

5213: Crear dispositivo para personas sordas (plataforma hardware Arduino)

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per

Marc Codina Barberà
i dirigit per
Pilar Gómez Sánchez
Bellaterra 16 de juny de 2013

El sotesignat, Pilar Gómez Sánchez

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Marc Codina Barberà

I per tal que consti firma la present.

Signat:

Bellaterra, 16 de juny de 2013

Contenido

Capítulo 1	3
Introducción	3
1.2. Objetivos	3
1.3. Motivación	4
1.4 Estructura de la memoria	4
Capítulo 2	6
Estado del arte	6
Capítulo 3	8
Análisis de la viabilidad	8
3.1 Estudio económico	8
3.1.1 Recursos hardware	8
3.1.2 Recursos humanos.....	8
3.1.3 Recursos software	8
3.1.4 Coste.....	9
3.2 Planificación temporal	10
Capítulo 4	12
Tecnología utilizada	12
4.1. Hardware	12
4.1.1. Plataforma de desarrollo	12
4.2. Software	21
4.2.1. Android.....	21
4.2.2 Arduino IDE	25
4.3 Tecnología de comunicaciones	27
4.3.1. Comparativa Bluetooth vs Zigbee.....	27
4.3.2 I2C.....	29
Capítulo 5	31
Análisis de requerimientos	31
5.1. Diseño	32
5.2. Implementación.....	37
5.2.1 Librerías Arduino	44
Capítulo 6	46
Pruebas	46
6.1. Comprobación perdida de datos en la comunicación.....	46
6.2. Establecer comunicación entre el módulo arduino y el smartphone.....	46
6.3. Chequear batería	47
6.4. Comprobar versiones de Android	47
Capítulo 7	48
Conclusiones	48
7.1. Trabajo futuro.....	49
Bibliografía	50

Índice de Figuras y tabla

Figura 1 Diagrama de Gantt.....	11
Figura 2 Primer prototipo Arduino.....	14
Figura 3 Arduino diecimila	16
Figura 4Arduino UNO	17
Figura 5 Arduino Shield con módulo Xbee conectado (Zigbee)	17
Figura 6 Módulo Blueetooth compatible con Arduino Shield	18
Figura 7 Modelo LCD utilizado.....	19
Figura 8 Motor y circuito asociado	20
Figura 9. Vista del HTC Dream	23
Figura 10 Vista del Samsung Galaxy S2	24
Figura 11Vista IDE Arduino	26
Figura 12 Comparativa tecnología wireless [18]	27
Figura 13 Esquema conexión I2C	30
Figura 14Mapa de situación dispositivos.....	32
Figura 15 Diagrama primera fase.....	34
Figura 16 Esquema general de la aplicación con Smartphone.....	36
Figura 17 Diagrama de flujo del envío de avisos	37
Figura 18 Esquema fase 1	38
Figura 19 Prototipo fase 1. LCD, motor y leds	40
Figura 20 Prototipo fase 2. Conexión del LCD y leds, y la conexión entre módulos por I2C	41
Figura 21 Menu de la aplicación.....	43
Tabla 1. Costes del proyecto	9

Capítulo 1

Introducción

A lo largo de los años, la sociedad ha intentado mejorar el estilo de vida de las personas. Han aparecido nuevos sistemas que mejoran la calidad de vida de la gente, facilitando tareas cotidianas. Pero, en esta sociedad no todos tenemos las mismas necesidades y cada día se depende más de las señales acústicas o visuales para la interacción con los distintos dispositivos disponibles. Es por ello que la gente con problemas auditivos o de visión, no siempre lo tienen fácil para poder usar estas mejoras.

Sin embargo, todos estos dispositivos que usan métodos sonoros o visuales pueden, mediante una modificación, incorporar un sistema de aviso por radio frecuencia que puede ser recibido por otro dispositivo. En la actualidad, existen varios sistemas de señalización para personas con algún problema, ya sea auditivo o visual, el principal inconveniente es que todos estos dispositivos son independientes entre ellos y es necesaria la utilización de distintas estaciones receptoras para usarlos todos.

1.2. Objetivos

Para dar solución a la gran variedad de receptores que existen en el mercado, el objetivo de este proyecto, es la de realizar un prototipo de un sistema de avisos para gente sorda o con problemas de oído. De esta forma se pretende conseguir una unificación de los distintos sistemas de señalización que existen en la actualidad

Utilizando las mejoras que ofrecen hoy en día los dispositivos smartphone, se diseñará una aplicación para uno de estos dispositivos el cual recibirá los avisos utilizando tecnología inalámbrica.

Mediante la utilización de la placa Arduino[1] se creará un sistema de sensores para captar

señales externas y enviarlas a un receptor central el cual se comunicará con el dispositivo Smartphone. Al tratarse de un prototipo, solo nos centraremos en la creación e implementación del sistema, pero la finalidad es la de, en un futuro, crear un sistema de tamaño reducido para poder ser usado en el día a día por las personas con deficiencia auditiva.

A parte del prototipo con Smartphone, una primera fase del proyecto, consistirá en la creación de un prototipo con pantalla LCD que sea totalmente autónomo. Con la idea de que en un futuro sea del tamaño de un reloj de pulsera, que sea cómodo y fácil de usar para el usuario.

1.3. Motivación

La motivación para la realización de este proyecto es la de poder diseñar un sistema electrónico trabajando con una plataforma OpenHardware, ampliar conocimientos tanto en el diseño e implementación de un proyecto, también saber que en un futuro todo el trabajo realizado pueda llegar a servir para facilitar un poco más la vida a todas esas personas con problemas auditivos.

1.4 Estructura de la memoria

A continuación se procederá a resumir el contenido de los apartados que conforman esta memoria.

- En el capítulo uno se hace una introducción sobre la importancia de la tecnología en nuestra sociedad. Y se describen los objetivos y motivaciones para la realización de este proyecto.
- En el capítulo dos se realiza una pequeña explicación de cómo está actualmente la tecnología orientada a personas con problemas auditivos.
- El capítulo tres se detallan los requisitos del proyecto, tanto software como hardware. Así como un análisis de costes para la realización del prototipo y la planificación temporal para la realización del mismo.
- En el cuarto capítulo se explica las características del hardware utilizado para la realización del proyecto, así como una breve explicación sobre los inicios de las tecnologías utilizadas.

- El capítulo cinco explica el análisis, diseño e implementación que se han seguido para la realización del prototipo.
- En el capítulo seis se describen las pruebas a las que se ha sometido el prototipo
- El capítulo siete se muestran las conclusiones extraídas de la realización del proyecto, así como de las futuras mejoras que se pueden implementar.

Capítulo 2

Estado del arte

Como se comenta en la introducción, en la actualidad, en el mercado existen distintos dispositivos que realizan varias funciones de las planteadas en este proyecto como pueden ser:

- Sistema de detección de humos/fuego. Avisan con una alarma sonora y luminosa a la persona con problemas de audición o sorda, a través de una estación receptora.
- Sistema de aviso de llamada a la puerta. Encienden y apagan distintas luces de la casa para avisar a la persona sobre el aviso de llamada, varían su frecuencia para dar distintos mensajes con el parpadeo de luces.
- Avisos en electrodoméstico. Avisan a una estación receptora sobre el funcionamiento de un electrodoméstico, como puede ser la finalización del horno microondas o el nivel de ruido de la lavadora.

El principal problema de estos sistemas es que son independientes uno de otros y necesitan distintas estaciones receptoras para cada uno de ellos. Hay algunos que ofrecen la posibilidad de realizar varios mensajes de alarma, para un variado número de dispositivos, pero tiene un coste un poco alto.

Con este proyecto pretendemos conseguir agrupar todos los posibles sistemas de aviso en un solo dispositivo que llevaría siempre encima la persona. Para llevarlo a cabo, en el mercado existen distintas plataformas de desarrollo hardware, como pueden ser: Arduino, Pingüino. [2]

En nuestro caso nos decantamos por Arduino, ya que nos ofrece una serie de módulos hardware con sus propias librerías, que facilitan el desarrollo de nuevos sistemas Arduino. El sistema Arduino es una plataforma de hardware opensource, lo que nos ofrece total libertad para poder trabajar con ella ya que nos facilita toda la documentación del dispositivo.

Además, nos ofrece una placa integrada con el microprocesador y las entradas y salidas de datos, ensamblada, mientras que “Pinguino” está más orientado a la realización por uno mismo del hardware que se necesite.

La plataforma Arduino nos permite una gran variedad de lenguajes para su programación. Y su coste es un poco inferior al de las otras plataformas de desarrollo. Cada pocos meses se realizan nuevas revisiones del hardware, lo que origina la posibilidad de elección entre una gran variedad de dispositivos de la familia Arduino.

Están apareciendo cada cierto tiempo nuevos sistemas de desarrollo basados en la idea de Arduino, gracias a la posibilidad de crear tu propio hardware que ofrece el concepto de open-hardware como puede ser Freeduino[3].

Freeduino empezó como un proyecto para publicar hardware que fuera compatible con Arduino y con las plataformas de desarrollo que aparecieron a partir de los diseños de Arduino. Con el tiempo el concepto Freeduino fue adoptando el significado de cualquier cosa que sea Arduino-compatible. De esta forma, cualquier diseño o desarrollo que sea compatible con Arduino, se conoce con el nombre de Freeduino.

Capítulo 3

Análisis de la viabilidad

3.1 Estudio económico

3.1.1 Recursos hardware

Como se ha comentado anteriormente, la realización de este proyecto se llevará a cabo mediante la plataforma hardware opensource Arduino. Usando distintos módulos de comunicaciones para poder desarrollar todas las funcionalidades que se necesitan.

En la primera fase del prototipo se usará el módulo de comunicaciones Xbee. Este módulo aportará la tecnología ZigBee a la plataforma Arduino, conectado mediante un Arduino shield a la placa Arduino. Hará falta una placa electrónica para conectar todos los dispositivos, así como resistencias, leds y display para visualizar distintos mensajes informativos.

Para la segunda fase, el usuario deberá de disponer de un Smartphone para la recepción de todas las señales de aviso, ya que se usará uno de estos dispositivos para la notificación. Aparte, también se usará un módulo Bluetooth para la comunicación entre la plataforma Arduino y el Smartphone.

3.1.2 Recursos humanos

Un jefe de proyecto, para controlar que el trabajo se desarolla según lo previsto.

Un analista-programador para desarrollar toda nuestra aplicación en la plataforma Arduino y conectar entre si los distintos módulos del proyecto para que interactúen.

3.1.3 Recursos software

La implementación del proyecto se realizará con el lenguaje propio de la plataforma, así que no tendremos coste en la compra de licencias.

3.1.4 Coste

Una vez descritos los diferentes recursos que se necesitan para implementar este proyecto, se detallará el coste de este proyecto:

TOTAL			
Recursos Hardware			
Módulo Arduino	20€	3 unidad	60€
Módulo Arduino Xbee shield	42€	3 unidades	126€
Módulo Bluetooth	40€	1 unidad	40€
SmartPhone	300€	1 unidad	300€
LCD	10€	1 unidad	10€
Recursos Humanos			
Jefe de proyecto	73€/h	20 horas	1460€
Analista-programador	40€/h	300 horas	12000€
TOTAL			13996€

Tabla 1. Costes del proyecto

Una vez terminado el prototipo, si se quisiera comercializar el producto en su primera fase, tal como se comenta en el apartado de objetivos del proyecto, esto sería un dispositivo del tamaño de un reloj de pulsera. Para ello se debería de utilizar una placa Arduino de tamaño reducido y una pantalla LCD que fuera de pequeño tamaño para mantener uno de los requisitos: la comodidad del usuario. El coste entonces sería 45€ para un display OLED de 4,5cm, en lugar de usar los LCD de un tamaño mayor (entre 8 y 10cm).

Como en la actualidad los smartphones no llevan la tecnología ZigBee incorporada, se seguirá

necesitando una estación base para hacer la conversión entre las dos tecnologías, el coste de comercialización de la misma, dependerá de la inclusión de una pantalla para mostrar información al usuario. Si en un futuro esta tecnología se incorpora en los dispositivos, se podrá prescindir de la estación base.

3.2 Planificación temporal

Para la realización de este proyecto se ha estimado los siguientes plazos.

- Se realizará un análisis de mercado para comprobar los distintos dispositivos existentes en la actualidad y un estudio de las distintas plataformas de desarrollo para la realización del proyecto. El tiempo de realización de este punto se estima en 3 semanas.
- Un análisis de requerimientos que podemos desglosar en análisis de la solución cuyo tiempo estimado es de 2 semanas; la realización del diseño, que llevará 2 meses, tiempo que se usará para buscar los mejores dispositivos para realizar la implementación del prototipo
- El tiempo estimado de implementación será de 2 meses para la parte de programación y 1 mes para la realización de pruebas.

En la siguiente figura se puede ver un diagrama de Gantt con el tiempo dedicado a cada parte del proyecto.

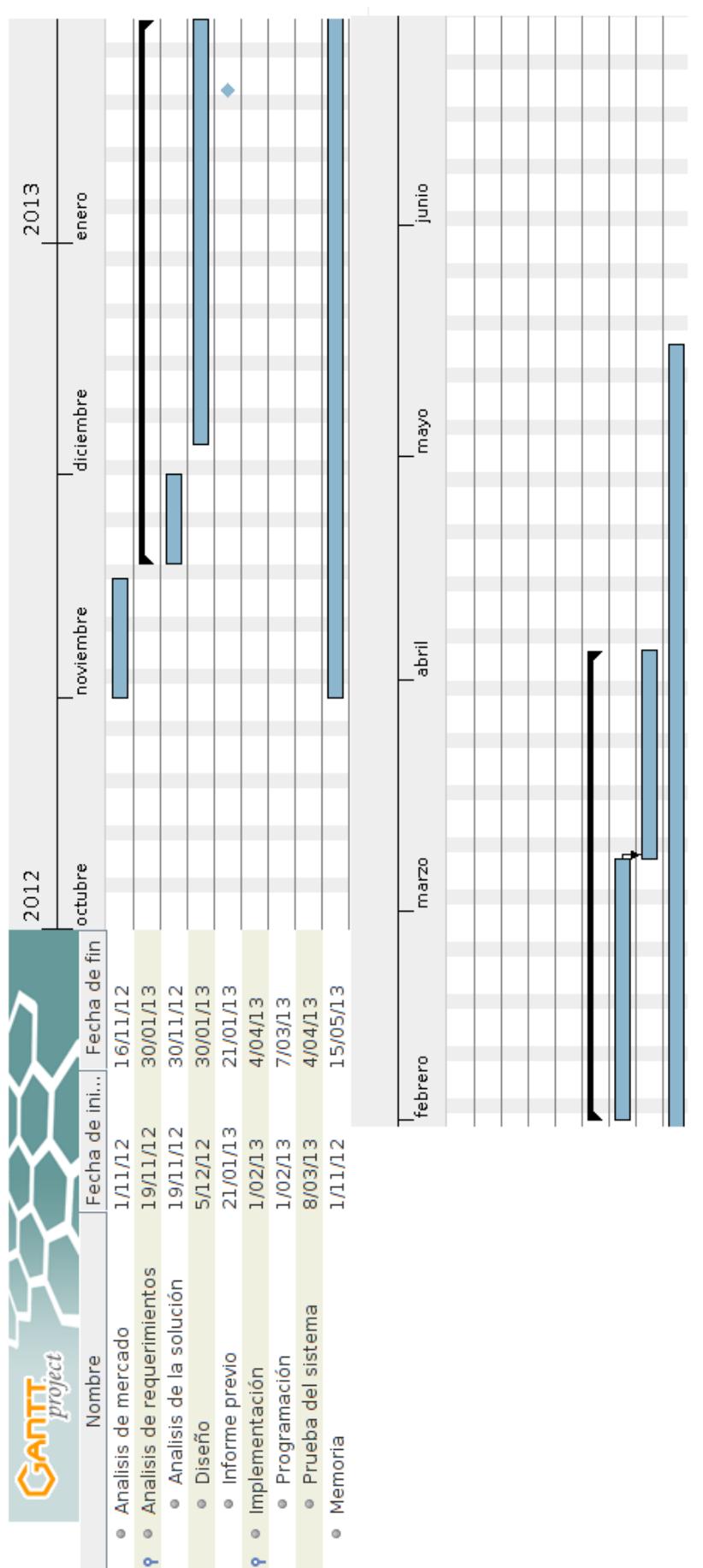


Figura 1 Diagrama de Gantt

Capítulo 4

Tecnología utilizada

En este capítulo se detallará el hardware y el software utilizado para la implementación del sistema centralizado de avisos, que consistirá a grandes rasgos en varios sistemas emisores que emitirán una señal recogida por un sistema receptor.

4.1. Hardware

En este proyecto se usarán distintos dispositivos hardware. La base para la creación de nuestro dispositivo será la plataforma de desarrollo opensource Arduino.

En concreto se usará un módulo Arduino UNO para realizar la gestión del dispositivo y se le conectará un módulo ZigBee [4] para otorgar al dispositivo soporte para conexiones inalámbricas.

La conexión entre el módulo ZigBee y la plataforma Arduino, se realizará mediante un módulo Xbee shield, que nos permite conectar el ZigBee con la plataforma de desarrollo.

4.1.1. Plataforma de desarrollo

4.1.1.1. Arduino

Arduino es una herramienta para hacer que los ordenadores puedan sentir y controlar el mundo físico a través de un ordenador. Es una plataforma de desarrollo de computación física (physical computing) de código abierto, basada en una placa con un sencillo microcontrolador y un entorno de desarrollo para crear software (programas) para la placa.

Se puede usar Arduino para crear objetos interactivos, leyendo datos de una gran variedad de interruptores y sensores y controlar multitud de tipos de luces, motores y otros actuadores físicos.

Los proyectos de Arduino pueden ser autónomos o comunicarse con un programa (software) que se ejecute en el ordenador (ej. Flash, Processing, MaxMSP). La placa puede montarla uno mismo o comprarla ya lista para usar, y el software de desarrollo es abierto y se puede descargar gratuitamente.

El lenguaje de programación de Arduino es una implementación de Wiring [5], una plataforma de computación física, que a su vez se basa en Processing, un entorno de programación multimedia. Este lenguaje, conocido como Wiring, es un conjunto de funciones que encapsulan el funcionamiento del hardware facilitando el uso del mismo, adicionalmente se puede utilizar las características del lenguaje C++ dentro del entorno de desarrollo lo que permite crear funciones, punteros, clases y objetos e incluso utilizar lenguaje máquina y otras características propias del compilador para micro controladores de la familia AVR fabricados por Atmel [6]. Esto da una gran flexibilidad al momento de crear proyectos complejos y gracias al entorno de desarrollo permite crear aplicaciones rápidamente. También es posible desarrollar librerías que pueden ser instaladas dentro del entorno de desarrollo y existe un gran número de ellas que permiten el manejo de servos, comunicación serial, pantallas LCD, GPS y otros componentes.

Las principales características de Arduino respecto a otras placas existentes, son:

- Asequibles, comparadas con las otras plataformas de microcontroladores, la placa Arduino tiene un coste de montaje inferior a cualquiera de las otras plataformas.
- Es multi-plataforma, funciona en distintos sistemas operativos, facilitando la programación de la plataforma. La gran mayoría de plataformas de microcontroladores, solo dan soporte al sistema operativo Windows.
- Facilidad de desarrollo, basado en Procesing, el entorno de programación es muy fácil para los principiantes y suficientemente flexible para los programadores avanzados.
- Ampliable y opensource, la gran ventaja de usar licencia libre, es que cualquier usuario puede aportar su conocimiento a ampliar la funcionalidad del producto, creando nuevas librerías. En el aspecto hardware, al estar publicado bajo licencia Creative Commons [7], los usuarios pueden implementar sus propios diseños de los módulos, mejorarlos o ampliarlos.

Breve historia de Arduino [8]

Desde el año 2005 en el que Massimo Banzi, ofrece a sus estudiantes del Interaction Design Institute Ivrea (IDII) una modesta herramienta, Arduino ha ido creciendo hasta revolucionar la electrónica de bajo coste.

En el año 2002, Massimo es contratado como profesor asociado en el IDII, para promover nuevas formas de diseño interactivo, conocido como computación física. A causa de los limitados fondos de los que disponía y de la necesidad de no depender del microcontrolador creado por Parallax, BASIC Stamp, se preguntó por qué no crear una plataforma que cubriera todas sus necesidades.

Contactó con un conocido suyo del MIT, que había desarrollado un lenguaje de programación llamado Processing. Una vez creada la base del futuro proyecto, buscaron qué licencia les permitiría desarrollar su proyecto open-source sin tener problemas de licencia, la solución fue el uso de Creative Commons.

Otro de los problemas que tuvieron que solucionar, fue el de encontrar los componentes que no provocaran problemas a la hora de interactuar con un PC o MAC, su idea era que el sistema fuera Plug&Play.

A día de hoy hay 13 placas con distintas configuraciones y 8 accesorios para expandir las posibilidades de la plataforma.

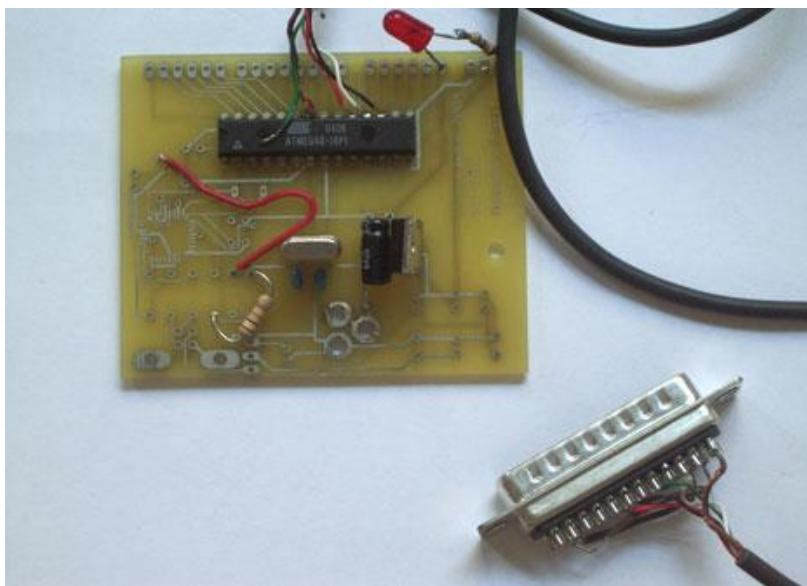


Figura 2 Primer prototipo Arduino

Arduinos utilizados

Después de comentar brevemente la historia de Arduino, a continuación comentaremos las características del hardware que se ha utilizado:

* Arduino diecimila

El Arduino diecimila es una placa microcontroladora basada en el microcontrolador ATmega168. Tiene 14 entradas/salidas digitales, 6 entradas analógicas y una conexión USB. Tiene una memoria flash de 16KB, un módulo SRAM de 1KB y un EEPROM de 512B.

La placa tiene distintos métodos de comunicación con otros dispositivos, como puede ser un puerto de comunicaciones serial. Tiene también un controlador ATmega8U2 para la comunicación serial a través del USB. Y mediante un Bootloader¹ se permite la carga de código. El dispositivo tiene un volumen de trabajo de 20 MIPS a una frecuencia de 20 MHz.

El Atmega 168 es un microcontrolador de 8 bits con arquitectura RISC, fabricado por la empresa Atmel[9]. Tiene 16KB de memoria flash con capacidad de lectura/escritura simultánea.

Para el prototipo, esta placa es la utilizada como emisor de datos hacia la base receptora. Se conecta el dispositivo a controlar a una de sus entradas para controlar en que momento tenemos un aviso y mandarlo mediante ZigBee al dispositivo receptor.

¹ Un bootloader es un gestor de arranque, es un pequeño programa que prepara todo lo necesario para que el sistema operativo de los dispositivos pueda funcionar correctamente.

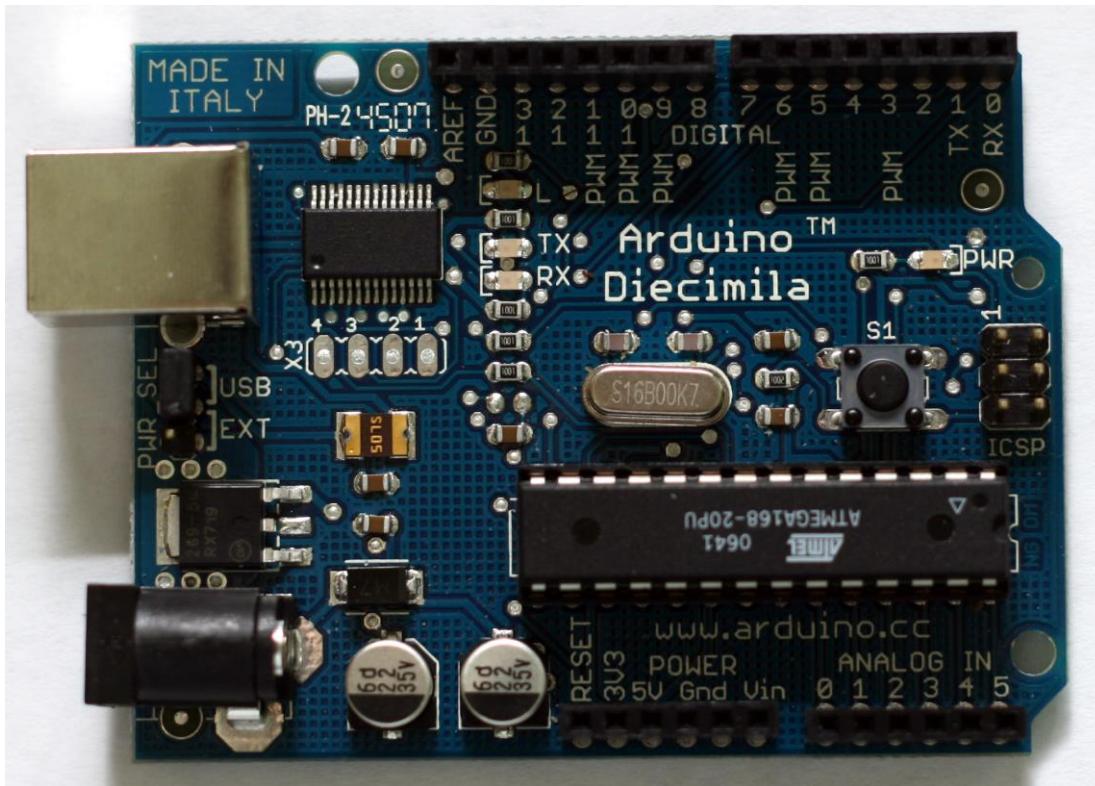


Figura 3 Arduino diecimila

* Arduino UNO

El Arduino Uno es una placa microcontroladora basada en ATmega328. Tiene 14 entradas/salidas digitales, 6 entradas analógicas, una USB connection, un conector de alimentación a 5V, una cabecera ICSP(In-Circuit Serial Programming) y un botón de reset, para reiniciar el programa cargado en memoria. A diferencia del anterior modelo, el modelo Uno, no usa el chip controlador FTDI para la comunicación USB-serial. En vez de eso, incorpora el Atmega16U2 (Atmega8U2 hasta la revisión 2) programado para funcionar como un convertidor USB-Serial.

Este modelo tiene una memoria flash de 32KB; 2KB de memoria SRAM y un EEPROM de 1KB, la velocidad de reloj del microcontrolador es de 16MHz.

En el prototipo, usamos dos de estas placas para realizar la función de base receptora-emisora. Uno de los módulos llevará conectado un módulo Zigbee mediante un Arduino shield. Con la tecnología inalámbrica ZigBee, recibiremos avisos del módulo emisor., que enviaremos a otra placa Arduino UNO mediante el bus de datos I2C. Esta otra placa tiene un módulo bluetooth conectado mediante otro módulo Arduino Shield.

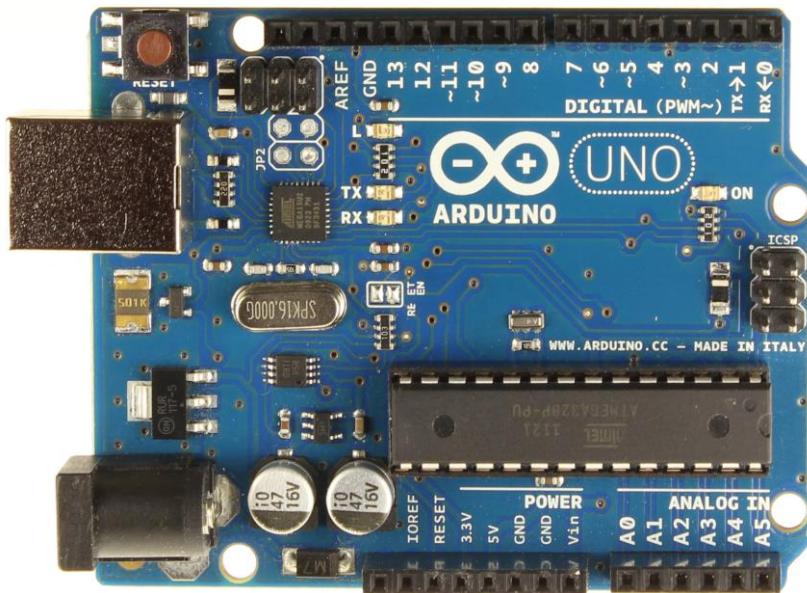


Figura 4 Arduino UNO

* Xbee Shield

Este módulo permite a una placa Arduino comunicarse de forma inalámbrica usando Zigbee. Nos permite comunicarnos hasta 30 metros en interiores y 90 metros al aire libre.

La comunicación se lleva a cabo mediante un módulo de comunicaciones inalámbricas basado en la especificación ZigBee, que definen una tecnología donde la intención es ser simple y de bajo coste comparada con otras tecnologías como pueden ser Bluetooth. Está orientado a conexiones por radio frecuencia, para aplicaciones que necesiten un bajo nivel de transferencia de datos y batería.

Este módulo aparte de permitir la conexión del módulo de comunicaciones ZigBee, también permite la conexión del módulo Bluetooth.

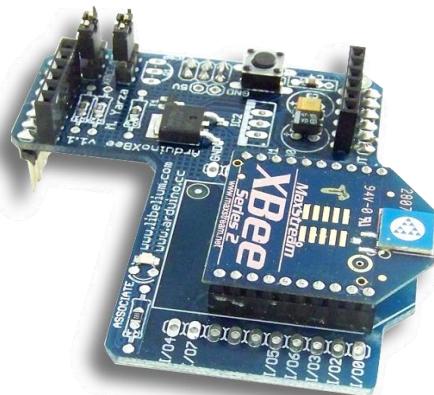


Figura 5 Arduino Shield con módulo Xbee conectado (Zigbee)

* Bluetooth [10]

Es una especificación industrial para redes que nos permite la transmisión de datos mediante radiofrecuencia en la banda de los 2,4GHz. La tecnología está orientada a dispositivos de bajo consumo y un alcance limitado.

Según la potencia de transmisión de los dispositivos, estos se catalogan en distintas clases [11]. Los de clase 1 tienen una potencia máxima de 100 mW (20 dBm) y un alcance de 100 metros, los de clase 2 la potencia es de 2.5 mW(4 dBm) y un alcance de 10 metros y los de clase 3 la potencia es de 1 mW(0 dBm) con un alcance de 1 metro.

Los dispositivos bluetooth pueden ser de distintas versiones, según la versión de las especificaciones que implementen, su velocidad de transmisión varia. Dependiendo de la versión se tiene un ancho de banda de distinto para la transmisión de los datos. La versión 1.2 tiene un ancho de banda de 1 Mbit/s, la versión 2.0 + EDR de 3 Mbit/s, la versión 3.0 + HS y la versión 4.0 de 24 Mbit/s. En este proyecto se usa la especificación 2.0 ya que es la que tiene un soporte más amplio en las versiones de Android con las que se trabajará.



Figura 6 Módulo Blueetooth compatible con Arduino Shield

Otro hardware utilizado

* LCD

Para la realización del prototipo, utilizaremos unas placas LCD de 20 caracteres y 2 líneas. Con un controlador KS0066U. Funciona con una corriente de 5V.

Consta de 16 pins de entrada, 8 para la alimentación, contraste y activación del dispositivo, y 8 entradas para la interactuar con el BUS de comunicación.

Este display está conectado a la estación base y nos muestra información sobre el estado de la estación, como puede ser el nivel de batería, así como mostrar los avisos recibidos. En una primera versión del prototipo se usaba como monitor para ver los avisos en el sistema portable. (En el capítulo 5 se explican las diferentes versiones del prototipo)

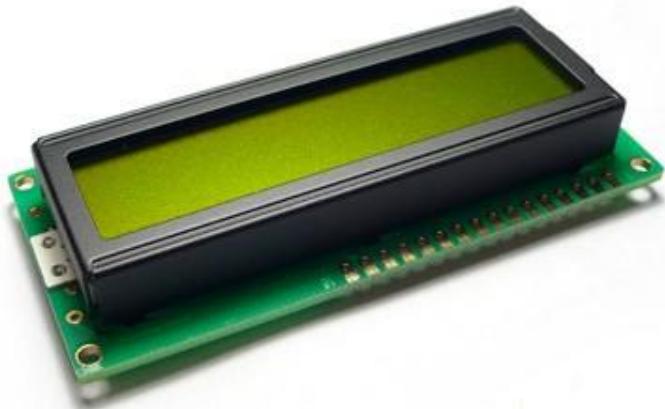


Figura 7 Modelo LCD utilizado

* Motor

El motor utilizado, con unas dimensiones muy reducidas, consigue un régimen de 10000 rpm, con un voltaje de funcionamiento de 3V y 75mA de potencia.

Al recibir una señal de aviso se pensó en la utilización de un pequeño motor, que transmitiera una vibración al usuario. Para el montaje se usa un pequeño interruptor de carga, que unirá nuestro prototipo con el motor.

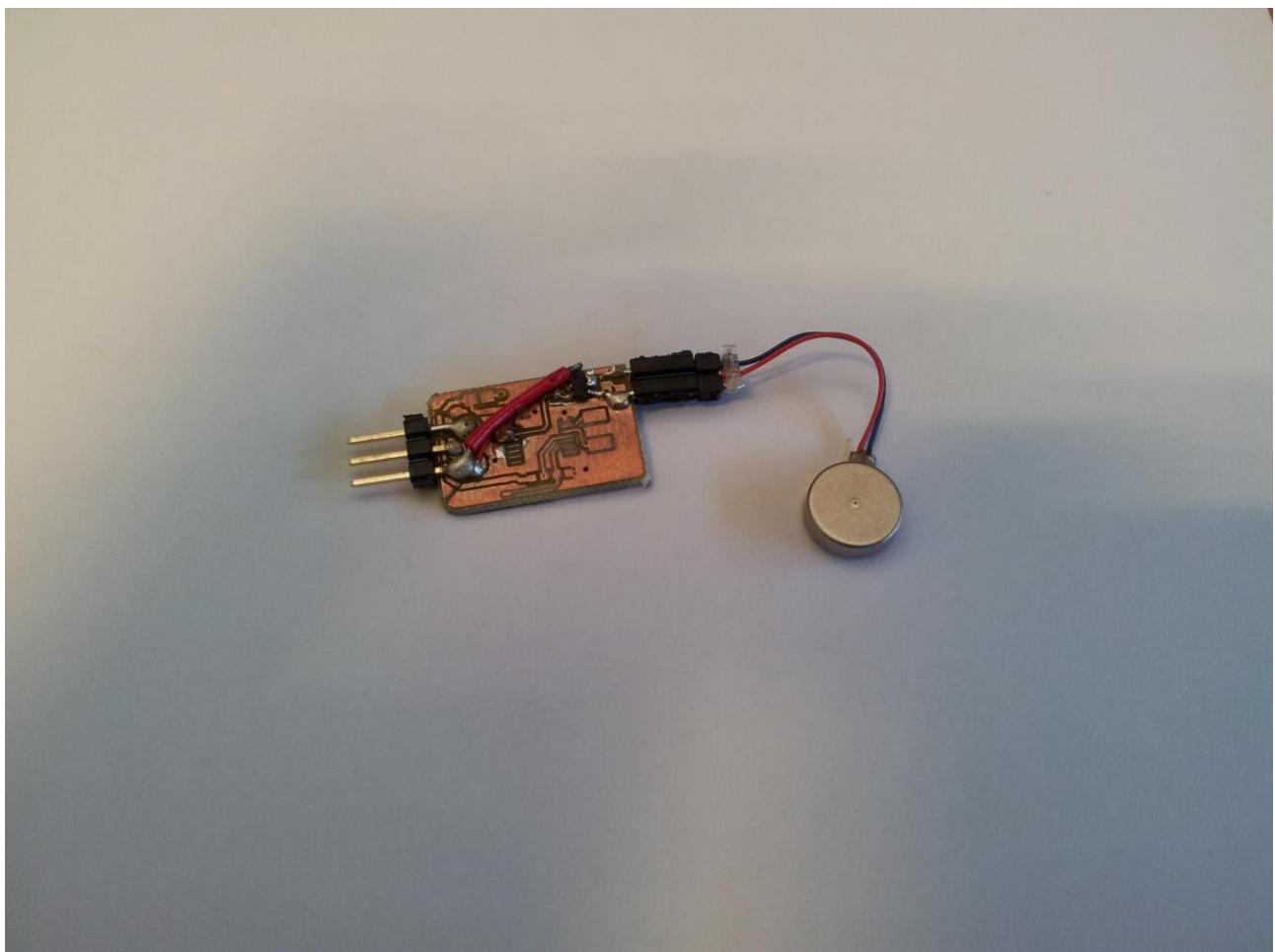


Figura 8 Motor y circuito asociado

4.2. Software

La plataforma Arduino nos permite trabajar con distintos lenguajes de programación. Tiene mucha variedad de librerías para interactuar entre el código de la plataforma y el lenguaje seleccionado para programar. La propia plataforma tiene un lenguaje propio de programación, el cual permite aprovechar las características de la placa. Como ya comentamos anteriormente este lenguaje usado por Arduino, es una implementación del lenguaje Wiring, y está basado en C/C++. Está lincado a través de la librería AVR libc [12] y permite el uso de cualquiera de sus funciones.

4.2.1. Android

Andorid [13], es un sistema operativo para dispositivos móviles y tablets desarrollado por la Open Handset Alliance y liderado por Google bajo un modelo de desarrollo FOSS (Free Open Source Software), que permite a los usuarios finales modificar, estudiar y mejorar el diseño del software mediante la disponibilidad del código fuente.

El sistema se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones sobre un núcleo de las bibliotecas de Java sobre una máquina virtual Dalvik [14] con compilación en tiempo de ejecución.

El núcleo del sistema es un kernel de linux sobre el que se hace correr el resto de aplicaciones. Al usar un núcleo muy adaptable, nos permite hacerlo funcionar en una gran variedad de dispositivos

Es un proyecto muy activo y actualmente está disponible la versión 4.2 del sistema. Más adelante se explican las mejoras de cada versión.

Breve historia

En octubre de 2003, Andy Rubin, Licenciado en Ciencias Informáticas, creó una pequeña empresa en Palo Alto, California, llamada Android Inc. Antes de su aventura en la creación de la empresa, estuvo varios años trabajando en Microsoft y Apple.

En agosto de 2005, dos años después de crear la empresa, Google la adquiere, y Andy pasa a formar parte de la plantilla de Google, a día de hoy ostenta el cargo de Vicepresidente de Ingeniería, supervisando el desarrollo de Android.

El 5 de noviembre de 2007 se formó la Open Handset Alliance, un consorcio de varias compañías (Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, T-Mobile y Texas Instruments) creado con el fin de desarrollar estándares abiertos para dispositivos móviles y ese mismo año, el 12 noviembre del 2007, se libera una versión beta del SDK para finalmente lanzar el primer teléfono con Android, el HTC Dream, el 22 de octubre del 2008. Posterior al lanzamiento nuevas compañías entran a formar parte del consorcio [15].

Dispositivos utilizados

En una segunda fase del proyecto se decide añadir un dispositivo Android, de esta forma se podrá usar la pantalla del dispositivo Android para mostrar los avisos que lleguen al usuario. Para hacer las pruebas se usa un Smartphone de la marca *HTC*, modelo *Dream*.

Este dispositivo lanzado en junio de 2009 fue el primer Android del mercado. Con una pantalla de 3,2 pulgadas y una resolución de 320x480 pixels. Tiene una memoria de 192 MB y un procesador Qualcomm 528Mhz ARM 11. La versión de bluetooth que implementa es la 2.0.

Al ser el primer dispositivo que salió al mercado con el sistema operativo Android, era el candidato perfecto para realizar las pruebas de compatibilidad con versiones antiguas del sistema. Actualmente está funcionando con la versión 2.2.



Figura 9. Vista del HTC Dream

Para realizar pruebas con nuevas versiones del sistema Android, se usa un terminal Galaxy S2 del fabricante Samsung. Este modelo aparece en abril de 2011 y actualmente está funcionando con la versión 4.1 del sistema Android.

Con una pantalla de 4,3 pulgadas y 480x800 pixels, el cambio respecto al dispositivo anterior es notorio. La memoria del dispositivo es de 1GB y un procesador Exynos Dual-core 1,2Ghz Cortex-A9. La versión que implementa del bluetooth es la 3.0

Con este dispositivo se quería probar que la implementación del código para la versión 2.2 del sistema Andorid fuera compatible con los sistemas posteriores.



Figura 10 Vista del Samsung Galaxy S2

4.2.2 Arduino IDE

La primera versión, 0001 [17], es liberada el 25 de agosto de 2005. Desde entonces, cada pocos meses han ido apareciendo nuevas versiones.

En cada revisión del IDE se van añadiendo nuevas funcionalidades y mejoras, así como un intento de hacer el IDE lo más universal posible para poder ser usado tanto en Windows como en Mac y GNU/Linux.

Se van añadiendo soporte para los nuevos dispositivos Arduino a medida que aparecen en el mercado, de esta forma se da total soporte a los nuevos microcontroladores desde el momento que están disponibles. También aparecen nuevas librerías y mejoras de las existentes.

En la figura siguiente, se puede ver el aspecto del entorno de desarrollo Arduino, en donde se aprecia un menú desplegable con los distintos tipos de placas que hay en la actualidad, cada una con su microcontrolador correspondiente. Para probar el código, solo se deberá de seleccionar la placa con la que estemos trabajando y el propio IDE carga el código en el microcontrolador.

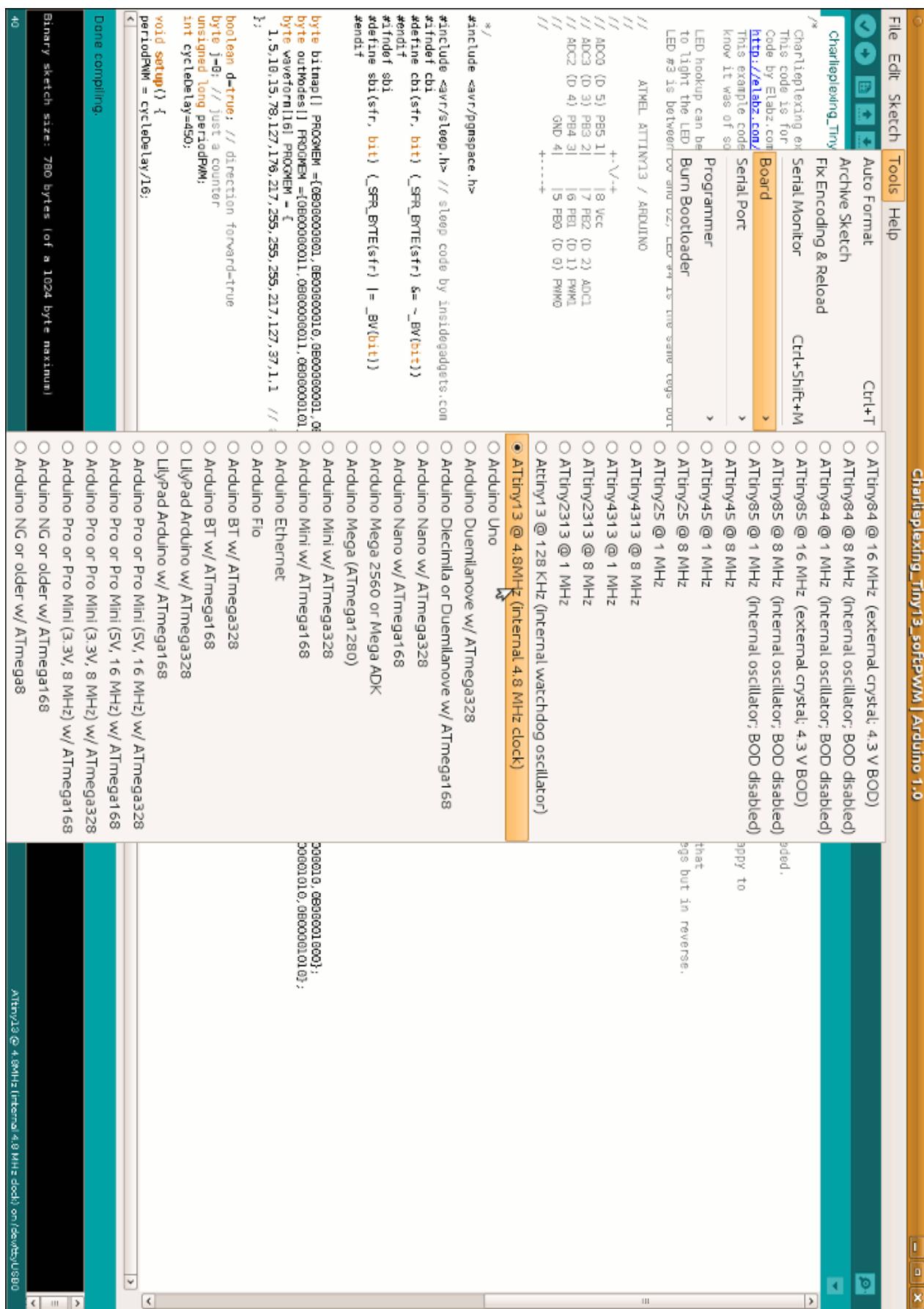


Figura 11Vista IDE Arduino

4.3 Tecnología de comunicaciones

4.3.1. Comparativa Bluetooth vs Zigbee

En la actualidad hay una gran variedad de tecnologías inalámbricas. Todas ofrecen unas características distintas como puede ser la velocidad de transmisión o el alcance entre dispositivos. A la hora de escoger una tecnología sin hilos, debemos saber qué características necesitamos y las ventajas e inconvenientes de cada tecnología.

La siguiente tabla muestra una comparativa de las distintas tecnologías wireless. En ella aparecen el alcance de cada tecnología y la velocidad de transferencia.

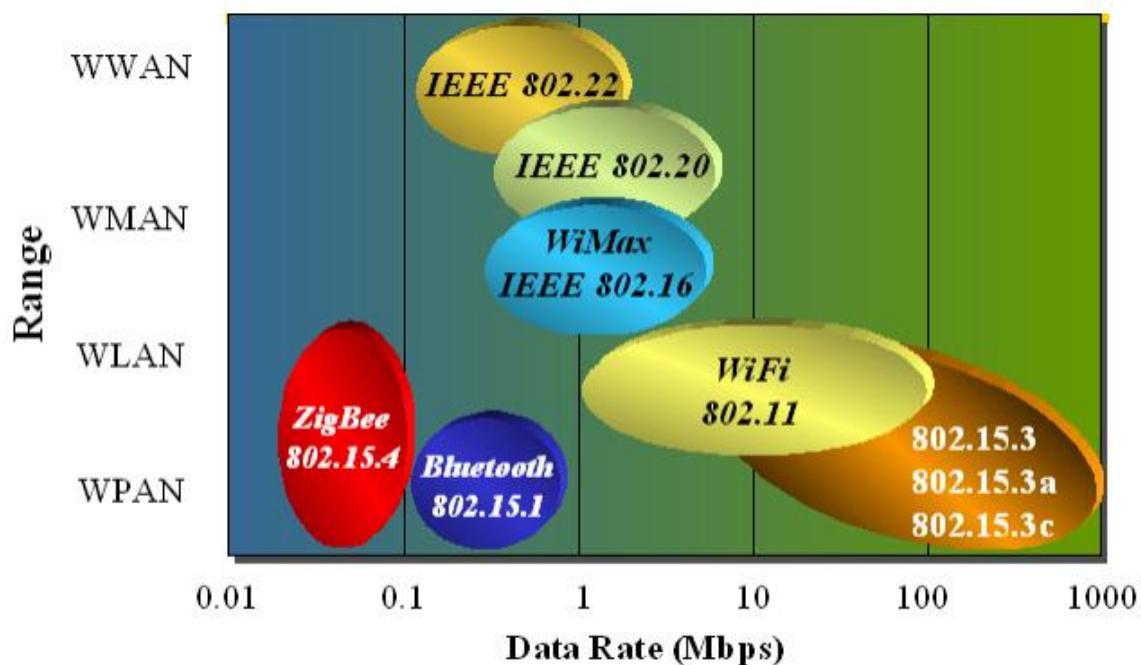


Figura 12 Comparativa tecnología wireless [18]

En este apartado, nos centraremos en comentar dos de estas tecnologías, que ofrecen unas características bastante parecidas.

En la actualidad, el sistema de comunicación bluetooth está presente en la mayoría de dispositivos móviles para facilitar la conexión con el mundo que nos rodea. Se calcula que hay más de 1000 millones de unidades en el mundo con esta tecnología y sigue en aumento.

Unos años más tarde de la aparición de Bluetooth, un consorcio de empresas presenta un nuevo estándar de comunicación a corto alcance, cuya característica más importante es su bajo consumo y un coste reducido. A esta nueva tecnología se la conoce como ZigBee. En la actualidad son más de 100 las empresas que componen esta alianza, algunas de las cuales son: *Texas instruments, Philips, Freescale* [19]

A diferencia del Bluetooth, la tecnología ZigBee ofrece un ancho de banda más pequeño con lo que limitamos su uso a solo mensajes de texto. Por el contrario, Bluetooth nos ofrece la posibilidad de transmitir video o audio.

La mayor diferencia entre ZigBee y bluetooth es el concepto de red. Mientras que ZigBee usa el concepto LAN (Local Area Network), Bluetooth usa el concepto PAN (Personal Area Network). La principal diferencia entre estos modos de red, es la cantidad de dispositivos que pueden conectarse entre sí. Bluetooth solo admite 7 nodos conectados con un dispositivo maestro, con ZigBee podemos llegar a conectar hasta 65535 dispositivos, con distintos tipos de tipología de red: en forma de estrella, como en Bluetooth; en forma de árbol, donde los dispositivos conectados cuelgan de otro dispositivo hasta llegar al maestro, etc.

Otro aspecto a tener en cuenta es la duración de las baterías. Con la tecnología ZigBee la vida de la batería tiene una estimación de 2 años, mientras que en Bluetooth, la vida de batería es de pocos días.

En cuanto a la seguridad, ZigBee puede usar cifrado AES 128 bits [20], que permite autenticar y cifrar las comunicaciones. Además existe un elemento en la red llamado Trust Center, que proporciona un mecanismo de seguridad en el que se utilizan dos tipos de claves de seguridad, la clave de enlace y la clave de red.

4.3.2 I2C

Aparte de la comunicación inalámbrica, otro de los protocolos que se han utilizado en el proyecto es el I2C [21]. Este protocolo servirá para realizar la comunicación entre los dos módulos Arduino que realizan la función de recibir a través de la tecnología ZigBee y enviar a través de Bluetooth.

A continuación se puede leer una breve historia de este protocolo de comunicaciones así como sus características.

Para la conexión de múltiples dispositivos a un microcontrolador, se conectaban de forma individual a los pines de entrada, de esta forma las placas terminaban con una gran cantidad de pistas y conexiones, lo que originaba un sobrecoste en la fabricación y ocasionaba problemas de interferencia o ruido entre las pistas.

A principios de los años 80, el fabricante Phillips diseñó un nuevo sistema de comunicaciones para dispositivos que estaba incorporado en el mismo circuito. El nombre de este nuevo bus de comunicación, Inter IC (Inter Integrated Circuits), llamado I2C o IIC.

Inicialmente la velocidad de transferencia de datos se definió en 100 kbit per segundo, la mayoría de aplicaciones no necesitaban más velocidad de transferencia. Para los dispositivos que necesitaran más velocidad, se introdujo un modo *fastmode*, con una velocidad de transferencia de 400 kbit por segundo. Ya en 1998 se realizó un nuevo aumento de velocidad de transferencia con el modo de alta velocidad, ofreciendo hasta 3.4 Mbit por segundo. Recientemente se ha propuesto un nuevo modo de transferencia llamado *fast mode plus*, que estaría entre el *fast mode* y el de alta velocidad.

Las características más significativas de este bus de comunicaciones son:

- Solo necesita dos líneas para realizar la comunicación.
- Para permitir la conexión entre distintos dispositivos, se diseñó un método de maestro/esclavo para realizar la comunicación. Cada dispositivo tiene un código de dirección que se selecciona mediante software, lo que hace que la relación entre los dispositivos sea permanente.

- A diferencia del RS232, no es necesario tener un ancho de banda estricto ya que el propio maestro genera un pulso de reloj.
- Tiene un sistema de colisiones implementado, lo que facilita la conexión de múltiples maestros.

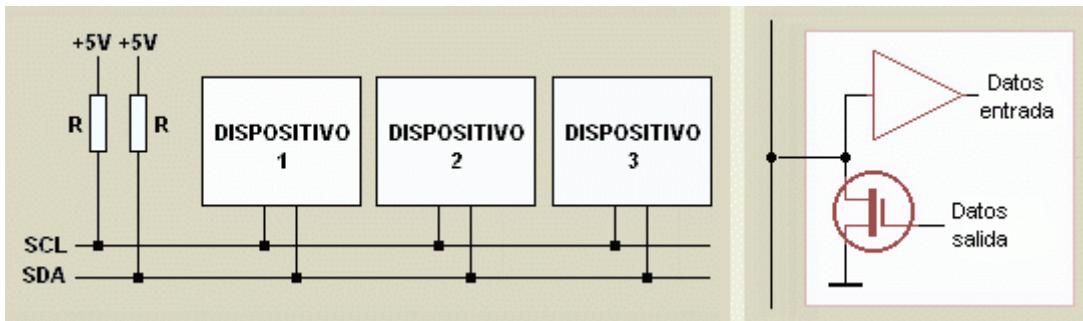


Figura 13 Esquema conexión I2C

En principio la cantidad de dispositivos que pueden conectarse no tiene límite siempre que la capacidad máxima no supere los 400pF (PicoFaradio).

El sistema tiene las siguientes señales, SCL (System clock), SDA (System Data) y GND (Masa). Las dos líneas de conexión son bidireccionales, por tanto pueden operar tanto de transmisor como de receptor de datos. Por eso todos los dispositivos se pueden considerar tanto maestros como esclavos, el dispositivo que realizará la función de maestro es el que inicia la transmisión y genera la señal de reloj (SCL).

Los dispositivos pueden saber que el bus está libre cuando tanto la señal SCL y SDA están activas. La transmisión de datos y direcciones se realiza mediante palabras de 8 bits. El primer byte que se envía, tiene 7 bits que componen la dirección del destinatario, a continuación se manda un bit de ACK reconociendo la solicitud de envío y que puede comunicarse.

Capítulo 5

Análisis de requerimientos

Después de explicar la tecnología que se utilizará en la implementación de este proyecto, seguidamente se explicarán los requisitos que ha de cumplir el proyecto para llevarse a cabo.

Se está buscando una forma de facilitar las cosas a un colectivo de personas con una discapacidad, en consecuencia, deberemos de crear el prototipo en base a este colectivo. En nuestro caso en particular, nuestro actor principal será una persona con problemas de oído. Todo el sistema estará pensado para que esta persona pueda interactuar con el sistema teniendo en cuenta su estado.

En la siguiente figura se puede ver una imagen de una casa, con distintas habitaciones, donde los emisores transmitirán hacia la base receptora y esta se comunicará con el dispositivo receptor que llevará el usuario. La base central es la que tiene los dos símbolos, el de ZigBee y el de bluetooth.



Figura 14 Mapa de situación dispositivos

De esta forma todas las señales deberán de ser visuales o físicas para que puedan ser entendidas. En un principio se pensó en la utilización de un dispositivo estilo reloj para que el usuario pudiera conocer de forma rápida lo que ocurría en la casa. Sin embargo, pensando que en la actualidad mucha gente ya dispone de Smartphone, se pensó en utilizar esta tecnología para el desarrollo del proyecto, de esta forma se usan las ventajas de disponer de este sistema para realizar todas las tareas de visualización de los mensajes. Con el uso del Smartphone, se consigue que sea igualmente cómodo para el usuario, que era otro de los requisitos del proyecto inicial.

5.1. Diseño

El proyecto se divide en dos fases. Una fase inicial en la que se quiere realizar un prototipo de un dispositivo de tamaño reducido que sea cómodo de llevar para el usuario. Esta primera fase constará de dos tipos de dispositivos, los dispositivos emisores y el receptor.

El dispositivo emisor, conectado a un electrodoméstico y timbre de la casa, estará en modo de espera hasta que sea activado. En ese momento enviará un aviso a nuestro receptor. Este aviso tendrá una cadena de datos en donde está el identificador del aviso.

Para hacerlo más modular, será el emisor el que lleve el identificador del aviso producido, de esta forma se podrá ampliar el número de emisores que hay sin tener que modificar nada del receptor.

El receptor, en esta primera fase, consistirá de un LCD conectado a la placa Arduino. Siempre estará esperando un aviso de uno de los emisores. En el momento que llegue un aviso, lo mostrará por pantalla y activará un pequeño motor que vibrará para avisar al usuario de este nuevo aviso. Una vez pasado un tiempo se borrará el aviso de pantalla hasta la llegada de un nuevo aviso. El dispositivo receptor también incluirá unos leds para indicar el estado de batería, a medida que se descarga la batería se irán apagando estos leds, hasta que sea necesario reemplazarla, momento en el que aparecerá un mensaje por pantalla.

En la siguiente figura se puede ver los estados del sistema emisor y receptor en esta primera fase del proyecto.

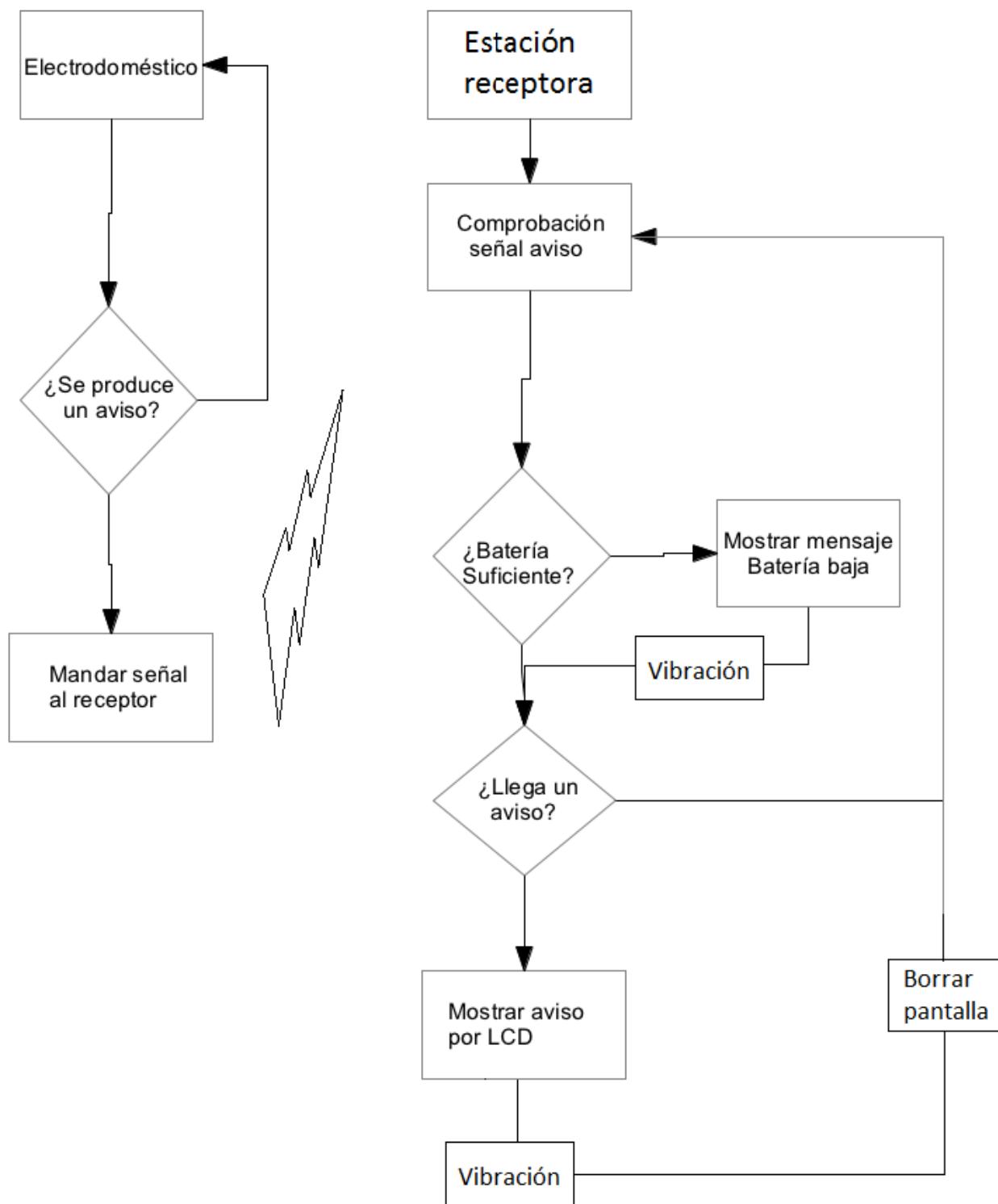


Figura 15 Diagrama primera fase

En una segunda fase del prototipo, el dispositivo receptor, hace las funciones de conversor entre la tecnología ZigBee y Bluetooth, de esta forma se introduce como dispositivo receptor final un Smartphone que llevará el usuario, aprovechando la tecnología de los smartphones de vibración y visualización de los mensajes.

A continuación se muestra un esquema general de la aplicación con la utilización de Smartphone. Este esquema representa la segunda fase del proyecto, donde puede apreciarse un dispositivo emisor conectado a, por ejemplo, un electrodoméstico; la base receptora, que está constantemente comprobando si hay algún aviso, y en caso de recibir un nuevo aviso, enviarlo al dispositivo de visualización, en este caso, un Smartphone.

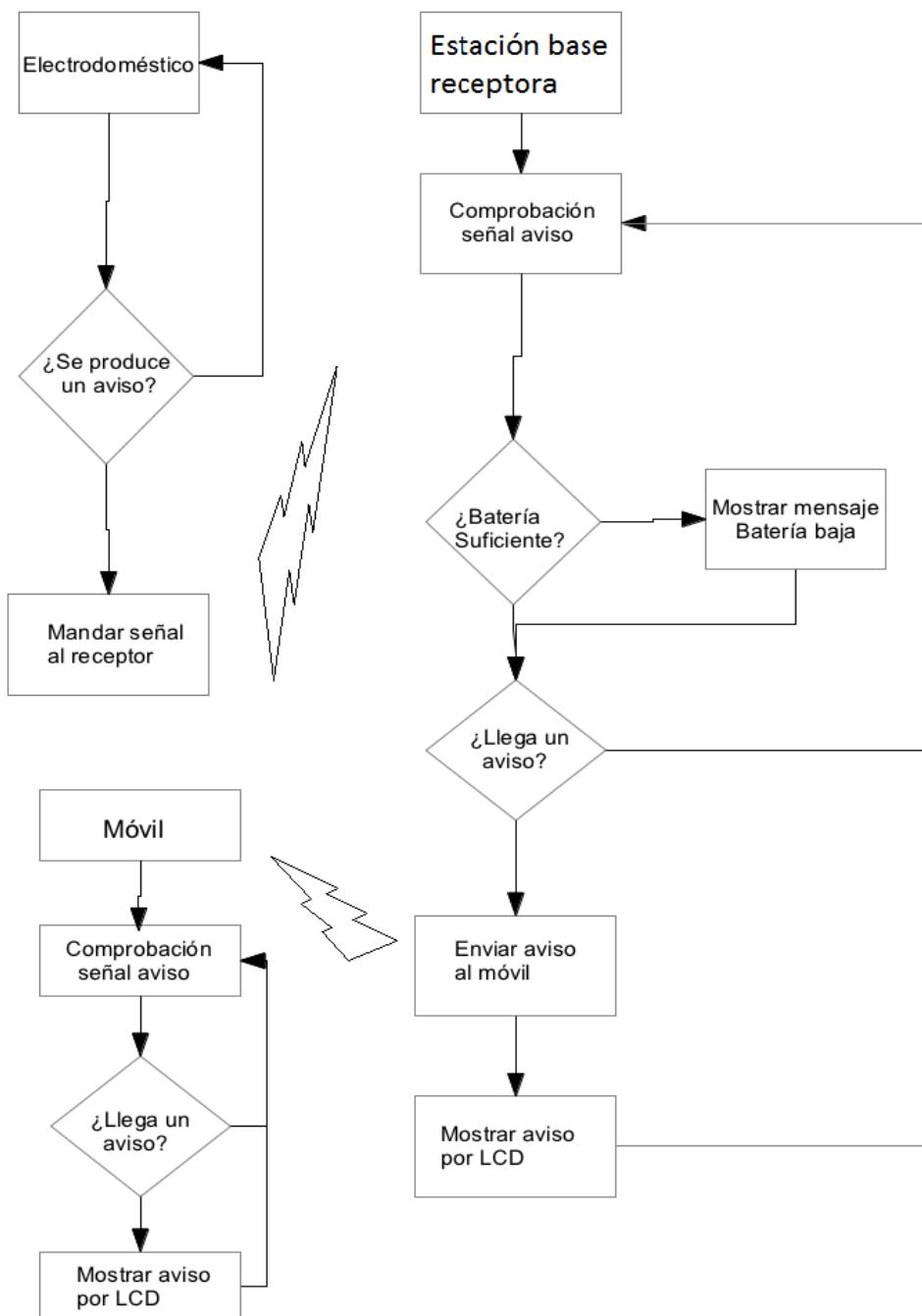


Figura 16 Esquema general de la aplicación con Smartphone

En la siguiente figura puede verse un diagrama de flujo del envío de los avisos, cuando se recibe un aviso se manda por ZigBee y el receptor responderá un mensaje de recibido, si pasado un tiempo no recibe la confirmación volverá a mandar el aviso y pasará al estado de recibir nuevos avisos.

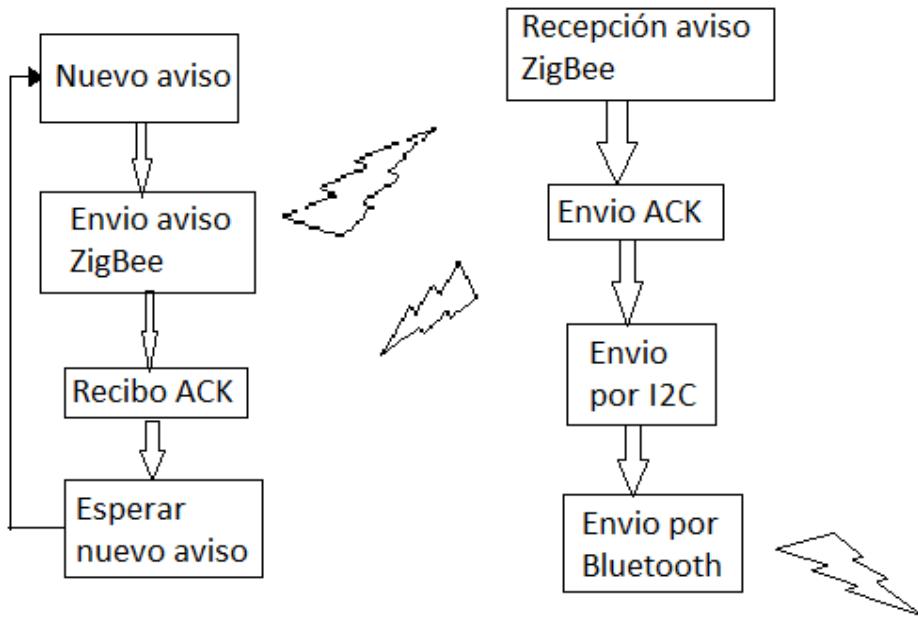


Figura 17 Diagrama de flujo del envío de avisos

5.2. Implementación

Como ya se ha comentado en el diseño el proyecto se ha realizado en dos fases distintas, una primera fase en la que se quiere conseguir la comunicación por ZigBee entre dos dispositivos Arduino. El primer prototipo consta de un módulo Arduino diecimila y un módulo Arduino UNO, ambos con el módulo Xbee shield y ZigBee conectado para las comunicaciones, una plataforma hará de emisor y la otra de receptor. El receptor tendrá un LCD conectado para la visualización de los avisos que lleguen, y para informar al usuario de los mensajes que el propio sistema pueda mostrar, como puede ser un mensaje de nivel bajo de batería. Mediante unos leds, conseguiremos dar una visualización rápida del estado de la batería y en caso de batería baja un led rojo informará de la necesidad de cambiarla o recargarla. Por pantalla también aparecerá un aviso de necesidad de cambiar batería. Para dar los avisos al usuario se conecta un pequeño motor, que da una señal por vibración al usuario. En la figura siguiente se muestra el esquema de conexiones entre la placa Arduino y los componentes: LCD, Motor y los distintos Leds.

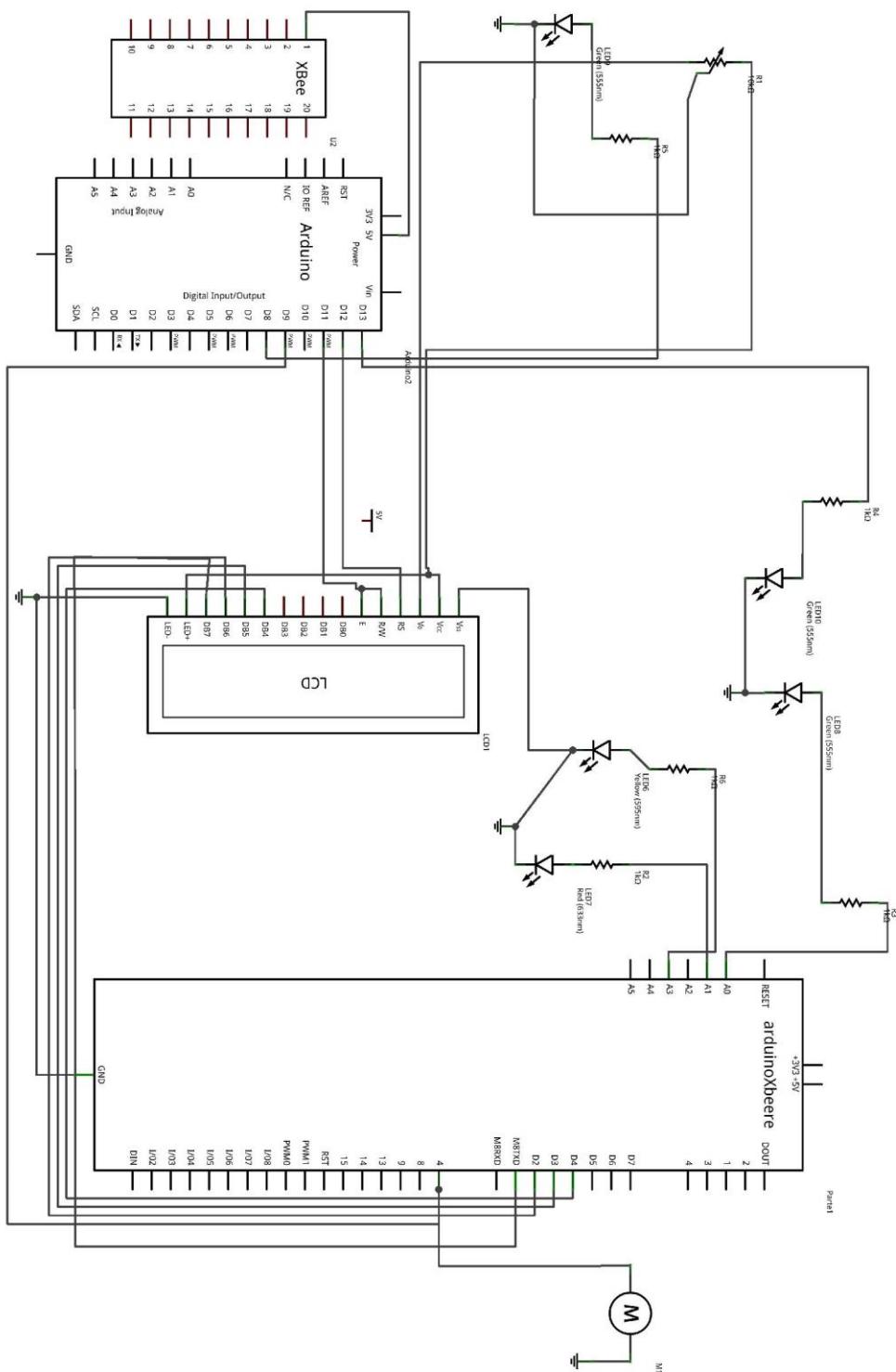


Figura 18 Esquema fase 1

La conexión del LCD con la plataforma debe de hacerse con los pines correctos para poder activar el display y posteriormente poder escribir datos. Los pines 1 y 2 son los de masa y voltaje de entrada. El pin 3 conectado a un potenciómetro, servirá para ajustar la iluminación del LCD, dependiendo del valor de resistencia que se le dé al potenciómetro, la imagen saldrá más o menos oscura. El pin 4 sirve para indicar si se quiere borrar, mover el cursor o por el contrario, queremos escribir datos. El pin 5 indicará si vamos a leer o escribir valores en el display, el pin 6 indica si el display está o no activo. Del 7 a 14 son los pines de datos, dependiendo del número de pines que se usen para escribir valores en el display, se puede usar 4 pines, del 14 al 11 que corresponden a las primeras 4 posiciones del bus de datos D0-D3 o usar los 8 pines del bus de datos, D0-D7. El pin 15 y 16 sirven para activar la luz de fondo de los LCD.

Se crea un sistema de batería para la alimentación de la plataforma receptora, ya que uno de los requisitos es que sea portátil y no dependa de estar conectado a una toma de corriente. Al necesitar una alimentación de entre 5V y 12V, se decide por conectar 6 baterías AAA para suministrar la energía necesaria para el funcionamiento de la estación receptora.

Para la estación emisora, como estará conectado en algún elemento fijo de la casa, podrá alimentarse mediante una toma de corriente o del propio dispositivo al que esté conectado.

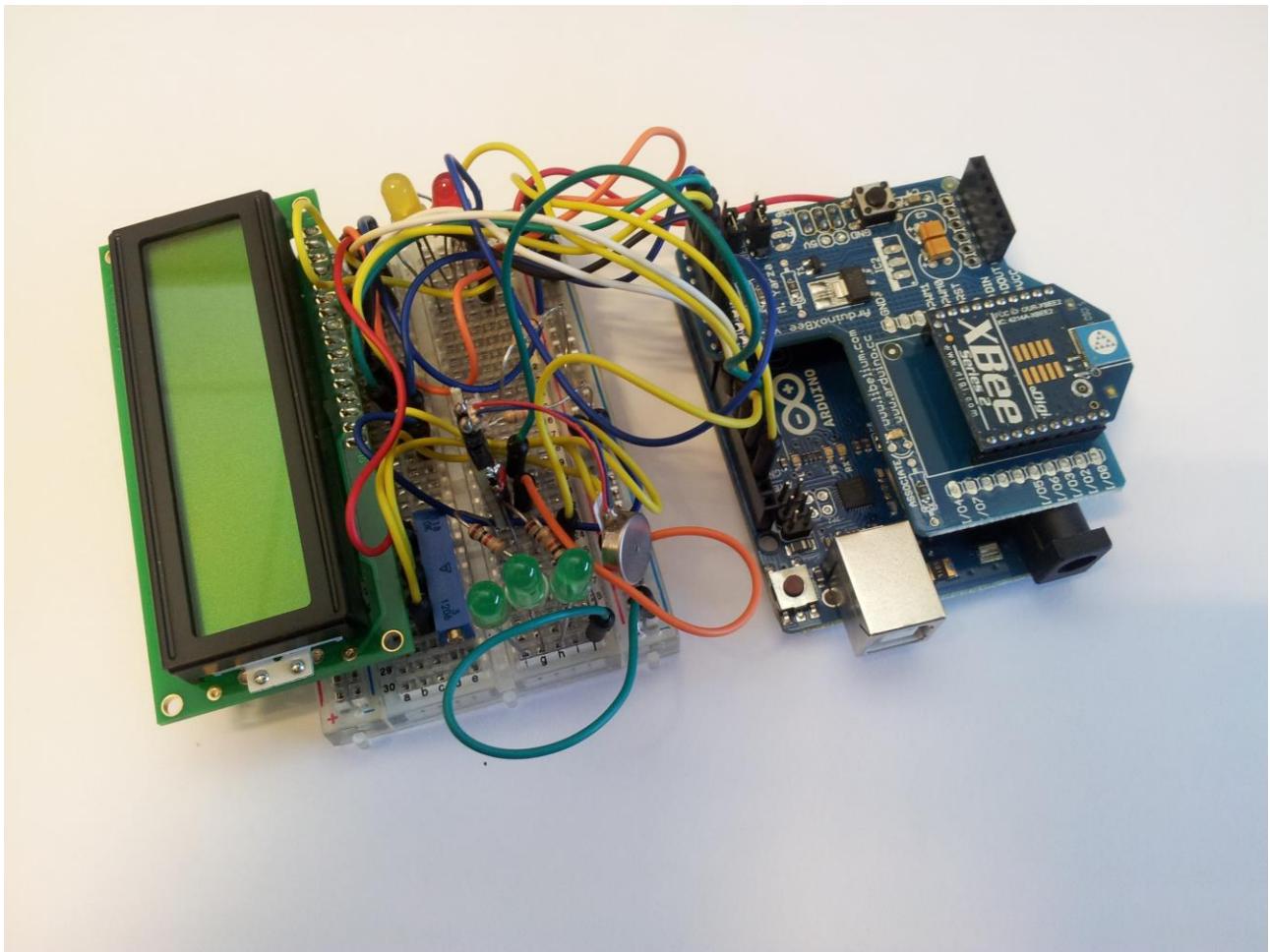


Figura 19 Prototipo fase 1. LCD, motor y leds

Una segunda fase del desarrollo consiste en la introducción de un elemento externo a la plataforma para la visualización de los mensajes, para ello se utiliza un Smartphone con tecnología bluetooth. Una vez la estación receptora reciba un mensaje, lo enviará a nuestro Smartphone utilizando un módulo bluetooth conectado a nuestro dispositivo Arduino.

Para los preparativos de esta fase del proyecto, se necesita realizar una conversión de la tecnología inalámbrica porque en la actualidad los Smartphone no tienen implementada la tecnología ZigBee, pero en cambio sí implementan la tecnología bluetooth.

Para realizar esta conversión utilizamos dos dispositivos Arduino UNO. Esta plataforma implementa el bus de comunicaciones I2C, por el cual podemos pasar mensajes entre los dos dispositivos. Mediante este bus de comunicaciones se tienen conectadas las dos plataformas Arduino. Cuando la plataforma con el módulo ZigBee conectado reciba un nuevo aviso, este será enviado por I2C a la otra plataforma, al tener esta segunda plataforma el módulo Bluetooth, mandará el aviso recibido al Smartphone.

Para la conexión del bus I2C, siguiendo las especificaciones hay que conectar entre sí dos entradas/salidas de cada una de las placas arduino y las dos placas deberán de compartir la misma conexión de masa para que se pueda realizar la sincronización. Una vez se consigue enviar un mensaje entre los dispositivos mediante el bus I2C, será necesario comprobar que la información que recibe el módulo ZigBee, llegue correctamente al módulo con el dispositivo bluetooth.

En esta parte utilizamos un tercer dispositivo Arduino, que realizará la función de emisor ZigBee. Este dispositivo puede ser el mismo que en la primera fase, ya que su función es únicamente la de estar esperando un aviso externo para enviarlo a la base receptora.

Conseguida la comunicación entre el emisor y el receptor, y la conexión por el bus de comunicación I2C, se procede a implementar una aplicación para el sistema Android que conecte con el dispositivo bluetooth y reciba los datos que este envía.

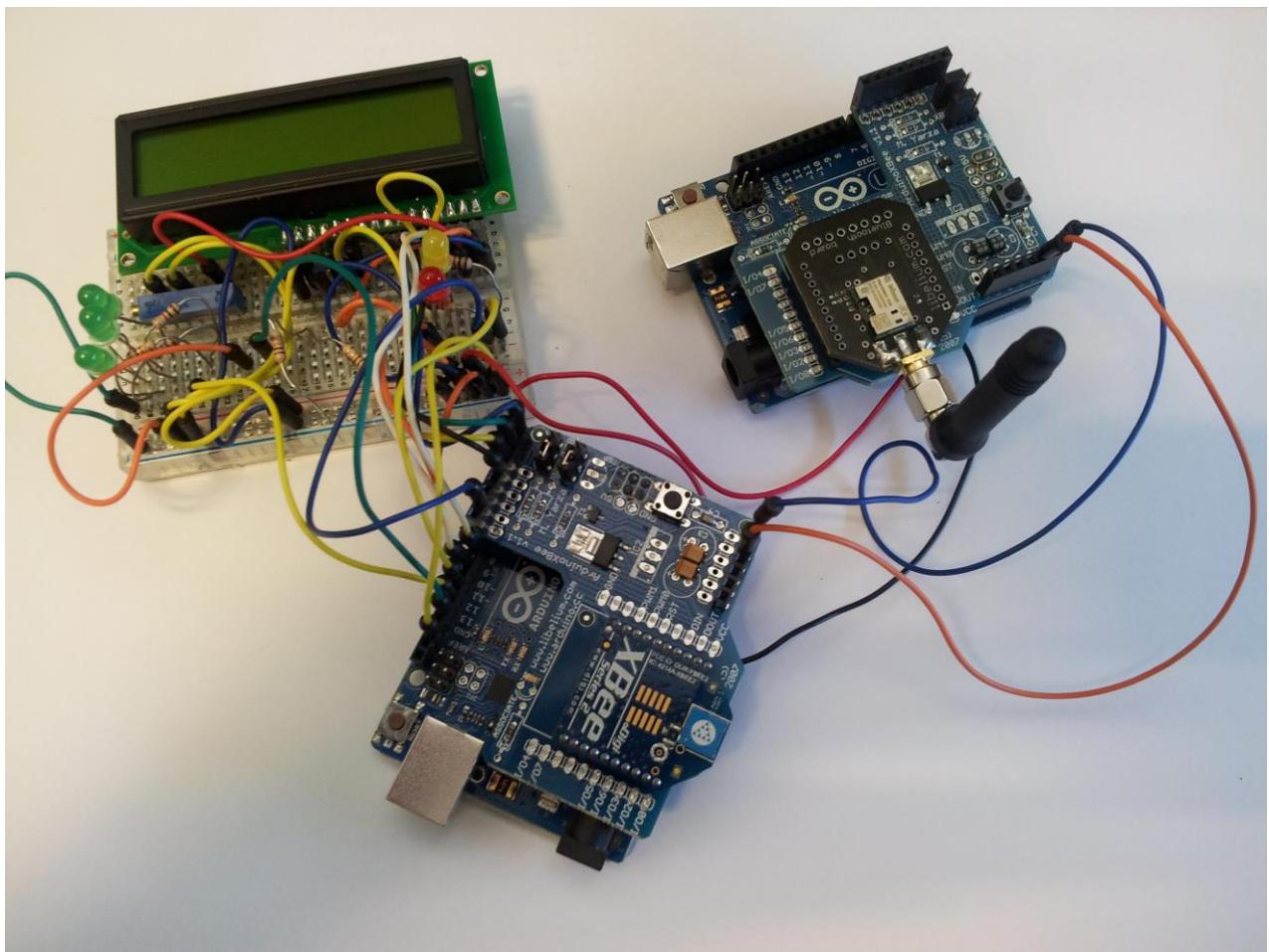


Figura 20 Prototipo fase 2. Conexión del LCD y leds, y la conexión entre módulos por I2C

Para ello se usa el entorno de desarrollo Eclipse que permite trabajar con el SDK de Android de una forma sencilla. Lo primero que hay que hacer una vez instalado Eclipse, es descargarse el Andorid manager de la página de desarrollo de Android, este plugin para eclipse ayuda a instalar todas las librerías necesarias del SDK de Android, así como crear máquinas virtuales para probar las aplicaciones sin tener que usar un dispositivo Android.

El Smartphone utilizado, usa el sistema operativo para dispositivos Android en su versión 2.2.6. La aplicación creada, escanea todos los dispositivos bluetooth de las cercanías y nos realiza una conexión con el dispositivo escogido. Se queda en modo de espera hasta que nos llegue una señal de aviso. Una vez llega un aviso, este es mostrado por pantalla y utilizando el sistema de vibración incorporado por los Smartphone, avisará al usuario.

En el desarrollo de la aplicación se han creado distintas clases para funcionar en el sistema Android. La clase Arduino es la que crea la nueva actividad, tiene el menú inicial, el cual permite activar o desactivar el Bluetooth y buscar un dispositivo Bluetooth para conectarse. Mediante llamadas al sistema se activa o desactiva el bluetooth del Smartphone. Cuando activamos el Bluetooth, automáticamente el dispositivo se hace visible durante 2 minutos por otros dispositivos Bluetooth de la zona.

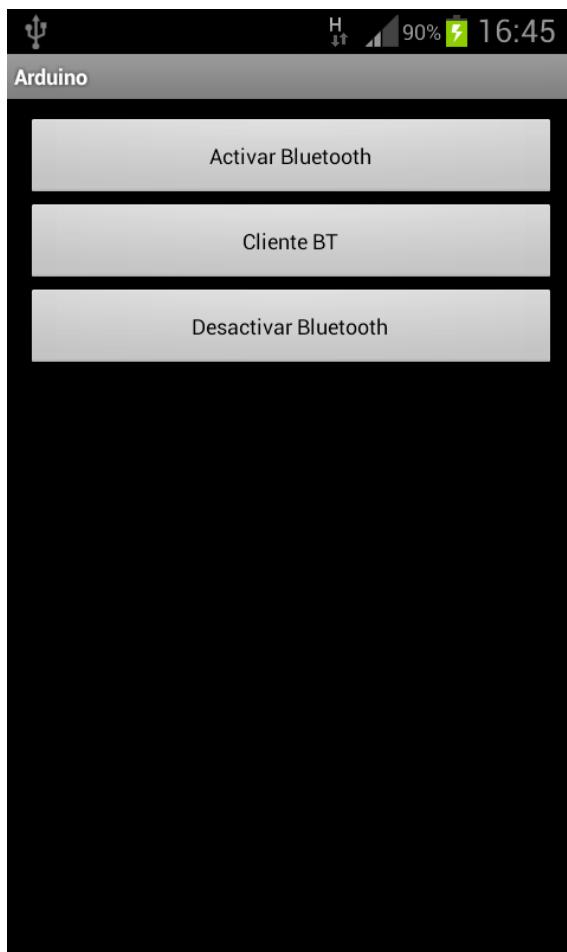


Figura 21 Menu de la aplicación

Cuando se pulsa en buscar dispositivo, se llama a la clase ClientSocket, esta clase lo primero que hará es comprobar si el bluetooth del dispositivo está activo, en caso contrario avisará de que hay que tenerlo activado para poder usar la aplicación. A continuación realiza una llamada a la clase DiscoveryActivity, esta clase nos muestra una pantalla en la que irán apareciendo los dispositivos que se van encontrando. Una vez terminado, se podrá seleccionar el dispositivo al que se quiera conectar. Cuando se selecciona el dispositivo se crea un socket de comunicaciones y se queda a la espera de que llegue un nuevo aviso, mostrándolo por pantalla y activar la vibración del Smartphone, de esta forma avisa al usuario de la llegada de los avisos.

La forma inicial de recepción de los mensajes se hacía mediante un buffer lo que provocaba la pérdida de datos en el momento de escribir, ya que si el dispositivo recibía información antes de inicializar de nuevo el buffer, la cadena de datos resultante era incompleta. Para ello se decidió directamente leer la cadena de datos, y que fuera el propio sistema Android el que realizará la conversión de bytes a cadena de caracteres.

5.2.1 Librerías Arduino

- LiquidCrystal

Esta librería nos permite controlar un dispositivo LCD compatible con el chip Hitachi HD44780 conectado a nuestro Arduino.

Nos ofrece soporte para inicializar el dispositivo y empezar a trabajar con él, ofreciendo la posibilidad de colocar el cursor en una posición de inicio, limpiar la pantalla, mostrar el cursor, mover los caracteres de izquierda a derecha de la pantalla.

Ejemplo uso:

```
LiquidCrystal lcd(12,11,10,5,4,3,2); //Inicialización del LCD, dependiendo de los parámetros, inicializaremos distintas
// conexiones del LCD, permitiendo el uso de 4 o 8 conexiones de datos según el
//numero de entradas usadas. Los valores representan las entradas de la placa
//Arduino. El pin 12, 11 y 10 hacen referencia al pin RS, RW y Enable del LCD.
//los pines 5,4,3,2 hacen referencia a los 4 pines de datos (d4,d4,d6,d7) del LCD.
//LiquidCrystal(rs, enable, d4, d5, d6, d7)
//LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)
//LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)
//LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)

void setup()
{
    lcd.print("hello, world!");
}

void loop() {}
```

- Wire

La librería wire permite la comunicación entre dos o más dispositivos a través del protocolo de comunicación I2C.

Algunas de sus funciones son las de Crear o finalizar una transmisión, escribir, leer o esperar una conexión.

Ejemplo uso:

```
byte val = 0;

void setup()
{
```

```

    Wire.begin(); // Inicializar bus i2c
}

void loop()
{
    Wire.beginTransmission(44); // transmite al dispositivo #44
    Wire.write(val); // Envía el valor byte
    Wire.endTransmission(); // finaliza transmisión

    val++;
    if(val == 64)
    {
        val = 0;
    }
    delay(500);
}

```

- SoftwareSerial

Esta librería permite la comunicación por el puerto serie de la placa Arduino. Aprovechando que el XBee se puede configurar como entrada/salida del puerto serie, podemos usar esta librería para la comunicación entre dispositivos.

Además, permite leer y escribir por el puerto de comunicaciones, esperar una petición de lectura/escritura, configurar la velocidad en la que se realizará la comunicación, dependiendo del dispositivo con el que nos comuniquemos la velocidad de transmisión puede ser de 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 31250, 38400, 57600, y 115200.

Ejemplo de uso de la librería SoftwareSerial:

```

SoftwareSerial mySerial(10,11); // Pin entrada y salida de datos

void setup()
{
    // Abertura de la comunicación serie y espera a que este abierto el puerto
    Serial.begin(57600);
    while(!Serial){
        ;//Solo algun modelo de Arduino lo necesita
    }
    Serial.println("Mensaje Uno!");
    // Configuramos la velocidad de transmisión del puerto
    mySerial.begin(4800);
    mySerial.println("Mensaje Dos");
}

void loop()// Se ejecutará infinitamente
{
    if(mySerial.available())
        Serial.write(mySerial.read());
    if(Serial.available())
        mySerial.write(Serial.read());
}

```

Capítulo 6

Pruebas

En este apartado se explicará las pruebas realizadas al dispositivo. Al ser un dispositivo que siempre está en espera de una actividad, fue probado de distintas formas para conseguir el funcionamiento deseado. Para realizar las pruebas, se conectó a una batería para poder trabajar en distintos ámbitos sin depender de donde se conectara el dispositivo emisor.

6.1. Comprobación perdida de datos en la comunicación

Se realizó una prueba consistente en intentar saturar la estación intermedia, que comunica los dispositivos ZigBee con nuestro terminal Bluetooth. Se realizó un pequeño programa que enviaba una trama a través del módulo ZibBee cada 500 milisegundos, el receptor recibía sin problemas y mostraba la trama correctamente. Una vez recibida la trama el módulo lo enviaba por I2C y quedaba a la espera de ser enviado por el dispositivo Arduino con el módulo Bluetooth al Smartphone. No se apreció ninguna pérdida de datos entre los distintos dispositivos.

6.2. Establecer comunicación entre el módulo arduino y el smartphone

Otra prueba consistía en enviar datos entre el módulo bluetooth conectado a la placa arduino y el receptor Bluetooth del smartphone, comprobando que llegaran de forma correcta. Hubo que cambiar la forma en la que se recibían los mensajes, ya que no se realizaba correctamente la recepción del mismo.

Al usar un sistema de buffer para ir cargando los caracteres recibidos, si el envío terminaba antes de que estuviera llenado el buffer o se recibía algún carácter extra, la impresión por pantalla de los avisos quedaba cortada y no era posible leer correctamente el mensaje. Para ello se decidió leer toda la información recibida y luego tratar esa información para mostrar el mensaje.

6.3. Chequear batería

Comprobación del funcionamiento del sistema de batería. Comprobamos que la carga de la batería es indicada correctamente por el circuito y mostrada por pantalla.

Se probó el sistema con distintas cargas de voltaje, para comprobar cuál era la respuesta, ya que el dispositivo tiene un rango de funcionamiento de distintos voltajes.

6.4. Comprobar versiones de Android

Las pruebas con el sistema Android, aparte de realizarlas con un dispositivo con la versión 2.2, se realizaron con un dispositivo Android 4.1 para comprobar la compatibilidad de la aplicación realizada.

Capítulo 7

Conclusiones

En este punto, se describen las conclusiones que se han obtenido una vez realizado el proyecto.

En primer lugar, podemos afirmar que el prototipo cumple con los requisitos especificados para este proyecto. Se ha conseguido centralizar los avisos enviados por varios dispositivos en una estación receptora. Otro de los requisitos era que el usuario estuviera informado en todo momento de lo que pasa en su entorno sin tener que mirar luces. Además, con la inclusión de la fase 2, se consigue otra forma de interactuar con el dispositivo.

Otra de las conclusiones que se han obtenido es que mediante la tecnología ZigBee se podrían conectar directamente todos los dispositivos al Smartphone si en un futuro implementaran esta tecnología. Esto abarataría el coste de implementación al ahorrarnos la estación intermedia que comunica el ZigBee y el Bluetooth. En caso de que un usuario no tuviera Smartphone, comercializando la fase 1 se tendría un dispositivo totalmente autónomo, para recibir los avisos que se producen dentro de la casa.

A nivel personal, al implementar una aplicación para el sistema Android se ha tenido que aprender un nuevo entorno de desarrollo con lo que se consigue uno de los objetivos personales que era el aprendizaje de nuevas tecnologías. El proyecto también ha servido para ampliar conocimiento en electrónica, ya que hubo que diseñar un circuito para la comprobación de la batería. También ha servido para conocer el mundo de los microcontroladores, ya que nunca había programado para un microcontrolador.

Terminado el proyecto, se entrará a valorar que módulo de Arduino sería el más indicado para nuestro caso, ya que hay distintos módulos de tamaño reducido, con características varias.

7.1. Trabajo futuro

Como trabajo futuro, están apareciendo unos dispositivos del tamaño de un reloj, que usan pantallas de tinta electrónica para la visualización de la información. Una idea sería usar uno de estos dispositivos para la comunicación con el usuario. Sería algo mucho más discreto y cómodo para llevar siempre encima.

En el mercado ya están disponibles dispositivos con estas características como por ejemplo el SmartWatch de Sony [22].

Como se ha comentado anteriormente, al usar la tecnología ZigBee, tenemos unas limitaciones en cuanto a comunicación con distintos dispositivos, al ser una tecnología de uso minoritario. Por esta razón, implementando el sistema con dispositivos wifi, que están mucho más extendidos, se podrían crear nuevas formas de comunicación entre el entorno y el usuario, pudiendo incorporar todo este sistema a la tecnología domótica. En cuanto la tecnología ZigBee esté más implementada, se mejoraría el consumo de los dispositivos.

Bibliografía

[1] Web Arduino [20 de mayo de 2013]

<http://www.arduino.cc/>

[2] Web proyecto Pinguino [13 de enero de 2013]

<http://www.hackinglab.org/pinguino/>

[3] Web proyecto Freeduino [3 de diciembre de 2012]

<http://www.freeduino.org/about.html>

[4] Zigbee [2 de marzo de 2013]

<http://www.zigbee.org/>

[5] Acerca del lenguaje Wiring [19 de diciembre de 2012]

<http://wiring.org.co/about.html>

[6] Web sobre la familia de chips AVR [27 de abril de 2013]

<http://www.atmel.com/products/microcontrollers/avr/default.aspx>

[7] Creative Commons [26 de enero de 2013]

<http://creativecommons.org/about>

[8] Historia Arduino [19 de febrero 2013]

<http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino/0>

[9] Información microcontrolador ATmega 168 [7 de mayo de 2013]

<http://www.atmel.com/devices/atmega168.aspx>

[10] Especificaciones de la tecnología bluetooth [11 de mayo de 2013]

<http://www.thewirelessdirectory.com/Bluetooth-Overview/Bluetooth-Specification.htm>

[11] Clases Bluetooth [30 de Abril de 2013]

<http://www.bluetooth.com/Pages/Basics.aspx>

[12] AVR Libc [15 de mayo de 2013]

<http://www.nongnu.org/avr-libc/>

[13] Android [9 de abril de 2013]

<http://www.tech2crack.com/history-android/>

[14] Dalvik [19 de marzo de 2013]

http://show.docjava.com/posterous/file/2012/12/10222640-The_Dalvik_Virtual_Machine.pdf

[15] Historia de Android [4 de mayo de 2013]

<http://www.theverge.com/2011/12/7/2585779/android-history>

[16] Versiones android [19 de enero de 2013]

<http://source.android.com/source/overview.html>

[17] Versiones IDE Arduino [23 de febrero de 2013]

<http://arduino.cc/en/Main/ReleaseNotes>

[18] Comparativa wireless [6 de mayo de 2013]

http://www.specifications.nl/zigbee/zigbee_UK.php

[19] Miembros de la alianza ZigBee [29 de mayo de 2013]

<http://www.zigbee.org/About/OurMembers.aspx>

[20] Cifrado ZigBee [1 de junio de 2013]

<http://www.digi.com/technology/rf-articles/wireless-zigbee>

[21] Bus I2C [16 de mayo de 2013]

<http://www.i2c-bus.org/>

[22] Especificaciones Smartwatch [15 de diciembre de 2012]

<http://www.sonymobile.com/es/products/accessories/smartwatch/specifications/>

Resumen. El trabajo expuesto en la presente memoria tiene como objetivo la creación de un prototipo de avisos para gente sorda. El sistema se encargará de facilitar la interactuación entre una persona con problemas auditivos y los señales sonoras que pueden hallarse en una casa.

El prototipo se ha desarrollado a partir de la plataforma hardware Arduino, un Smartphone con sistema operativo Android y la tecnología de comunicaciones inalámbrica Bluetooth y ZigBee.

Resum. El treball exposat en aquesta memòria té com a objectiu la creació d'un prototip d'avisos per a gent sorda. El sistema s'encarregarà de facilitar la interactuació entre una persona amb problemes auditius i els senyals sonors que poden trobar-se en una casa.

El prototip s'ha desenvolupat a partir de la plataforma maquinari Arduino, un Smartphone amb sistema operatiu Android i la tecnologia de comunicacions sense fils Bluetooth i ZigBee.

Abstract. The work described in this report is intended to create a prototype of notices for deaf people. The system is responsible for facilitate the interaction between a deaf person and sound signals that can be found in a house.

The prototype has been developed from the Arduino hardware platform, a smartphone with Android operating system and Bluetooth wireless communications technology and ZigBee.