerómetro. Puede ser útil para comprobar validez de los datos, u obtener información adicional.

4.1.2.3. Presencia

De forma similar a como ocurría en el caso anterior, la salida de este sensor se encuentra conectada a un CN, por lo que será necesario atender a la rutina de interrupción `IntCN ()` y, en base a esto, definir las acciones que sean necesarias.

No obstante, puesto que este puerto también es accesible bajo demanda, creamos la función `BOOL getPIR ()`, la cual devuelve `TRUE` si halla presencia, o `FALSE`, en caso contrario.

4.1.2.4. Luminosidad

Como ya conocemos, la salida del sensor de luminosidad es una tensión proporcional a la fotocorriente causada por la iluminación del dispositivo, por lo que se conecta a un puerto ADC del microcontrolador, para que éste haga la conversión, a 10 bits y en el intervalo 0 V–3.3 V, de la tensión que le llega.

Para consultar el valor de luminosidad, basta con llamar a `int getLum ()`, el cual solicita realizar la medida al ADC. Esta función devuelve un entero que representa la tensión.

Todas estas medidas son posibles gracias al uso de la librería `adc.h`, la cual contiene las funciones necesarias tanto para inicializar el ADC como para realizar las lecturas.

4.1.3. Actuadores

Análogamente, procedemos a presentar las funciones que requieren los actuadores incorporados.

4.1.3.1. Emisor infrarrojo

La comunicación infrarroja se implementa bajo diversos estándares, puesto que cada fabricante define su propio protocolo de comunicaciones entre sus equipos. Como muestra de ello se presentan, en la figura 4.2, las tramas procedentes de mandos a distancia de fabricantes distintos.

No obstante, estos estándares sí unifican el hecho de que para este tipo de comunicaciones se usan modulaciones basadas en el ancho de pulsos sobre portadoras de 34 kHz, y siguiendo un esquema temporal similar al que se muestra en la figura 4.3.

La decisión de diseño tomada en esta HAL ha sido la de usar un *buffer*, puesto que previmos que en la implementación final del sistema se tendría la necesidad de enviar varios comandos seguidos. El tamaño del mismo puede controlarse por la constante `BUFFER_MAX`. Además, con este método conseguimos que la transmisión por el puerto Ir no bloquee el programa principal, puesto que el encadenamiento de varias puede originar un paro significativo en la ejecución.

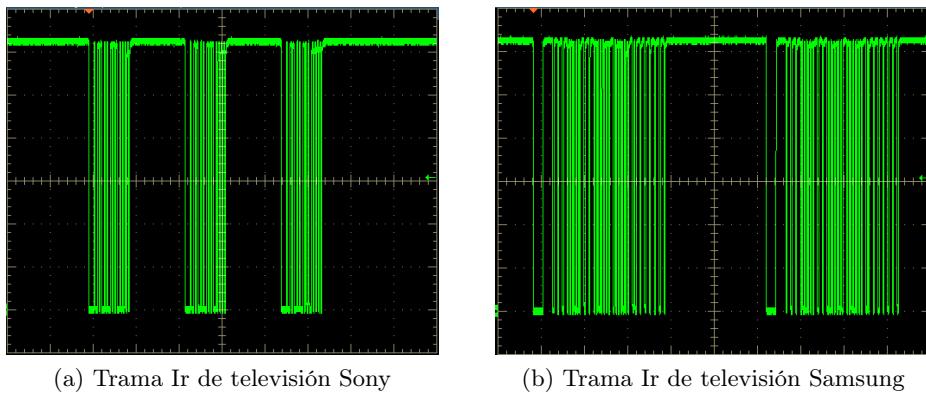


Figura 4.2: Tramas de comunicación infrarroja de distintos fabricantes.

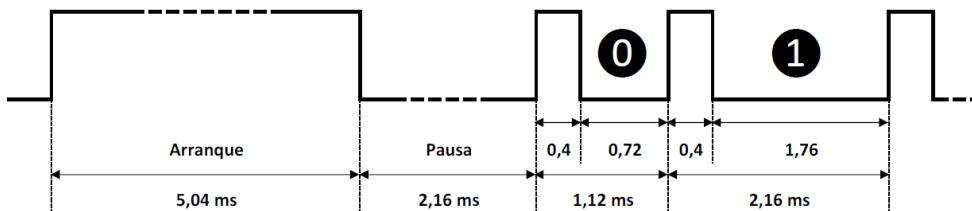


Figura 4.3: Temporalización de la comunicación infrarroja.

Como resultado de requerir una modulación, será necesario emplear de nuevo el temporizador interno, con la frecuencia suficiente que permita obtener una señal pulsada de 34 kHz.

La función `void sendIR (INT8 device, INT8 param1, INT8 param2)` añade comandos a la cola de transmisión. El parámetro `device` fija el dispositivo, mientras que `param1` y `param2` establecen los nuevos valores que deben aplicarse, y que dependerán del dispositivo. Se recomienda fijar estos valores por medio de etiquetas.

Las funciones, no exportadas, que apoyan esta tarea son:

- **void protocoloIR ()**. Incluye la máquina de estados que gobierna la transmisión de los datos. Su misión es que, en cada interrupción del temporizador, se ponga en la salida del GPIO el valor que corresponda, según la posición en la que nos encontramos.
Al ser dependiente de la aplicación, deberán implementarse tantos como dispositivos a controlar.
 - **int mandarUno ()**, **int mandarCero ()**. Puesto que estos dependen de la trama y el hecho de enviar un “0” o un “1” implica una parte de modulación y otra de silencio, decidimos crear dos funciones que simplifiquen esta tarea, dada la cantidad de veces que tendrán que ejecutarse.

4.1.3.2. Buzzer

El buzzer se activa mediante una señal rectangular periódica, cuya frecuencia determina además la frecuencia del tono acústico que percibimos. Hemos decidido implementar dos tonos que se alternan, con el motivo de tratar de imitar el sonido caracte-

terístico de una alarma. Las frecuencias de ambas serían $f_{High} = 2$ kHz y $f_{Low} = 1,5$ kHz, aproximadamente.

Para este actuador volveremos a requerir la interrupción periódica, la cual, como consecuencia de lo estudiado en la sección 4.1.3.1, es capaz de soportar las frecuencias que se precisan.

Se han desarrollado dos funciones, `void buzzerOn ()` y `void buzzerOff ()`, que gobiernan el control del mismo. En este sentido, la primera hace que el buzzer empiece a sonar y la segunda lo detiene.

4.1.3.3. LEDs

Como se expuso en la sección 3.1.2.3, se han introducido dos LED que permiten mostrar distintas señales, en función del contexto y del sistema final. Las funciones que producen su modificación son las que siguen:

- `LedOn (sensorLed s1)`, que enciende el LED `s1`.
- `LedOff (sensorLed s1)`, en el que se fuerza el apagado del LED `s1`.
- `LedToggle (sensorLed s1)`, que conmuta el estado del LED `s1`.

`sensorLed` es un tipo definido que consta de los valores `GREEN` y `RED`, los cuales identifican al LED por su color, y `BOTH`, que indica que se actúe sobre ambos.

4.2. Aplicación de prueba

Se ha desarrollado una aplicación demostrativa con el objeto de que ilustre el correcto funcionamiento de la placa diseñada en combinación con el *software* implementado.

No cabe olvidar que la meta del trabajo reside en incluir nuestra *shield* dentro de un nodo de una CWSN, por lo tanto es necesario que las placas implantadas en distintos nodos sean capaces de compartir información entre ellas.

En este sentido, se desarrolla una aplicación que consta de los siguientes pasos:

1. Se ejecutan en primer lugar las rutinas de inicialización del sistema, tanto del cNGD como de la placa de Sensores y Actuadores.
2. Se establecen las comunicaciones, siguiendo el protocolo P2P (*Peer-to-Peer*) Uni-cast¹, y se deciden los canales por los que se realizará la comunicación.
3. Se ejecuta el programa de prueba, comprobando periódicamente:
 - Si la temperatura alcanza una temperatura máxima predefinida, se mandan instrucciones al aire acondicionado por vía Ir para que se encienda. En caso de que baje de una temperatura mínima, se mandaría el comando con efecto opuesto.
 - Un giro de la placa, que será detectado por el acelerómetro, disparará la alarma del nodo vecino hasta que vuelva a su posición de reposo.
 - En caso de presencia detectada, se ilumina el LED verde; y en caso contrario, se enciende el rojo.

¹En un único sentido, lo que implica la existencia de un nodo transmisor y un nodo receptor.

- Mientras el nivel de luz esté por debajo de un umbral predefinido, sonará el buzzer del nodo emisor.

Remarcar que, para esta aplicación de prueba, la consulta del valor de los sensores se realiza mediante *polling* de periodo 3 segundos (salvo el de presencia, que genera interrupciones). Esto quiere decir que obtenemos un nuevo dato cada 3 segundos, lo que hace que las actuaciones se realicen coincidiendo con este periodo. Además, esta aplicación genera tramas en las que se muestran los datos leídos por los sensores y algunas actuaciones. Para esto, hemos hecho uso de la interfaz RS-232 y de las librerías ya desarrolladas con este fin.

El archivo que contiene este *software* se denomina `MainAppCode_NodeTest.c`, el cual es el responsable de llamar a las funciones de inicialización y de albergar el bucle para que se ejecuten estas acciones de forma continua. Además, es necesario definir la función de cada nodo en particular, como corresponde a una aplicación P2P Unicast.

Este programa se ha diseñado bajo la idea de que ilustre el funcionamiento de todos los sensores y actuadores que se incluyen en la placa, de forma que una variación de los mismos implique alguna actuación sobre el entorno.

4.3. Herramientas

Al igual que en casos anteriores, presentamos las herramientas utilizadas para la realización del *software* de tanto la HAL como de la demo.

4.3.1. MPLAB X

MPLAB X es la IDE (*Integrated Development Environment*) desarrollada por Microchip para la programación de sus microcontroladores. Está basado en NetBeans IDE, *open-source* y de Oracle, e incluye integrado el compilador de C para sus dispositivos. Entre algunas de las opciones que ofrece destacamos:

- Varias herramientas de depuración.
- *Parsing* y control de sintaxis en tiempo real.
- Hipervínculos que permiten una navegación rápida para acceder a las declaraciones.

Más información puede ser consultada en el manual [43].

4.3.2. Programador: ICD 3

El ICD (*In-Circuit Debugger*) es el dispositivo que permite programar el microcontrolador. En nuestro caso usaremos el ICD 3 [44], el cual ha sido desarrollado para los productos de Microchip. Como característica adicional, este dispositivo puede ser usado, en combinación con el MPLAB, para la depuración en tiempo real del *software* corriendo en el MCU, permitiendo hasta 6 puntos de parada. Este hecho resultará crucial en los siguientes apartados.

Capítulo 5

Pruebas y Resultados

*Hay una forma de hacerlo mejor
-encuéntrala.*

Thomas Alva Edison

RESUMEN: En este capítulo se presentan las pruebas que fueron realizadas a los diferentes elementos que conforman nuestro sistema, desde el doble punto de vista del *hardware* y del *software*. Los resultados de las mismas, así como las particularidades encontradas, son igualmente detalladas. Finalmente se resumen las principales herramientas usadas con este propósito.

5.1. Hardware

A continuación se presentan las pruebas que se realizaron desde el punto de vista del comportamiento *hardware* a tanto los dispositivos como a la placa en la que se enmarcan.

5.1.1. Sensores y Actuadores

Detallamos en primer lugar las pruebas realizadas a los sensores de forma individual antes de ser incluidos, comenzando por el sensor de **Luminosidad**. Para verificar el correcto funcionamiento del mismo, se realizó el montaje detallado en la figura 3.5, y se comprobó que, para distintos niveles de iluminación (aplicándole una fuente lumínica externa o cubriendo el detector) obteníamos a la salida distintos niveles de tensión, medidos en el osciloscopio.

Asimismo, observamos que la adición del condensador reducía el ruido por rizado de alta frecuencia cuando la iluminación provenía de fuentes fluorescentes.

A continuación procedemos a evaluar el sensor de **Presencia**. El circuito que implementamos es el de la figura 3.4, alimentado con las mismas tensiones que tendría en la placa, para lo cual hicimos uso de una fuente de alimentación. Conectando la sonda al pin de salida, observamos que la señal se ponía a nivel alto cuando pasábamos la mano sobre él, y que volvía a nivel bajo cuando la quitábamos.

No obstante, nos percatamos de que la adición de la resistencia de *pull-down* nos introducía un efecto de carga en el nivel alto, si bien la tensión resultante no era

mucho menor de la que se debía obtener al ser la resistencia de valor alto, por lo que concluimos la prueba satisfactoriamente.

Las pruebas correspondientes a los sensores de **Temperatura** y **Acelerómetro** se exponen en el subapartado de *software*, ya que necesitan de la interfaz I²C para la obtención de resultados o modificación de parámetros.

En cuanto a los actuadores, comenzamos detallando el **Emisor Infrarrojo**. Para las pruebas con este dispositivo se hizo necesario un detector para la longitud de onda del mismo, por dos motivos. En primer lugar porque la luz que emite no es visible para el ojo humano y, en segundo lugar, aunque lo fuera, la modulación usada haría imposible la comprobación del buen funcionamiento. Es por ello que usamos el detector Vishay TSOP1138 [45], siguiendo un montaje similar al que encontramos en la figura 5.1.

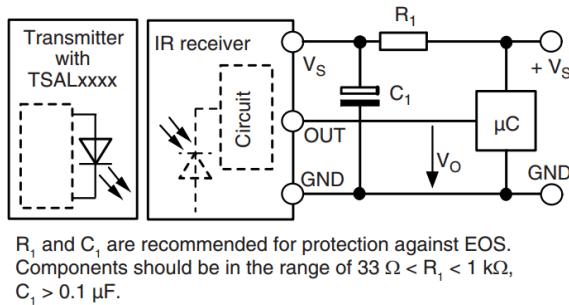


Figura 5.1: Detector infrarrojo Vishay TSOP1138, obtenido de [45].

Conectamos la salida de este detector a la sonda del osciloscopio, y alimentamos el circuito del Ir como ilustra la figura 3.6. Observamos que la salida del detector está a nivel alto cuando no recibe nada, y pasa a nivel bajo durante un instante cuando recibe señal de forma continua. Este comportamiento es el esperado, por lo que la prueba es satisfactoria.

Con los **LEDs** de propósito general, realizamos una prueba similar, alimentando el circuito de la figura 3.8. En este caso todo es más sencillo, puesto que la luz emitida por estos LED sí pertenece al rango visible, por lo que observamos que lucen perfectamente.

No fue necesario realizar una prueba similar con el **Buzzer**, puesto que cuando este componente fue recibido ya estaba desarrollado el *software* para el mismo, así que se probó en combinación con este.

5.1.2. PCB

Las pruebas realizadas en el ámbito de la PCB fueron generalmente test eléctricos para comprobar la soldadura, o las conexiones entre pistas. Para ello se hizo indispensable la ayuda de un multímetro, que será detallado más abajo, en la sección 5.3.2. Las mismas se detallan a continuación.

- En primer lugar se realizó una inspección óptica, comprobando que el trazado de la placa era el esperado.

- Cuando se soldaba un dispositivo, como puede ser un sensor, un actuador o un componente, se realizaron pruebas de corto, en las cuales comprobamos que las conexiones entre patas del componente eran nulas, y que había cortocircuitos entre el pad del mismo y el siguiente punto de conexión en el circuito.
- En caso de soldar resistencias, se usó el multímetro en posición de medir esta magnitud, de forma que comprobamos que el valor que aparecía en el medidor era similar al que debíamos obtener, con lo que verificábamos tanto una buena conexión como la inserción de la resistencia de valor adecuado.
- Acciones similares eran seguidas con los condensadores, buscando los mismos propósitos que en el caso anterior. Para ello, se configuró el multímetro en posición de medir capacidades.

Una vez satisfechas todas las pruebas, la PCB final ya podría ser sometida a pruebas de comportamiento del sistema.

5.2. Software

En este subapartado se detallan las pruebas hechas sobre el *software* desarrollado, que ha sido objeto de estudio en el capítulo 4. Las hemos clasificado en dos subapartados, las de la HAL independientes y las que se han desarrollado en combinación con los sensores y los actuadores.

5.2.1. HAL

Las pruebas realizadas sobre la HAL consistieron en medidas sobre las distintas interfaces que iban a formar parte de nuestro sistema. Pasamos a detallarlas a continuación.

En primer lugar probamos el **ADC**, cuya prueba consistió en formar un divisor de tensión por medio de resistencias conectadas a V_{CC} y a GND, e introducir la tensión en distintos puntos en el ADC, de forma que comprobamos que la palabra obtenida era la que correspondía.

En efecto, verificamos que las palabras en hexadecimal obtenidas cuando probamos con diferentes valores eran las que debían corresponder, por lo que determinamos que el funcionamiento era el correcto.

Para los **GPIO** de entrada, la prueba realizada para cada uno de ellos fue la de forzar un nivel alto, seguido de uno bajo, sobre las mismas, comprobando que el programa detectaba correctamente el nivel en el que se encontraba.

Análogamente para los de salida, se programó un nivel alto seguido de uno bajo, con puntos de parada, verificando por medio del osciloscopio que se obtenía el citado estado cuando la ejecución del programa llegaba a los puntos donde se debían variar.

Los actuadores que requieren la intervención de un temporizador, como son el **Emisor Infrarrojo** o el **Buzzer**, fueron probados de la siguiente manera. Se obtuvo la salida modulada que debían presentar, una señal cuadrada periódica de frecuencia 34 kHz para el caso del Ir, y 2 y 1.5 kHz para el caso del buzzer, y se introdujo esa señal en el osciloscopio, de forma que se midió la frecuencia.

5.2.2. Funciones para el manejo de los sensores y actuadores

Por último, se presentan las pruebas realizadas a las funciones desarrolladas para el control de los sensores y actuadores. En este punto los sensores y actuadores ya estaban soldados y la placa colocada sobre el cNGD, de forma que estos tests se llevaran a cabo bajo las mismas condiciones que tendrán en su entorno final.

En primer lugar presentamos las pruebas realizadas al sensor de **Temperatura**. En la sección 4.1.2.4 presentamos el conjunto de funciones que necesita:

- **getTemp**. Comprobamos que el sensor respondía a esta instrucción, y que el valor de temperatura devuelto se correspondía con el del entorno. Un aumento de temperatura fue forzado, y volvimos a repetir la medición, esta vez con un resultado superior.
- **setTempResolution**. Tras enviar este comando, procedimos a solicitar de nuevo la temperatura actual, observando en dicha medida que su precisión había variado conforme se le había indicado.
- **setTempAlert** y **getTempAlert**. Conocida la temperatura ambiente en el momento, e iniciando la prueba en una situación de no alarma, modificamos el valor de los registros de forma tal que este valor pasara a estar en la zona de peligro. Con la segunda función, comprobamos efectivamente que bajo estas nuevas condiciones se producía la alarma.

Seguidamente procedemos a evaluar el sensor de **Presencia**. Como se dijo en su sección del capítulo 4, el estado de este sensor puede ser evaluado bien mediante *polling*, preguntando periódicamente, o bien mediante la interrupción del puerto CN.

En el primer caso, vimos que la función **getPIR** devolvía valores correctos en función de si detectaba presencia o no. Valor que, a su vez, estaba comprobándose paralelamente en el osciloscopio.

En el caso de la interrupción, dejamos corriendo el programa y situamos un punto de parada en la rutina de atención a la interrupción. Cuando el sensor detectó presencia, el programa entró en la misma, por lo que verificamos el funcionamiento correcto.

Siguiendo con el sensor de **Luminosidad**, llevamos un proceso similar al visto en la sección 5.2.1, comprobando que la tensión que se obtiene a la salida del sensor (medida por el osciloscopio) y la palabra resultante de la llamada a la función **getLum** son concordantes; siendo así en efecto.

No se han podido comprobar todas las funciones del **Acelerómetro**, puesto que la versión definitiva no llegó hasta pocos días antes del fin de este proyecto. No obstante, sí hemos podido comprobar la función **getAcc** y las auxiliares que acceden a las variables locales, necesarias para la aplicación de prueba. Es por ello que somos optimistas respecto a la operación del sensor, conociendo además que los periféricos del MCU que usa el acelerómetro sí funcionan, ya que la interfaz I²C ha sido testeada con éxito con el sensor de temperatura, y los CN también han mostrado un comportamiento positivo en las pruebas desarrolladas con el PIR.

En cuanto a los actuadores, comenzamos evaluando el **Emisor Infrarrojo**. Decidimos implementar el protocolo del aire acondicionado, debido a que la placa dispone de sensor de temperatura y puede conocer, por tanto, el valor de la misma en tiempo real.

Evaluando trabajos previos [46], obtenemos que la información en las tramas del aire acondicionado se organiza de la siguiente manera: En primer lugar en 2 bytes se codifica el dispositivo, seguido de un byte de protección frente a errores, el comando a mandar –con tamaño variable– y, por último, su código redundante. La figura 5.2 muestra en detalle la información contenida en la trama.



Figura 5.2: Contenido de la trama del aire acondicionado.

Los bytes que forman las tramas fueron decodificados por medio de la inspección de la trama enviada por el mando original, usando el receptor y presentando dicha trama en el osciloscopio, realizando ingeniería inversa. La figura 5.3 muestra en detalle este proceso con una trama. A partir del análisis de varias tramas en las que se modificaba algún parámetro (conocido), procedimos a identificar la posición de los mismos dentro de la trama, observando manualmente qué bits eran los que cambiaban su estado. Para este propósito requerimos el uso de una hoja de cálculo.

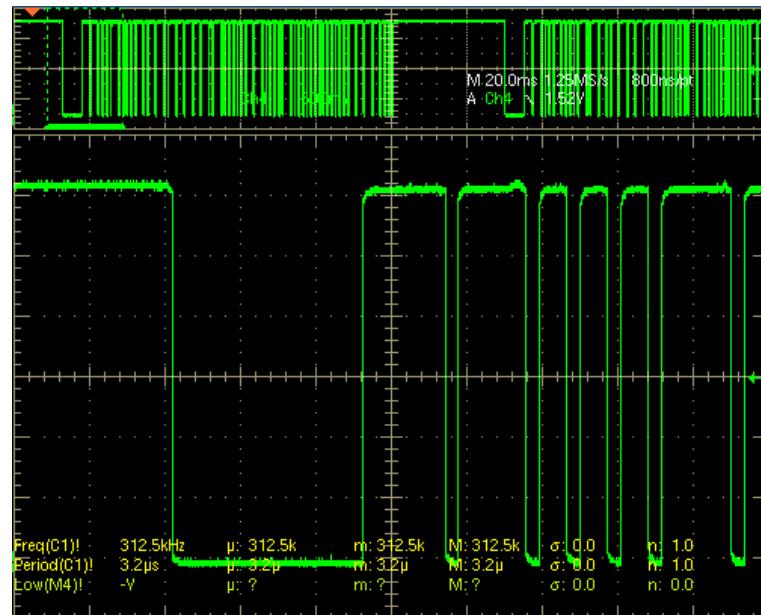


Figura 5.3: Detalle del comienzo de trama infrarroja.

La función `void protocoloAA ()` implementa el protocolo del aire acondicionado. Asimismo, en la función `sendIR`, expuesta en el apartado 4.1.3.1, los argumentos serían `AA` para el dispositivo, `modo` (que indica si se quiere entrar en modo verano, `AA_Summer`, o invierno, `AA_Winter`) para el primer parámetro, y `estado` (que enciende el aparato, `AA_On`, o lo apaga, `AA_Off`) para el segundo.

Las pruebas realizadas fueron el envío de los comandos “Enciende Aire Acondicionado, modo Verano” y “Apaga Aire Acondicionado, modo Verano”. Estas tramas fueron

reconocidas correctamente por el receptor de infrarrojos descrito en el apartado 5.1.1, y dieron lugar al encendido (y posterior apagado) de la máquina de aire acondicionado presente en el laboratorio.

Por último, las pruebas realizadas al **Buzzer** y **LEDs** fueron el mandar el encendido y el apagado de los mismos, con las funciones ya descritas. En el caso del Buzzer, se obtuvieron dos pitidos que se alternaban cada medio segundo; y, para los LEDs, comprobamos que `ledOn` mantiene encendido el LED, `ledOff` lo apaga incondicionalmente, y `ledToggle` lo commuta, apagándolo o encendiéndolo según el caso.

5.3. Herramientas

Para la realización de estas pruebas ha sido necesario usar una serie de equipos que permitan obtener las medidas, como ha sido visto a lo largo del capítulo, por lo que se ahondará en ellas en los siguientes apartados.

Además, ha sido necesario usar las herramientas expuestas en la sección 4.3, dado que hemos hecho igualmente pruebas al *software*.

5.3.1. Osciloscopio

El osciloscopio nos permite ver la forma de onda de las señales temporales que se introducen en él. En nuestro laboratorio encontramos el **Tektronix TDS5054B**, el cual nos permite, además, las siguientes funciones, útiles en nuestro trabajo.

- *Cuatro canales de entrada*, con lo que se pueden introducir cuatro señales simultáneas.
- *Alto periodo de muestreo*, de hasta 5 millones de muestras por segundo.
- *Medidas de parámetros de señal*, como pueden ser la frecuencia en señales periódicas, la amplitud, entre otras.
- *Disparo*, que permite que las señales periódicas permanecen congeladas en la pantalla. De especial ayuda resultó el modo *single*, el cual únicamente esperaba a recibir una trama, reteniéndola en la pantalla y permitiendo su evaluación.
- *Cursores*, que ofrecen la posibilidad de realizar medidas de distancia entre dos puntos, útil para calcular los tiempos de nivel alto y bajo en las señales moduladas.
- *Zoom*, que amplía el sector que se desea evaluar para tener una mejor visión del mismo, usado en tramas largas como las del Ir.
- *Capturas de pantalla*, de forma que los resultados son exportables y presentables en otros documentos.

5.3.2. Multímetro

El multímetro es un dispositivo que permite medir las principales magnitudes eléctricas, como son la tensión, la corriente, la resistencia o la capacidad. Especialmente de utilidad han sido las dos últimas, como ha sido detallado.

Capítulo 6

Conclusiones y líneas futuras

*Busca un trabajo que te guste,
y no tendrás que trabajar un sólo día de tu vida.*

Confucio

RESUMEN: En este último capítulo se ofrece una revisión de todo el proyecto, detallando los hitos logrados y las metas alcanzadas. Asimismo, se presentan los estudios futuros surgidos de la implementación de este trabajo.

6.1. Conclusiones

La principal meta de este trabajo era diseñar e implementar una placa con sensores y actuadores que se incluya dentro de la plataforma cNGD y que ayudara al estudio de las CWSN, además de realizarlas bajo unas constantes de bajo coste y bajo consumo.

Adicionalmente, era necesario desarrollar *software* que abstrajera el *hardware* para su uso desde la aplicación. igualmente buscando la eficiencia y la simplicidad a la hora de implementar el código, y modularidad con el objeto de que la misma fuera versátil y ampliable en un futuro.

Los sensores y actuadores que se han buscado se caracterizan por su polivalencia, en lo que se ha pensado una placa de propósito general, su variedad, puesto que permiten medir y actuar sobre diferentes aspectos del entorno, y su tamaño, de forma que fueran integrables en un reducido espacio.

En referencia a lo que, en un principio, se expuso en los objetivos, mostramos los logros obtenidos a lo largo del trabajo:

- *Documentación.* Se consideraron los trabajos previos realizados en esta materia, centrándonos especialmente en el desarrollo de la plataforma que sirve como base de este trabajo. Esto puede encontrarse en los capítulos 1 y 2.
Asimismo, se han usado las herramientas que han sido precisas para la elaboración del mismo, tanto *hardware* como *software*, referidas a lo largo del trabajo.
- *Diseño de la shield.* Como ha sido expuesto, se han buscado sensores y actuadores bajo las premisas de bajo coste, consumo y con el objeto de desarrollar una placa de propósito general. Este trabajo se detalla en el capítulo 2.

Adicionalmente, se realizó el diseño de la placa de expansión y se añadieron tanto los sensores y actuadores como los componentes adicionales necesarios, lo cual se encuentra en el capítulo 3.

- *Implementación y prueba de la placa de expansión.* Se llevó a cabo la implementación del diseño en PCB, trabajo detallado en el capítulo 3. Además, los componentes fueron soldados sobre la misma y fueron probados, tanto de forma individual como de forma conjunta. Las pruebas se encuentran en el capítulo 5.
- *Desarrollo del software.* La HAL desarrollada permite consultar el estado, realizar medidas y modificar parámetros de configuración de todos los sensores y actuadores que se encuentran sobre la placa. El capítulo 4 muestra este trabajo. Asimismo, se han realizado pruebas tanto en vacío como con los sensores y actuadores conectados, verificando en ambos casos el funcionamiento correcto de los mismos, y que pueden encontrarse en el capítulo 5.
- *Aplicación.* Se diseñó una aplicación de prueba que mostrara la integración de los trabajos realizados. Más detalles se encuentran en el capítulo 4.

6.2. Líneas futuras

En base al desarrollo llevado a lo largo del trabajo, observamos ciertas vías de investigación futuras, que presentamos con objeto de mejorar el sistema desde el punto de vista tanto del aumento de funciones como del control y el conocimiento del sistema. Las exponemos a continuación.

- *Realización de pruebas exhaustivas a la placa de expansión.* Estas pruebas conllevarían la caracterización real del comportamiento de la placa, mediante la medición del consumo, la estabilidad, la precisión de las medidas, entre otras.
- *Desarrollo de una placa de expansión con sensores y/o actuadores ad-hoc.* En la misma habría dispositivos que sirvieran a un propósito más específico. Sensores como pueden ser de humedad, gas, GPS (*Global Positioning System*), RFID (*Radio Frequency Identification*), etc.; y actuadores como relés, controladores para puertas automáticas, GSM (*Global System for Mobile*), entre otros.
- *Implementación de software para emisor infrarrojo,* que trate de automatizar el proceso de aprendizaje de nuevos comandos para nuevos electrodomésticos, integrando tantos protocolos como sea posible.
- *Pasarela de presentación de datos en tiempo real.* Trata de un sistema que presentara, bajo una interfaz web, los datos de todos los sensores y actuadores de la red, y poder realizar acciones sobre los mismos.
- *Despliegue de una red de sensores.* Al hilo de lo que se habló en la Introducción, el despliegue de una de estas redes es prioritario, puesto que permite obtener datos y mediciones reales, de forma que puedan ser usadas en investigación.
- *Cajeado del nodo con la placa de expansión.* Se realizaría un diseño que fuera compatible con la captación de medidas y la transmisión de información entre nodos y, a su vez, cumpliera una misión de protección del mismo.

Referencias

- [1] Office of Spectrum Management, *United States Frequency Allocations. The radio spectrum.* U.S. Government Printing Office, Aug 2011.
- [2] Mitola, J. III; Maguire, G.Q. Jr, “Cognitive radio: making software radios more personal,” *Personal Communications*, vol. 6, pp. 13 – 18, Aug 1999.
- [3] Mitola, J. III, *Cognitive Radio An Integrated Agent Architecture for Software Defined Radio, PhD. Dissertation.* Royal Institute of Technology, Kista, Sweden, May 2000.
- [4] Handziski, V.; Köpke, A.; Willig, A.; Wolisz, A., “Twist: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks,” *REALMAN '06 Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pp. 63 – 70, 2006.
- [5] “LOG-a-TEC by SensorLab.” <http://log-a-tec.eu/>. Último acceso el 16-5-2014.
- [6] Newman, T.R.; He, A.; Gaeddert, J.; Hilburn, B.; Bose, T.; Reed, J.H., “Virginia tech cognitive radio network testbed and open source cognitive radio framework,” *Testbeds and Research Infrastructures for the Development of Networks & Communities and Workshops, 2009. TridentCom 2009. 5th International Conference on*, pp. 1 – 3, Apr 2009.
- [7] “FIT/CorteXlab: Cognitive radio testbed.” <http://www.cortexlab.fr/>. Último acceso el 16-5-2014.
- [8] López, F.; Romero, E.; Blesa, J.; Villanueva, D.; Araujo, A., “Cognitive Wireless Sensor Network Device for AAL Scenarios,” *Lecture Notes in Computer Science*, vol. 6693, pp. 116 – 121, 2011.
- [9] Tena, A., *Development of a Multiple RF interfaced platform for Cognitive Wireless Sensor Networks. PFC.* Laboratorio de Sistemas Integrados, ETSIT-UPM, Nov 2013.
- [10] Romero, E.; Blesa, J.; Tena, A.; Jara, G.; Domingo, J.; Araujo, A., “Cognitive Test-bed for Wireless Sensor Networks,” *Dynamic Spectrum Access Networks (DySPAN), 2014 IEEE International Symposium on*, pp. 346 – 349, Apr 2014.
- [11] Jara, G., *Diseño e implementación de una arquitectura para la gestión de comunicaciones de una red de sensores inalámbricas cognitiva. PFC.* Laboratorio de Sistemas Integrados, ETSIT-UPM, Sep 2013.

- [12] Domingo, J., *Diseño, optimización y prueba de un nodo para una red de sensores inalámbrica con capacidades cognitivas.* PFC. Laboratorio de Sistemas Integrados, ETSIT-UPM, Feb 2013.
- [13] “Texit.” <http://gaia.fdi.ucm.es/research/texit/>. Último acceso el 21-4-2014.
- [14] “Latex.” <http://en.wikibooks.org/wiki/TeX/>. Último acceso el 10-5-2014.
- [15] Microchip Technology Inc., *PIC32MX5XX/6XX/7XX Family Data Sheet*. 2009.
- [16] Microchip Technology Inc., *AN1066 Microchip MiWiTM Wireless Networking Protocol Stack*. 2007.
- [17] Microchip Technology Inc., *MRF49XA Data Sheet, ISM Band Sub-GHz RF Transceiver*. 2009.
- [18] Microchip Technology Inc., *MRF24J40MA Data Sheet 2.4 GHz, IEEE Std. 802.15.4TM RF Transceiver Module*. 2008.
- [19] “Farnell España, Componentes Electrónicos.” <http://es.farnell.com/>. Último acceso el 3-5-2014.
- [20] “RS Components.” <http://es.rs-online.com/web/>. Último acceso el 3-5-2014.
- [21] “Digi-Key Corp. Electronic Components Distributor.” <http://www.digikey.es/>. Último acceso el 3-5-2014.
- [22] “SparkFun Electronics.” <https://www.sparkfun.com/>. Último acceso el 13-5-2014.
- [23] Blanco, J.M.; Serrano, J.J.; Sanz, A., *Instrumentación Electrónica*. Fundación Rogelio Segovia para el Desarrollo de las Telecomunicaciones, ETSIT-UPM, Ene 2006.
- [24] “Analog Devices. Low Voltage Temperature Sensor TMP36. Data Sheet.” http://www.analog.com/static/imported-files/data_sheets/TMP35_36_37.pdf. Último acceso el 3-5-2014.
- [25] “Microchip. Low-Power Lineal Active ThermistorTM MCP9701A. Data Sheet.” <http://ww1.microchip.com/downloads/en/DeviceDoc/21942e.pdf>. Último acceso el 3-5-2014.
- [26] “Texas Instruments. LM20 Temperature Sensor. Data Sheet.” <http://www.ti.com.cn/lit/ds/symlink/lm20.pdf>. Último acceso el 4-5-2014.
- [27] “Texas Instruments. Digital Temperature Sensor with I²CTM Interface TMP101. Data Sheet.” <http://www.ti.com/lit/ds/symlink/tmp101.pdf>. Último acceso el 4-5-2014.
- [28] “Analog Devices. 10-Bit Digital Temperature Sensor AD7814. Data Sheet.” http://www.analog.com/static/imported-files/data_sheets/AD7814.pdf. Último acceso el 4-5-2014.

- [29] "Microchip. 2-Wire High-Accuracy Temperature Sensor MCP9800. Data Sheet." <http://ww1.microchip.com/downloads/en/DeviceDoc/21909d.pdf>. Último acceso el 4-5-2014.
- [30] "Xtrinsic. Digital Accelerometer MMA8453Q. Data Sheet." http://cache.freescale.com/files/sensors/doc/data_sheet/MMA8453Q.pdf. Último acceso el 13-5-2014.
- [31] "Analog Devices. Digital Accelerometer ADXL345. Data Sheet." http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf. Último acceso el 13-5-2014.
- [32] "ST. LIS3DH MEMS Ultra Low-Power Nano Accelerometer. Data Sheet." <http://www.st.com/web/en/resource/technical/document/datasheet/CD00274221.pdf>. Último acceso el 13-5-2014.
- [33] "Murata. Pyroelectric Infrared Sensors. Data Sheet." <http://www.murata.com/products/catalog/pdf/s21e.pdf>. Último acceso el 13-5-2014.
- [34] "Panasonic. High-sensitive human detection sensor. Data Sheet." <http://pewa.panasonic.com/assets/pcsd/catalog/papirs-ekmb-catalog.pdf>. Último acceso el 13-5-2014.
- [35] "Panasonic. Passive infrared human detection sensor. Data Sheet." <http://pewa.panasonic.com/assets/pcsd/catalog/napion-catalog.pdf>. Último acceso el 13-5-2014.
- [36] "Everlight. Ambient Light Sensor ALS-PT19. Data Sheet." http://www.everlight.com/datasheets/ALS-PT19-315C-L177-TR_datasheet.pdf. Último acceso el 13-5-2014.
- [37] "Avago. APDS-9005 Miniature Surface-Mount Ambient Light Photo Sensor. Data Sheet." <http://www.avagotech.com/docs/AV02-0080EN>. Último acceso el 13-5-2014.
- [38] "Osram. High Accuracy Ambient Light Sensor SFH 5711. Data Sheet." http://www.osram-os.com/Graphics/XPic6/00039059_0.pdf/Appnote%20for%20Ambient%20light%20sensor%20SFH%205711.pdf. Último acceso el 13-5-2014.
- [39] "Vishay. Infrared Emmiting Diode. Data Sheet." <http://www.vishay.com/docs/81055/tsus520.pdf>. Último acceso el 15-6-2014.
- [40] "TDK. Piezoelectronic Buzzers PS Series. Data Sheet." http://product.tdk.com/en/catalog/datasheets/ef532_ps.pdf. Último acceso el 15-6-2014.
- [41] "JBC. Rework Station 6000. Instructions manual." http://www.jbctools.com/pdf/manual_AM6000_230v.pdf. Último acceso el 17-6-2014.
- [42] Kernighan, B.W; Ritchie, D.M., *The C programming language*. Prentice Hall PTR, 1988.
- [43] Microchip Technology Inc., *MPLAB X Integrated Development Environment User's Guide*. 2011.

- [44] Microchip Technology Inc., *MPLAB ICD 3 In-Circuit Debugger User's Guide For MPLAB X IDE*. 2010.
- [45] “Vishay. IR Receiver Modules for Remote Control Systems. Data Sheet.” <http://www.farnell.com/datasheets/1699774.pdf>. Último acceso el 22-6-2014.
- [46] Blesa, J., *Análisis y diseño de aplicaciones para la gestión de edificios singulares basadas en el protocolo de redes inalámbricas de área personal ZigBeeTM. PFC*. Laboratorio de Sistemas Integrados, ETSIT-UPM, Sep 2008.

Lista de Acrónimos

ADC	<i>Analog-to-Digital Converter</i>
AMI	<i>Ambient Intelligence</i>
CN	<i>Change Notification</i>
CNGD	<i>Cognitive Next Generation Device</i>
CR	<i>Cognitive Radio</i>
CWSN	<i>Cognitive Wireless Sensor Network</i>
DIE	<i>Departamento de Ingeniería Electrónica</i>
EDA	<i>Electronic Design Automation</i>
ETSIT	<i>Escuela Técnica Superior de Ingenieros en Telecomunicación</i>
FCD	<i>First Cognitive Device</i>
GPIO	<i>General Purpose Input-Output</i>
GPS	<i>Global Positioning System</i>
GSM	<i>Global System for Mobile</i>
HAL	<i>Hardware Abstraction Layer</i>
I ² C	<i>Inter-Integrated Circuit</i>
ICD	<i>In-Circuit Debugger</i>
IDE	<i>Integrated Development Environment</i>
IR	<i>Infrarrojo</i>
ISM	<i>Industrial, Scientific and Medical</i>
LDR	<i>Light Dependent Resistor</i>
LED	<i>Light-Emitting Diode</i>
LSI	<i>Laboratorio de Sistemas Integrados</i>
MCU	<i>Micro-controller Unit</i>
MEMS	<i>Micro-Electro-Mechanical System</i>

NTC	<i>Negative Temperature Coefficient</i>
OTAP	<i>Over The Air Programming</i>
OTG	<i>On-The-Go</i>
P2P	<i>Peer-to-Peer</i>
PCB	<i>Printed Circuit Board</i>
PIR	<i>Passive Infrared</i>
PTC	<i>Positive Temperature Coefficient</i>
RAM	<i>Random Access Memory</i>
RFID	<i>Radio Frequency Identification</i>
RI	<i>Radio Interface</i>
SMD	<i>Surface-Mount Devices</i>
SPI	<i>Serial Peripheral Interface</i>
TH	<i>Through Hole</i>
UART	<i>Universal Asynchronous Receiver-Transmitter</i>
UPM	<i>Universidad Politécnica de Madrid</i>
USB	<i>Universal Serial Bus</i>
WSN	<i>Wireless Sensor Network</i>