



Trabajo Fin de Máster
Máster en Ingeniería Electrónica

Sistema de interacción basado en RFID para salas de estimulación multisensorial

Centro Politécnico Superior
UNIVERSIDAD DE ZARAGOZA

Diciembre 2010

Autor: José Miguel Idiago Valero
Director: Roberto José Casas Nebra
Co-Director: Álvaro Marco Marco

Departamento:
Ingeniería electrónica y comunicaciones.
Área de tecnología electrónica.

Agradecimientos

A todos mis compañeros de Tecnodiscap que desinteresadamente me han ayudado a realizar este proyecto.

Resumen

Este Proyecto de Fin de Carrera se ha realizado en Tecnodiscap, grupo de investigación del I3A, dentro de la línea de investigación en salas de estimulación multisensorial.

Las salas de estimulación multisensorial son aulas situadas en colegios de educación especial con materiales diseñados para que los niños estén expuestos a estímulos controlados, con el objetivo final de favorecer su nivel de integración sensorial facilitando así los aprendizajes básicos. Se ha detectado en las instalaciones actuales una falta de posibilidades de interacción del usuario con el entorno, lo que supone una limitación en el desarrollo de las sesiones terapéuticas.

En este proyecto se enmarca dentro del desarrollo de un sistema que permita la interacción del usuario con los diferentes elementos que pueden estar presentes en una sala de estimulación multisensorial. En concreto se ha diseñado un dispositivo basado en RFID y Zigbee que permite la interacción del usuario con otros elementos presentes en el aula. Se pretende que el usuario pueda llevar encima el dispositivo para que cuando se acerque a determinados objetos el sistema lo reconozca y sea capaz de producir un estímulo en el usuario mediante la activación de otros elementos presentes en la sala como luces, sonidos o proyecciones...

Además del diseño hardware se ha integrado el dispositivo en un driver zigbee, que permite la auto-detección del dispositivo. Este driver funciona bajo un framework (OSGi) que permite el manejo e interconexión de dispositivos y servicios, de esta manera el objeto software creado por el driver puede interactuar con otros dispositivos o servicios.

Por otro lado se ha contribuido al diseño de la arquitectura software/middleware del sistema completo, tratando de que quede abierto a la incorporación de más elementos en futuros desarrollos en función de las necesidades expresadas por los usuarios. También se ha realizado una colaboración con alumnos de diseño industrial en el diseño del interfaz de usuario, además se ha dirigido un proyecto fin de carrera en el que siguiendo la línea de investigación en aulas de estimulación multisensorial se ha desarrollado un controlador de luces para poder controlar la iluminación del aula.

El objetivo principal ha sido añadir un elemento nuevo a las salas de estimulación multisensorial e integrarlo junto con otros dispositivos que se están desarrollando en paralelo, pero a la vez ha obtenido una visión clara del proceso de diseño de un dispositivo, desde la electrónica básica, pasando por el firmware, el funcionamiento de un driver y el control software del dispositivo.

Índice general

1. Introducción	3
1.1. Salas de Estimulación Multisensorial	3
1.2. Objetivos del Proyecto	4
1.3. Estado del Arte	5
1.4. Diagrama Temporal del Trabajo	6
2. Desarrollo Hardware	9
2.1. Descripción del Hardware	9
3. Firmware y Middleware	15
3.1. Firmware	15
3.1.1. Comunicación con el Lector de Skyetek	16
3.1.2. Comunicación Zigbee	17
3.1.3. Perfil ZigBee-Tecnodiscap	18
3.1.4. Estructura de Mensajes	18
3.2. Middleware	20
3.2.1. OSGi (Open Service Gateway Initiative)	20
3.2.2. OSGi4AmI	21
3.2.3. Relación entre OSGi4AmI y Perfil ZigBee-Tecnodiscap	24
3.2.4. Integración en Driver Zigbee de Tecnodiscap	25
5. Otros Trabajos Realizados	33
5.1. Colaboración con Alumnos y Profesores de Diseño Industrial	33

5.2. Dirección del Proyecto Fin De Carrera	35
6. Conclusiones y trabajos futuros	37
6.1. Conclusiones	37
6.2. Futuros Trabajos	38
ANEXOS	43
A. Salas de Estimulación Multisensorial	43
A.1. Estimulación Multisensorial	43
A.2. Aula Multisensorial	44
A.3. Áreas de estimulación multisensorial	44
A.4. Materiales disponibles	46
B. Tecnología RFID	51
B.1. ¿Qué es la Tecnología RFID?	51
B.2. ¿Cómo Funciona?	52
B.3. Beneficios de la tecnología RFID	57
C. Protocolo Zigbee	59
C.1. Estándar IEEE 802.15.4	59
C.2. Tipos de dispositivos	60
C.3. Topologías y funcionamiento de la red	61
C.4. Comunicaciones	62
C.5. Mensajes de datos	64
C.6. Seguridad	64
C.7. Perfiles	64
C.8. Campos de aplicación	65
D. Esquemático y PCB	69

Lista de Figuras

1.1. Diagrama temporal de las tareas realizadas	7
2.1. Diagrama de la Tag v2	10
2.2. Esquema dsPic	10
2.3. Módulo ZigBee	11
2.4. Diagrama de Conexiones	13
2.5. Lector Skyetek M1-mini	13
2.6. Dispositivo para Aulas de Estimulación Multisensorial	14
3.1. Comunicación con el lector RFID	16
3.2. Petición: Microcontrolador →Lector	16
3.3. Respuest: Lector →Microcontrolador	17
3.4. Estructura de un paquete tipo Comando	19
3.5. Estructura de un paquete tipo Evento	20
3.6. Jerarquización de los dispositivos	23
3.7. Ejemplo de Arquitectura con OSGi y OSGi4AmI	23
3.8. Endpoints Implementados	25
3.9. Clusters Implementados	25
3.10. Algunos Comandos y Eventos Implementados	26
3.11. Interfaz OSGi4AmI Device	27
3.12. Interfaz OSGi4AmI Sensor	27
A.1. Columna de Burbujas	47

A.2. Tormenta de Colores	47
A.3. Sala Equipada con Proyector y Fibras Ópticas	48
A.4. Elementos Pasivos en una Sala de Estimulación Multisensorial	49
B.1. Etiqueta RFID	52
B.2. Esquema general de funcionamiento de la tecnología RFID	54
B.3. Etiqueta Pasiva	55
B.4. Otro Ejemplo de Etiqueta Pasiva	55
B.5. Etiqueta Activa	56
C.1. Diagrama de la estructura funcional de Zigbee sobre IEEE 802.15.4	60
C.2. - Esquema de las distintas topologías posibles con ZigBee	62
C.3. Reparto de direcciones en una red de cinco hijos por nodo.	63
C.4. Perfiles ZigBee	65
C.5. Perfiles ZigBee	66
C.6. Comparativa de diferentes estándares inalámbricos en función de su ancho de banda	66
D.1. Esquemático del dispositivo	70
D.2. PCB del dispositivo	70

Lista de Tablas

B.1. Bandas de frecuencia utilizadas en la tecnología RFID	56
C.1. Comparativa entre las distintas tecnologías inalámbricas	67

Introducción

Este proyecto se ha realizado dentro del grupo de investigación Tecnodiscap de la Universidad de Zaragoza. El objetivo del grupo Tecnodiscap es la mejora de la calidad de vida de las personas con algún tipo de discapacidad, dependencia o necesidades especiales mediante servicios basados en la aplicación de las Tecnologías de la Información y Comunicaciones (TIC). Una nueva línea de investigación del grupo corresponde a la mejora de las salas de estimulación multisensorial presentes en muchos de los colegios de educación especial con los que se colabora.

Como explicaremos más adelante se han detectado un serie de necesidades por parte de los educadores de dichos centros que podrían ser cubiertas con el uso de tecnología. Este proyecto pretende dar solución a algunas de esas necesidades y junto con otros desarrollos que se están llevando a cabo en el grupo, ofrecer nuevas posibilidades a los profesores de educación especial gracias a la ayuda de las TIC.

A lo largo de este capítulo se introducir el concepto de Estimulación Multisensorial para después plantear los objetivos del proyecto, por último se hará un breve resumen sobre el estado del arte en cuanto a dispositivos interactivos con tecnología RFID.

1.1. Salas de Estimulación Multisensorial

La estimulación multisensorial es un instrumento utilizado con el objetivo de mejorar las condiciones de vida de las personas con discapacidad. Para ello se recurre a medios y estrategias que trabajan las capacidades más básicas del ser humano: las sensaciones, la percepción y la integración sensorial.

Las salas «snoezelen», o multisensoriales, están repletas de luces, olores, sonidos, colores... Surgen de una iniciativa creada en Holanda para estimular a personas con discapacidades psíquicas. Con estos tratamientos no se pretende curar a estos discapacitados, pero sí hacer que disfruten, mejoren sus capacidades cognitivas y de relación, así como lograr que se encuentren mejor.

Sobre todo en los niños pequeños, se busca reforzar su desarrollo favoreciendo la integración de la información sensorial que reciben, ayudando en sus aprendizajes y su relación con el entorno.

Se trabajan los sentidos en un ambiente de estímulos controlados, donde se facilitan la exploración, el descubrimiento y el disfrute de diferentes experiencias sensoriales. Llegando a experimentar sensaciones intensas con la posibilidad de expresar emociones contenidas. Se busca un despertar sensorial a través de la propia experiencia sensorial.

Puede encontrarse información ampliada sobre salas de estimulación multisensorial en los anexos.

1.2. Objetivos del Proyecto

Los profesores de educación especial han expresado al grupo Tecnodiscap la carencia de interactividad en los elementos que se encuentran actualmente disponibles para aulas de estimulación multisensorial. Desde el grupo Tecnodiscap se ha detectado que esta carencia de interactividad entre elementos puede ser enmendada mediante el uso de tecnologías como ZigBee, RFID, sensores inerciales ... Es por eso que una de las nuevas líneas de investigación del grupo Tecnodiscap se centra en el desarrollo de un sistema interactivo que mediante el desarrollo de dispositivos y software de control mejore la experiencia de los usuarios de aulas de estimulación multisensorial.

Al hilo de esta línea de investigación se plantea este trabajo fin de máster, en él se pretende sentar las bases de lo que sería el sistema de interacción entre elementos de aulas de estimulación multisensorial. Para ello se va a desarrollar un dispositivo de interacción física y motriz que permita reconocer la interacción del usuario con elementos de la sala mediante la incorporación de tecnología RFID, a la vez que se monitoriza la actividad del usuario mediante sensores inerciales. El dispositivo permitirá interactuar con otros elementos de la sala gracias a que se integrará en un sistema middleware basado en OSGi.

Además del diseño hardware y del firmware de control de los distintos sensores, se ha implementado perfil ZigBee privado que permite las comunicaciones entre el nodo y el PC (Coordina-

dor) permitiendo la integración del dispositivo en un driver zigbee desarrollado por Tecnodiscap. El perfil privado de comunicaciones ha sido desarrollado por Tecnodiscap y se explicará en los siguientes capítulos. Por otra parte la integración del dispositivo en el driver zigbee permitirá el uso del dispositivo en un entorno OSGi. Como se verá más adelante OSGi es un framework para programación en Java que se ajusta muy bien a la arquitectura software necesaria para el desarrollo del sistema de interacción para aulas de estimulación multisensorial.

Otro objetivo fundamental del proyecto es comprender el proceso de creación de un nuevo dispositivo desde la electrónica básica, pasando por el firmware, hasta su integración en un sistema software. También es objetivo de este proyecto la asimilación de conceptos y metodologías de trabajo que permitan el diseño e implementación de sistemas modulares y ampliables.

Otros objetivos del proyecto son la profundización en las tecnologías ZigBee y RFID, así como conocer las familia de dispositivos dsPIC de Microchip.

1.3. Estado del Arte

En esta sección se va a hacer un repaso a proyectos y publicaciones relacionadas con tecnologías inalámbricas y RFID destinadas a fines educativos o lúdicos.

Una de las primeras referencias encontradas es [14] en la que se un lector RFID integrado en un guante como nosotros, pero en este artículo no se implementaba ninguna solución inalámbrica, lo que obliga al usuario a estar conectado por un cable al ordenador. El desarrollo más similar al nuestro es el encontrado en [13] en el que se propone integrar el lector RFID en un guante o un brazalete pero la comunicación se realiza por bluetooth en lugar de ZigBee como proponemos nosotros, bluetooth permite una mayor ancho de banda en la comunicación pero el coste a pagar es un mayor consumo, además bluetooth sólo permite conexiones tipo estrella entre los dispositivos, por otra parte ZigBee permite una red tipo mesh(malla), este tipo de red se ajusta mucho mejor a un sistema de dispositivos que interactúan unos con otros. En este último artículo aparte del desarrollo del propio dispositivo se proponen una serie de juegos por ejemplo el descrito en [12]. Este juego se basa en la lectura de tags RFID mediante lectores incorporados en la ropa de los niños. Este juego escapa al alcance de un aula de estimulación multisensorial por la necesidad de que los participantes tengan unas capacidades cognitivas prácticamente plenas, pero es un buen ejemplo de integración de la tecnología RFID en la educación y puede servir de inspiración a los profesores de educación especial para sacar todo el potencia posible de la tecnología RFID.

Al hilo de aplicaciones de RFID para juegos encontramos propuestas como la de [5] en la que se propone integra además de el lector RFID un acelerómetro para incrementar las posibilidades de juego. Como se verá más adelante nosotros también hemos integrado un acelerómetro, pero no se ha implementado una aplicación de reconocimiento de gestos tan específica como la mostrada en ese artículo. Otra aplicación de RFID para juegos la encontramos en [6], a destacar que aparte de usar un lector RFID y una comunicación Bluetooth se han basado en la plataforma Arduino y en LEGO Mindstrom.

Apartándonos un poco de nuestra aplicación encontramos otros ejemplos de interacción con RFID en [11], donde a partir de un dispositivo simular al nuestro se ha implementado un sistema de reconocimiento de gestos basado en la lectura de tags RFID dispuestas sobre el pecho y el abdomen del usuario. Por otra parte encontramos en [9] y [10] dos ejemplos del uso de tecnología RFID integrada en guantes o brazaletes que permiten interactuar con objetos de la vida cotidiana. Por último, en [17] encontramos un sistema de guiado basado en un lector de tags integrado en el zapato, que a su vez está conectado por bluetooth a un dispositivo tipo PDA, este sistema funciona gracias a la lectura de tags que previamente se han colocado en el suelo.

De todos estos artículos se han obtenido ideas y mejoras que al final han conducido al dispositivo desarrollado. Destacar que la principal diferencia entre nuestro dispositivo y los expuestos en esta sección es el uso de tecnología ZigBee y la integración del dispositivo en una red de tipo mesh.

1.4. Diagrama Temporal del Trabajo

En la Figura [1.1] se muestra el diagrama de Gantt de las tareas llevadas a cabo durante la realización del trabajo fin de máster.

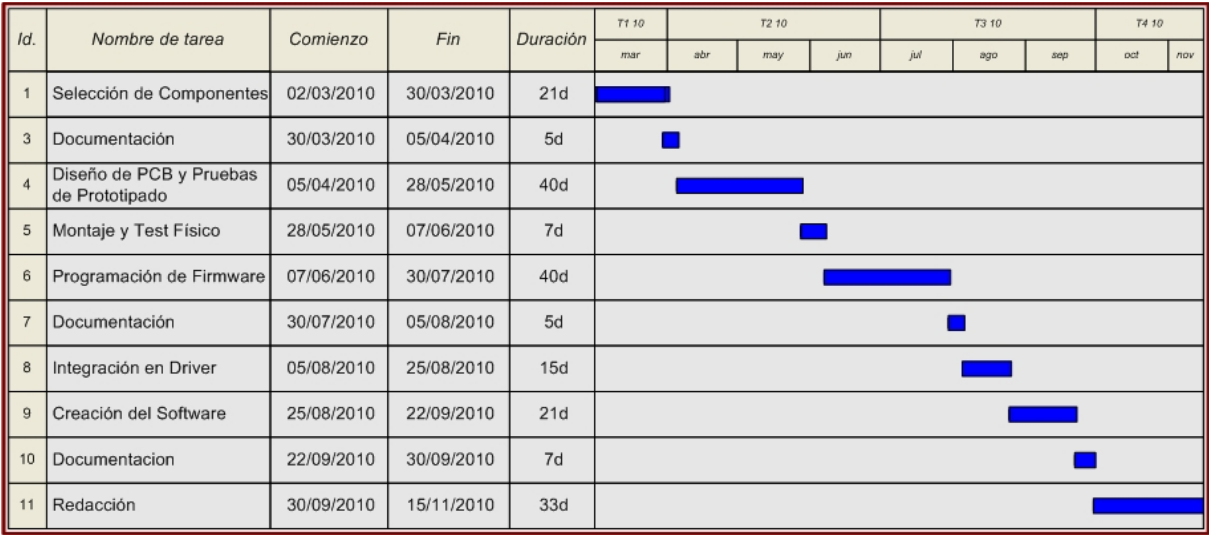


Figura 1.1: Diagrama temporal de las tareas realizadas

Capítulo 2

Desarrollo Hardware

En este capítulo vamos a explicar la parte hardware del dispositivo, las principales necesidades del dispositivo son la comunicación inalámbrica y el tamaño reducido. Estas características son fundamentales para que el dispositivo no resulte incomodo en su uso. Otras características que queremos para el dispositivo son: algún tipo de interfaz con el usuario y capacidad de monitorización básica de actividad o movimiento del usuario.

Para la comunicación inalámbrica se ha elegido el estándar ZigBee, este estándar se ajusta perfectamente a la aplicación ya que está pensado para maximizar la duración de las baterías de los dispositivos, además el grupo Tecnodiscap tiene experiencia en el desarrollo de dispositivos basados en este estándar. Por otra parte para reducir el tamaño al máximo se han utilizado componentes de montaje superficial y se ha prestado especial cuidado al realizar la PCB del dispositivo. Respecto al tema de sensorización se han integrado el lector RFID, un acelerómetro de tres ejes y un sensor de temperatura, con estos sensores se pretende ser capaz de monitorizar y detectar ciertas actividades del usuario de cara a enviar eventos al sistema global para permitir la interacción con otros elementos de la sala.

2.1. Descripción del Hardware

En la Figura [2.1] diagrama de bloques se muestran las distintas partes que integran el dispositivo.

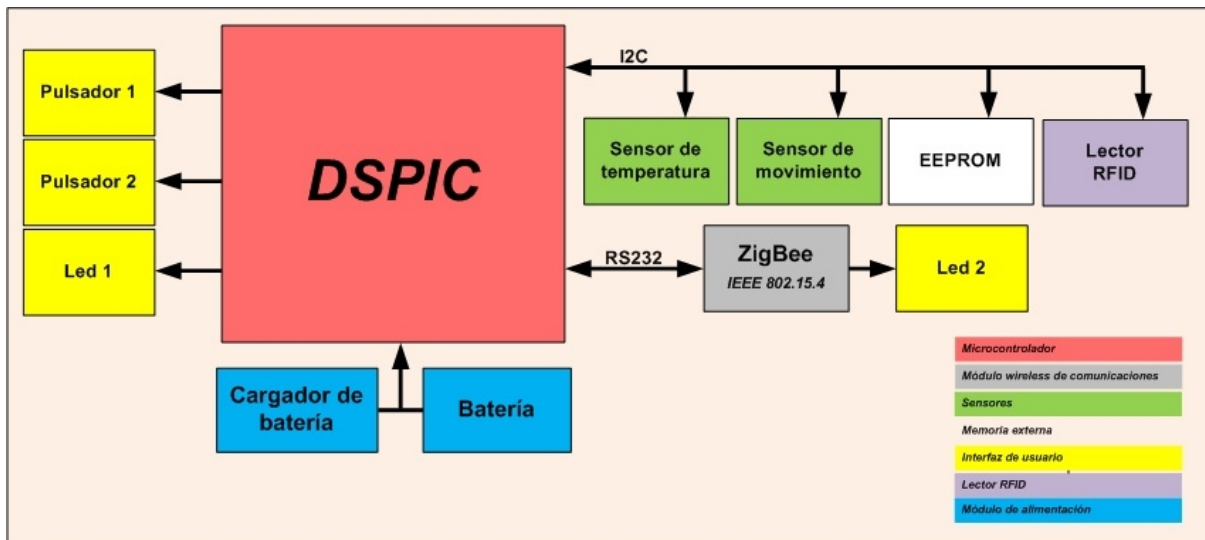


Figura 2.1: Diagrama de la Tag v2

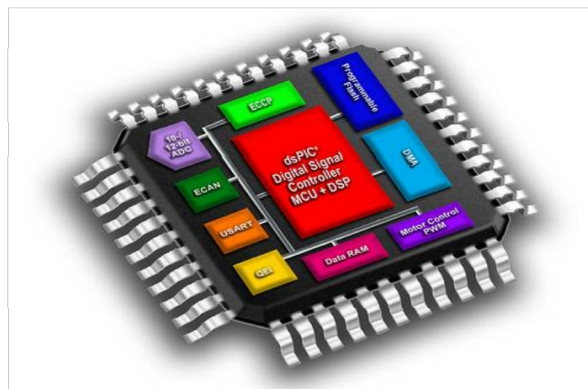


Figura 2.2: Esquema dsPic

Microcontrolador

Se utiliza un dsPIC33FJ32GP202 de Microchip (Figura [2.2]). Se trata de un microcontrolador de 16 bits que incluye una unidad de procesamiento de señales digitales (DSP). Integrar tanto el control de todos los dispositivos como el sistema de procesamiento y tiene una velocidad de procesamiento de señales óptima y permite la comunicación con el sensor de temperatura, el acelerómetro, la memoria y el lector RFID de forma sencilla gracias a su bus I2C.



Figura 2.3: Módulo ZigBee

Módulo wireless de comunicaciones

Para las comunicaciones ZigBee se ha optado por el módulo ETRX3 de Telegesis (Figura [2.3]). Este módulo permite reprogramar el procesador EM357 interno de manera personalizada o utilizar la API proporcionada por el fabricante. Esta API permite, mediante comandos AT, gestionar la red ZigBee de una forma sencilla.

Sensores integrados en la placa

Sensor digital de temperatura LM75B de National Semiconductor que suministra a través del bus I2C la temperatura ambiente codificada en 9 bits.

Acelerómetro LIS302DL de STMicroelectronics que suministra a través del bus I2C los valores de aceleración correspondiente a cada uno de los ejes X, Y y Z.

Lector RFID

De cara a mejorar la interacción con el lector se decidió incorporar a la placa principal un conector para tener el bus i2c accesible, a este conector se conectará un lector RFID de la empresa Skyetek, concretamente el modelo M1-mini. Se ha elegido este lector por su reducido tamaño, su comunicación i2c que se ajusta perfectamente a las características de nuestro microcontrolador y por la frecuencia de trabajo (13.56 MHz) que es muy apropiada en cuanto a distancia de lectura de nuestra aplicación. En las figuras [2.5] [2.6] podemos ver tanto el lector de forma individual como el dispositivo final ya ensamblado.

Destacar en la Figura [2.6] que se han unido la placa principal con el lector mediante cuatro cables trenzados (Vcc, GND, SDA y SCK) y el lector se ha encapsulado en una funda de plástico de la utilizada para coleccionismo de monedas. El diagrama de conexiones puede verse en la

Figura [2.4].

Memoria externa

Se utiliza una memoria M24M01-RMN6 de STMicroelectronics. Se trata de una EEPROM de 1Mbit que se comunica vía serie por I2C y se utilizará para almacenar los datos obtenidos tras el procesamiento de las muestras de aceleración generadas por el acelerómetro.

Interfaz de usuario

Se incluyen dos pulsadores y dos leds para interactuar con el usuario. Posibles usos serían informar del estado bajo de la batería mediante la iluminación o parpadeo de un led o el envío de un determinado mensaje si se aprieta el pulsador.

Módulo de alimentación

- **Carga de las baterías**

Las baterías de litio necesitan un sistema de carga especial que controle el voltaje e intensidad suministrada; ello se hace mediante el chip Microchip MCP73812T-420I/OT, el cual, además, protege las baterías contra autodescarga. Estos sistemas facilitan en gran medida la gestión de baterías de iones de litio que requieren unas condiciones de carga muy especiales para no deteriorarse. Las baterías elegidas van a ser las usadas en el iPod nano, ya que su precio es muy reducido debido a la cantidad que se fabrican, además de incorporar un circuito de protección y monitorización que la protegen de un daño prematuro.

- **Alimentación externa**

La alimentación externa se puede realizar mediante un conector microUSB. Se han instalado este sistema de carga en previsión de que en pocos años todos los móviles deberán tener un cargador común mediante microUSB, así este dispositivo se adelanta a ese evento y ya permite la conexión de este tipo de cargadores cuyo precio será muy reducido al ser un estándar y sobre todo se podrán encontrar en cualquier centro comercial.

- **Regulación de voltaje**

El voltaje suministrado por la batería de litio depende de su estado de carga, por lo que se necesita un regulador de voltaje capaz de asegurar que la alimentación que llega al resto

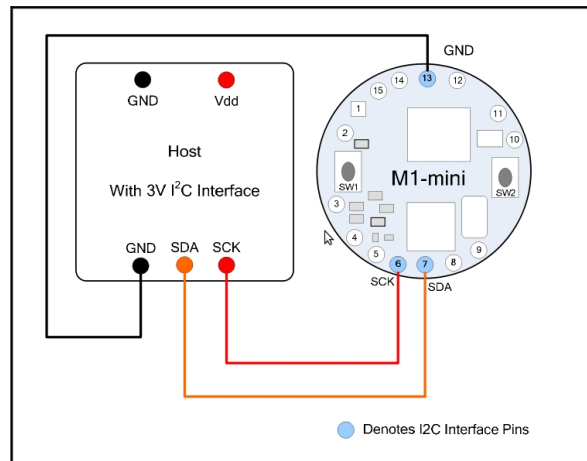


Figura 2.4: Diagrama de Conexiones

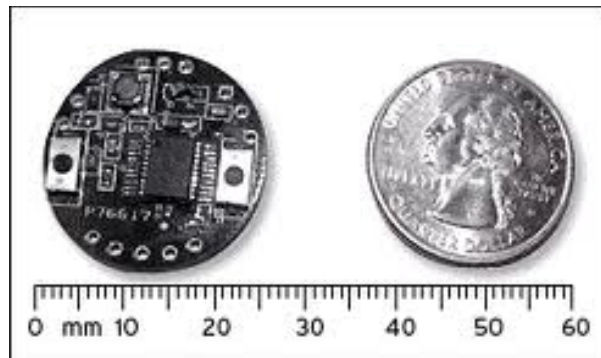


Figura 2.5: Lector Skyetek M1-mini

del circuito es siempre de 3.3V. El circuito integrado usado es el TPS79333DBVRG4. Al ser su tensión de dropout muy baja, el sistema se puede alimentar a una tensión de tan sólo 3.5V, lo que asegura un aprovechamiento máximo de las baterías. Además, dispone de una entrada de habilitación que conectada a otro circuito integrado, el MCP111T, desconecta la alimentación del circuito cuando la tensión de la batería no es la adecuada, con lo que puede provocar un mal funcionamiento del sistema.

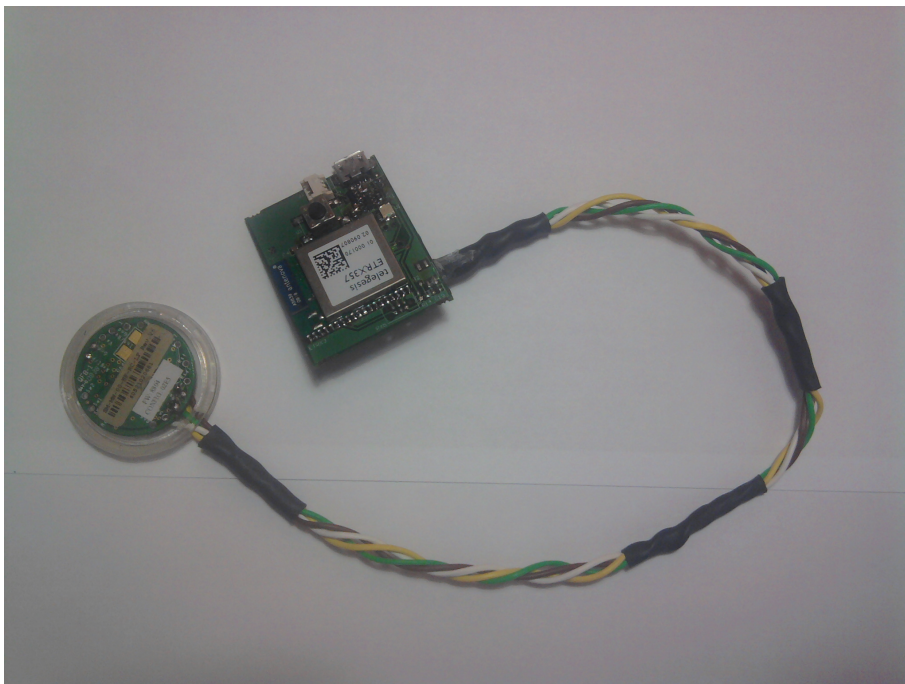


Figura 2.6: Dispositivo para Aulas de Estimulación Multisensorial

Capítulo 3

Firmware y Middleware

El objetivo de este capítulo es el de exponer las diferentes partes del middleware y firmware que se han usado e implementado en el proyecto. Para comenzar explicaremos brevemente como se realizan tanto las lecturas de las tags RFID y el envío de datos por zigbee, la interacción con los sensores de aceleración y temperatura así como la lectura y escritura en la memoria externa no se han descrito, ya que se trata de una comunicación i2c más simple que la realizada con el lector RFID. De ahí pasaremos a explicar el funcionamiento del perfil privado ZigBee desarrollado por Tecnodiscap, este perfil permite la comunicación entre los nodos de una red ZigBee . En la segunda parte del capítulo explicaremos la capa más baja del software también llamada middleware, ahí introduciremos herramientas como OSGi y OSGi4AmI imprescindibles para realizar una aplicación modular y reutilizable. Por último explicaremos la relación entre el perfil privado ZigBee desarrollado por Tecnodiscap y la ontología OSGi4AmI, en esta última parte expondremos la forma en la que se ha introducido nuestro dispositivo en el driver ZigBee desarrollado por el grupo Tecnodiscap.

3.1. Firmware

En este primer apartado primero haremos una breve introducción a la comunicación entre el lector y el microcontrolador, para luego pasar a ver unas pinceladas de los comandos AT con los que se gobierna el módulo ZigBee, por último introduciremos la comunicación llevada a cabo entre el dispositivo y el PC gracias a la implementación del perfil privado Zigbee utilizado en Tecnodiscap, que a partir de ahora nos referiremos a él como Perfil ZigBee-Tecnodiscap.



Figura 3.1: Comunicación con el lector RFID

MSG LEN	FLAGS	COMMAND	RID	TAG TYPE	TID request	AFI	STARTING BLOCK	NUMBER OF BLOCKS	DATA	CRC
8 bits	8 bits	8 bits	8 bits	8 bits	64 bits	8 bits	8 bits	8 bits	n*8 bits	16 bits

Figura 3.2: Petición: Microcontrolador → Lector

3.1.1. Comunicación con el Lector de Skyetek

En el anexo B se ha realizado una introducción a la tecnología RFID, en el se detallan los diferentes elementos que involucra, como lectores etiquetas... Es este sub-apartado simplemente vamos a describir cómo se realiza la comunicación entre el microcontrolador y el lector a través de el bus i2c.

Como ya hemos dicho en el capítulo anterior el lector elegido para nuestra aplicación es el modelo m1-Mini de la empresa Skyetek. Se trata de un dispositivo i2c por lo que obedecerá a comandos enviados desde el el controlador i2c que posee el micro.

I2c es un protocolo de comunicación entre dispositivos que obedece al patrón Maestro-Esclavo. Cada dispositivo i2c posee una dirección de escritura y otra para lectura, cuando el micro quiere escribir sobre un dispositivo, pone en el bus la dirección de escritura del dispositivo y a continuación los datos a escribir, si por el contrario quiere leer algún dato de un dispositivo pone en el bus la dirección de lectura del dispositivo y es entonces cuando el dispositivo puede poner los datos en el bus.

El diagrama de comunicación con el lector puede verse en la Figura [3.1]. Skyetek ha desarrollado un protocolo interno para estructurar los datos que se intercambian entre micro y lector. En las Figuras [3.2] y [3.3] puede verse la estructura de básica de un petición y una respuesta.

Los campos coloreados en gris son opcionales, su aparición viene condicionada por el valor del byte de fags, el campo COMMAND indica si el comando es una lectura del ID de la etiqueta o una escritura o lectura de la memoria de la etiqueta. Los campos RID y TID sirven para ejecutar ordenes sobre un lector en concreto (RID) o una etiqueta en concreto (TID). AFI se

RESPONSE					
MSG LEN	RESPONSE CODE	RID	TAG TYPE	RESPONSE DATA	CRC
8 bits	8 bits	8 bits	8 bits	n * 8 bits	16 bits

Figura 3.3: Respuest: Lector →Microcontrolador

utiliza para ejecutar algunas funciones especificas de algún tipo de etiqueta. STARTING BLOCK y NUMBRES OF BLOCKS se utilizan para lecturas y escrituras en la memoria interna que algunas tags poseen. Por último DATA son los datos específicos de cada petición. Para más informacion revisar el documento completo del protocolo de Skyetek [2].

3.1.2. Comunicación Zigbee

El el capítulo anterior ya presentamos el módulo ETRX3 de Telegesis, al comunicarse con el microcontrolador mediante UART y comando AT, resulta bastante sencillo de manejar. En la documentación de Telegesis [3] podemos encontrar una completa descripción de todos los comandos disponibles. Como es conocido la comunicación por UART es uno de los métodos más simples de interconexión de dispositivos, la comunicación se basa en el envío bytes de forma asíncrona de la misma forma que se realiza en una comunicación serie en un PC.

A continuación se va a hacer una pequeña enumeración de los comandos más importantes usados para transmitir y recibir datos.

- **AT+BCASTB:** Transmite un mensaje de broadcast a toda la red.
- **AT+UCASTB:** Transmite un mensaje a un nodo de la red.
- **AT+SCASTB:** Transmite un mensaje al sumidero(en general el coordinador).
- **AT+JN:** Permite asociarse a una red ZigBee.
- **AT+SN:** Realiza una búsqueda de nodos en la red.
- **AT+EN:** Crea la red ZigBee.

...

Con estos comandos el establecimiento de la red y la comunicación entre nodos se reduce a la utilización de una serie de instrucciones quedando de una manera casi transparente la

arquitectura de la red formada.

3.1.3. Perfil ZigBee-Tecnodiscap

El Módulo de comunicaciones ZigBee de Telegesis usado en el desarrollo permite un envío de mensajes entre los diferentes nodos de la red, a la forma de estructurar la información que va dentro de estos mensaje la hemos llamado Perfil ZigBee-Tecnodiscap. El Perfil ZigBee-Tecnodiscap está basado en las especificaciones de la ZigBee Cluster Library y en la especificación de las capas altas del stack ZigBee.

Al hilo de lo explicado en el anexo dedicado a la tecnología ZigBee, el Perfil ZigBee-Tecnodiscap podría considerarse un perfil privado desarrollado por el grupo Tecnodiscap. El Perfil ZigBee-Tecnodiscap siguiendo las directrices de la ZigBee Alliance define diferentes «endpoints», «clusters» comandos y eventos. A continuación vamos a hacer una breve explicación de cada uno de estos conceptos.

Según el Perfil ZigBee-Tecnodiscap, cada dispositivo puede implementar diferentes endpoints, cada endpoint representa una funcionalidad individual que implementa el dispositivo, así por ejemplo, un sensor domótico que es capaz de sensar luz y temperatura, implementaría como poco dos endpoints, uno representaría el sensado de luz y el otro el de temperatura. Como se verá más adelante, la separación de funcionalidades por endpoints permite el acceso individual a cada una de las funcionalidades que puede dar cada endpoint.

Dentro de cada endpoint se pueden tener implementados diferentes clusters, cada cluster especifica una funcionalidad concreta dentro de un endpoint. Siguiendo con el ejemplo anterior del dispositivo sensor de luz y temperatura, dentro del endpoint que representa al sensor de temperatura podría estar implementado un cluster que emitiera una alarma(evento) cuando la temperatura superase un umbral. Los eventos y comandos son los mensajes específicos de cada cluster.

3.1.4. Estructura de Mensajes

A continuación se va a describir la estructura de los mensajes que intercambian los nodos ZigBee que obedecen al Perfil ZigBee-Tecnodiscap. Destacar que hay dos tipos de mensaje: eventos y comandos.

8 Bytes	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	N Bytes
ADDR_TO	END_POINT	PROFILE_ID	CLUSTER_ID	LENGTH	COMMAND	DATA
HEADER				PAYLOAD		

Figura 3.4: Estructura de un paquete tipo Comando

Comandos

Lo primero es definir que es un comando. Se denomina comando a todo aquel mensaje que parte del ordenador de control o de la pasarela residencial y tiene como destino cualquier nodo de la red.

En la Figura [3.4] muestra la estructura que tiene un comando tipo en la red implementada. A continuación se explicará brevemente que significa cada una de las partes que conforman el comando. ADDR_TO es la dirección a la que va remitido el mensaje, en nuestro caso no es necesario implementarlo porque el módulo ZigBee ya lo integra automáticamente igual que el PROFILE_ID que es el perfil del dispositivo. La parte denominada como END_POINT es el identificador, hexadecimal, del endpoint al que va dirigido el mensaje. CLUSTER_ID es la identificación del grupo de atributos en la que se encuadra el mensaje. Estos cuatro componentes del mensaje conforman la parte denominada como HEADER, esta parte es siempre fija, es decir su tamaño es siempre el mismo y no se puede obviar a la hora de mandar un comando. El resto del mensaje es la parte denominada PAYLOAD. Esta parte es de tamaño variable aunque también tendrá que ir siempre en el mensaje. La primera de sus partes LENGTH es la que define el tamaño del comando que se le envía al nodo, pero solo se expresa la longitud del PAYLOAD, puesto que como ya se ha explicado es la única parte del mensaje con tamaño variable. La segunda de sus partes es la denominada como COMMAND que como su propio nombre indica es la identificación del comando que se manda al nodo correspondiente. Y por último iría la parte denominada DATA, esta parte es la de tamaño variable, puesto que su longitud variará dependiendo del comando que se mande, en esta parte del mensaje se mandan los parámetros necesarios para su interpretación.

Eventos

Igual que en el caso anterior lo primero es definir qué se entiende por evento. Se denomina evento a todo aquel mensaje que parte de un nodo de la red y tiene como destino el ordenador

8 Bytes	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	N Bytes
ADDR_FROM	END_POINT	PROFILE_ID	CLUSTER_ID	LENGTH	EVENT	DATA
HEADER				PAYLOAD		

Figura 3.5: Estructura de un paquete tipo Evento

de control o la pasarela residencial.

Como se puede ver en la Figura [3.5] la estructura es la misma cambiando pocas partes, por lo que solo explicaran aquellas partes que se diferencian con respecto a los comandos. La primera de las diferencias es que en lugar de mandar el nodo de destino, mandamos el nodo de origen. Por eso la primera de las partes que como en el caso anterior está en la capa transparente al módulo ZigBee, por lo que no hay que programarla, es la denominada ADDR_FROM. El segundo cambio seria el parámetro EVENT siendo este, como su propio nombre indica, el evento que se manda, para notificar cualquier circunstancia ocurrida, al coordinador de la red.

3.2. Middleware

En esta sección se van a introducir conceptos y herramientas usadas para el desarrollo del middleware. En los anexos se puede encontrar información ampliada sobre Java, OSGi y otros aspectos nombrados en esta sección como pueda ser middleware y SOA.

3.2.1. OSGi (Open Service Gateway Initiative)

Open Services Gateway Initiative (OSGi) es un grupo de trabajo que surgió en Marzo de 1999, cuyo principal impulsor es Sun Microsystems. El resto de los miembros fundadores son: Alcatel, Cable and Wireless, Electricité de France, Enron Communications, Ericsson, IBM, Liberate Technologies, Lucent Technologies, Motorola, Nortel Networks, Oracle, Philips Electronics, Sybase and Toshiba. Actualmente tiene unos treinta socios.

El objetivo es definir y promover un estándar abierto para permitir conectar los servicios ofrecidos en redes metropolitanas(WAN) a redes de locales(LAN) o domóticas(LON). Esto permitirá la conexión de la próxima generación de aparatos inteligentes que se puedan encontrar en un hogar(oficina) con los servicios externos a la casa(oficina) ofrecidos a través de Internet. De esta forma, Proveedores de Internet(ISP), operadores de red y fabricantes de equipos pueden

ofrecer una amplia gama de servicios a los usuarios finales utilizando todos la misma pasarela.

La pasarela de servicios (Services Gateway) es un servidor embebido que se inserta en la propia red para conectar la red Internet externa con los clientes internos. La SG se inserta entre la red de Proveedores de Servicios y la red domótica, de tal forma que se separa la topología interna y externa de ambas redes. Esta SG debe ser capaz de manejar tanto flujo de datos como multimedia.

OSGi aporta a Java lo que se conoce como "Modularidad dinámica", esto quiere decir que podemos instalar, actualizar o eliminar componentes de software ^{.en} caliente", además es capaz de descubrir librerías o aplicaciones de forma dinámica y utilizarlos en nuestros módulos o aplicativos.

En resumen, lo que OSGi aporta a Java es por una parte la capacidad de instalar módulos (bundles OSGi) de una manera dinámica permitiendo así encenderlos y apagarlos tanto local como remotamente. Por otra parte OSGi nos permite que cada modulo publique objetos software (servicios o dispositivos) que pueden ser usados por el resto de módulos instalados.

Vamos a poner un ejemplo ilustrativo aplicado a aulas de estimulación multisensorial: Supongamos que hemos desarrollado dos dispositivos hardware con sus respectivos módulos OSGi que publican los respectivos objetos software correspondientes con cada dispositivo hardware. Supongamos que los dispositivos son unas luces tele-controladas por zigbee y una pantalla táctil. También supondremos que hemos desarrollado otro módulo OSGi (servicio) que permite el control de las luces con la pantalla táctil. Vamos a suponer que un tiempo después desarrollamos el guante con lector RFID del que se ocupa este proyecto, la ventaja de OSGi es que no tendríamos que cambiar todo el software, ni siquiera sería necesario detener la ejecución del sistema, nos bastaría con desarrollar tanto el módulo que publique el nuevo dispositivo como el servicio/modulo que le de funcionalidad (por ejemplo el control de las luces mediante la lectura de tags). Además estos módulos podrían ser instalados de forma remota en la maquina presente en colegio y desde ese momento podrían ser usados en la sala.

3.2.2. OSGi4AmI

OSGi4AmI es una ontología para integración de dispositivos es sistemas basados en inteligencia ambiental, su nombre es debido a que se ha implementado como una colección de interfaces Java para su uso en desarrollo con tecnología OSGi.

OSGi4AmI define una formalización de los dispositivos que podemos encontrar en un ambiente inteligente, de forma que puedan ser utilizados por las aplicaciones de forma transparente, independientemente del fabricante del dispositivo, y centrándose sólo en la funcionalidad que requieren las aplicaciones.

OSGi4AmI pretende proporcionar una representación virtual de los dispositivos presentes en el entorno físico, de forma que puedan ser utilizados por las aplicaciones del ambiente inteligente sin preocuparse de otra cosa que de la funcionalidad requerida del dispositivo.

OSGi4AmI no busca definir todas las características que puede tener un dispositivo, sino sólo aquellas que realmente son relevantes para las aplicaciones, dejando explícitamente a un lado detalles de funcionamiento y configuración de los dispositivos, que en la práctica no son relevantes a las aplicaciones.

No es extraño que un mismo dispositivo físico pueda ofrecer distintas funciones relacionadas en mayor o menor medida. En esos casos, no tendremos un dispositivo virtual equivalente al dispositivo real, sino que existirán varios dispositivos simples que ofrecen la funcionalidad completa del dispositivo original.

Además de considerar dispositivos particulares, OSGi4AmI contempla atributos y funcionalidades que puedan ser comunes a dispositivos diferentes, agrupándolos en «clusters», de forma que una implementación particular de un dispositivo concreto pueda seleccionar aquellos clusters que pueda ofrecer.

OSGi4AmI propone las bases para definir formalmente un dispositivo, pero da la libertad para definir nuevos dispositivos, incluso individuales, combinando las funcionalidades contempladas por los distintos clusters.

La ontología OSGi4AmI propone una jerarquización en niveles de los dispositivos, aproximadamente como se ve en la figura [\[3.6\]](#)

Todos los dispositivos del entorno inteligente se ajustan a la definición del dispositivo básico BaseDevice, y en función del tipo de dispositivo concreto del que se trate, se ajustarán también a la definición de Sensor, Actuator o SimpleHMI, o de otros tipos que se pudieran definir.

OSGi4AmI permite abstraernos de la tecnología con la que estén implementados los dispositivos permitiendo intercambiar dispositivos con la misma funcionalidad aunque estén implementados en tecnologías diferentes.

La figura [\[3.7\]](#) muestra la arquitectura del sistema de interacción para aulas de estimulación multisensorial, en él puede verse cómo se utiliza OSGi y OSGi4AmI para combinar dos tecnologías

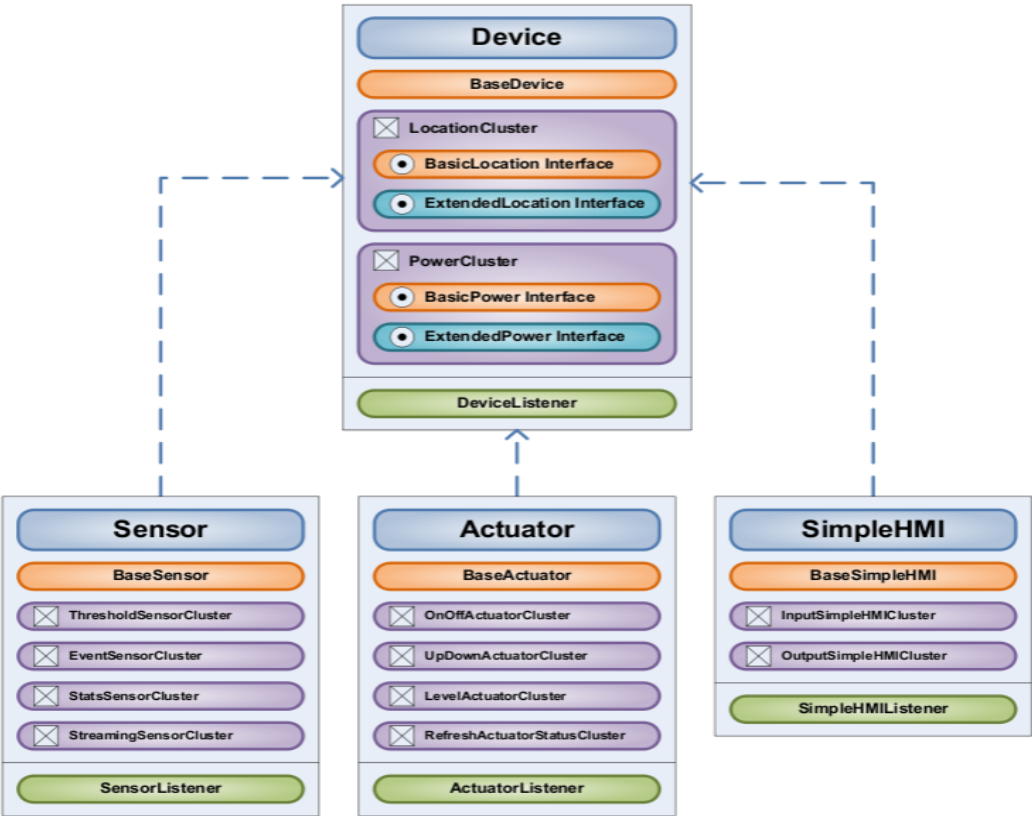


Figura 3.6: Jerarquización de los dispositivos

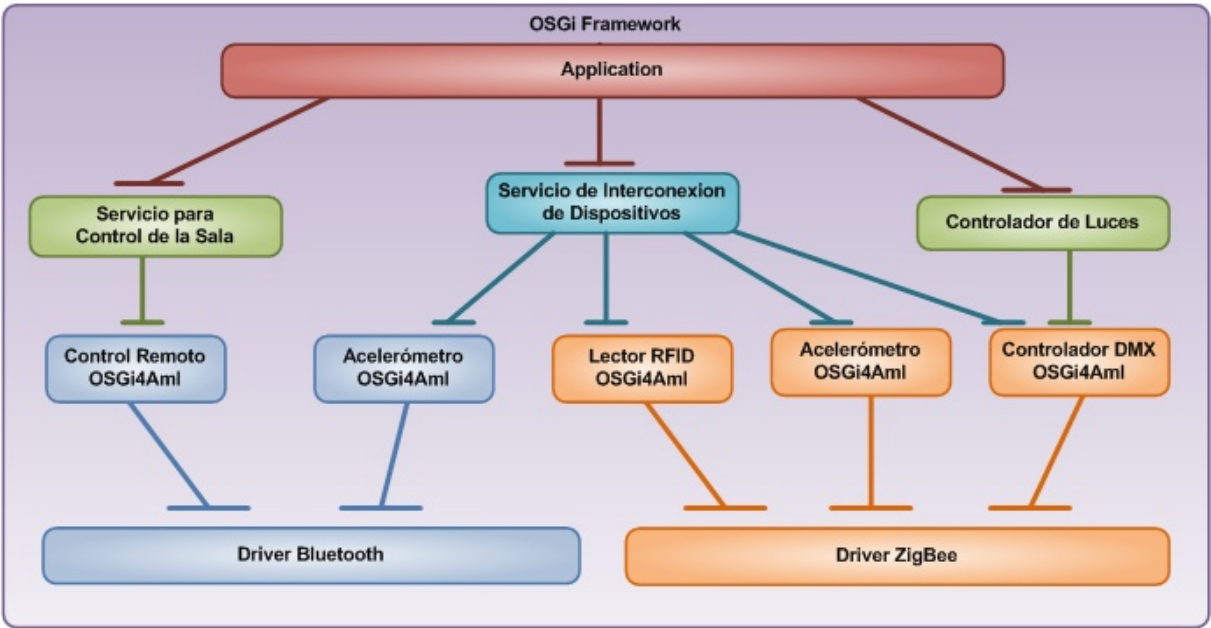


Figura 3.7: Ejemplo de Arquitectura con OSGi y OSGi4Aml

diferentes: Bluetooth y Zigbee. En la figura se puede ver que existe un driver para cada una de las tecnologías que se ocupa de publicar los diferentes dispositivos físicos como dispositivos software en el framework. Gracias a OSGi4AmI los servicios en capas superiores pueden hacer uso de los dispositivos sin tener en cuenta la tecnología en la que están implementados. Por ejemplo, el servicio de interconexión de dispositivos puede utilizar tanto un acelerómetro basado en Zigbee como otro proveniente de tecnología bluetooth al publicarse ambos con el mismo interfaz OSGi4AmI. Puede encontrarse mas información sobre OSGi4AmI en [\[4\]](#)

3.2.3. Relación entre OSGi4AmI y Perfil ZigBee-Tecnodiscap

Como hemos explicado en apartados anteriores, un dispositivo ZigBee que obedece al Perfil ZigBee-Tecnodiscap implementa diferentes endpoints atendiendo a las funcionalidades individuales que ofrece y a su vez dentro de cada endpoint implementa diversos clusters que dan funcionalidad específica a cada «sub-dispositivo».

Los diferentes clusters que puede implementar un endpoint están muy relacionados con la filosofía que persigue la ontología OSGi4AmI. Por una parte, todo dispositivo definido en OSGi4AmI hereda las propiedades y métodos del interfaz «Device», en este interfaz se describen métodos para la habilitación y des-habilitación del dispositivo, así como métodos para fijar y obtener información general del dispositivo. Por otra parte, existe un cluster llamado «Device_Cluster» definido en el Perfil ZigBee-Tecnodiscap que permite en el dispositivo físico realizar las operaciones de habilitación, recuperación de información... Puede verse que de esta manera queda enlazadas las operaciones que puede realizar el dispositivo físico con las que permite realizar el objeto software que lo controla en el middleware.

Para tratar de aclarar la relación entre Perfil ZigBee-Tecnodiscap y OSGi4AmI, vamos a describir tanto los endpoints y cluster implementados por nuestro dispositivo, así como los interfaces OSGi4AmI que implementara el objeto software. De esta manera se podrá ver la estrecha relación que guardan unos con otros.

En las Figuras [\[3.8\]](#), [\[3.9\]](#) y [\[3.10\]](#) se encuentra las tablas de endpoints, clusters y algunos de los comandos y eventos implementados en el dispositivo desarrollado junto con los identificadores de categoría y de dispositivo.

En el listado de endpoints destaca que el endpoint O4A_BASE_DEVICE es obligatorio implementarlo en cualquier dispositivo, es el endpoint que se encarga de de informar al software del

END_POINT		CATEGORY		DEVICE	
0x10	O4A_BASE_DEVICE	0x01	O4A_DEVICE	0x01	O4A_DEVICE
0x11	O4A_DEVICE	0x10	SENSOR	0X01	TEMPERATURE
0X12	O4A_DEVICE	0X30	SIMPLE_HMI	0X01	TWO_BUTTON
0X13	O4A_DEVICE	0X10	SENSOR	0X02	ACCELEROMETER
0X14	O4A_DEVICE	0X10	SENSOR	0X03	FALL DETECTOR
0X15	RFID_READER	0X10	SENSOR	0X0C	RFID_SKYETEK

Figura 3.8: Endpoints Implementados

0x10 BASE_DEVICE	0X11 TEMPERATURE	0X12 SIMPLE_HMI	0X12 ACCELEROMETER	0X14 FALL_DETECTOR	0X15 RFID_READER
DEVICE	DEVICE	DEVICE	DEVICE	DEVICE	DEVICE
POWER	SENSOR	SIMPLE_HMI	SENSOR	SENSOR_CUSTOM	SENSOR
			SENSOR_STREAMING	SENSOR_EVENT	SENSOR_STREAMING

Figura 3.9: Clusters Implementados

resto de endpoints implementados en el dispositivo. El resto de endpoints describen las diversas funcionalidades individuales en las que se puede dividir el dispositivo. En estas tablas se puede ver la relación entre el Perfil ZigBee-Tecnodiscap y OSGi4AmI, todo endpoint implementa el cluster DEVICE equivalente al interfaz Device de OSGi4AmI del que heredan todo los dispositivos. A su vez cada endpoint (a excepción del O4A_BASE_DEVICE) implementa uno de los siguientes clusters: SENSOR, ACTUATOR o SIMPLE_HMI, correspondientes a uno de los tres subtipos de dispositivos OSGi4AmI. Para aclarar más estos últimos conceptos puede compararse la definición de los interfaces OSGi4AmI: Device, Sensor (Figuras [3.11] y [3.12]) con los métodos y eventos de los clusters DEVICE y SENSOR (Figura [3.10]). Puede verse una correspondencia directa de métodos.

Como resumen decir que OSGi4AmI es una ontología que permite la abstracción del hardware en la capa middleware y el Perfil ZigBee-Tecnodiscap en la forma en la que el middleware se comunica con el hardware, pero diseñados de una manera concordante permiten realizar comunicaciones eficientes debido a que cuando se requiere alguna información del hardware sólo se transmite la información mínima, alargando así la vida de la batería del dispositivo.

3.2.4. Integración en Driver Zigbee de Tecnodiscap

Una vez explicados los conceptos de Perfil ZigBee-Tecnodiscap y OSGi4AmI e introducidos los clusters y endpoints Zigbee junto con los interfaces implementados, falta por explicar el

CLUSTER: DEVICE (Comandos)	
0x00	GET_TYPE
0X01	GET_CLUSTERS
0X02	PING
0X03	RESET
0X04	ECHO
0X05	GET_STATE
0X06	SET_STATE
0X07	GET_ID
0X08	SET_ID
0X09	GET_DESCRIPTION
0X0A	SET_DESCRIPTION

CLUSTER: DEVICE (Eventos)	
0x00	ACK
0X01	NOT_SOPPORTED
0X02	ERROR
0X03	DEV_TYPE
0X04	DEV_CLUSTERS
0X05	ECHO
0X06	DEV_ID
0X07	DEV_DESCRIPTION
0X08	DEV_STATE
0X09	GET_DESCRIPTION
0X0A	SET_DESCRIPTION

CLUSTER: SENSOR (Comandos)	
0x00	GET_VALUE

CLUSTER: SENSOR (Eventos)	
0x00	SENSOR_VALUE

CLUSTER: SENSOR_STREAMING (Comandos)	
0x00	START_STREAMING
0X01	STOP_STREAMING
0X02	GET_STREAMING_CONFIG
0X03	SET_STREAMING_CONFIG

CLUSTER: SENSOR_STREAMING (Eventos)	
0x00	STREAMING_PACKET
0X01	STREAMING_STARTED
0X02	STREAMING_STTOPED
0X03	STREAMING_CONFIG

Figura 3.10: Algunos Comandos y Eventos Implementados


```
package es.unizar.tecnodiscap.osgi4ami.device;

import java.util.Map;

public interface Device {

    public static final String DEVICE_CATEGORY = "DEVICE_CATEGORY";
    public static final String DEVICE_DESCRIPTION = "DEVICE_DESCRIPTION";
    public static final String DEVICE_SERIAL = "DEVICE_SERIAL";
    public static final String DEVICE_PROVIDER = "DEVICE_PROVIDER";
    public static final String DEVICE_DRIVER_ID = "DEVICE_DRIVER_ID";

    public String getDeviceId ();

    public String getDeviceDescription ();

    public int setDeviceDescription (String description);

    public int disable ();

    public int enable ();

    public boolean isEnabled ();

    public int registerDeviceListener (DeviceListener listener);

    public int unregisterDeviceListener (DeviceListener listener);

    public String[] getClusters ();

    public Map getAttributes();

}
```

Figura 3.11: Interfaz OSGi4AmI Device

```
package es.unizar.tecnodiscap.osgi4ami.device.sensor;

import es.unizar.tecnodiscap.osgi4ami.device.Device;

import java.util.Date;
import java.util.Map;

public interface Sensor extends Device {

    public static final String VALUE_KEY = "VALUE";

    public Map getValue ();

    public Date getValueTimestamp ();

    public int refreshValue ();

    public int registerSensorListener (SensorListener listener);

    public int unregisterSensorListener (SensorListener listener);

}
```

Figura 3.12: Interfaz OSGi4AmI Sensor

funcionamiento del driver ZigBee desarrollado por el grupo Tecnodiscap.

El driver Zigbee consiste en el middleware que se encarga de gestionar la red ZigBee (crearla, descubrir los dispositivos existentes al inicio, gestionar la aparición de nuevos dispositivos...), también es función de este driver la creación de los objetos software (dispositivos software) que controlan cada dispositivo presente en la red. Por último el driver publica en el framework OSGi los distintos dispositivos software para que puedan ser usados por los servicios de capas mas altas del software.

El driver esta desarrollado de tal manera que resulte sencilla la incorporación de nuevos dispositivos desarrollados. Para ello cuando se descubre un nuevo dispositivo se le ejecuta el comando `get_type` perteneciente al endpoint `BASA_DEVICE` y cluster `DEVICE` que devuelve los endpoint implementados en el dispositivo. Contesta información el driver ya es capaz de montar cada uno de los subdispositivos que integran el dispositivo.

Como es obvio para que el driver sea capaz de montar los diversos dispositivos software, primero han de estar programadas cada una de las clases que los representan. Pero esta tarea también resulta sencilla gracias a que todas estas clases extienden la clases genérica `device`, esto nos da acceso al stream de datos que llega por el puerto serie por lo que sólo queda realizar el procesado de estos datos.

Para finalizar recalcar que el driver zigbee está implementado de tal manera que se puede implementar un nuevo dispositivo de una manera simple gracias a su implementación modular.

Capítulo 4

Desarrollo Software

En este capítulo vamos a explicar el software creado para la utilización del dispositivo en aulas de estimulación multisensorial. El software desarrollado se ha realizado conjuntamente con Daniel Domínguez ya que él también ha desarrollado un dispositivo para aulas de estimulación multisensorial como proyecto fin de carrera. La dirección de ese proyecto fin de carrera también ha sido tarea de este trabajo fin de máster, las tareas llevadas a cabo durante esa dirección serán explicadas en el siguientes capítulo.

4.1. Aplicación de interacción entre dispositivos

En capítulos anteriores hemos visto como a través del framework OSGi y de la ontología OSGi4AmI hemos sentados las bases para la creación de un sistema que permita añadir a las aulas de estimulación multisensorial dispositivos capaces de interactuar entre ellos y con el usuario.

En esta sección vamos a mostrar una aplicación desarrollada para permitir a los profesores de educación especial configurar los diferentes dispositivos instalados en la sala para que interactúen unos con otros. Esta idea surge de la colaboración realizada con alumnos de diseño industrial, esta colaboración entre el grupo Tecnodiscap y los alumnos y profesores de diseño industrial se expondrá en el siguiente capítulo.

En la Figura [4.1] puede verse el interfaz de usuario de la aplicación, como puede observarse se ha tratado de simplificar al máximo el interfaz dado que los profesores de educación especial no tienen porque estar habituados al manejo del ordenador.

Se ha tratado de implementar un sistema intuitivo en el que el profesor sólo tiene que relacio-

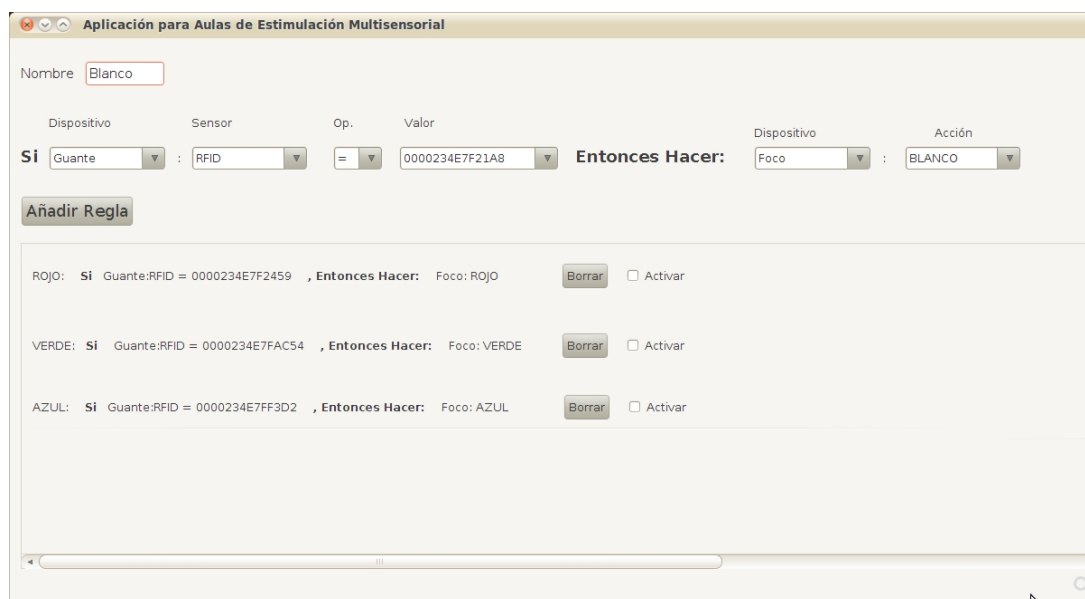


Figura 4.1: Interfaz de usuario de la aplicación

nar un dispositivo tipo «sensor» con un dispositivo tipo «actuador». Se busca que el profesional tenga que relacionar una acción del usuario con una reacción del sistema. Por ejemplo, si el usuario toca o se acerca a algún objeto marcado con una etiqueta RFID concreta que el sistema reaccione cambiando el color de la luz ambiental.

Esto se consigue gracias a una lista de reglas activas y un método para añadir nuevas reglas. Como puede verse en la Figura [4.1], en la parte inferior de la aplicación puede verse una lista de reglas que pueden estar activas o no, a lección del profesor. También es posible añadir nuevas reglas y borrar las ya existentes. Para añadir nuevas reglas se utiliza la parte superior de la aplicación, en ella puede verse diferentes listas desplegables que permiten elegir los dispositivos sensores y actuadores para crear las nuevas reglas.

El profesor debería primero elegir primero el dispositivo sensor, en esa lista aparecerá un nombre amigable para cada dispositivo desarrollado, por ejemplo para el dispositivo desarrollado en este trabajo fin de máster un nombre apropiado podría ser «Guante», una vez elegido el dispositivo, hay que elegir la acción del usuario a detectar, ya que puede darse el caso de que se pueda detectar diferentes acciones del usuario, siguiendo con el ejemplo del guante en la siguiente lista desplegable aparecerían las siguientes opciones: «RFID» o «Actividad», con RFID nos referimos a que se va a detectar la interacción de usuario con objetos marcados con etiquetas RFID; con Actividad nos referimos a que se va a detectar el nivel de actividad mediante el



Figura 4.2: Panel para interacción entre RFID y luces

acelerómetro que incorpora el dispositivo. Para concretar un poco más la acción a detectar se han añadido las dos listas desplegables siguientes, en la primera hay que elegir el un operador lógico, por ejemplo: $=$, $>$, $<$... en la siguiente hay que concretar el valor de comparación. Vamos a poner dos ejemplos aclaratorios, en el caso de haber seleccionado el dispositivo Guante y el sensor RFID, el operador sólo podría ser el simbolo « $=$ », y el campo valor debería contener el identificador de una etiqueta RFID. Por otro lado si se hubiera seleccionado el dispositivo Guante y el sensor «Actividad», podría utilizarse cualquier operador lógico, ya que en la lista de posibles valores sólo aparecerían valores cuantitativos del estilo: poca actividad, actividad media, alta actividad ... Por último sólo quedaría elegir el dispositivo sobre el que actuar y la acción que debe realizar dicho actuador, por ejemplo podríamos decir que se cambien las luces a un determinado color.

En la Figura [4.2] se puede ver como con varias cartulinas de colores y tags RFID se ha realizado un panel para trabajar con la aplicación anterior. Tan sólo habría que incluir cuatro reglas en la aplicación para que cuando se detecte una de las cuatro etiquetas presentes en la cartulina se cambiase el color de las luces.

Otros Trabajos Realizados

En este capítulo se van a describir otros trabajos realizados de forma paralela al desarrollo del dispositivo principal. Primero expondremos la colaboración llevada a cabo entre el grupo Tecnodiscap y profesores y alumnos de Ingeniería Técnica en Diseño Industrial, esta colaboración ha consistido en la generación de ideas para el interfaz de usuario del sistema para aulas de estimulación multisensorial. También se ha llevado a cabo la dirección de un proyecto fin de carrera de la titulación de Ingeniería en Telecomunicación, el proyecto ha sido realizado por Daniel Domínguez Guillén y ha consistido en desarrollar un controlador de luces DMX inalámbrico mediante tecnología ZigBee. Al igual este trabajo fin de máster, el proyecto de Daniel también esta orientado a su uso en aulas de estimulación multisensorial. Al final de este capítulo describiremos las tareas llevadas a cabo en la dirección de este proyecto.

5.1. Colaboración con Alumnos y Profesores de Diseño Industrial

Al comienzo del curso 2009-2010 se mantuvieron contactos con el profesor Rubén Rebollar del Área de Proyectos del Departamento de Diseño y Fabricación para realizar una colaboración con los alumnos de la asignatura Interacción Usuario-Producto de la titulación de Ingeniería Técnica Industrial en Diseño Industrial.

El objetivo de la colaboración era el de obtener una serie de propuestas sobre cómo debería ser el interfaz de usuario de un futuro sistema que mejorara las actuales aulas de estimulación multisensorial, a la vez que los alumnos se enfrentaban a una situación de diseño más cercana a la realidad y que les permitía tratar con futuros usuarios de sus diseños.

Para comenzar con la colaboración primero se realizó una presentación en la que se expuso a los alumnos el concepto de aulas de estimulación multisensorial y se presentaron los principales elementos con los que se cuenta actualmente. Se les presentaron las principales dificultades a las que se enfrentan los profesores de educación especial (principalmente falta de interactividad entre elementos), también se les propuso posibles elementos tecnológicos a incorporar que el grupo Tecnodiscap considera factibles o fácilmente desarrollables.

Una vez introducidos en el tema se convocó una sesión de brainstorming con el objetivo de que los alumnos pensasen posibles interacciones entre los elementos y las tecnologías que se les había presentado. De esa sesión de brainstorming se fijaron una serie de elementos que debían ser controlados por un hipotético sistema informático/electrónico. El trabajo de asignatura de estos alumnos consistió en diseñar por grupos un interfaz de usuario para este hipotético sistema.

Por concretar un poco más las especificaciones de ese interfaz, el trabajo consistía en diseñar el interfaz gráfico de la parte de planificación de sesiones para cada niño, junto con el diseño de un interfaz para controlar el sistema dentro de la sala una vez se está trabajando con un niño en concreto.

Antes de comenzar con el diseño de los interfaces se realizó una visita al colegio de educación especial Alborada para ver de cerca una sala de estimulación multisensorial y poder ver cómo los profesores de educación especial desarrollan su trabajo.

Una vez realizadas unas primeras versiones de los interfaces de cada grupo se realizó una prueba con usuarios reales, en concreto los profesores del colegio Alborada. Esta prueba sirvió para realizar ajustes en cuanto a funcionalidad y usabilidad de cada una de las propuestas.

Las propuestas de los alumnos en cuanto a la planificación de las sesiones giraron principalmente en torno a la planificación del uso de elementos dentro de un orden temporal, al estilo de un diagrama de Gantt. Con respecto al interfaz de control dentro de la sala hubo menos unanimidad en la solución y las variedades de propuestas fueron desde un control basado en dispositivos tipo PDA o teléfonos táctiles, pasando por controles basados en mando a distancia tipo Wii hasta controles por voz mediante dispositivos tipo «headset bluetooth».

Al final del curso cada grupo realizó una presentación de su desarrollo así como una valoración de la experiencia de trabajo sobre un caso práctico real. En general la experiencia resultó muy positiva entre alumnos y profesores, resultando enriquecedora tanto para los alumnos como para el grupo Tecnodiscap.

5.2. Dirección del Proyecto Fin De Carrera

Durante toda la realización de este trabajo fin de máster se ha estado realizando en paralelo la dirección de un proyecto fin de carrera de la titulación de Ingeniería en Telecomunicación, el proyecto ha sido realizado por Daniel Domínguez Guillén y Roberto Casas Nebra ha sido su Ponente.

El objetivo principal del proyecto ha sido desarrollar un puente ZigBee a DMX para integrar el control de luces dentro de abanico de dispositivos disponibles para usar en aulas de estimulación multisensorial. Durante el proyecto se ha abordado desde la elección de componentes y el diseño de la PCB, la programación del firmware para obedecer al Perfil ZigBee-Tecnodiscap y la posterior inclusión del dispositivo en el driver ZigBee y su publicación en el framework OSGi gracias a los interfaces OSGi4AmI.

Destacar que la tecnología empleada en el microcontrolador ha supuesto una novedad dentro del grupo Tecnodiscap, en el grupo estamos habituados a desarrollar dispositivos sobre tecnología PIC o dsPIC de la empresa Microchip, para el proyecto de Daniel se decidió usar una nueva tecnología emergente de la empresa Cypress llamada PSoC. La característica principal de esta tecnología es que aparte del simple microcontrolador tiene integrado en el chip una serie de bloques analógicos y digitales programables, de ahí su nombre (Programable System on Chip).

Durante todo el proyecto se ha tratado de que el alumno adquiriese conocimientos más allá del puro diseño electrónico, se ha tratado de infundir metodologías de trabajo para que los diseños sean modulares y ampliables. También se ha tratado de mostrar todo el ciclo de diseño de un producto electrónico, desde la electrónica básica, pasando por el firmware y su interacción con un driver, hasta llegar a capas más altas del software donde se realiza las aplicaciones de usuario. En este último aspecto destacar que el alumno ha adquirido una buena base sobre programación orientada a objetos en lenguaje Java.

Respecto a la aportación personal de la dirección de este proyecto destacar que ha servido para aprender más sobre coordinación y gestión de tareas, además de el aporte la satisfacción personal al transmitir conocimientos a una persona y ver su evolución conforme pasa el tiempo.

El proyecto de Daniel Domínguez tiene por título: « Control vía Zigbee de dispositivos de iluminación con protocolo DMX para salas de estimulación multisensorial » y será presentado en la convocatoria de Diciembre de 2010 ante el tribunal de Electrónica.

Conclusiones y trabajos futuros

6.1. Conclusiones

El objetivo principal de este proyecto era el desarrollo de un nuevo dispositivo para su utilización en aulas de estimulación multisensorial. Dicho dispositivo se ha desarrollado tratando de maximizar las posibilidades de interacción con otros elementos presentes en la sala. Es interacción es la principal demanda de los profesores de educación especial que trabajan con niños en estas salas.

Atendiendo a esa capacidad de interacción el dispositivo cuenta con un lector RFID capaz de reconocer la interacción del usuario con cualquier objeto etiquetado con una tag RFID. Esta característica supone que cuando un usuario se acerque, toque o agarre un objeto se podrá activará algún elemento de la sala capaz de provocar en el usuario un estímulo identificable dadas sus capacidades físicas y psíquicas.

Para poder monitorizar otros parámetros relativos al usuario se han incorporado otros sensores como por ejemplo un acelerómetro en tres ejes capaz de enviar al sistema un streaming de aceleración, con esta información el sistema puede enviar eventos en función del nivel de actividad del niño.

Así pues el objetivo de incorporar un nuevo dispositivo que aumente la capacidad de interacción en salas de estimulación multisensorial puede darse por cumplido.

Otro objetivo de este proyecto era el fijar la arquitectura software de un sistema que permitiese la interacción entre elementos de una sala de estimulación multisensorial. Esta arquitectura ha sido diseñada en torno a un framework de programación Java llamado OSGi. OSGi se ajusta

perfectamente a las características que podría tener un sistema de sensores y actuadores que interactúan entre si. A parte de OSGi el sistema se fundamenta en un driver ZigBee que publica dispositivos conforme son dados de alta en la red ZigBee. Si a esto añadimos la capacidad de abstracción sobre la tecnología hardware que nos brinda la ontología OSGi4AmI obtenemos una arquitectura robusta que permite el desarrollo de dispositivos y su incorporación al sistema de una manera modular y que se ajusta muy bien a las características necesarias en una aula de estimulación multisensorial.

Basándonos en la arquitectura antes expuesta se ha desarrollado una aplicación básica que permite a los profesores de educación especial configurar los dispositivos presentes en el aula para que interactúen entre ellos. Esta configuración se basa en generar una serie de reglas del tipo «Si X entonces hacer Y». Esta aplicación puede ser mejorada más adelante conforme se vayan desarrollando nuevos dispositivos.

Con todo esto se ha sentado las bases para desarrollar un sistema completo de dispositivos interactivos para aulas de estimulación multisensorial, el siguiente paso es validar el trabajo realiza con los profesores de educación especial a la vez que se van desarrollando nuevos dispositivos que complementen el sistema.

Como tarea secundaria en este trabajo se ha llevado a cabo la dirección de un proyecto fin de carrera, el proyecto consistía en desarrollar un actuador de luces también para aulas de estimulación multisensorial. La experiencia ha resultado sumamente grata ya que el proyectando comenzó sin tener conocimientos sobre ciertas tecnologías como ZigBee, PSoC, Java...y ha acabado teniendo un nivel aceptable en la mayoría de ellas.

Para concluir apuntar que la experiencia de diseñar un sistema que auna software y hardware para una aplicación real, en la que hemos tratado con diferentes personas con diferentes perfiles profesionales ha resultado sumamente gratificante y enriquecedora.

6.2. Futuros Trabajos

La primera tarea después del desarrollo de los dispositivos y la aplicación de control es la validación con usuarios. Antes de proseguir es conveniente mostrar la aplicación a los profesores de educación especial y recoger sus opiniones y consejos de mejora.

Por otra parte es necesario el desarrollo de nuevos dispositivos que aumenten las posibilidades de interacción del sistema. Para ello el grupo Tecnodiscap se encuentra en estos momentos

realizando propuestas de proyectos de investigación para obtener financiación para acometer los desarrollos.

Una vez aparezcan nuevos dispositivos y ayudándonos del «feedback» que nos puedan dar los profesores de educación especial, habrá que acometer mejoras en la aplicación de configuración y desarrollar algún interfaz que permita al profesor controlar la sala en tiempo real mientras trabaja con un niño.

Por otra parte y centrándonos en el dispositivo desarrollado una posible mejora es la de integrar en el firmware dispositivo la capacidad de monitorización de movimientos del usuario. Quizás esto requiera de la incorporación de nuevos sensores inerciales como giróscopos o magnetómetros.

