

Comunicación a través del Protocolo Zigbee con Nios II

Játiva José, Cabello Lissette, Ponguillo Ronald Ing.
Facultad de Ingeniería Eléctrica y Computación.
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador
{jjativa, lisacabe, rponguil}@espol.edu.ec

Resumen

El presente trabajo pretende dar a conocer la versatilidad de los sistemas embebidos aplicados en las FPGAs para la elaboración de proyectos que se acoplen a cada una de las necesidades y aplicaciones que requiere el usuario. En nuestro caso implementaremos el protocolo Zigbee, basado en comunicaciones inalámbricas de área personal con bajo consumo de energía y baja tasa de transmisión de datos, para que, junto al sistema embebido en la FPGA de Altera, se pueda formar un sistema de seguridad eficiente.

Para la programación y procesamiento de la información utilizamos la tarjeta DE2 de Altera, la cual cuenta con un sin número de elementos, entre esos se encuentra el dispositivo FPGA Cyclone II que ofrece memorias embebidas, también da la posibilidad de generar el procesador embebido Nios II gracias a las herramientas de SOPC Builder en Quartus II y así permitimos implementar una variedad de proyectos de diseño para el desarrollo de sistemas digitales sofisticados.

El trabajo "Comunicación a través del protocolo Zigbee con Nios II" tiene como objeto complementar estas dos tecnologías en una red de seguridad, debido a que en la actualidad por el alto índice de inseguridad estos sistemas son muy utilizados en hogares, oficinas y locales comerciales, teniendo así un alto índice de crecimiento primordialmente en el ámbito de la domótica.

Palabras Claves: Comunicaciones Inalámbricas, Zigbee, FPGA, Nios II.

Abstract

This paper aims to show the versatility of embedded systems implemented in FPGAs for the development of projects that fit every application the user requires. In our case we will implement the Zigbee protocol, based on personal area wireless communications with low power consumption and low data transmission rate, so that, together with the system embedded in the FPGA from Altera, can form an efficient security system.

For programming and data processing the Altera DE2 board, which has a number of elements, such is the Cyclone II FPGA offering embedded memories, it also gives the possibility to generate the Nios II embedded processor by SOPC Builder tools in Quartus II and thus enable us to implement a variety of design projects for the development of sophisticated digital systems.

The paper "Communication via Zigbee protocol with Nios II" aims to complement these two technologies in a safety net, because at present by the high rate of insecurity these systems are widely used in homes, offices and commercial premises thus have a high rate of growth primarily in the field of home automation.

Keywords: Wireless Networks, Zigbee, FPGA, Nios II.

1. Introducción

En la actualidad la tecnología ha avanzado de tal manera que busca la implementación de sistemas que trabajen a una mayor velocidad, procesen mayor cantidad de información y que faciliten la interacción con el usuario.

Es por esto que actualmente se trabaja con los sistemas basados en FPGA (Field Programmable Gate Array), las cuales permiten a los usuarios programarlas de tal forma que cumplan especificaciones definidas por los mismos y no exista

la atadura definida por fabricantes de diversas soluciones.

Con este dispositivo nos podemos valer de nuestra creatividad e ingenio para desarrollar soluciones o también podemos incorporar soluciones de terceros, que complementándolas tendremos un sistema más poderoso personalizado para nuestras necesidades.

Al juntarla con otra tecnología como es Zigbee, que hoy se está utilizando mucho en lo que es domótica, podremos encontrar diversas soluciones a problemas tales como seguridad, comunicación, comodidades y muchas otras necesidades de la vida cotidiana.

En este proyecto se busca encontrar una solución sencilla pero a la vez muy necesaria a lo que es la seguridad en nuestros hogares, tal como es el ingreso de intrusos a las mismas.

2. Objetivos

2.1. Objetivo Principal

Se desea realizar la implementación de un sistema, el cual contenga un hardware capaz de realizar comunicación inalámbrica por medio del protocolo Zigbee, para poder transferir señales provenientes de un sistema de seguridad. Estas señales enviadas por el sistema Zigbee serán a su vez enviadas a nuestro procesador embebido NIOS II para finalmente encender una alarma.

2.2. Objetivos Específicos

- Obtener conocimientos acerca del funcionamiento de los FPGA, con la finalidad de utilizarlo como controlador del sistema de seguridad.
- Aprender cómo realizar una comunicación y enviar una trama entre dispositivos por medio del protocolo Zigbee.
- Configurar y acoplar sensores de movimiento a un módulo X-bee.
- Diseñar un sistema de detección de movimiento que emita una señal de respuesta a manera de alarma.

3. Metodología

Nuestra red de seguridad cuenta con un sistema infrarrojo de movimientos el cual, al estar combinado con un transmisor inalámbrico de tecnología Zigbee nos evita realizar ciertas variaciones en la infraestructura de la casa, ya que solo es necesario seleccionar el lugar y colocarlos, de igual manera estos transmisores economizan el uso de energía y solo requieren de una batería de 3.3 v.

El transmisor está configurado de tal forma que envía dos tipos de respuestas diferentes, dependiendo estas de la variación que exista en el sensor de movimiento.

Estos transmisores inalámbricos tienen un alcance aproximado de 300 metros dentro del hogar, por lo que da la ventaja de tener varios sensores y un solo procesador de información.

Acoplado a nuestro procesador de información se encuentra un receptor Zigbee el cual capta la señal enviada por los transmisores y la transmite a la tarjeta.

Nuestra tarjeta ahora hace las veces de procesador de información y toma las diferentes decisiones

dependiendo de los datos que sean recibidos por medio del receptor Zigbee. A la tarjeta se encuentra acoplada una alarma, la cual si es el caso, empieza a funcionar para notificar que existe algo detectado por el sensor de movimiento.

Para desactivar el sistema de alarma, se debe introducir la contraseña que ha sido previamente validada por el dueño del hogar la primera vez que se inicializa el sistema de seguridad.

4. Marco Teórico

4.1. Tarjeta DE2 de Altera

La Tarjeta DE2 de Altera fue diseñada por profesores para profesores. Es un vehículo ideal para el aprendizaje de la lógica digital, la organización de computadores y el estudio de las FPGAs.

Con un FPGA Cyclone II 2C35 de Altera, la tarjeta está diseñada para la universidad y su uso en los laboratorios. Es adecuada para una amplia gama de ejercicios en los cursos sobre lógica digital y la organización de computadores, desde tareas simples que ilustran los conceptos fundamentales de diseños hasta los más avanzados.

Tabla 1. Información sobre la tarjeta DE2 de Altera

Características	Descripción
FPGA	Cyclone II EP2C35F672C6 con EPCS16 16-Mbit
Interfaces E/S	USB-Blaster para la configuración FPGA E/S de Audio Micrófono Salida de Video (VGA 10-bit DAC) Entrada de Video (NTSC/PAL/Multi-formato) RS232 Puerto Infrarrojo PS/2 10/100 Ethernet USB 2.0 (tipo A y tipo B) Puertos de Expansión
Memoria	8 MB SDRAM, 512 KB SRAM, 4 MB Flash Puerto para la tarjeta SD
Displays	Ocho displays de 7-segmentos Display LCD de 16 x 2
Switches y LEDs	18 Switches 19 LEDs rojos 9 LEDs verdes

	4 botoneras
Clocks	50 MHz clock 27 MHz clock Entrada SMA de clock



Figura 1. Tarjeta DE2 de Altera

4.2. FPGA

Es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada 'in situ' mediante un lenguaje de programación especializado.

La lógica programable puede reproducir desde funciones tan sencillas como las llevadas a cabo por una puerta lógica o un sistema combinatorial, hasta complejos sistemas en un chip.

Muchos FPGA modernos soportan la reconfiguración parcial del sistema, permitiendo que una parte del diseño sea reprogramada, mientras las demás partes siguen funcionando. Este es el principio de la idea de la «computación reconfigurable», o los «sistemas reconfigurables».



Figura 2. FPGA Cyclone II

4.3. Nios II

El procesador Nios II es un procesador RISC para propósitos generales con las siguientes características:

- Conjunto de instrucciones de 32 bits, canal de datos y espacio de dirección.
- 32 registros para propósitos generales.
- Instrucciones dedicadas para calcular productos de multiplicación de 64 bits y 128 bits.
- Acceso a variedad de periféricos, interfaces y memorias.
- Unidad de protección de memoria opcional (MPU).
- Unidad de manejo de memoria (MMU).
- Rendimiento hasta 250DMIPS.

Un procesador Nios II es equivalente a un microcontrolador que incluye un procesador y una combinación de periféricos y memoria en un solo chip. El sistema de un procesador Nios II contiene un núcleo procesador Nios II, un conjunto de periféricos, memoria e interfaces, todas estas implementadas en un solo dispositivo de Altera. Como una familia de microcontrolador, todos los sistemas del procesador Nios II usan un conjunto de instrucciones consistentes y un modelo de programación.

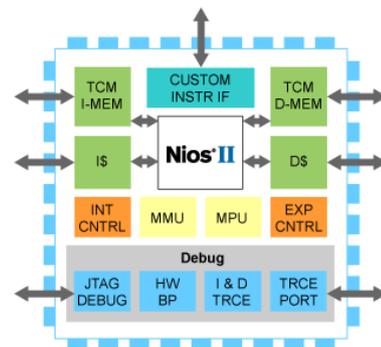


Figura 3. Nios II

4.4. Protocolo Zigbee

Es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal.

Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

En principio, el ámbito donde esta tecnología se desarrolla con más fuerza es en domótica.

Se definen tres tipos distintos de dispositivo Zigbee según su papel en la red:

- Coordinador Zigbee: Es el tipo de dispositivo más completo. Debe existir uno por red. Sus funciones son las de encargarse de controlar la red y los

caminos que deben seguir los dispositivos para conectarse entre ellos.

- Router Zigbee: Interconecta dispositivos separados en la topología de la red, además de ofrecer un nivel de aplicación para la ejecución de código de usuario.
- Dispositivo final: Posee la funcionalidad necesaria para comunicarse con su nodo padre (el coordinador o un router), pero no puede transmitir información destinada a otros dispositivos. De esta forma, este tipo de nodo puede estar dormido la mayor parte del tiempo, aumentando la vida media de sus baterías. Un dispositivo final tiene requerimientos mínimos de memoria y es por tanto significativamente más barato.

4.4.1. Módulo de transmisión inalámbrica Xbee.

Los módulos Xbee proveen 2 formas amigables de comunicación, como lo son la transmisión serial transparente (modo AT) y el modo API. Los módulos Xbee pueden ser configurados desde la PC y estos pueden comunicarse en arquitecturas punto a punto, punto a multipunto o en una red mesh.



Figura 4. Módulo XBee Pro

5. Diseño de la Solución

5.1. Bloque Emisor

En este bloque se combinan el sensor de movimiento y el emisor Zigbee. Este sensor de movimiento tiene un circuito tipo relé, el cual nos sirve para identificar el momento en el que se detecta algún movimiento dentro del área que este cubre.

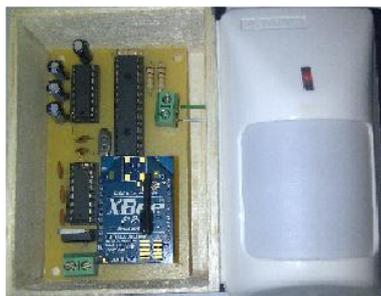


Figura 5. Bloque emisor

5.1.1. Sensor de movimiento. El sensor que utilizamos es el CometPet, el cual tiene ciertas características de alcance y de reconocimiento de objetos, como por ejemplo, este alcanza una distancia máxima de hasta 8 metros, cubriendo un radio de 90°, de igual manera cuenta con un sensor Infrarrojo pasivo anti mascotas de hasta 20 kg (42 lbs), estamos hablando de animales domésticos, lo cual lo convierte en un sensor ideal para domótica, también tiene protección contra la luz blanca, contador de pulsos seleccionables y un prisma óptico LED.

5.2. Bloque Receptor

En este bloque se combinan el receptor Zigbee y el procesador de información DE2. Nuestro receptor se encuentra todo el tiempo transmitiendo la información recibida a la tarjeta DE2 por medio de una comunicación serial.



Figura 6. Bloque receptor

5.2.1. Configuración Xbee. Una manera sencilla de hacerlo es utilizando el programa X-CTU, provisto por Digi, el cual se puede descargar de internet y es completamente gratuito, este ayuda a configurar muchas de las características de los módulos Xbee tales como velocidad de transmisión y canales, también puede actualizar sus firmwares y realizar todas las pruebas necesarias para verificar el buen funcionamiento de los mismos.

5.2.2. Protocolo de comunicación RS232. Es el tipo de conexión por el cual se configuran los módulos Xbee, consiste en un conector tipo DB-25 de 25 pines, aunque es normal encontrar la versión de 9 pines DB-9, más barato e incluso más extendido para cierto tipo de periféricos. Las señales con las que trabaja este puerto serial son digitales, los pines que portan los datos son RXD y TXD, los demás se encargan de otros trabajos, DTR indica que el ordenador está encendido, DSR que el aparato conectado a dicho puerto está encendido, RTS que el ordenador puede recibir datos, CTS que el aparato conectado puede recibir datos y DCD detecta que existe una comunicación, presencia de datos.

Tabla 2. Información de pines del conector DB9

#	Pin	E/S	Función
1			Tierra de chasis
2	RXD	E	Recibir datos
3	TXD	S	Transmitir datos
4	DTR	S	Terminal de datos listo
5	SG		Tierra señal
6	DSR	E	Equipo de datos listo
7	RTS	S	Solicitud de envío
8	CTS	E	Libre para envío
9	RI	S	Timbre telefónico

5.3. Bloque Procesador

Este bloque es el encargado de recibir los datos enviados desde el Coordinador, procesar dicha información y tomar decisiones en cuanto a las acciones a realizarse.

La tarjeta DE2 de Altera tiene múltiples beneficios como ya lo hemos mencionado anteriormente. El procesador Nios II y la herramienta SOPC Builder trabajan en conjunto con la DE2, de manera que la generación del bloque “procesador” se consigue a través de la herramienta SOPC Builder del Quartus II de Altera, y se configuran todos los periféricos necesarios en dicho bloque para así utilizarlo, éste es probado con el entorno Nios II IDE.

5.3.1. Configuración en Quartus II. Para empezar a trabajar con la tarjeta DE2 de Altera debemos definir el hardware que se va a utilizar. Gracias a las herramientas del Quartus II y a SOPC Builder esta configuración es sumamente sencilla.

Con SOPC Builder se puede armar el hardware dependiendo de los componentes vamos a utilizar.

En necesario ir seleccionando cada elemento a utilizar, agregarlo al hardware que estamos creando y configurar cada elemento de acuerdo a nuestra necesidad.

En la figura 6 se muestra una imagen del hardware que elaboramos para nuestro proyecto, el cual debe ser compilado para que genere un archivo .sof y un archivo .pft que será utilizado posteriormente por el Nios II IDE

Use	Conn.	Module Name	Description	Clock	Base	End
<input checked="" type="checkbox"/>		cpu_0	Nios II Processor	clk_0		
		instruction_master	Avalon Memory Mapped Master		IPQ 0	IPQ 31
		data_master	Avalon Memory Mapped Master		0x00001800	0x00001fff
		flag_debug_module	Avalon Memory Mapped Slave		0x00001800	0x00001fff
<input checked="" type="checkbox"/>		onchip_memory2_0	On-Chip Memory (RAM or ROM)		0x00000000	0x00000fff
		s1	Avalon Memory Mapped Slave		0x00000000	0x00000fff
<input checked="" type="checkbox"/>		switches	Parallel Port		0x00001000	0x0000100f
		avalon_paralel_port_s.	Avalon Memory Mapped Slave		0x00001000	0x0000100f
<input checked="" type="checkbox"/>		green_leds	Parallel Port		0x00001010	0x0000101f
		avalon_paralel_port_s.	Avalon Memory Mapped Slave		0x00001010	0x0000101f
<input checked="" type="checkbox"/>		red_leds	Parallel Port		0x00001020	0x0000102f
		avalon_paralel_port_s.	Avalon Memory Mapped Slave		0x00001020	0x0000102f
<input checked="" type="checkbox"/>		zigbee_input	Parallel Port		0x00001030	0x0000103f
		avalon_paralel_port_s.	Avalon Memory Mapped Slave		0x00001030	0x0000103f
<input checked="" type="checkbox"/>		alarm_output	Parallel Port		0x00001040	0x0000104f
		avalon_paralel_port_s.	Avalon Memory Mapped Slave		0x00001040	0x0000104f

Figura 6. Hardware del Procesador

5.3.2. Configuración en Nios II IDE. Primero se debe seleccionar el archivo .pft generado por el programa de Quartus II para que el IDE sepa con qué tipo de dispositivo se esta trabajando.

Luego se debe realizar la generación del código en Lenguaje C con la que el dispositivo trabajara he interactuará con el usuario.

Finalmente se debe cargar el Hardware en la tarjeta, para esta labor, el IDE pose una herramienta que permite cargar el archivo .sof, a continuación se muestra una imagen mostrando esta pantalla.

Luego de esto se copia el código que se generó y se podrá ver funcionando la tarjeta.

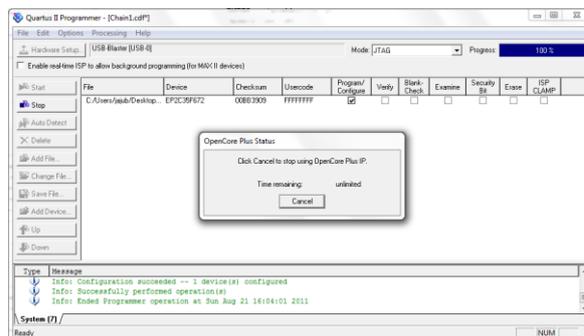


Figura 7. Cargando el Hardware

5.3.3. Código del programa principal. De una manera resumida mostramos parte del código de nuestro programa, así como ejemplos de definición de librerías, prototipos de funciones y algunas variables globales.

```

/* librerias */
#include "address_map.h"
#include "nios2_ctrl_reg_macros.h"
#include "259macros.h"
#include "io.h"
#include "system.h"
...

/* prototipos de las funciones */
void Update_red_LED(void);
void Update_UARTs(int);
void LimpiarLCD(int, char *);
...

/* variables globales */
volatile unsigned int *pLEDG = (volatile
unsigned int *)0x10000010;
volatile unsigned int *pSWITCH = (volatile
int green_LED_pattern = 0;
int display_toggle = 0;
int salir = 0;
...

/* programa principal */
int main(void)
{
volatile int * pushbutton_ptr = (int *)
PUSHBUTTON_BASE;
unsigned int contrasena=0;

```

```
unsigned int sw=0;
unsigned int ENTER=0;
unsigned int ENTER2=0;
unsigned int ENTER3=0;

while (1)
{
    *pLEDG=0;
    if (contrasena == 0)
    {
        LimpiarLCD(17, "Enter a New Pass");
        while (ENTER==0)
        {
            Update_red_LED ( );
        }
    }
}
```

5.4. Bloque de Salidas

Este bloque es el encargado activar las diferentes salidas del sistema dependiendo de los requerimientos, señales y datos ingresados al sistema, los cuales ya han sido previamente procesados.

En este sistema se manejan 2 dispositivos de salida. Al momento de detectar algún tipo de movimiento, mientras que el sistema de alarma se encuentra encendido, se activa un circuito de alarma junto con una serie de luces intermitentes, el cual se desactivará únicamente ingresando la clave correcta del sistema.

6. Resultados

6.1. Comunicación entre dispositivos XBee

Una vez configurados los dispositivos XBee están listos para empezar la transmisión de datos entre sí, en la Figura 8 podemos apreciar la comunicación entre el Coordinador y el Router Zigbee

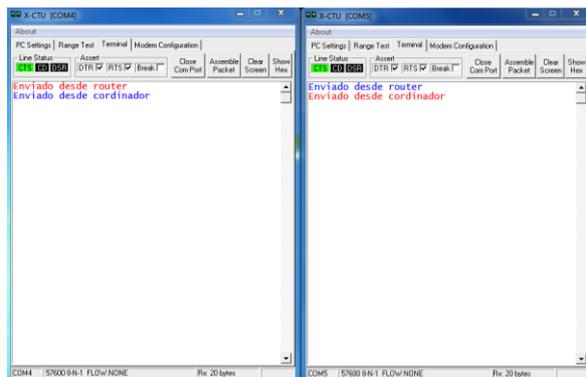


Figura 8. Comunicación entre los dispositivos XBee

6.2. Interacción entre el usuario y la tarjeta

A continuación detallamos todo el proceso de interacción del usuario con la tarjeta DE2.

Una vez cargado el hardware necesario en la tarjeta aparecerá un mensaje indicándonos que la operación

se llevó a cabo con éxito, seguido de esto procedemos a compilar el programa en NIOS II y una vez concluida la compilación aparecerá un mensaje de bienvenida en la tarjeta, lo cual solo ocurrirá al momento de encenderla o resetearla



Figura 9. Mensaje de bienvenida

Este mensaje de bienvenida tiene una duración de 3 segundos, pasado este tiempo el sistema pide que se ingrese la clave por primera vez para grabarla en memoria.

Esta contraseña es ingresada mediante los switches de la tarjeta por lo que se pueden utilizar hasta los 18 existentes para su creación. La manera de guardarla es levantando los switches seleccionados por el usuario y presionando el botón Enter, una vez hecho esto aparecerá el mensaje indicando que la contraseña ha sido guardada.

Si deseamos activar el sistema de alarma debemos presionar el botón Enter, el sistema nos da 5 segundos para poder salir del lugar que estamos dejando asegurado sin que la alarma empiece a sonar detectándonos mientras salimos. Luego de esto el sistema se activa y se enciende un LED verde indicándolo.

Cuando el sensor detecta algún movimiento se enciende el LED y comienza la transmisión, en ese momento el sistema le da al usuario 5 segundos para poder desactivar la alarma antes de encender la sirena.



Figura 10. Sensor de movimiento detectando

Si la alarma no es desactivada aparecerá un mensaje indicando que hay un intruso en el lugar, se encenderán los LEDs verdes de forma intermitente y empezará a sonar la sirena, para desactivar la alarma hay que ingresar la contraseña, en caso de que esté incorrecta el sistema presentará un mensaje de error, si la contraseña ingresada es correcta la sirena dejará de sonar, los LEDs se apagarán y aparecerá un mensaje indicando que el sistema está nuevamente disponible para ser activado.



Figura 11. Alarma lista para usar

7. Conclusiones

1. La tarjeta Altera DE2 puede ser utilizada como controlador principal del sistema de seguridad. Pudimos programarla para que realice la coordinación de la información receptada por el módulo Zigbee y dependiendo de esta trama ejecute el sistema de alarma. También se la pudo programar para que interactúe con el usuario de una manera muy sencilla gracias a sus botoneras, leds, display, lcd que incluye.
2. Podemos concluir que es posible realizar una comunicación entre dispositivos utilizando el protocolo Zigbee, lo que se necesita es la formación de una trama para que esta pueda ser enviada por un elemento emisor. En este caso puede ser generado por la tarjeta DE2 para realizar las pruebas.
3. Para acoplar el sensor de movimiento a módulos XBee fue necesario la utilización de un pequeño microcontrolador para generar la trama necesaria para que pueda ser enviado por el transmisor XBee. Se debe realizar una configuración básica emisor-receptor para que funcione.
4. Se realizó el sistema de detección de movimiento con el sensor y el emisor XBee. La tarjeta DE2 trabajo como controlador del sistema y dependiendo de la respuesta del sensor pudo manejar el encendido de un circuito de sirena acoplado a su E/S en el puerto de expansión.

8. Agradecimiento

Agradecemos a nuestros profesores que nos han ayudado y sabido guiar para la realización de este proyecto, así como el apoyo incondicional de nuestros padres y hermanos.

9. Referencias

- [1] Miguel Ángel Freire Rubio, Introducción al lenguaje VHDL, Universidad Politécnica de Madrid Departamento de Sistemas Electrónicos y de Control, Marzo 2010
- [2] EUITI Bilbao, Libro Electrónico de VHDL, Departamento de Electrónica y Telecomunicaciones, Marzo 2010
- [3] Altera, Tarjeta DE2 Altera, http://www.altera.com/education/univ/materials/boards/de2/unv-de2-board.html?GSA_pos=1&WT.oss_r=1&WT.oss=e2, Abril 2011.
- [4] Configuración RS232 en proto (comunicación), <http://www.recercat.net/bitstream/2072/13081/1/FC+Ivan+Barneda.pdf>, Abril 2011
- [5] Configuración de los módulos Xbee, <https://forja.rediris.es/docman/view.php/720/1132/Tutorial%20XBee.pdf>, Abril 2011
- [6] Cableado emisor y receptor Zigbee <http://www.forosdeelectronica.com/f24/comunicar-usb-zigbee-24120/>, Abril 2011
- [7] Computadora básica, test de funcionamiento, http://www.altera.com/education/univ/support/examples/unv-example-systems.html?GSA_pos=9&WT.oss_r=1&WT.oss=example%20c%20code%20nios%20ii%20rs232, Agosto 2011