



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“Control de un Robot usando un sistema Embebido
basado en el procesador NIOS II”**

TESINA DE SEMINARIO

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE:
INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN
ELECTRÓNICA
INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN
ELECTRÓNICA
INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN
INDUSTRIAL**

Presentado por:

JOSE ANTONIO CALDERON CEPEDA

HECTOR FELIPE DIAZ VERA

LENIN FABRIZIO CHOEZ GUTIERREZ

**Guayaquil – Ecuador
2012**

AGRADECIMIENTO

A mi madre, a mi esposa, a mi hija

José Antonio

A mis padres por estar siempre pendiente de mí,
a mi esposa por ser quien está allí apoyándome, y
a mi hijo que me da aliento para seguir

Héctor Felipe

A mis padres, a mi hermano, por el apoyo que
me han dado en todo momento y al profesor
el Ing. Ronald Ponguillo por el apoyo brindado

Lenín Fabrizio

DEDICATORIAS

A mi madre, pilar fundamental de mi vida, quien
me enseñó los valores más importantes sobre
los que se debe desarrollar una persona

José Antonio

A mi madre Isabel, a mi padre Héctor
a mi esposa Ana y a mi hijo Héctor Felipe

Héctor Felipe

A nuestro señor Dios todopoderoso, a mis padres
Claudio Chóez y Ana Gutiérrez, a mis abuelos,
a mi hermano Javier y a mi hijo Lenín Matías.

Lenín Fabrizio

TRIBUNAL DE SUSTENTACIÓN

Ing. Ronald Ponguillo Intriago

PROFESOR DEL SEMINARIO DE GRADUACIÓN

Ing. Carlos Valdivieso Armendáriz

PROFESOR DELEGADO POR EL DECANO

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este trabajo, nos corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”. (Reglamento de exámenes y títulos profesionales de la ESPOL)

Felipe Díaz Vera

José Calderón Cepeda

Lenin Choez Gutiérrez

RESUMEN

El presente trabajo es una herramienta para desarrollo e investigación de la robótica en la ESPOL.

Pretende hacer uso de tecnologías aprendidas e investigadas para promover en la Universidad la implementación a la medida de soluciones robóticas y no pensando en el actual comportamiento de consumo, que nos dice lo que tenemos que comprar y no lo que tenemos que diseñar.

Es un impulso y motivación al desarrollo de soluciones ingenieriles, con tecnología de punta como son los FPGA, sistemas embebidos, comunicaciones inalámbricas de bajo coste pero de eficiencia máxima, trata de romper el paradigma de que lo hecho es lo que hago e idealizar a la población Politécnica a usar los conocimientos adquiridos para crear soluciones y no solo consumir.

INDICE GENERAL

RESUMEN.....	vii
INDICE GENERAL.....	ix
ABREVIATURAS.....	xiii
INDICE DE FIGURAS.....	xv
INDICE DE TABLAS.....	xvii
INTRODUCCIÓN.....	xviii
Capítulo 1.....	1
1. GENERALIDADES.....	1
1.1. ALCANCE Y LIMITACIONES DEL PROYECTO.....	1
1.2. OBJETIVOS.....	2
1.1.1. Objetivos generales.....	2
1.1.2. OBJETIVOS ESPECIFICOS.....	2
1.3. ANTECEDENTES.....	3
1.4. IDENTIFICACION DEL PROBLEMA.....	3

	3
1.5. DESCRIPCION BREVE DE LA SOLUCION	4
Capítulo 2.....	5
2. MARCO TEORICO.....	5
2.1. INTRODUCCION AL PROCESADOR NIOS II.....	5
2.2. CARACTERISTICAS Y ARQUITECTURA.....	5
2.3. TARJETAS DE DESARROLLO Y EDUCACIÓN DE ALTERA.....	8
2.4. FPGA	11
2.4.1. Definición	11
2.5. Características.....	13
2.6. Aplicaciones.....	14
2.7. PROCESADORES EMBEBIDOS CONFIGURABLES.....	15
2.7.1. Procesadores soft-core	15
2.8. PROTOCOLO DE COMUNICACIÓN RS-232.....	15
2.9. TECNOLOGÍAS INALÁMBRICAS	16
2.9.1. Protocolo Zigbee.....	16
2.9.2. Coordinador ZigBee (ZigBee Coordinator ZC).....	19

2.9.3.	Router ZigBee(ZigBee Router ZR)	20
2.9.4.	Dispositivo final (ZigBee En Device ZED)	20
2.10.	Dispositivo de funcionalidad completa (FFD)	21
2.11.	Dispositivos de funcionalidad reducida (RFD)	21
2.12.	Módulos de transmisión inalámbrica Xbee.....	22
2.13.	ROBOTICA	22
2.13.1.	CONCEPTOS GENERALES	23
2.13.2.	DEFINICION DE ROBOT	23
2.13.3.	MORFOLOGIA DE UN ROBOT	24
2.13.4.	ROBOT MOVIL TERRESTRE	25
2.13.5.	LOCOMOCION CON RUEDAS	25
2.13.6.	Diferencial	27
2.13.7.	Diseño sincronizado	27
2.13.8.	Diseño de triciclo y coche.....	28
2.14.	MOTORES Y SISTEMA DE ENERGIA	28
2.15.	PUENTE H.....	29

Capítulo 3.....	32
3. DISEÑO E IMPLEMENTACION	32
3.1. DISEÑO DE LA ARQUITECTURA DEL ROBOT	32
3.2. MORFOLOGIA	33
3.3. SISTEMA DE ENERGIA	36
3.4. DRIVER Y MOTORES	37
3.5. COMUNICACIÓN	40
3.6. LOGICA DE CONTROL	40
3.7. DISEÑO DEL HARDWARE EN SOPC BUILDER.....	41
3.8. COMPILACION EN QUARTUS II	51
3.9. ASIGNACION DE PINES EN LA FPGA	51
3.10. CONFIGURACION DE LOS MODULOS XBEE	59
3.11. CODIGO DEL PROGRAMA PRINCIPAL	63
3.12. PCB Y ENSAMBLADO FINAL	65
Capítulo 4.....	70
4. PRUEBAS Y RESULTADOS	70

4.1. CASOS DE PRUEBAS..... 71

4.2. RESULTADOS 74

4.3. CONCLUSIONES 75

4.4. RECOMENDACIONES 76

CONCLUSIONES

RECOMENDACIONES

BIBLIOGRAFIA

ANEXOS

ABREVIATURAS

FPGA	Field Programmable Gate Array
DE2	Development and Education Board
RS-232	Recommended Standard 232
E/S	Entrada y Salida
JP1	Expansion Header
GPIO	General Purpose Input/Output
UART	Universal Asynchronous Receiver-Transmitter
ASIC	Application Specific Integrated Circuit
SOPC	System on a Programmable Chip
PCB	Printed Circuit Board
RTS	Request to send
CTS	Clear to send

INDICE DE FIGURAS

Figura 1: Tablero de desarrollo DE2 de Altera.....	9
Figura 2: Diagrama de bloques de la tarjeta DE2.....	11
Figura 3: Arquitectura de un FPGA	12
Figura 4: Diseño de aplicaciones para ZigBee.....	17
Figura 5: Arquitectura de la tecnología ZigBee.....	18
Figura 6: Modelo de red ZigBee	20
Figura 7: Módulos XBee.....	22
Figura 8: Ejemplo de robot movido por ruedas	26
Figura 9: Robot con rueda loca.....	27
Figura 10: Ejemplos de varios tipos de motores	29
Figura 11: Configuración de puente en H.....	30
Figura 12: Los dos estados básicos del circuito.....	30
Figura 13: Diagrama de bloques del robot	32
Figura 14: Prueba de modem	33
Figura 15: Vista del robot desde arriba	34
Figura 16: Vista interna del robot.....	35
Figura 17: Driver.....	36
Figura 18: Regulador de 5V	37
Figura 19: Valores de los motores.....	38
Figura 20: Mecánica del componente L298	39

Figura 21: Dimensiones del motor lateral.....	39
Figura 22: Diseño del motor frente	40
Figura 23: Captura de pantalla sistema embebido	42
Figura 24: Valores de procesador seleccionado.....	43
Figura 25: Configuración JTAG UART	44
Figura 26: Configuración de memoria	45
Figura 27: Controlador SDRAM.....	46
Figura 28: Configuración del reloj del sistema	47
Figura 29: Configuración UART.....	48
Figura 30: Definición de puerto paralelo.....	49
Figura 31: Paramterización del timer.....	50
Figura 32: Captura de pantalla de valores de compulación	51
Figura 33: Configuración de la conexión serial	60
Figura 34: Pantalla de valores de dispositivo inalámbrico.....	61
Figura 35: Parametrización de dispositivos XBee.....	62
Figura 36: Prueba de estados de conexión.....	63
Figura 37: Código PRINCIPAL.C.....	64
Figura 38: Definición de librería PRINCIPAL.H.....	65
Figura 39: Esquemático del circuito adicional.....	66
Figura 40: PCB terminado con distribución de elementos.....	67
Figura 41: PCB a doble cara, este es el top layer del PCB.....	68
Figura 42: Bottom Layer del PCB	69

10

Figura 43: Pantalla de programa de prueba..... 71

INDICE DE TABLAS

Tabla 1: Características de los motores.....	38
Tabla 2: Tabla de asignación de pines del circuito	59

INTRODUCCIÓN

El desarrollo de la robótica es una de las expresiones de ingeniería mas desafiantes de los últimos tiempos.

Esta requiere establecer los parámetros de integración de muchas disciplinas que aporten a desarrollo de prototipos cada vez más inteligentes, autónomos y útiles.

Desapegándonos de los estándares, aportando a la educación e investigación, hemos pretendido aportar con nuestro conocimiento en el desarrollo de lo que pretende ser una plataforma que impulse el inventivo de los estudiantes de la ESPOL, para aportar como ha sido costumbre desde siempre con herramientas de desarrollo para el País.

Capítulo 1

1. GENERALIDADES

Dado que la robótica no se ha desarrollado en la ESPOL, donde las materias introductorias no requieren tener conocimientos de electrónica, comunicaciones, inteligencia artificial, ni nada que le permita a los estudiantes e ingenieros realizar diseños de robots, proponemos este trabajo para abrir el camino a la investigación y desarrollo de artefactos con características que vayan de acuerdo a un marco mas real que pueda enfrentar la problemática nacional.

Hemos utilizado tecnología que converge a la realidad actual,por un lado a nivel de sistemas digitales, la tecnología de FPGAs y por otro lado la realidad cada vez mas notoria de sistemas embebidos, de manera que con este trabajo se intenta romper el paradigma nacional de consumir lo realizado y ponemos una semilla para entrar en la era de la siembra y explotación de nuestro talento.

1.1. ALCANCE Y LIMITACIONES DEL PROYECTO

Se creará un robot móvil terrestre como plataforma de desarrollo de proyectos de robótica futuros, no será autónomo, ni tendrá inteligencia propia, pues para esto se controlará remotamente el artefacto utilizando un computador con un software instalado que envíe mediante comunicación serial las órdenes que entienda el robot.

Se caracterizará por ser de baja potencia lo que nos lleve a tener para el movimiento una velocidad reducida.

1.2. OBJETIVOS

1.1.1. OBJETIVOS GENERALES

- Construir un robot móvil terrestre.
- Implementar una plataforma de desarrollo de robots móviles terrestres teleoperado para aprendizaje e investigación.
- Introducir en la ESPOL la tecnología de diseño OnChip para dar soluciones a problemas expertos puntuales.

1.1.2. OBJETIVOS ESPECIFICOS

- Aplicar tecnología inalámbrica para la comunicación con agentes remotos.
- Usar tecnología de punta basada en FPGA y sistemas embebidos.

1.3. ANTECEDENTES

Teniendo como base el concepto universalmente aceptado sobre robótica, definida por así decir, como una rama de la tecnología que se dedica al diseño, construcción, operación, disposición estructural, manufactura y aplicación de robots; y a sabiendas que en la ESPOL se introduce a los futuros ingenieros al mundo de la robótica con la materia Fundamentos de Robótica, que si bien da las bases del conocimiento general de la Robótica, no se realiza investigación, ni diseño de prototipo, que permitan a la ESPOL ser representantes de esta multidisciplinaria tendencia tecnológica que permitan a la ESPOL aportar al País con herramientas de apoyo a su producción.

1.4. IDENTIFICACION DEL PROBLEMA

Las soluciones robóticas, vistas en el medio, implican mayormente lo que podríamos definir como “unir y pegar”. Si bien se tienen herramientas de soporte para poder producir elementos tecnológicos de avanzada, el entorno ecuatoriano nos impulsa a consumir las soluciones provistas por agentes externos al medio, inclusive no adaptados a la realidad de las necesidades de nuestro País.

Es por esto que para sentar las bases para el desarrollo de la robótica, se debe impulsar la realización de tecnologías desarrollada por nuestros

ingenieros, donde apliquen las ciencias y disciplinas que aprendieron en la Universidad y no utilizar solamente los puzles tecnológicos que nos envían del exterior.

1.5. DESCRIPCION BREVE DE LA SOLUCION

Hemos diseñado un robot controlado remotamente de manera inalámbrica que integra tecnología de sistemas digitales, sistemas embebidos, comunicaciones inalámbricas, arquitectura de computadores.

Se ha utilizado un FPGA de Altera integrado en la tarjeta de desarrollo DE2 de la empresa mencionada, donde se ha implementado un sistema embebido conformado entre muchos componente por un procesador en software o soft-processor llamado NIOS II en su versión estándar.

Este robot también posee un driver para controlar los motores para el movimiento diferencial, que permiten el del robot en cualquier dirección desarrollado mediante un puente en H L298.

La comunicación con el exterior que es quien envía las ordenes se realizo mediante la utilización del protocolo ZIGBee, con lo cual se puede de manera inalámbrica enviar ordenes al artefacto.

Capítulo 2

2. MARCO TEORICO

2.1. INTRODUCCION AL PROCESADOR NIOS II

NIOS II es un Soft Processor de 32bits diseñado por ALTERA para la familia de sus FPGAs. Se utiliza en un amplio rango de aplicaciones embebidas desde DSP hasta sistemas de control.

2.2. CARACTERISTICAS Y ARQUITECTURA

Tiene una arquitectura RISC soft-core, que es implementada enteramente con lógica programable y bloques de memoria de ALTERA. La naturaleza del soft-core del NIOS II permite a los diseñadores especificar y generar personalizaciones del NIOS II, que se acoplen exactamente a sus aplicaciones específicas.

NIOS II es ofrecido en 3 diferentes configuraciones, fast, standard y economy.

NIOS II/f

Diseñado para máximo desempeño, posee las siguientes características:

- Instrucciones separadas y cache de datos (512 B a 64kB)
- MMU o MPU opcional
- Acceso hasta 2GB de memoria externa
- Memoria opcional para instrucciones y datos
- Seis etapas de pipeline
- Multiplicación en hardware de un solo ciclo
- División por hardware opcional
- Predicción de saltos dinámica
- Hasta 256 instrucciones personalizadas y aceleradores de hardware ilimitados
- Modulo de depuración JTAG
- Modulo opcional JTAG debugger mejorado, incluyendo hardware breakpoint, data triggers y trazabilidad en tiempo real.

NIOS II/s

Diseñado para mantener un balance entre desempeño y costo, se caracteriza por:

- Cache de instrucciones
- Hasta 2GB de direccionamiento de memoria externa
- Memoria opcional para instrucciones y datos
- Cinco etapas de pipeline
- Predicción de saltos estática
- Multiplicación, división y desplazamiento por hardware
- Hasta 256 instrucciones personalizadas
- Módulo de depuración JTAG

NIOS II/e

Está diseñado para utilizarse en pequeñas soluciones, eficiente por el bajo costo en aplicaciones con FPGA Cyclone II, se caracteriza por:

- Hasta 2GB de direccionamiento externo de memoria
- Modulo de depuración JTAG
- Sistema completo en poco menos que 700 LEs
- Mejoras en debug opcionales
- Hasta 256 instrucciones personalizadas

- Libre de licencia.

Los diseñadores de hardware basado en NIOS II usan el SOPC(System on a programmable chip) Builder, un componente para Quartus II, para configurar y generar el sistema NIOS. El GUI permite a los usuarios escoger el conjunto de características del NIOS II y agregar periféricos y bloques I/O como son timers, controladores de memoria, interfaces seriales, etc. y embeberlos en el sistema.

2.3. TARJETAS DE DESARROLLO Y EDUCACIÓN DE ALTERA

DE2 de Altera

La tarjeta para desarrollo y educación Altera DE2 fue diseñada por profesores para profesores. Es una herramienta ideal para el aprendizaje sobre lógica digital, organización de computadores y FPGA. Es usada en un amplio rango de ejercicios de lógica digital y organización de computadoras.

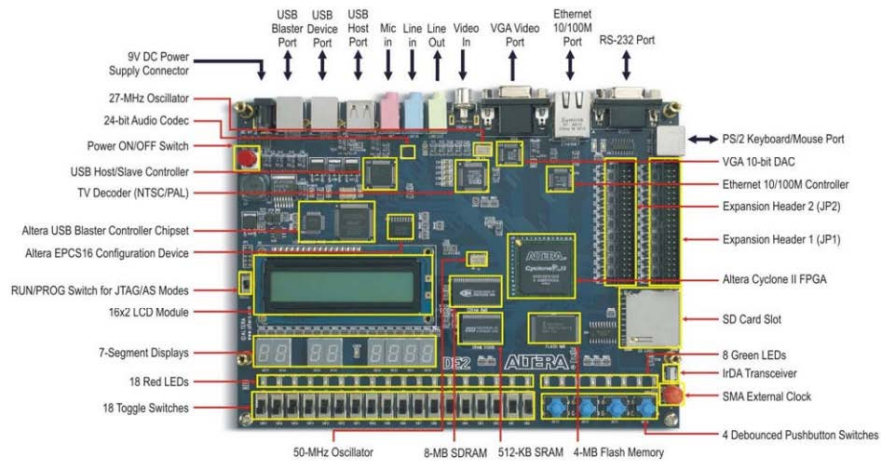


Figura 1: Tablero de desarrollo DE2 de Altera [1]

La figura arriba muestra los diferentes componentes de la tarjeta DE2, que se describen a continuación:

- FPGA Altera Cyclone® II 2C35
- EPCS16, memoria de configuración serial de Altera
- USB Blaster para programación y control de usuario. Ambas programaciones mediante JTAG y AS son soportadas
- 512 Kbyte SRAM
- 8 Mbyte SDRAM
- 4 Mbyte de memoria Flash

- Socket SD Card
- 4 interruptores pulsadores
- 18 interruptores de palanca
- 18 LEDs rojos
- 8 LEDs verdes
- Fuentes de reloj de 50 MHz y 27 MHz
- Línea de entrada de audio con calidad de 24 bit y conector para micrófono
- Controlador Ethernet de 10/100 Mbps
- Conector USB tipo A y B con controlador host/slave
- RS-232 en interface de 9 pines
- Conector PS/2
- IrDA transceiver
- Dos slot de expansión

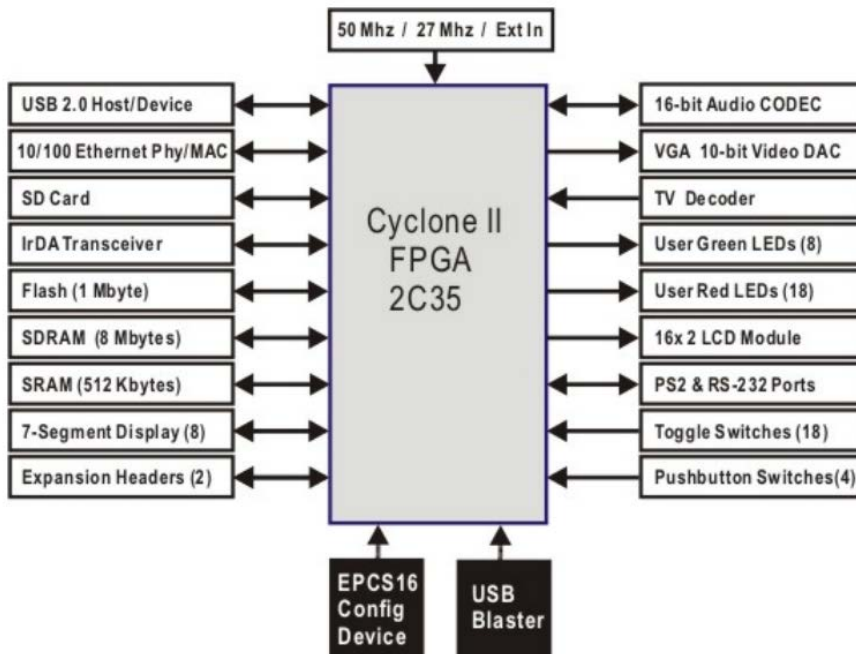


Figura 2: Diagrama de bloques de la tarjeta DE2 [2]

2.4. FPGA

2.4.1. DEFINICIÓN

Es un dispositivo semiconductor compuesto por bloques de lógica que se configura o programan mediante un lenguaje descriptor especializado. La programación interconecta estos bloques de lógica creando funcionalidades específicas, todo esto 'in situ'.

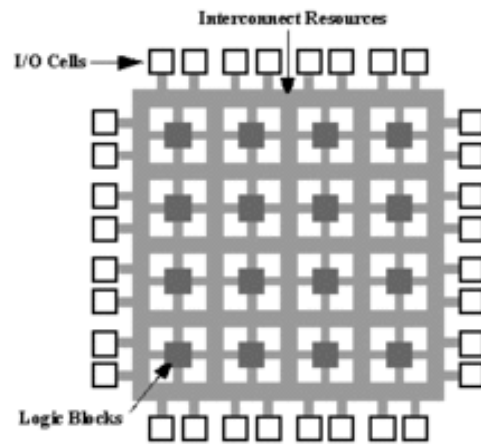


Figura 3: Arquitectura de un FPGA [3]

Mediante esta programación se puede crear desde funciones básicas como las llevadas a cabo por una puerta lógica o un sistema combinatorial pero también se puede crear complejos componentes e incluso sistemas en un chip.

Los FPGAs tienen como característica entre tantas de ser reprogramables, lo que reduce costos por diseño, costos en desarrollo pues este tiempo es mucho menor que con otras tecnologías.

Un FPGA puede desempeñar cualquier función lógica que se necesite, debido a que los bloques lógicos e interconexiones pueden ser programados por el desarrollador después que han sido

manufacturados según la necesidad del ingeniero de desarrollo de sistemas.

Como tendencia tecnológica se muestra en los últimos tiempos que las interconexiones de bloques lógicos configurables del FPGA se combinan con microprocesadores y periféricos con el objetivo de desarrollar un Sistema Programable 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».

Existe código fuente disponible (bajo licencia GNU GPL) de sistemas como microprocesadores, microcontroladores, filtros, módulos de comunicaciones y memorias, entre otros. Estos códigos se llaman cores.

2.5. CARACTERÍSTICAS

Una jerarquía de interconexiones programables permite a los bloques lógicos de un FPGA ser interconectados según la necesidad del diseñador del sistema, algo parecido a un breadboard programable.

Se pueden integrar con gran cantidad de circuitos lógicos, lo que permite tener diseños escalables y adaptables a cualquier necesidad.

Las velocidades de estos dispositivos son muy altas inclusive pueden llegar a la actualidad a 28Gb/s¹

Existe un número considerable de aplicaciones que dan soporte del desarrollo con FPGA, incluso herramientas CAD como es Quartus II

2.6. APLICACIONES

Cualquier circuito de aplicación específica puede ser implementado en un FPGA, siempre y cuando esta disponga de los recursos necesarios. Las aplicaciones donde más comúnmente se utilizan los FPGA incluyen a los DSP (procesamiento digital de señales), radio definido por software, sistemas aeroespaciales y de defensa, prototipos de ASICs, sistemas de imágenes para medicina, sistemas de visión para computadoras, reconocimiento de voz, bioinformática, emulación de hardware de computadora, entre otras. Cabe notar que su uso en otras áreas es cada vez mayor, sobre todo en aquellas aplicaciones que requieren un alto grado de paralelismo.

¹ <http://www.neoteo.com/xilinx-7-fpga-mayor-capacidad-del-mundo>

Una tendencia reciente ha sido combinar los bloques lógicos e interconexiones de FPGA con microprocesadores y periféricos relacionados para formar un sistema programable en un chip.

2.7. PROCESADORES EMBEBIDOS CONFIGURABLES

2.7.1. PROCESADORES SOFT-CORE

Pueden configurarse en software agregando puertos de I/O, temporizadores, módulos de comunicación varios, memoria, etc., para obtener una computadora de aplicación específica o microcontrolador.

Además se pueden crear cualquier modulo usando un lenguaje de descripción de hardware y agregarlo como parte del sistema en chip.

Permite la creación de nuevas instrucciones por parte del usuario.

La asignación de pines es también flexible, lo que permite hacer modificaciones al sistema completo sin necesidad de cambiar de hardware.

2.8. PROTOCOLO DE COMUNICACIÓN RS-232

Recommended Standard 232, conocido como RS-232 es un protocolo que define una norma para intercambio de una serie de datos binarios entre un equipo terminal de datos (DTE) y un equipo de comunicación

de datos (DCE), aunque existen otras que también usan interfaz RS-232.

El RS-232 tiene como interface de conexión común un conector DB-25 o de 25 pines pero también de una interface de 9 pines o DB-9, comúnmente encontrado en un PC.

La interface RS-232 fue diseñada inicialmente para realizar la impresión de documentos en distancias cortas de hasta 15 metros, y con velocidades de comunicación lentas de 20Kbps. EL protocolo soporta la comunicación síncrona y asíncrona y con canales simplex, half duplex y full duplex. El canal usado full duplex es controlado por las líneas de handshaking del RS-232 que se usa para resolver problemas que se puedan presentar con este tipo de operación.

Si un dispositivo RS-232 procesa datos a una velocidad menor que el otro las líneas de handshaking RTS y CTS se encargan de controlar este flujo.

2.9. TECNOLOGÍAS INALÁMBRICAS

2.9.1. PROTOCOLO ZIGBEE

ZigBee es el nombre de la especificación de un conjunto de alto nivel de comunicación inalámbrica para la utilización con radiodifusión

digital de bajo consumo, se basa en la norma IEEE 802.15.4 de redes inalámbricas de área personal (WPAN). Se lo concibió para usarse en aplicaciones que requieran comunicaciones seguras de baja tasa de envío de datos como de maximización de la vida de las baterías.

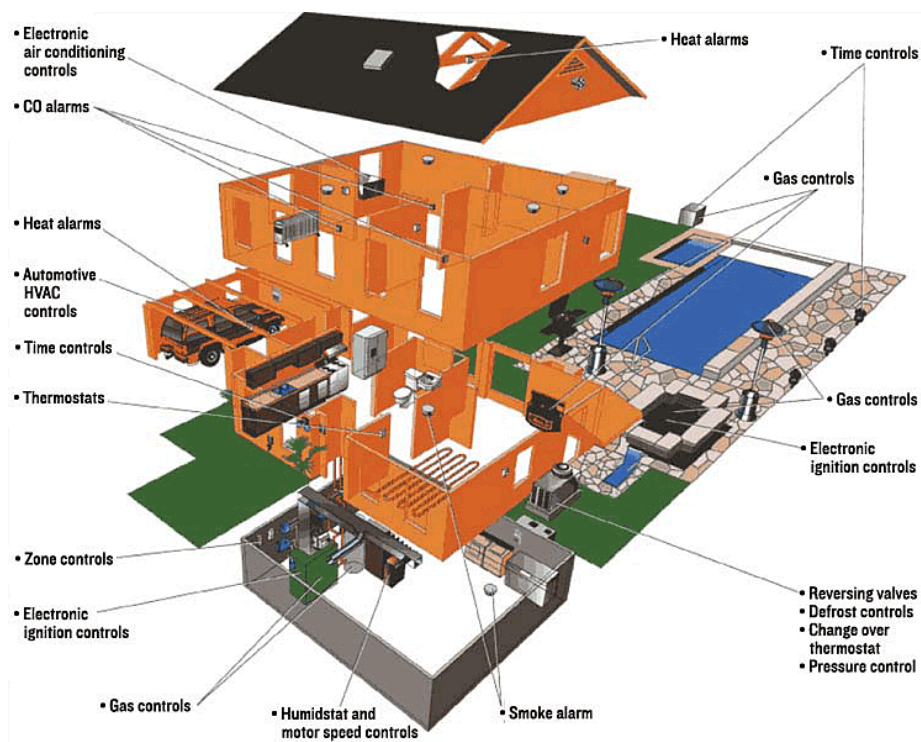


Figura 4: Diseño de aplicaciones para ZigBee [4]

El ámbito donde mayormente se encuentran aplicaciones desarrolladas es en la DOMOTICA, la razón de ello:

- Su bajo consumo de energía

- Su topología de red en malla
- Su fácil integración

ZigBee usa la banda ISM (Industrial, Scientific and Medical), en concreto, 868 MHz en Europa, 915 MHz en Estados Unidos y 2.4 GHz en todo el mundo, sin embargo se optó por la frecuencia 2.4 GHz por ser libre en todo el mundo.

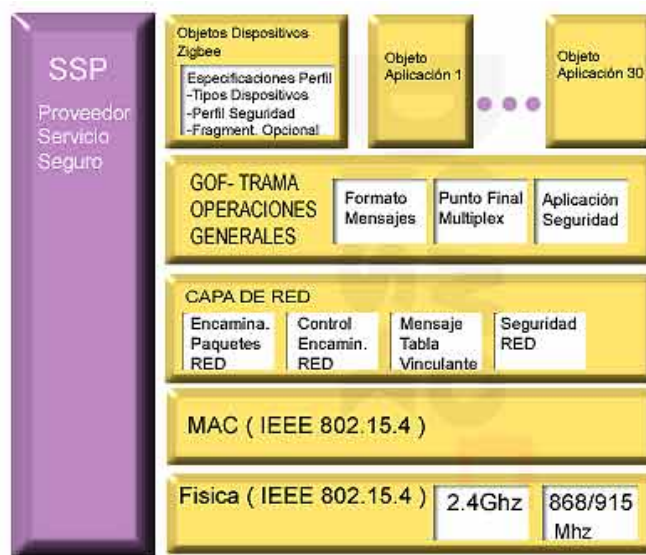


Figura 5: Arquitectura de la tecnología ZigBee [5]

El nodo ZigBee más completo requiere cerca del 10% del hardware de un nodo Bluetooth o WIFI típico, no obstante el tamaño del código en

si es mayor y se acerca al 50% del tamaño del Bluetooth. Existen dispositivos de hasta 128Kb de almacenamiento.

A nivel de precios en el 2006 ZigBee costaba cerca de USD 1,00 para el transceptor y para el radio, procesador y memoria unos 3 dólares. Mientras que el Bluetooth en grandes volúmenes se lo encontraba de 4 a 6 dólares.

Los protocolos ZigBee están definidos para usarse en aplicaciones encastradas con requerimientos de baja transmisión de datos y poco consumo energético. Puede utilizarse para control industrial, sensores empotrados, recolectores de datos, para seguridades como es detección de humo, intrusos o domótica en general. La red en conjunto utilizará una cantidad muy pequeña de energía de forma que cada dispositivo individual tenga autonomía de hasta 5 años.

Según el papel en la red de los dispositivos ZigBee, encontramos:

2.9.2. COORDINADOR ZIGBEE (ZIGBEE COORDINATOR ZC)

Debe existir uno por red, se encarga de controlar la red y los caminos que deben seguir los dispositivos para conectarse entre ellos. Es el dispositivo más completo.

2.9.3. ROUTER ZIGBEE(ZIGBEE ROUTER ZR)

Interconecta dispositivos separados en la topología de la red, además de ofrecer una capa de aplicación para la ejecución de código de usuario.

2.9.4. DISPOSITIVO FINAL (ZIGBEE EN DEVICE ZED)

Posee la funcionalidad necesaria para comunicarse con un nodo padre sea coordinador o router, pero no transmite información a otros dispositivos. De esta forma el nodo puede estar dormido la mayor parte del tiempo, aumentando la vida media de la batería. Es relativamente barato debido a que no tiene grandes requerimientos de memoria.

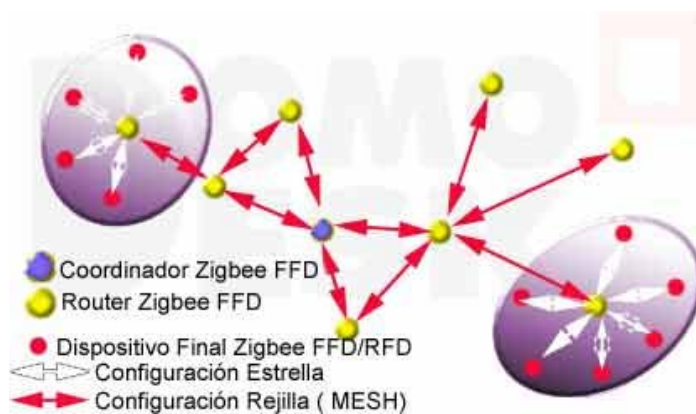


Figura 6: Modelo de red ZigBee [6]

Basándose en la funcionalidad, se clasifican en:

2.10. DISPOSITIVO DE FUNCIONALIDAD COMPLETA (FFD)

También conocidos como nodo activo. Es capaz de recibir mensajes en formato 802.15.4. Puede funcionar como Coordinador o Router, o puede ser usado en dispositivos de red que actúe de interface con usuarios.

2.11. DISPOSITIVOS DE FUNCIONALIDAD REDUCIDA (RFD)

Conocido como nodo pasivo, tiene capacidades y funcionalidad limitada. Básicamente son los sensores/actuadores de la red.



Figura 7: Módulos XBee [7]

2.12. MÓDULOS DE TRANSMISIÓN INALÁMBRICA XBEE

XBee es la marca comercial de Digi International para una familia de módulos de radio compatibles. XBee puede ser usado con 4 conexiones mínimas, las cuales son, power (3.3V), ground, data in y data out. También se puede usar líneas reset y sleep.

Muchas familias de XBee tienen algunas otras señales como control de flujo, I/O, A/D e indicadores de líneas integradas.

2.13. ROBOTICA

2.13.1. CONCEPTOS GENERALES

La Robótica es una ciencia de investigación, estudio y tecnología de los robots. Se ocupa del diseño, manufactura y aplicaciones de robots. Combina diversas disciplinas como son la mecánica, electrónica, informática, inteligencia artificial y la ingeniería de control.

2.13.2. DEFINICION DE ROBOT

Un robot es un agente artificial, mecánico o virtual. Es una maquina usada para realizar un trabajo automáticamente y que es controlada por una computadora.

Un robot para ser considerado como tal debe presentar algunas propiedades:

- No es natural, ha sido creado artificialmente
- Puede sentir su entorno
- Puede manipular cosas de su entorno
- Tiene cierta inteligencia o habilidad para tomar decisiones basadas en ambientes o en una secuencia pre programada automáticamente.
- Es programable

- Puede moverse en uno o más ejes de rotación o traslación.
- Puede realizar movimientos coordinados.

Joseph Engelberger, pionero de la industria robótica dijo: "No puedo definir un robot, pero reconozco uno cuando lo veo.

2.13.3. MORFOLOGIA DE UN ROBOT

La morfología de un robot describe la estructura de componentes que lo conforman, Un robot está formado por elementos que definen su diseño, como la estructura mecánica, transmisiones, sistema de accionamiento, sistema sensorial, sistemas de control y elementos terminales.

Mecánicamente un robot está formado por una serie de elementos o eslabones unidos mediante articulaciones que permiten un movimiento relativo entre cada dos eslabones consecutivos.

Los robots se clasifican según varios autores de diferente forma, pues no existe un consenso en este tema, pero las más aceptadas son:

Según su cronología

En varias generaciones que contemplan la evolución de los mismos, comenzando por manipuladores, robots de aprendizaje, robot de control sensorizado y robots inteligentes.

Según su arquitectura

Aquí se definen a los robots basados en su configuración, se definen los poliarticulados, donde se encuentran los robots industriales, los manipuladores, cartesianos; todos estos caracterizados por ser sedentarios, pues sus extremidades suelen ser limitadas en su desplazamiento que está determinado por un sistema de coordenadas. También se encuentran los móviles, que tienen gran capacidad de desplazamiento, generalmente basados en un sistema locomotor de tipo rodante. Los andróides, que intentan actualmente reproducir total o parcialmente la forma y el comportamiento de la cinemática humana. Los zoomórficos, constituyen una clase caracterizada principalmente por sus sistemas de locomoción que imitan a diversos seres vivos. Por último los híbridos, que corresponden a aquellos de difícil clasificación cuya estructura se sitúa en alguna de las anteriores expuestas.

2.13.4. ROBOT MOVIL TERRESTRE

Es un robot de movimiento terrestre, existe una gran variedad de modos de moverse sobre una superficie sólida. Los más comunes son las ruedas, las cadenas y las patas.

2.13.5. LOCOMOCION CON RUEDAS

Los vehículos con rueda son con mucho los más populares por razones prácticas. Los robots con ruedas son más sencillos y más fáciles de construir, la carga que pueden soportar es comúnmente mayor. A esto se puede añadir la posibilidad de transformar vehículos de ruedas de radio control para usarlos como bases de robots.



Figura 8: Ejemplo de robot movido por ruedas [8]

La principal desventaja de las ruedas es su empleo en terreno irregular en el que se comportan bastante mal. Normalmente un vehículo de ruedas podrá sobrepasar un obstáculo que tenga una altura no superior al radio de sus ruedas.

Existen varios diseños de ruedas para elegir cuando se quiere construir una plataforma móvil sobre ruedas: diferencial, sincronizada, triciclo y de coche.

2.13.6. DIFERENCIAL

Es uno de los sistemas menos complicados de locomoción. Básicamente el robot puede ir recto, girar sobre si mismo y trazar curvas. Como dificultad tenemos el equilibrio del robot, hay que buscarle un apoyo adicional a las dos ruedas ya existentes. Esto se consigue mediante una o dos ruedas de apoyo añadidas en un diseño triangular o romboidal. Cuando los motores encuentran diferentes resistencias las velocidades de los motores varían provocando que el robot no vaya recto a pesar de definir el movimiento así.

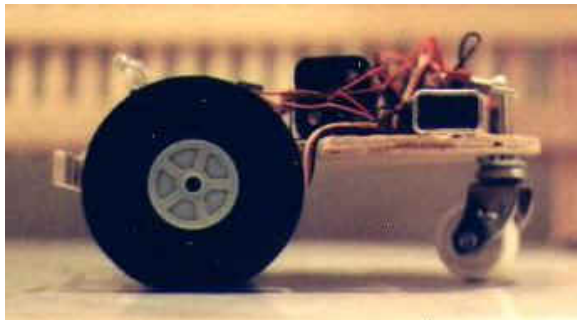


Figura 9: Robot con rueda loca [9]

2.13.7. DISEÑO SINCRONIZADO

El diseño sincronizado basado generalmente en tres ruedas, las ruedas generalmente siempre apuntan en la misma dirección, para poder girar el robot gira simultáneamente todas las ruedas en un eje vertical, de modo que la dirección del robot cambia.

2.13.8. DISEÑO DE TRICICLO Y COCHE

Proporciona buena estabilidad, el diseño en triciclo tiene unas prestaciones similares con la ventaja de ser mecánicamente más simple que el coche que necesita alguna unión entre las ruedas direccionales. En general en estos dos diseños las ruedas direccionales no son motrices, y no es necesario controlar la velocidad de las ruedas para que el robot se mantenga recto.

2.14. MOTORES Y SISTEMA DE ENERGIA

Al momento de elegir un motor para nuestro robot, se deben considerar ciertos parámetros como la velocidad, el torque, el frenado, la inercia y el modo de control.

Existen diferentes tipos de motores entre los que tenemos:

- Motores de corriente continua
- Motores de corriente continua con reductores
- Servomotores de los utilizados en modelismo y radiocontrol

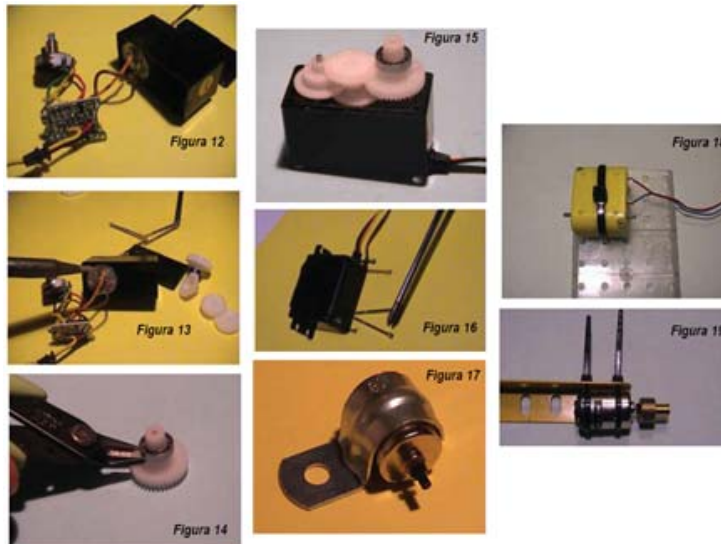


Figura 10: Ejemplos de varios tipos de motores [10]

2.15. PUENTE H

Un Puente H o Puente en H es un circuito electrónico usado para mover en ambos sentidos a motores eléctricos DC, usados mayormente en robótica y como convertidores de potencia. Están disponibles como circuitos integrados pero pueden construirse igualmente a partir de componentes discretos.

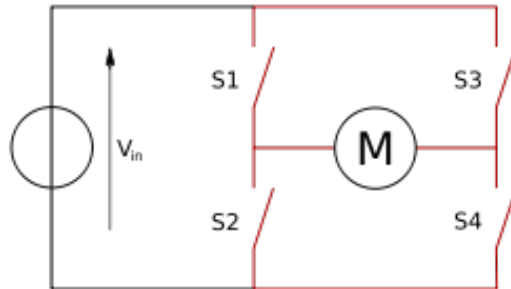


Figura 11: Configuración de puente en H [11]

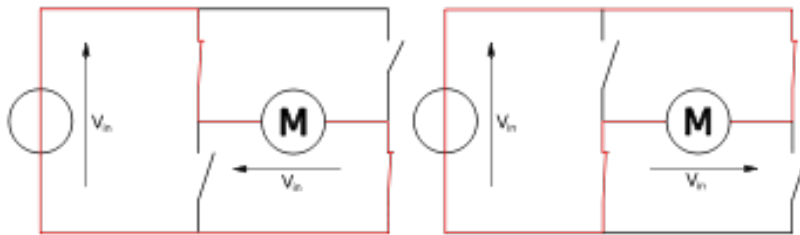


Figura 12: Los dos estados básicos del circuito [12]

Este término "puente H" proviene de su representación gráfica. Se construye con 4 interruptores. Cuando los interruptores S1 y S4 están cerrados y S2 y S3 abiertos se aplica un voltaje positivo en el motor haciéndolo girar. Abriendo los interruptores S1 y S4 y cerrando los S2 y S3 el voltaje en el motor se invierte ocasionando el giro en el sentido inverso del motor.

Debe estar claro que los interruptores S1 y S2 nunca podrán estar cerrados al mismo tiempo, ni el S3 y S4 porque esto provocaría un cortocircuito en la fuente de tensión.

Capítulo 3

3. DISEÑO E IMPLEMENTACION

3.1. DISEÑO DE LA ARQUITECTURA DEL ROBOT

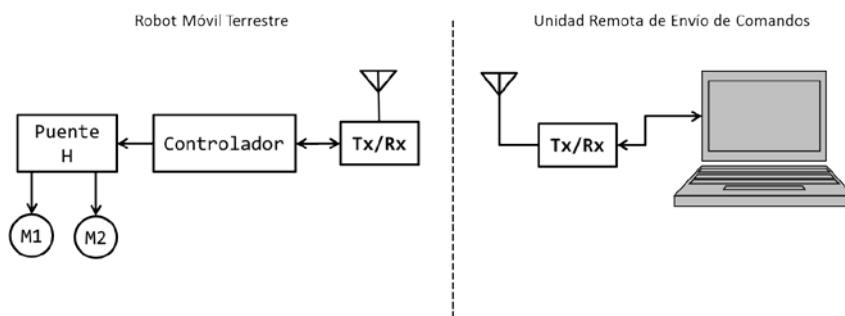


Figura 13: Diagrama de bloques del robot

La imagen arriba muestra los componentes del sistema robótico construido. La parte izquierda del dibujo presenta los módulos que conforman el robot móvil terrestre, caracterizado por estar formado de un componente transmisor/receptor de comandos (XBee XB24), un controlador integrado en la tarjeta de desarrollo DE2 de Altera, un puente H (L298) y dos motores DC.

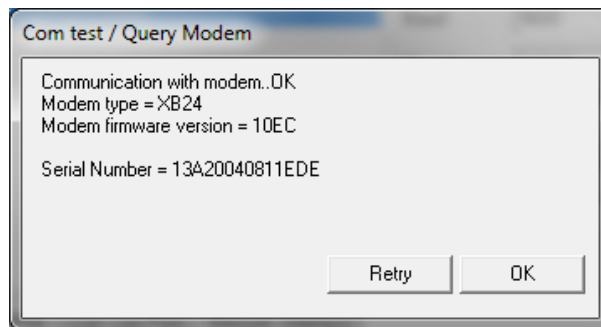


Figura 14: Prueba de modem

Por otro lado, la Unidad Remota de Envío de Comandos o componente presentado a la derecha del diagrama, esta formado por un módulo de envío de órdenes (XBee XB24) y un computador que envía las ordenes mediante el dispositivo inalámbrico, conectándose a este mediante RS232.

3.2. MORFOLOGIA

El Sistema Robótico que se construyó se lo puede dividir en dos componentes, por un lado el robot propiamente dicho, que tiene como características esenciales; ser terrestre, con locomoción por ruedas, basada en dos ruedas de manejo diferencial y una rueda loca. Además de tener comunicación con el centro de mando de manera inalámbrica mediante una interface ZigBEE implementada en un módulo XBee XB24.

Por otra parte el segundo componente se basa en una computadora embebida en la tarjeta DE2 de Altera, basada en el procesador NIOS II.



Figura 15: Vista del robot desde arriba

La figura 15 mostrada muestra el robot visto desde arriba, donde se puede apreciar fácilmente las dos ruedas de manejo diferencial, la tarjeta DE2 de Altera. También se puede ver el módulo XBee XB24 con su antena para la comunicación con la interface de computadora.



Figura 16: Vista interna del robot

En la figura X se puede ver la parte interna del robot; podemos desconectar la tarjeta DE2, ver los dos motores para controlar el movimiento diferencial de las ruedas, la tarjeta controladora que implementa el puente en H L298, conectado y alimentado todo el circuito por la batería recargable de 9.6V, 800mAh.



Figura 17: Driver

3.3. SISTEMA DE ENERGIA

El robot construido se alimenta de energía por un sistema de baterías recargables de tipo NiMH, que entrega 800mAh. Esta batería esta formada por 8 pilas arregladas de manera serial que contribuyen con 1.2V cada una para proporcionar un total de 9.6V.

La batería permite alimentar a la tarjeta DE2 y a los motores del robot, permitiendo la autonomía a nivel de alimentación de energía.

Se utiliza un regulador de tres terminales 7805 de 5V para proporcionar un correcto voltaje a los circuitos que trabajan a baja potencia y que funcionan a 5V.

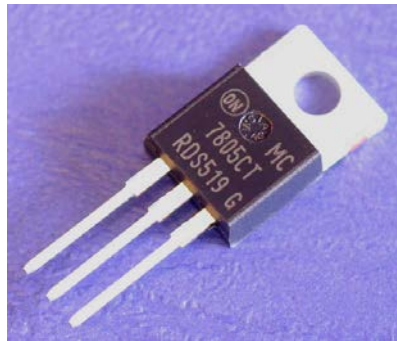


Figura 18: Regulador de 5V

3.4. DRIVER Y MOTORES

Comentario [R1]: Voy por aquí

Para el movimiento del robot, que tiene la característica de moverse hacia adelante, atrás, izquierda, derecha, utiliza un puente H L298 con driver incluido de máximo 2A, que permite general el control del sentido de giro de los motores.

El motor utilizado tiene las siguientes características de placa:

No.	ITEM	ESPECIFICACION
1	Tensión nominal	12V
2	Corriente sin carga	0.12A Max
3	Velocidad sin carga	29 RPM

4	Relación de engranajes	250:1
5	Velocidad nominal	24 RPM
6	Corriente nominal	0.29A Max
7	Torque nominal	5.8 Kg*cm

Tabla 1: Características de los motores

Las características técnicas del puente H L298 utilizado son:

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_o	Peak Output Current (each Channel)		
	- Non Repetitive (t = 100 μ s)	3	A
	- Repetitive (80% on -20% off; t _{on} = 10ms)	2.5	A
	- DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation (T _{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	°C
T_{slg}, T_j	Storage and Junction Temperature	-40 to 150	°C

Figura 19: Valores de los motores [13]

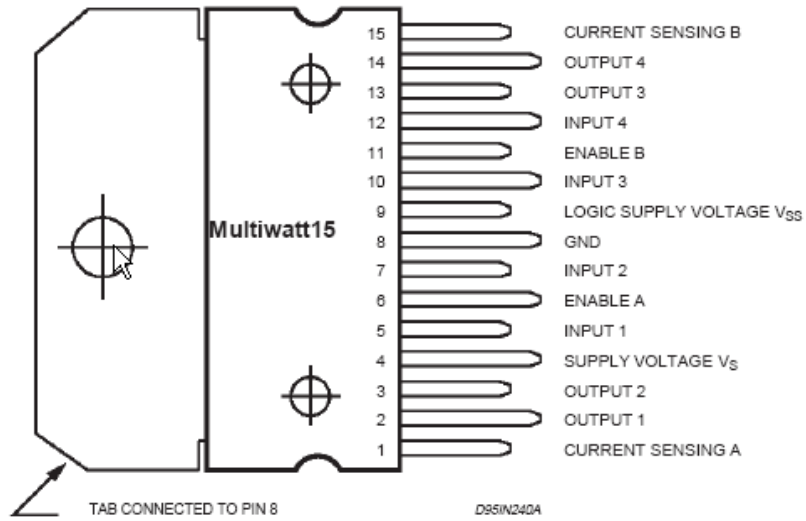


Figura 20: Mecánica del componente L298 [14]

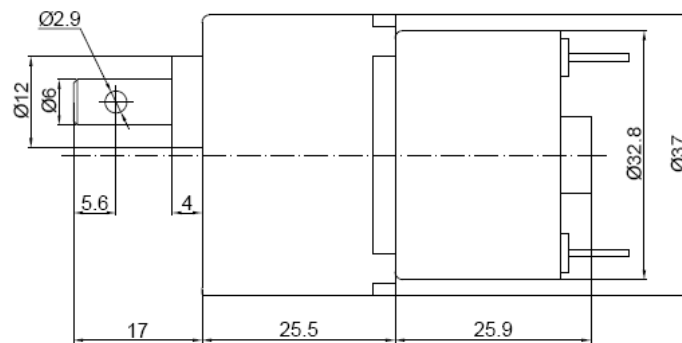


Figura 21: Dimensiones del motor lateral [15]

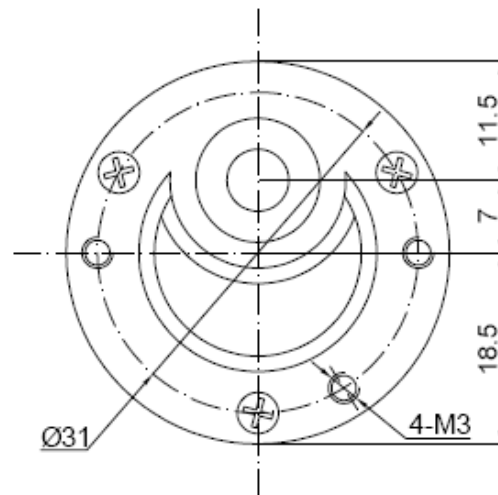


Figura 22: Diseño del motor frente [16]

3.5. COMUNICACIÓN

El robot construido es denominado como teleoperado, pues es controlado remotamente mediante las instrucciones de un computador, que envía las señales usando la implementación definida por el estándar IEEE 802.15.4 que se integra en el dispositivo de Digi Internacional, XBee XB24 10ED (ZigBee). Como tal se entiende que maneja comunicación inalámbrica y serial, en el que por el lado del PC se configura un dispositivo como coordinador y en el robot como dispositivo terminal o nodo.

3.6. LOGICA DE CONTROL

La lógica de control esta implementada en la tarjeta DE2 de Altera que integra un FPGA Cyclone II, en este se ha diseñado esta capa basándonos en la arquitectura del soft-processor NIOS II, donde se ha descrito la construcción de la computadora embebida.

3.7. DISEÑO DEL HARDWARE EN SOPC BUILDER

SOPC Builder o “System on a Programmable Chip Builder” es un aplicativo de Altera que automatiza la conexión de componentes soft-hardware, para crear sistemas de computación completos que corren en cualquiera FPGA.

En su arquitectura se encuentra una librería de componentes prefabricados que incluye el “soft-processor”, NIOS II, controladores de memoria, interfaces y periféricos.

Las capturas de pantalla mostradas abajo, representan el trabajo realizado en la construcción de la computadora embebida, mostrando cada uno de los componentes personalizados.

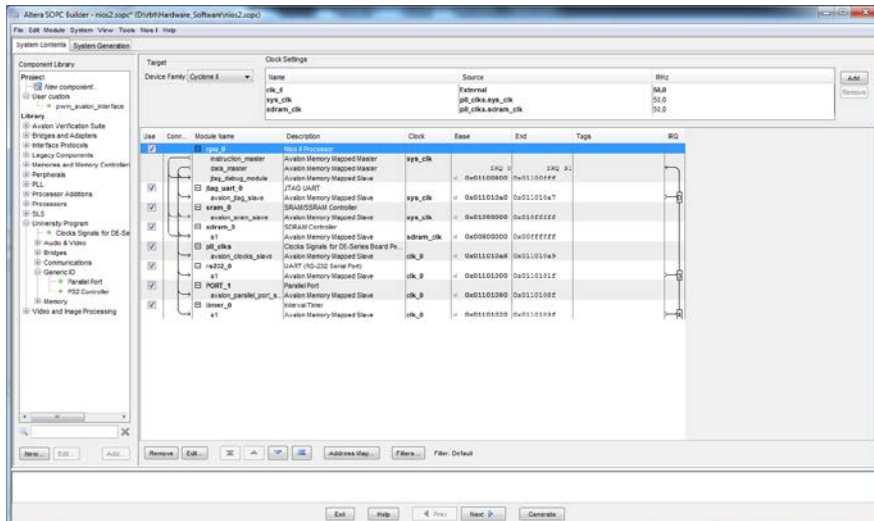


Figura 23: Captura de pantalla sistema embebido

La figura 23 muestra una captura de pantalla del SOPC Builder de Altera donde se puede revisar los componentes del sistema embebido diseñado. Podemos observar que esta conformado por un soft-processor NIOS II Processor, un JTAG UART, un controlador de memoria SRAM/SSRAM, un controlador SDRAM, un controlador de señales de reloj, un controlador UART para comunicación serial, un puerto paralelo y un timer.

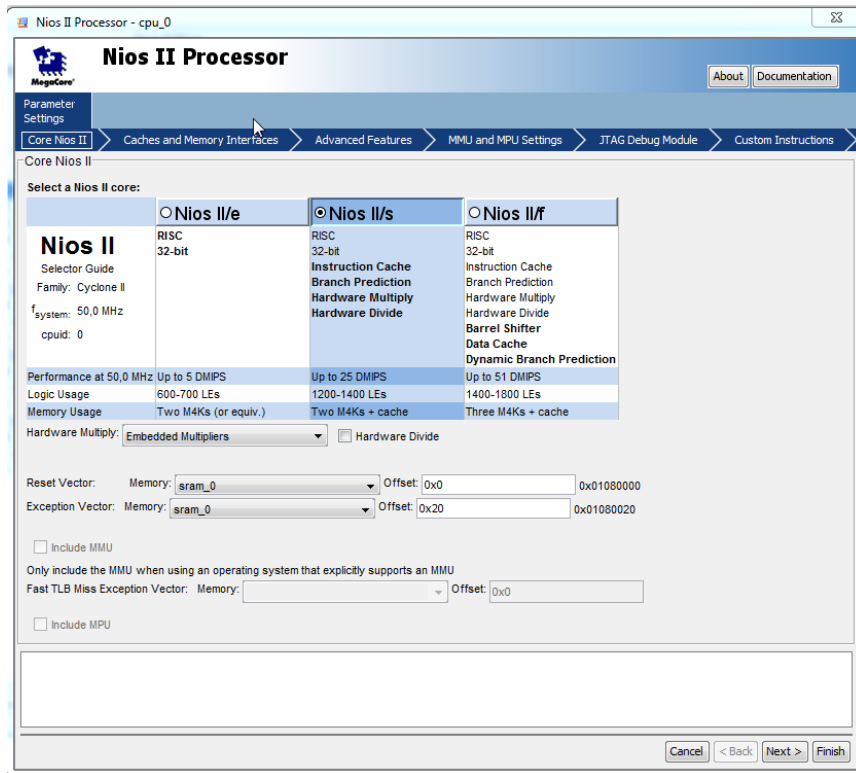


Figura 24: Valores de procesador seleccionado

La figura 24 muestra la versión del procesador seleccionado, el cual es NIOS II/s, se puede observar también la configuración del Reset Vector y del Exception Vector.

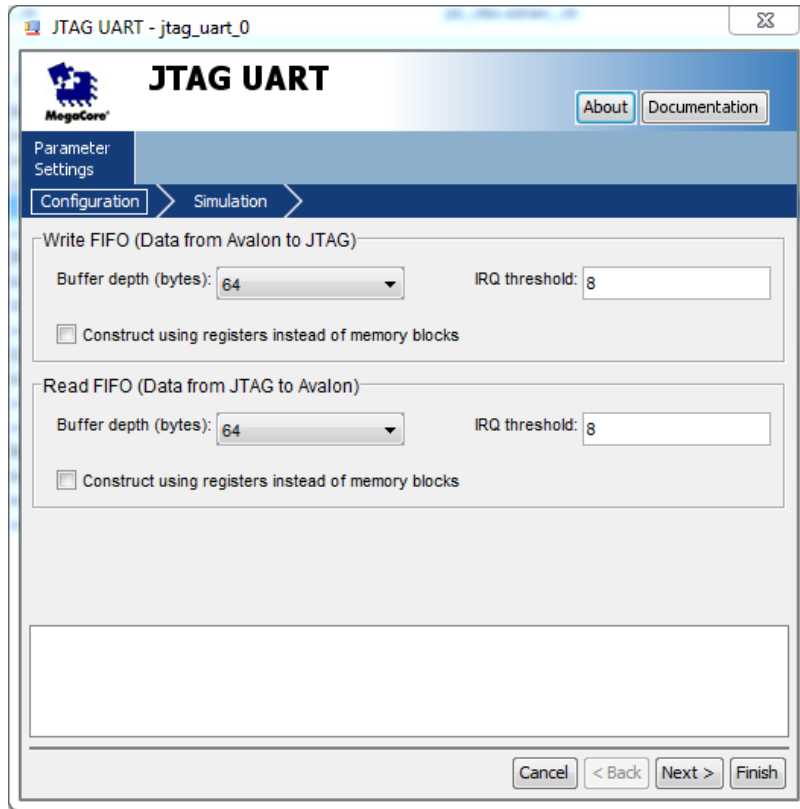


Figura 25: Configuración JTAG UART

La figura 25, muestra la interface para conectar la tarjeta con la aplicación NIOS IDE que es donde se programa en lenguaje C, el cual se carga a la computadora embebida mediante la conexión física por USB usando USB Blaster.

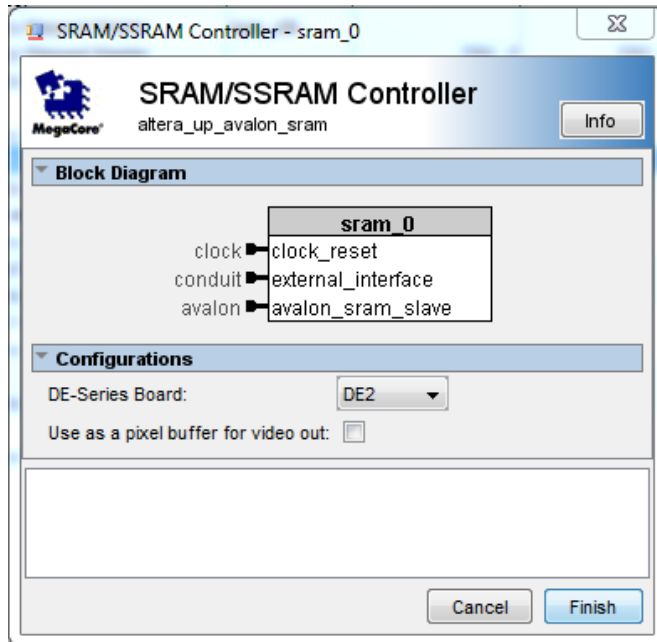


Figura 26: Configuración de memoria

En la figura 26 podemos ver el diagrama de bloque de la memoria principal.

The image shows a software window titled "SDRAM Controller - sdram_0". The window has a blue header with the "MegaCore" logo and the title "SDRAM Controller". There are "About" and "Documentation" buttons in the top right. Below the header, there are two tabs: "Memory Profile" (selected) and "Timing".

The main configuration area is divided into several sections:

- Presets:** A dropdown menu set to "Custom".
- Data width:** A section containing a "Bits" dropdown menu set to "16".
- Architecture:** A section containing "Chip select" and "Banks" dropdown menus, both set to "1" and "4" respectively.
- Address widths:** A section containing "Row" and "Column" input fields, both set to "12" and "8" respectively.
- Share pins via tristate bridge:** A section containing a checkbox "Controller shares dq/dqm/addr I/O pins" (unchecked) and a "Tristate bridge selection" dropdown menu.
- Generic memory model (simulation only):** A section containing a checkbox "Include a functional memory model in the system testbench" (checked).

At the bottom of the configuration area, the following text is displayed:

Memory size = 8 MBytes
4194304 x 16
64 Mbits

At the bottom of the window, there are four buttons: "Cancel", "< Back", "Next >", and "Finish".

Figura 27: Controlador SDRAM

Podemos ver en la figura 27, la configuración de la memoria para programas y datos.

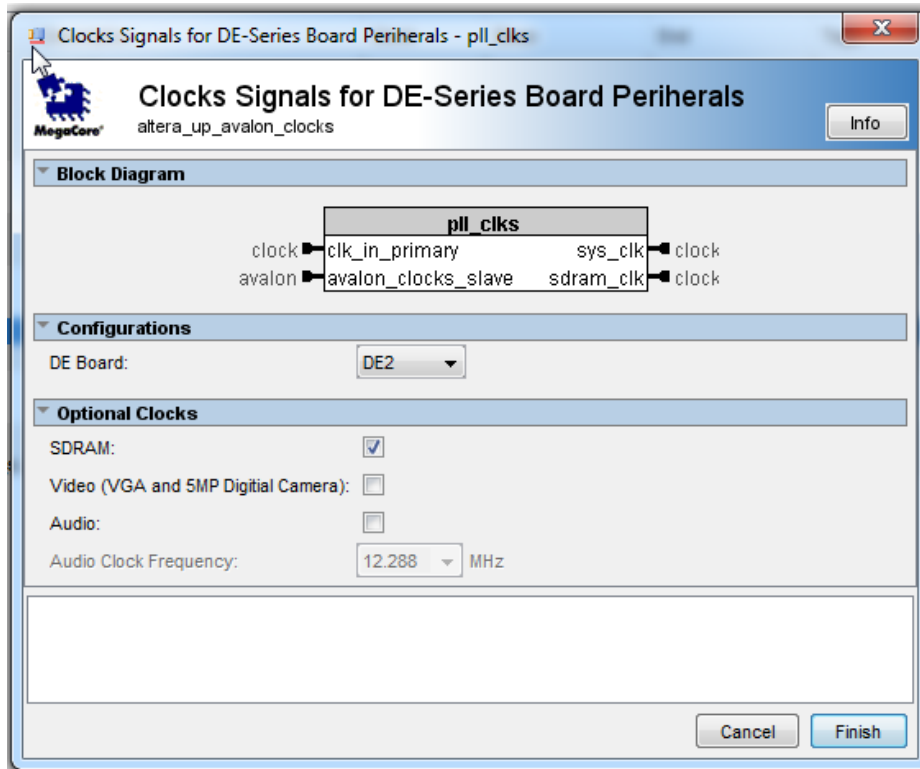


Figura 28: Configuración del reloj del sistema

La figura 28 indica la configuración del reloj del sistema, para multiplicar o dividir frecuencias de reloj, la tarjeta tiene un reloj de 50mhz que es el sysclock, utilizado solo para mantener la señal de reloj estable.

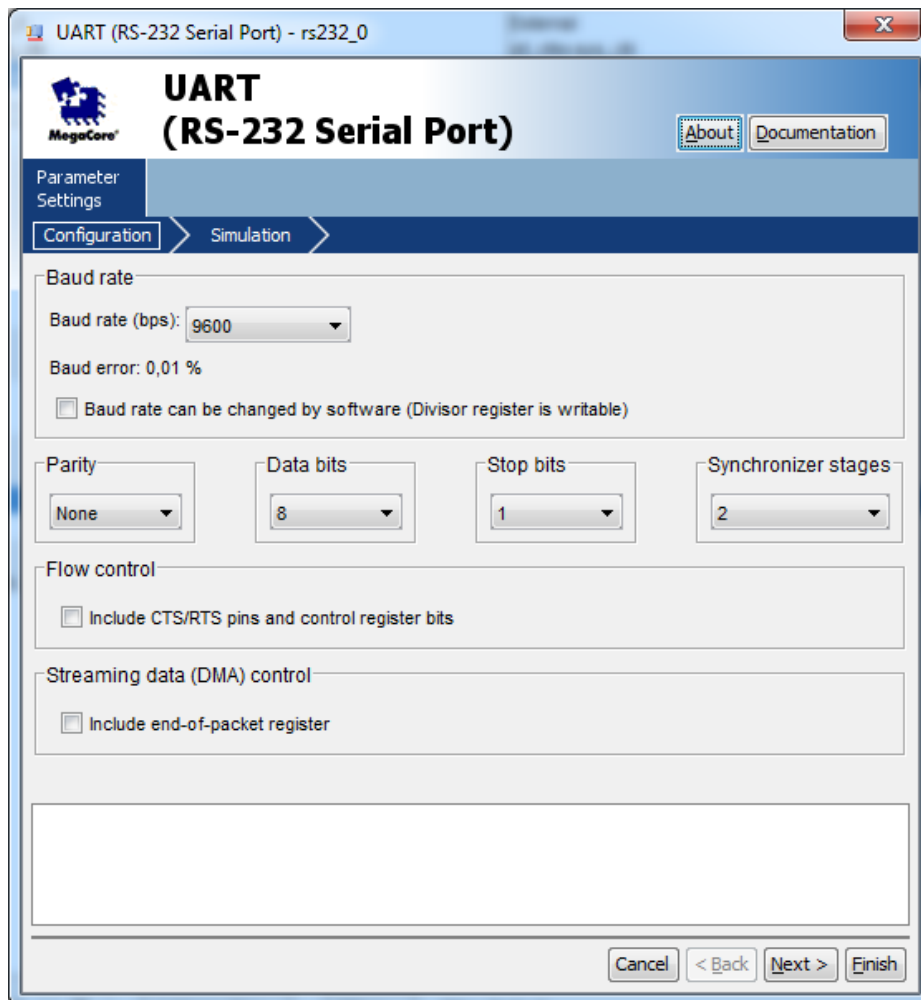


Figura 29: Configuración UART

La figura 29 muestra la configuración UART del sistema para comunicación mediante RS-232. Se puede apreciar los valores de configuración de la comunicación serial del lado del hardware implementado.

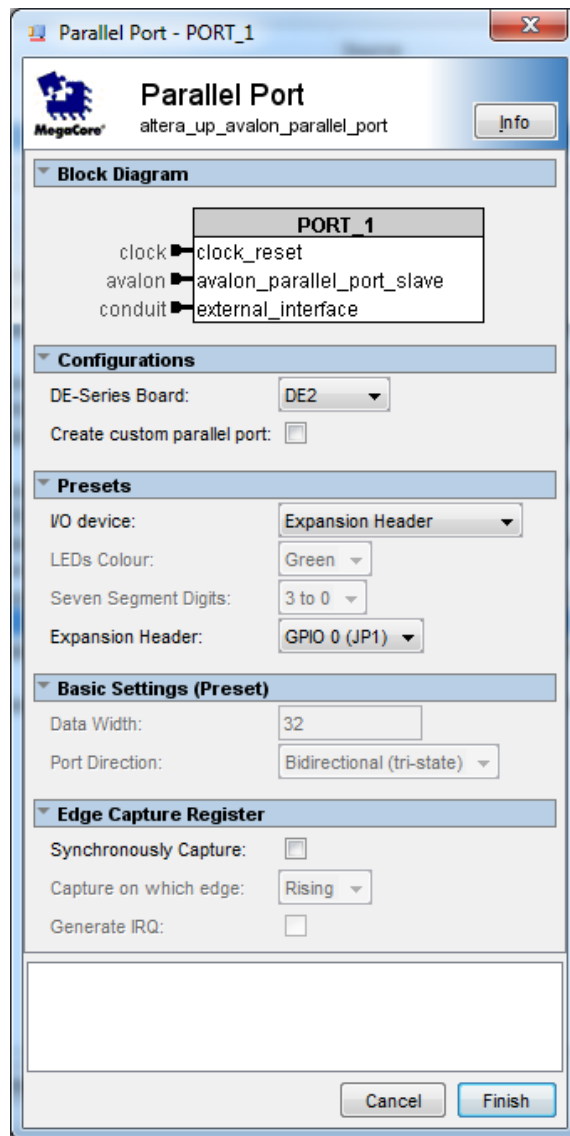


Figura 30: Definición de puerto paralelo

La figura 30 indica los parámetros de personalización de la interface de comunicación paralela, el cual es de entrada/salida, el cual físicamente es uno de los puertos de expansión de la DE2 y es de 32 bits.

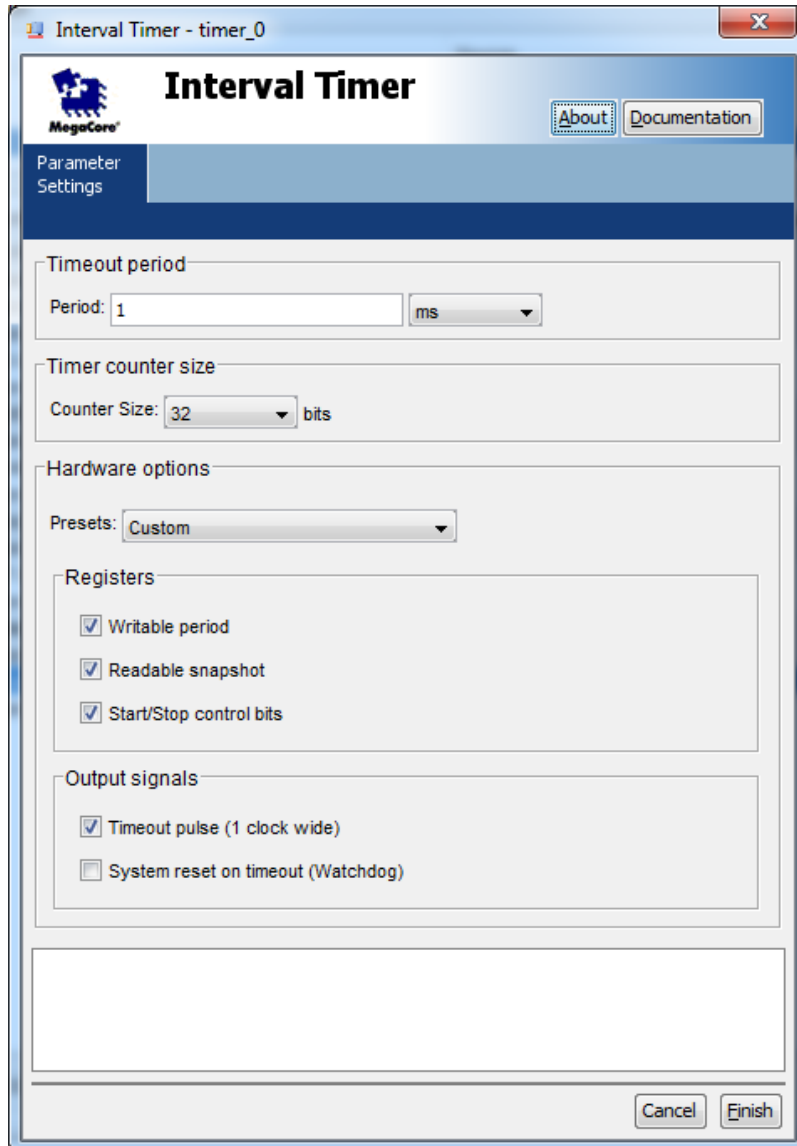


Figura 31: Paramterización del timer

La figura 31 muestra el Interval Timer usado para generar las funciones de temporización.

3.8. COMPILACION EN QUARTUS II

Una vez realizado el proyecto y ajustado los parametros para mejorar las funcionalidades del mismo, se procede a compilar por última vez el prototipo con la herramienta Quartus II y se obtiene los siguientes resultados de la compilación, mostrados en la figura 32.

Flow Status	Successful - Sun May 20 12:40:37 2012
Quartus II Version	9.1 Build 222 10/21/2009 SJ Full Version
Revision Name	pwm_source
Top-level Entity Name	pwm_source
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Met timing requirements	Yes
Total logic elements	5,178 / 33,216 (16 %)
Total combinational functions	4,229 / 33,216 (13 %)
Dedicated logic registers	3,287 / 33,216 (10 %)
Total registers	3339
Total pins	137 / 475 (29 %)
Total virtual pins	0
Total memory bits	46,720 / 483,840 (10 %)
Embedded Multiplier 9-bit elements	4 / 70 (6 %)
Total PLLs	1 / 4 (25 %)

Figura 32: Captura de pantalla de valores de compilación

3.9. ASIGNACION DE PINES EN LA FPGA

SEÑAL	PIN	I/O	BANCO	VREF GROUP
CLOCK_50	PIN_N2	2	B2_N1	3.3-V LVTTTL (default)

DRAM_ADDR[0]	PIN_T6	1	B1_N0	3.3-V LVTTTL (default)
DRAM_ADDR[1]	PIN_V4	1	B1_N0	3.3-V LVTTTL (default)
DRAM_ADDR[2]	PIN_V3	1	B1_N0	3.3-V LVTTTL (default)
DRAM_ADDR[3]	PIN_W2	1	B1_N0	3.3-V LVTTTL (default)
DRAM_ADDR[4]	PIN_W1	1	B1_N0	3.3-V LVTTTL (default)
DRAM_ADDR[5]	PIN_U6	1	B1_N1	3.3-V LVTTTL (default)
DRAM_ADDR[6]	PIN_U7	1	B1_N1	3.3-V LVTTTL (default)
DRAM_ADDR[7]	PIN_U5	1	B1_N1	3.3-V LVTTTL (default)
DRAM_ADDR[8]	PIN_W4	1	B1_N1	3.3-V LVTTTL (default)
DRAM_ADDR[9]	PIN_W3	1	B1_N1	3.3-V LVTTTL (default)
DRAM_ADDR[10]	PIN_Y1	1	B1_N1	3.3-V LVTTTL (default)
DRAM_ADDR[11]	PIN_V5	1	B1_N1	3.3-V LVTTTL (default)
DRAM_BA_0	PIN_AE2	1	B1_N1	3.3-V LVTTTL (default)
DRAM_BA_1	PIN_AE3	1	B1_N1	3.3-V LVTTTL (default)
DRAM_CAS_N	PIN_AB3	1	B1_N1	3.3-V LVTTTL (default)

DRAM_CKE	PIN_AA6	1	B1_N1	3.3-V LVTTTL (default)
DRAM_CLK	PIN_AA7	1	B1_N1	3.3-V LVTTTL (default)
DRAM_CS_N	PIN_AC3	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[0]	PIN_V6	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[1]	PIN_AA2	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[2]	PIN_AA1	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[3]	PIN_Y3	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[4]	PIN_Y4	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[5]	PIN_R8	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[6]	PIN_T8	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[7]	PIN_V7	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[8]	PIN_W6	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[9]	PIN_AB2	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[10]	PIN_AB1	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[11]	PIN_AA4	1	B1_N1	3.3-V LVTTTL (default)

DRAM_DQ[12]	PIN_AA3	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[13]	PIN_AC2	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[14]	PIN_AC1	1	B1_N1	3.3-V LVTTTL (default)
DRAM_DQ[15]	PIN_AA5	1	B1_N1	3.3-V LVTTTL (default)
DRAM_LDQM	PIN_AD2	1	B1_N1	3.3-V LVTTTL (default)
DRAM_UDQM	PIN_Y5	1	B1_N1	3.3-V LVTTTL (default)
DRAM_RAS_N	PIN_AB4	1	B1_N1	3.3-V LVTTTL (default)
DRAM_WE_N	PIN_AD3	1	B1_N1	3.3-V LVTTTL (default)
UART_RXD	PIN_K25	5	B5_N1	3.3-V LVTTTL (default)
UART_TXD	PIN_K26	5	B5_N1	3.3-V LVTTTL (default)
SRAM_ADDR[0]	PIN_AE4	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[1]	PIN_AF4	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[2]	PIN_AC5	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[3]	PIN_AC6	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[4]	PIN_AD4	8	B8_N1	3.3-V LVTTTL (default)

SRAM_ADDR[5]	PIN_AD5	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[6]	PIN_AE5	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[7]	PIN_AF5	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[8]	PIN_AD6	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[9]	PIN_AD7	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[10]	PIN_V10	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[11]	PIN_V9	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[12]	PIN_AC7	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[13]	PIN_W8	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[14]	PIN_W10	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[15]	PIN_Y10	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[16]	PIN_AB8	8	B8_N1	3.3-V LVTTTL (default)
SRAM_ADDR[17]	PIN_AC8	8	B8_N1	3.3-V LVTTTL (default)
SRAM_DQ[0]	PIN_AD8	8	B8_N1	3.3-V LVTTTL (default)
SRAM_DQ[1]	PIN_AE6	8	B8_N1	3.3-V LVTTTL (default)

SRAM_DQ[2]	PIN_AF6	8	B8_N1	3.3-V LVTTTL (default)
SRAM_DQ[3]	PIN_AA9	8	B8_N1	3.3-V LVTTTL (default)
SRAM_DQ[4]	PIN_AA10	8	B8_N1	3.3-V LVTTTL (default)
SRAM_DQ[5]	PIN_AB10	8	B8_N1	3.3-V LVTTTL (default)
SRAM_DQ[6]	PIN_AA11	8	B8_N1	3.3-V LVTTTL (default)
SRAM_DQ[7]	PIN_Y11	8	B8_N1	3.3-V LVTTTL (default)
SRAM_DQ[8]	PIN_AE7	8	B8_N1	3.3-V LVTTTL (default)
SRAM_DQ[9]	PIN_AF7	8	B8_N1	3.3-V LVTTTL (default)
SRAM_DQ[10]	PIN_AE8	8	B8_N0	3.3-V LVTTTL (default)
SRAM_DQ[11]	PIN_AF8	8	B8_N0	3.3-V LVTTTL (default)
SRAM_DQ[12]	PIN_W11	8	B8_N0	3.3-V LVTTTL (default)
SRAM_DQ[13]	PIN_W12	8	B8_N0	3.3-V LVTTTL (default)
SRAM_DQ[14]	PIN_AC9	8	B8_N0	3.3-V LVTTTL (default)
SRAM_DQ[15]	PIN_AC10	8	B8_N0	3.3-V LVTTTL (default)
SRAM_WE_N	PIN_AE10	8	B8_N0	3.3-V LVTTTL (default)

SRAM_OE_N	PIN_AD10	8	B8_N0	3.3-V LVTTTL (default)
SRAM_UB_N	PIN_AF9	8	B8_N0	3.3-V LVTTTL (default)
SRAM_LB_N	PIN_AE9	8	B8_N0	3.3-V LVTTTL (default)
SRAM_CE_N	PIN_AC11	8	B8_N0	3.3-V LVTTTL (default)
SW[0]	PIN_N25	5	B5_N1	3.3-V LVTTTL (default)
SW[1]	PIN_N26	5	B5_N1	3.3-V LVTTTL (default)
SW[2]	PIN_P25	6	B6_N0	3.3-V LVTTTL (default)
SW[3]	PIN_AE14	7	B7_N1	3.3-V LVTTTL (default)
SW[4]	PIN_AF14	7	B7_N1	3.3-V LVTTTL (default)
SW[5]	PIN_AD13	8	B8_N0	3.3-V LVTTTL (default)
SW[6]	PIN_AC13	8	B8_N0	3.3-V LVTTTL (default)
SW[7]	PIN_C13	3	B3_N0	3.3-V LVTTTL (default)
SW[8]	PIN_B13	4	B4_N1	3.3-V LVTTTL (default)
SW[9]	PIN_A13	4	B4_N1	3.3-V LVTTTL (default)
SW[10]	PIN_N1	2	B2_N1	3.3-V LVTTTL (default)

SW[11]	PIN_P1	1	B1_N0	3.3-V LVTTTL (default)
SW[12]	PIN_P2	1	B1_N0	3.3-V LVTTTL (default)
SW[13]	PIN_T7	1	B1_N0	3.3-V LVTTTL (default)
SW[14]	PIN_U3	1	B1_N0	3.3-V LVTTTL (default)
SW[15]	PIN_U4	1	B1_N0	3.3-V LVTTTL (default)
SW[16]	PIN_V1	1	B1_N0	3.3-V LVTTTL (default)
SW[17]	PIN_V2	1	B1_N0	3.3-V LVTTTL (default)
KEY[0]	PIN_G26	5	B5_N0	3.3-V LVTTTL (default)
KEY[1]	PIN_N23	5	B5_N1	3.3-V LVTTTL (default)
KEY[2]	PIN_P23	6	B6_N0	3.3-V LVTTTL (default)
KEY[3]	PIN_W26	6	B6_N1	3.3-V LVTTTL (default)
PWM_1	PIN_AF23	7	B7_N0	3.3-V LVTTTL (default)
PWM_2	PIN_AB21	7	B7_N0	3.3-V LVTTTL (default)
PORT_1[0]	PIN_D25	5	B5_N0	3.3-V LVTTTL (default)
PORT_1[1]	PIN_J22	5	B5_N0	3.3-V LVTTTL (default)

PORT_1[2]	PIN_E26	5	B5_N0	3.3-V LVTTTL (default)
PORT_1[3]	PIN_E25	5	B5_N0	3.3-V LVTTTL (default)
PORT_1[4]	PIN_F24	5	B5_N0	3.3-V LVTTTL (default)
PORT_1[5]	PIN_F23	5	B5_N0	3.3-V LVTTTL (default)

Tabla 2: Tabla de asignación de pines del circuito

3.10. CONFIGURACION DE LOS MODULOS XBEE

Mediante la aplicación X-CTU provista por Digi International, podemos configurar los módulos de XBee a utilizar.

Una vez instalado y conectado el módulo de XBee en el computador que trabajará como anfitrión, procederemos a ejecutar el programa, buscándolo en el menú Inicio de Windows

El programa buscará automáticamente los puertos COM que se encuentren disponibles en el sistema, sean virtuales o físicos y los listará de manera que podamos seleccionar como muestra la figura el puerto a usar.

Se deberá establecer los parámetros de baudios, control de flujo, bit de datos, paridad y bit de parada; tal como se muestra en la figura 33.

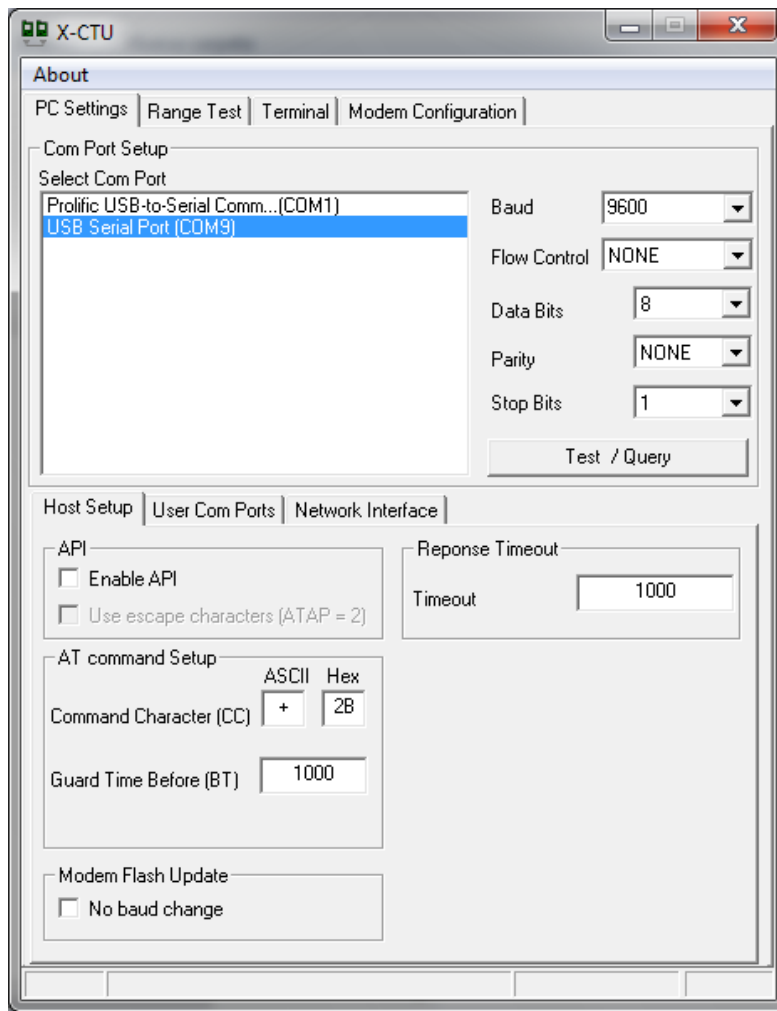


Figura 33: Configuración de la conexión serial

Una vez realizada la configuración de estos parámetros, se presionara el botón Test/Query, el cual lanzará el cuadro de dialogo mostrado en la figura, que indica el estado de conexión con el modem, tipo, firmware y numero de serie del dispositivo XBee conectado en el puerto testado.

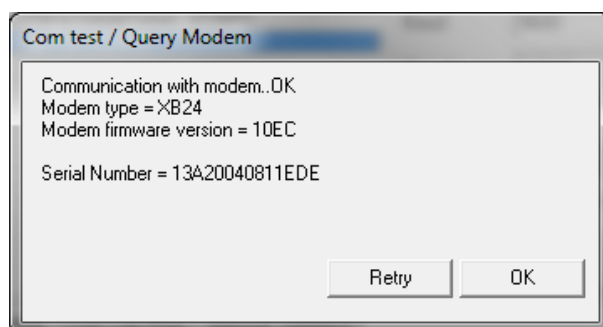


Figura 34: Pantalla de valores de dispositivo inalambrico

Procedemos a hacer clic en la ficha Modem Configuration, donde podemos cambiar los parametros del modem para indicar los valores de este. Al presionar el boton Restore podemos recuperar las configuraciones que hayan sido guardadas anteriormente.

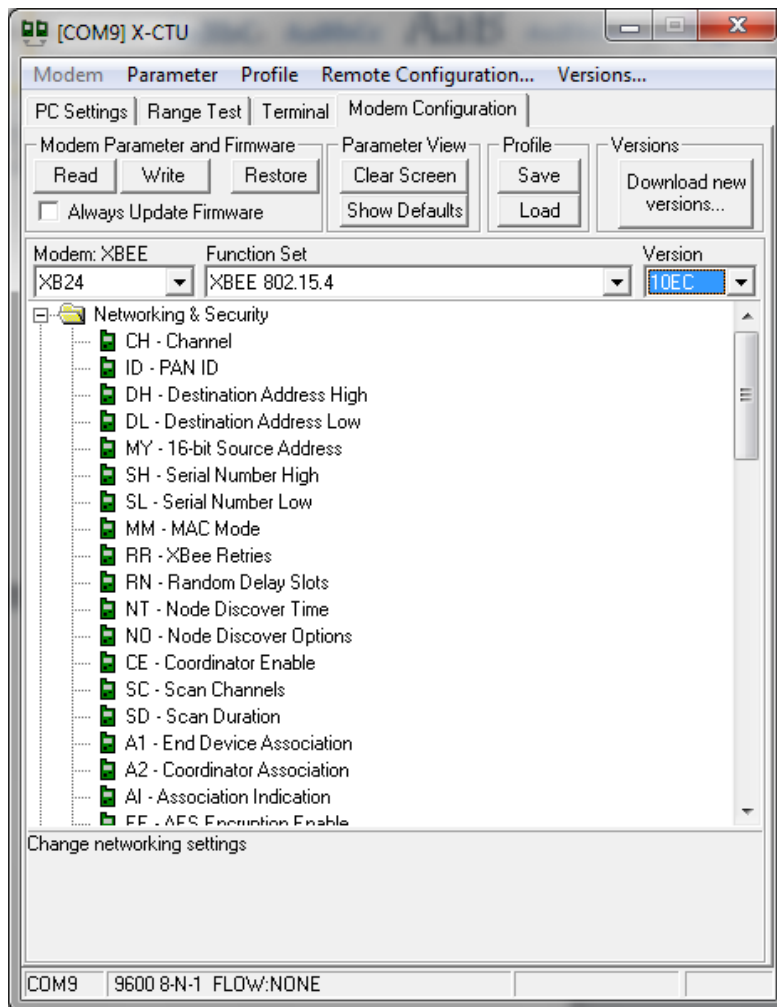


Figura 35: Parametrización de dispositivos XBee

En la ventana Terminal se muestra el estado de conexión del dispositivo.

Nos indica que las líneas de handshaking Clear to Send esta activa.

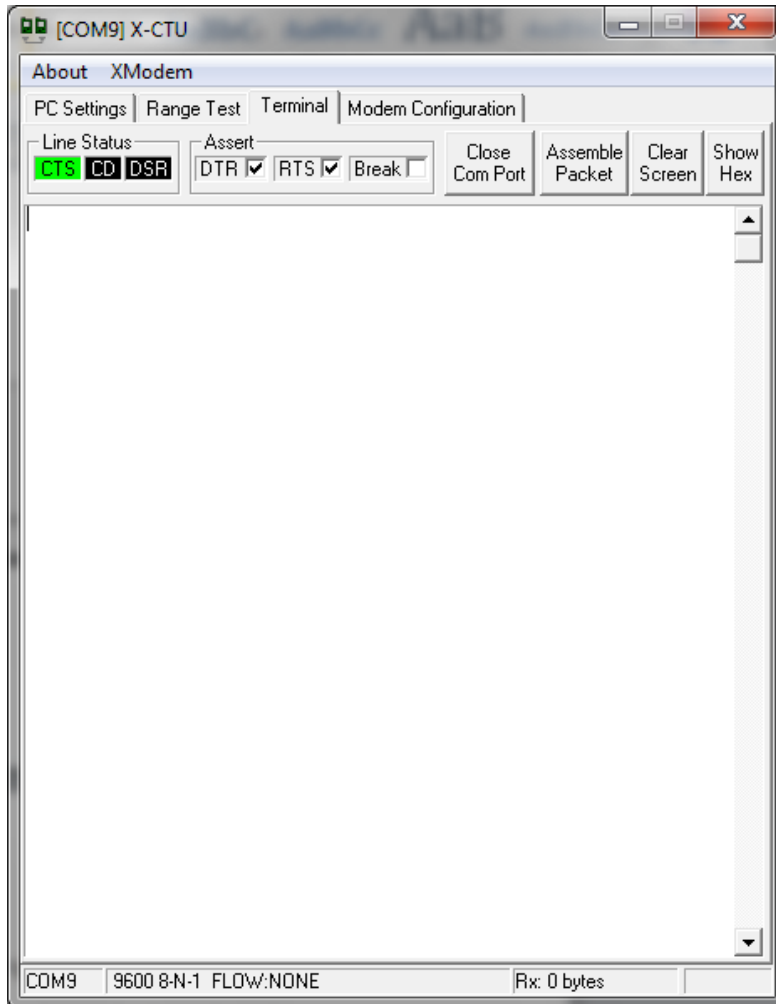


Figura 36: Prueba de estados de conexión

3.11. CODIGO DEL PROGRAMA PRINCIPAL

La tarjeta programable DE2 envía señales de activación al controlador formado por el puente en H L298 que define su comportamiento gracias a la implementación en lenguaje C de las funciones que envían al puerto

serial RS232 los comandos indicados para producir el movimiento de los motores.

El código abajo indicado, es la implementación del módulo principal de comunicación entre los componentes del robot. Aquí se puede observar la utilización del dispositivo rs232 al cual llamamos COM y que se encuentra en la ruta /dev/rs232_0 de la tarjeta de desarrollo DE2.

```
#include <stdio.h>
#include "system.h"
#include "principal.h"
//Dispositivo serial RS232 de la tarjeta controladora
#define COM "/dev/rs232_0"

int main()
{
    inicializar();
    unsigned int comando;
    FILE* fp;
    //Abrimos la interface serial para escribir en ella
    fp = fopen (COM,"r+");
    printf ("Comunicacion RS232 exitosa...\n");
    while (1)
    {
        //Captura de ordenes mediante interface serial
        comando = getc(fp);
        movimiento(comando);
        usleep(200000);
    }
    return 0;
}
```

Figura 37: Código PRINCIPAL.C

Se ha creado una librería principal.h para implementar y declarar los recursos de programación del dispositivo serial de la DE2. La función

movimiento utilizada en el programa principal, recibe un entero sin signo que indica el código orden de movimiento para el robot, código que se envía al puerto abierto indicado en el programa principal.

```

#ifndef PRINCIPAL_H_
#define PRINCIPAL_H_

volatile unsigned int *PORT1 = (unsigned int *) PORT_1_BASE;
volatile unsigned int *PORT1_DIR = PORT_1_BASE+4;

void inicializar ( void )
{ //Seteo de bits del PORT1 como salidas
  *PORT1_DIR = 0xffffffff;
}

void movimiento(unsigned int orden)
{ //0.- Detener ;1.- adelante ;2.- atras 3.- derecha ;4.- izquierda
  switch (orden){
    case (48):
      *PORT1 = 0x00;
      break;
    case (49):
      *PORT1 = 0x36;
      break;
    case (50):
      *PORT1 = 0x39;
      break;
    case (51):
      *PORT1 = 0x3A;
      break;
    case (52):
      *PORT1 = 0x35;
      break;
    default:
      break;
  }
}
#endif /* PRINCIPAL_H_ */

```

Figura 38: Definición de librería PRINCIPAL.H

3.12. PCB Y ENSAMBLADO FINAL

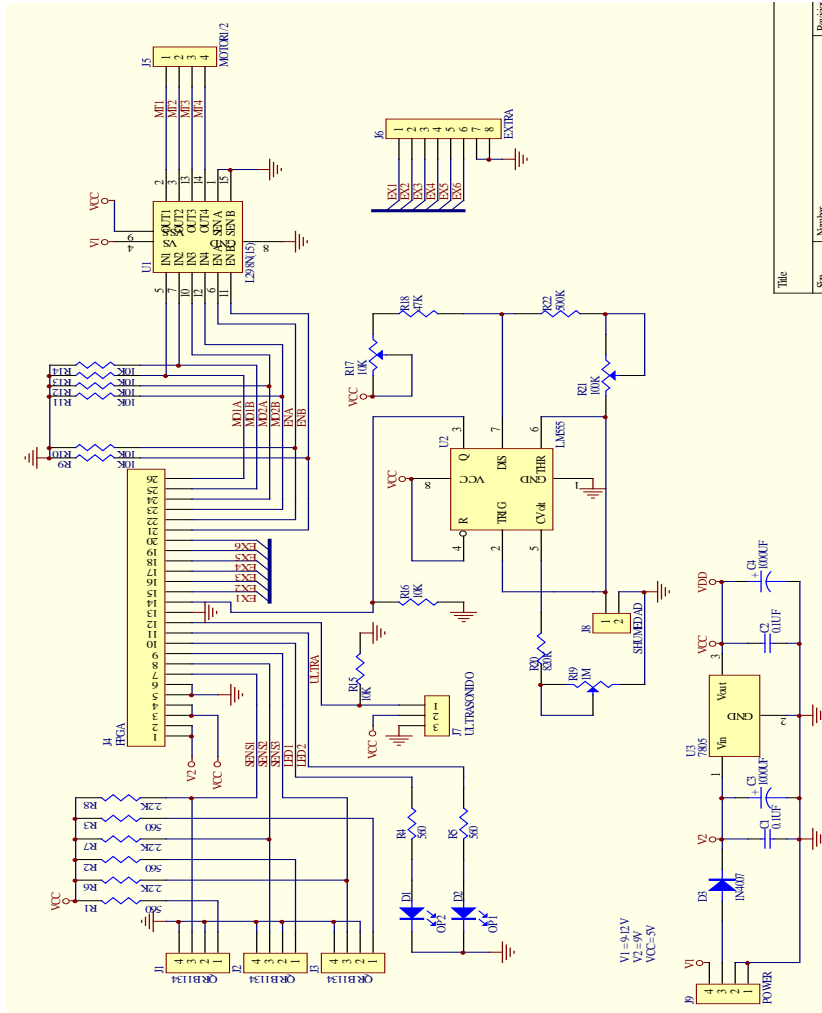


Figura 39: Esquemático del circuito adicional

El gráfico arriba mostrado es el esquemático de la implementación del PCB del robot adicional, en la capa del controlador. Se observa conformado por un regulador 7805 de tres terminales que provee 5V al controlador.

También se representa al puente en H L298, con las conexiones hacia la parte lógica de la tarjeta DE2 de Altera y conexiones hacia la sección de fuerza que controla los motores.

Se muestran también los conectores de la interface con la tarjeta DE2.

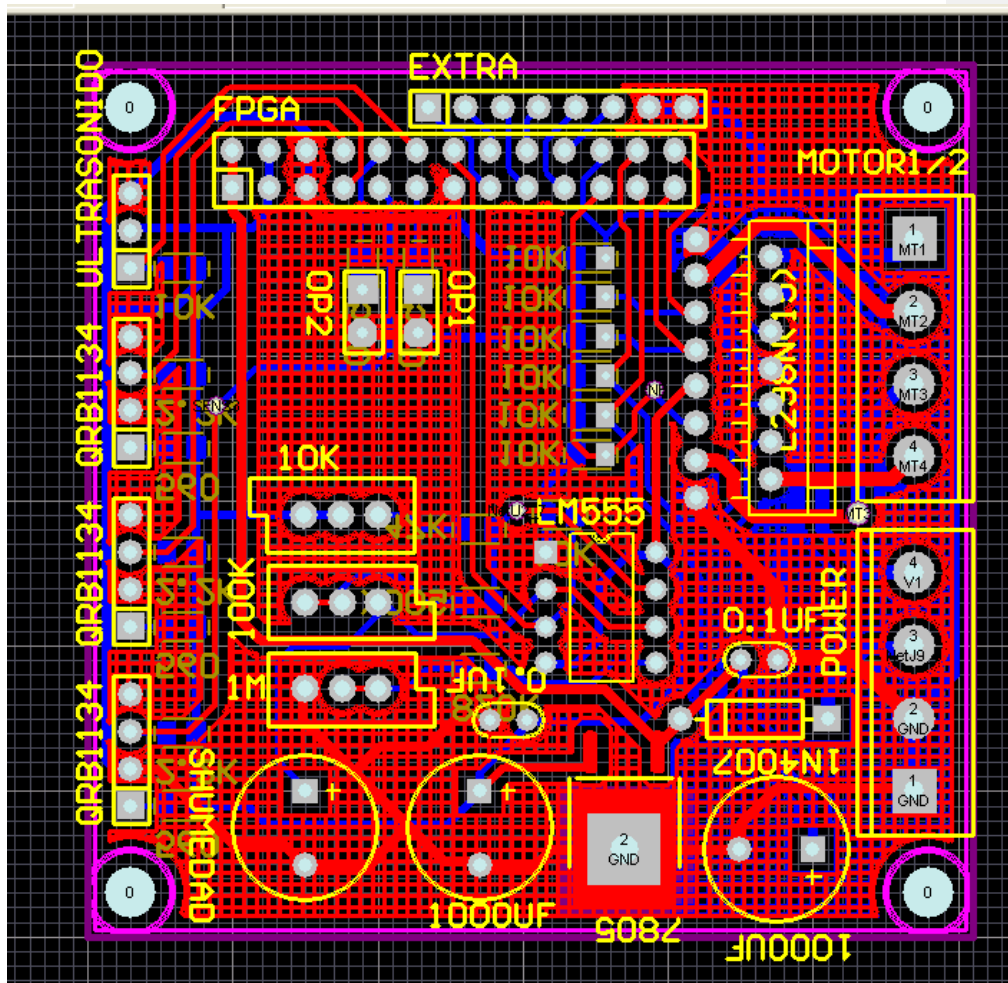


Figura 40: PCB terminado con distribución de elementos

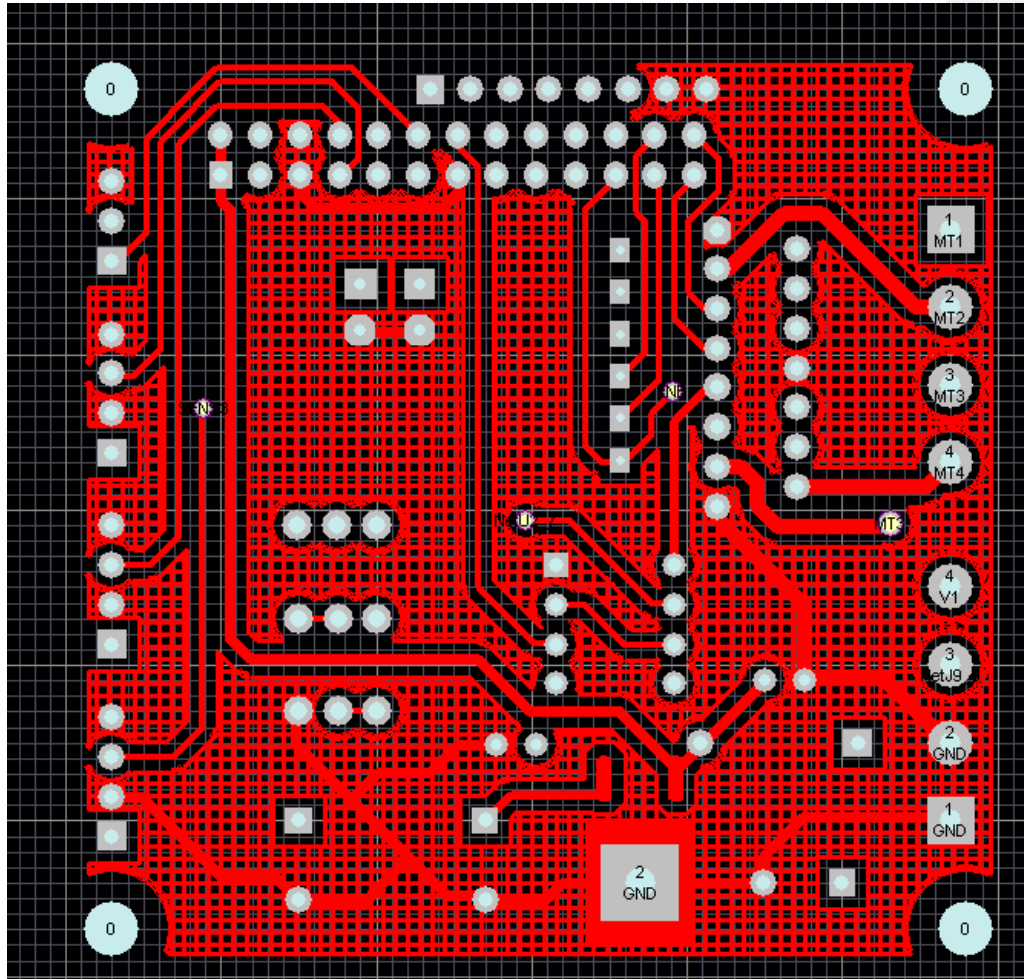


Figura 41: PCB a doble cara, este es el top layer del PCB

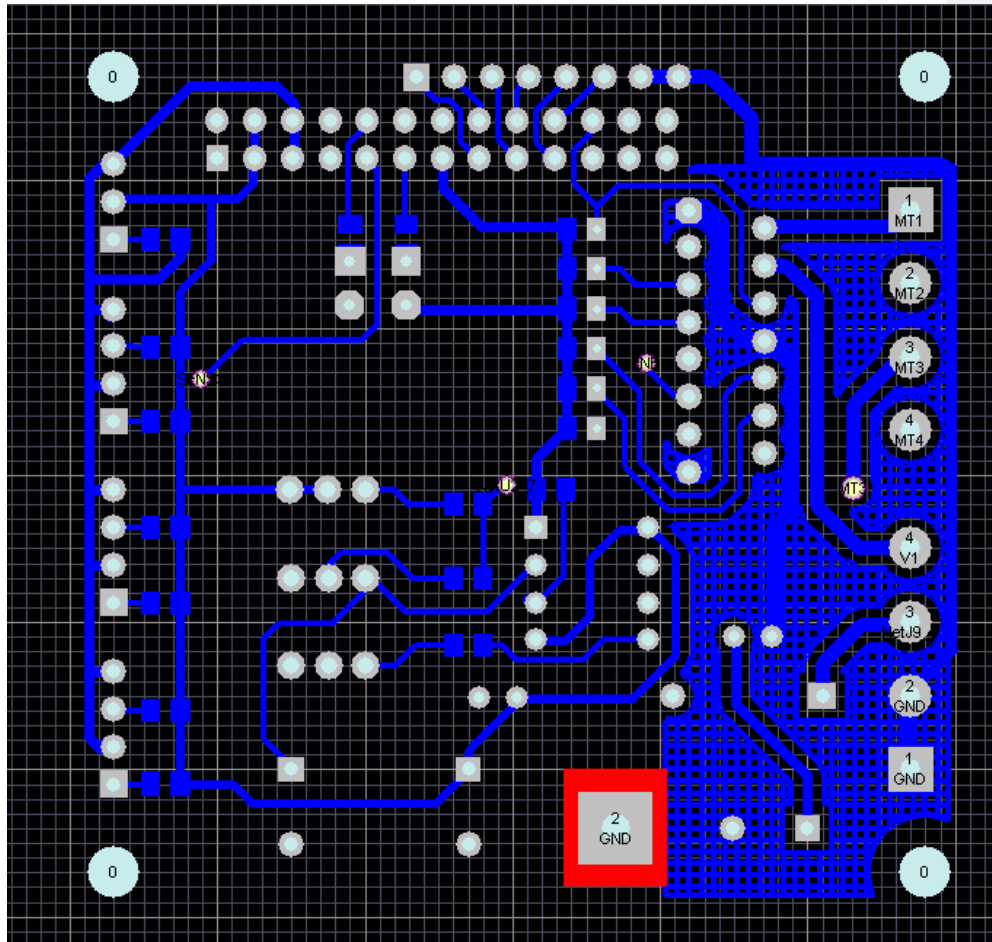


Figura 42: Bottom Layer del PCB

Capítulo 4

4. PRUEBAS Y RESULTADOS

Este capítulo resume lo realizado en las pruebas del producto, como muestra también los resultados de estas evaluaciones. Vemos como el robot es controlado mediante la interface XBee.

Para la realización de las pruebas se creó un sistema llamado Control Robot en .Net usando C# como lenguaje de programación, el cual envía por el puerto serial seleccionado la bandera que activará el movimiento del robot.

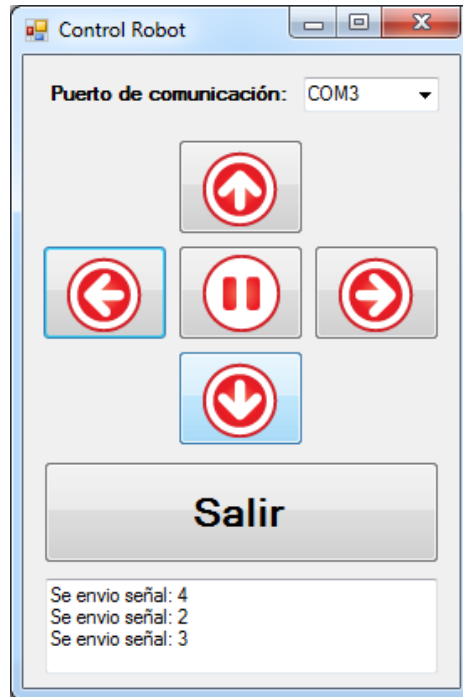


Figura 43: Pantalla de programa de prueba

4.1. CASOS DE PRUEBAS

Con nuestra herramienta desarrollada, se procedió a realizar las pruebas que se describen a continuación, para todos los casos las superficies de prueba fueron lisas e irregulares, además que los comandos a enviados fueron:

- 1.- Adelante
- 2.- Atrás

3.- Derecha

4.- Izquierda

0.- Detener

DISTANCIA (m)	ESCENARIO	COMANDO	RESULTADO
Con línea de vista			
1	El robot se desplaza hacia adelante	1	OK
1	El robot se desplaza hacia atrás	2	OK
1	El robot gira hacia la derecha	3	OK
1	El robot gira a la izquierda	4	OK
1	El robot se detiene	0	OK
5	El robot se desplaza hacia adelante	1	OK
5	El robot se desplaza hacia atrás	2	OK
5	El robot gira hacia la derecha	3	OK
5	El robot gira a la izquierda	4	OK
5	El robot se detiene	0	OK
10	El robot se desplaza hacia adelante	1	OK
10	El robot se desplaza hacia atrás	2	OK
10	El robot gira hacia la derecha	3	OK
10	El robot gira a la izquierda	4	OK
10	El robot se detiene	0	OK
15	El robot se desplaza hacia adelante	1	OK
15	El robot se desplaza hacia atrás	2	OK

15	El robot gira hacia la derecha	3	OK
15	El robot gira a la izquierda	4	OK
15	El robot se detiene	0	OK
Con obstáculos línea de vista			
1	El robot se desplaza hacia adelante	1	OK
1	El robot se desplaza hacia atrás	2	OK
1	El robot gira hacia la derecha	3	OK
1	El robot gira a la izquierda	4	OK
1	El robot se detiene	0	OK
5	El robot se desplaza hacia adelante	1	OK
5	El robot se desplaza hacia atrás	2	OK
5	El robot gira hacia la derecha	3	OK
5	El robot gira a la izquierda	4	OK
5	El robot se detiene	0	OK
10	El robot se desplaza hacia adelante	1	OK
10	El robot se desplaza hacia atrás	2	OK
10	El robot gira hacia la derecha	3	OK
10	El robot gira a la izquierda	4	OK
10	El robot se detiene	0	OK
15	El robot se desplaza hacia adelante	1	NO OK
15	El robot se desplaza hacia atrás	2	NO OK
15	El robot gira hacia la derecha	3	NO OK
15	El robot gira a la izquierda	4	NO OK
15	El robot se detiene	0	NO OK

4.2. RESULTADOS

De los resultados obtenidos podemos indicar que el robot tiene una buena comunicación con la central de control mientras existas línea de vista hasta aproximadamente 15 metros, y en promedio 10 metros en ambientes como casas donde las paredes generan un obstáculo para la comunicación.

CONCLUSIONES

1. El sistema robótico implementado cumple con el objetivo de crear un prototipo plataforma de desarrollo y aprendizaje de tecnologías aplicables a la robótica.
2. La velocidad de comunicación es aceptable, pues no existen retardos visibles en la ejecución de la orden por medio del robot.

RECOMENDACIONES

1. Para diseño de mejoras futuras a este proyecto se puede considerar aumentar la potencia ofrecida por las baterías colocando arreglos de estas que puedan ofrecer mayor voltaje y corriente, debido a que en las pruebas se observó la descarga de la batería usada en tiempos breves de 5 a 10 minutos de operación.
2. También se puede considerar aumentar el rango de distancia cubierta por el dispositivo XBee XB24.
3. Debido a que el proyecto es una plataforma robótica básica para futuras investigaciones, se pueden instalar varias opciones de sensores y dispositivos de entrada y salida adicionales y sus respectivos programas de control para con ello utilizar la plataforma implementada.

BIBLIOGRAFIA

WIKIPEDIA, Especificaciones del protocolo ZigBee (disponible en http://es.wikipedia.org/wiki/ZigBee_%28especificaci%C3%B3n%29 consultado el 12 de diciembre de 2011)

JOBSCOCHIN, Tecnologías inalámbricas del futuro (disponible en <http://jobscochin.com/zigbee-%E2%80%93an-emerging-technology-future> consultado el 15 de febrero de 2012)

WIKIPEDIA, Conceptos de robótica (disponible en <http://es.wikipedia.org/wiki/Rob%C3%B3tica> consultado el 14 de enero de 2012)

ALEGSA, Definiciones comunes de robots, (disponible en <http://www.alegsa.com.ar/Dic/robot.php> consultado el 13 de Enero de 2012)

ALTERA, Manual de tarjeta de desarrollo DE2 (disponible en ftp://ftp.altera.com/up/pub/Webdocs/DE2_UserManual.pdf consultado el 4 de enero de 2012)

ALTERA, Características de la tarjeta de desarrollo DE2 (disponible en <http://www.altera.com/education/univ/materials/boards/de2/unv-de2-board.html> consultado el 15 de noviembre de 2011)

- [1] Digitales-ITESI, Programación de la tarjeta DE-2 de Altera (disponible en http://digitales-itesi.blogspot.com/2010_08_01_archive.html, consultado el 12 de diciembre de 2011)

- [2] SCRIBD, Altera Develop and Education Board, (disponible en <http://es.scribd.com/doc/39591828/DE2-Reference-Manual>, consultado el 12 de diciembre de 2011)

- [3] WIKIPEDIA, Field Programable Field Array, (disponible en http://es.wikipedia.org/wiki/Field_Programmable_Gate_Array, consultado el 15 de diciembre de 2011)

- [4] MACROARTS, Life Just Better, (disponible en, <http://www.macroarts.com/hot/wholehousecontrols.html>, consultado el 14 de Enero de 2012)

- [5] DOMODESK, Arquitectura ZigBee (disponible en, <http://www.domodesk.com/content.aspx?co=97&t=146&c=43>, consultado el 12 de Enero de 2012)

- [6] DOMODESK, Modelo de Red ZigBee (disponible en, <http://www.domodesk.com/content.aspx?co=97&t=146&c=43>, consultado el 12 de Enero de 2012)

- [7] VICTRONEWS, Implementacion de redes XBee, (disponible en http://victronews.wordpress.com/2012/01/18/implementacion-de-redes-xbee/prd-xbee-modules-fam_lg/, consultado el 15 de Febrero 2012)
- [8] AVANCES EN ROBOTICA, Robots Móviles, (disponible en <http://avancesenrobotica.blogspot.com/2011/07/robots-moviles-disenio.html>, consultados el 15 de Febrero de 2012)
- [9] AVANCES EN ROBOTICA, Robots Móviles, (disponible en <http://avancesenrobotica.blogspot.com/2011/07/robots-moviles-disenio.html>, consultados el 15 de Febrero de 2012)
- [10] WEBELECTRONICA, Motores Y Servos par Robots, (disponible en <http://www.webelectronica.com.ar/news29/nota05.htm>, consultado el 21 de diciembre de 2010)
- [11] WIKIPEDIA, H Bridge, (disponible en http://en.wikipedia.org/wiki/H_bridge, consultado el 13 de Enero de 2012)
- [12] WIKIPEDIA, H Bridge, (disponible en http://en.wikipedia.org/wiki/H_bridge, consultado el 13 de Enero de 2012)

- [13] L298 DATASHEET, (disponible en http://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf, consultado el 21 de diciembre de 2011)
- [14] L298 DATASHEET, (disponible en http://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf, consultado el 21 de diciembre de 2011)
- [15] SINOTECH, Datacheet de B38-12V29RPM
- [16] SINOTECH, Datacheet de B38-12V29RPM