

# HERRAMIENTA PARA LA IMPLANTACIÓN HARDWARE DE CONTROLADORES SOBRE FPGAs.

José Javier Martínez Álvarez  
Grupo de Neurotecnología, Control y Robótica. (NEUROCOR)  
Universidad Politécnica de Cartagena.  
[Jjavier.martinez@upct.es](mailto:Jjavier.martinez@upct.es)

Antonio Guerrero González.  
Grupo de Neurotecnología, Control y Robótica. (NEUROCOR)  
Universidad Politécnica de Cartagena.  
[Antonio.guerrero@ucpt.es](mailto:Antonio.guerrero@ucpt.es)

Juan López Coronado.  
Grupo de Neurotecnología, Control y Robótica. (NEUROCOR)  
Universidad Politécnica de Cartagena.  
[Jl.coronado@upct.es](mailto:Jl.coronado@upct.es)

## Resumen

*En este artículo se presenta una herramienta diseñada para facilitar la implementación hardware sobre FPGAs de controladores para sistemas de tiempo real. En este trabajo se muestran las características fundamentales de esta herramienta, sus detalles constructivos y su capacidad operativa. Finalmente se presenta un ejemplo en el que se muestra el proceso de implementación de un controlador para gobernar el posicionamiento de un motor de corriente continua.*

**Palabras Clave:** Implementación hardware, FPGAs, sistemas de tiempo real.

## 1 INTRODUCCIÓN.

Las FPGAs representan una tecnología muy interesante para aplicaciones de control en tiempo real cuando se requiera implantar un determinado algoritmo de control sobre una determinada máquina o dispositivo. Además suponen una solución muy adecuada para utilizar tecnologías ASIC en pequeños prototipos con costos reducidos de fabricación. Desde el punto de vista de las aplicaciones en robótica, son tecnologías muy interesantes ya que permiten implantar directamente algoritmos de control en hardware de una forma rápida y económica.

Este tipo de tecnología se está comenzando a utilizar en el grupo NEUROCOR de la UPCT para la realización de dispositivos mecatrónicos a los que se les han implantado algoritmos específicos de control con un hardware mínimo. Estos controladores

integrados no solamente llevan a cabo el algoritmo de control sino que a su vez realizan otras tareas secundarias como las de comunicación con gran ancho de banda y tratamiento de sensores y actuadores. Estos controladores hardware implantados sobre FPGAs, son relativamente complicados de implementar y esto nos indujo a crear una herramienta que nos permitiera simplificar esta tarea.

En este artículo se presentan en un primer lugar los detalles constructivos de esta herramienta. A continuación se da una muestra de sus posibilidades mediante un ejemplo en el que se implementa en hardware un algoritmo de control.

## 2 ESPECIFICACIONES PARA EL DISEÑO.

Inicialmente, se establecieron una serie de especificaciones con el objeto de obtener un sistema con las siguientes características:

### - Sistema de Propósito General y multifuncional.

El sistema debe adaptarse a cualquier ambiente de trabajo y debe ser útil tanto en aplicaciones de tipo industrial así como en aplicaciones de tipo educativo como pueden ser el de introducción al mundo de las FPGAs, configuración, evaluación e incluso manejo de buses.

### - Sistema abierto.

Esto significa que por un lado, el sistema se ha ideado para que sea cómoda y completamente reconfigurable por el usuario desde el exterior y a

través del software del sistema. Esto permite que haciendo uso de las herramientas del sistema, se puede controlar y modificar la arquitectura interna del hardware a nuestro gusto o necesidad. Una vez determinada la nueva configuración hardware, es posible crear una nueva aplicación software a medida, que interactuando con la nueva arquitectura hardware creada, consiga dar toda la potencia a la aplicación desarrollada y conseguir de esta manera llegar a una filosofía de independencia y simbiosis entre hardware y software.

**- Plataforma Hardware reconfigurable por software.**

Que conectada al BUS ISA XT, sea capaz interactuar con el exterior o con ella misma (hardware interno del cliente) con el fin de desarrollar e implementar aplicaciones de diferente índole y carácter basadas en dispositivos FPGAs. Para poder cumplir con las especificaciones anteriores de ser sistema reconfigurable y de propósito general (abierto) , la plataforma debe incluir (gracias a las características de estos dispositivos reconfigurables), dos FPGAs; La primera, la FPGA Gestora, se encargará de asumir las tareas de gestión y comunicación del bus mediante el hardware apropiado (cuya filosofía en lo que coniere a la parte de comunicación consiste en la compartición de recursos de almacenamiento (ejm. memorias de doble puerto); y la segunda, la FPGA Cliente, que haciendo uso de esos recursos de comunicación para poder comunicarse con el PC, se destina a inferir el hardware del usuario cliente que también será reconfigurable y que determinará consecuentemente la aplicación cliente.

**- Plataforma Software.**

Capacitada para poder comunicarse con la plataforma hardware y establecer la configuración del sistema, esta, además de que nos permita el control del hardware debería tener la facultad de ser de libre acceso por cualquier usuario (para cumplir con las exigencia de sistema abierto y propósito general) y poder recibir o transmitir la información necesaria entre el exterior y el PC en tiempo real.

**3 DIAGRAMA DE BLOQUES DEL SISTEMA.**

En el diagrama de bloques de la figura 1 se muestra la solución que se ha adoptado para el desarrollo del sistema. En este, se pueden observar tres partes fundamentales, que son : una plataforma Hardware, una Interface de comunicación y una plataforma Software. A continuación se muestran las

características principales de cada una de estas partes.

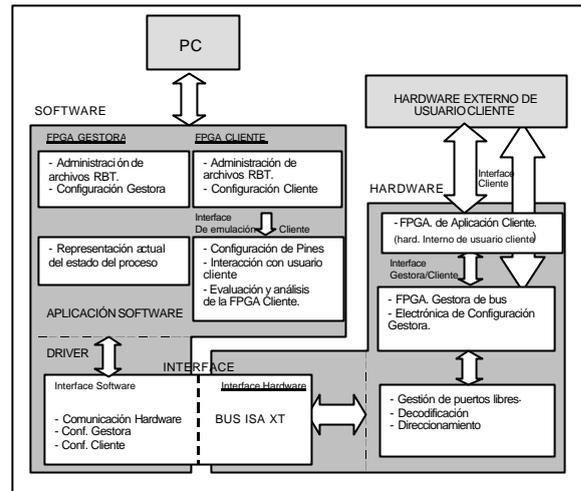


Figura 1: Diagrama de bloques del sistema.

La plataforma Hardware es una tarjeta, capaz de interactuar con el PC y con el Hardware del usuario cliente a través de las respectivas interfaces de usuario y bus del PC. Esta plataforma en cuestión, consta de las siguientes partes:

**- Interface con PC (Bus\_isa XT).**

Esta interface proporciona el enlace necesario para poder configurar y controlar el sistema desde el ordenador. Las características de esta interface, determinan la potencia y flexibilidad en los recursos de comunicación entre el sistema (hardware Cliente) y el ordenador. La interface bus isa XT supone la solución de interconexión entre el sistema y el PC.

**- Gestión, Decodificación y Direccionamiento de puertos.**

Esta parte conforma el núcleo de unión entre el hardware del sistema y la interface del PC. Fundamentalmente, su misión consiste en gestionar la compatibilidad de nuestro Hardware con el bus del PC. Para llevarlo a cabo, básicamente lo que hace, es elegir un espacio de trabajo libre dentro de la zona memoria del PC reservada a puertos de entrada/salida y conseguir su direccionamiento y administración por medio de una adecuada etapa de decodificación.

**- FPGA Cliente.**

La cual podrá ser configurada por un programa cliente. Este programa cliente determinará el uso y la aplicación del sistema. Por otro lado, el tipo y complejidad del programa, corre a cuenta y criterio del diseñador en cuestión y depende del tipo de aplicación para la cual se destine el sistema. El tipo

de aplicación puede abarcar un amplio rango de posibilidades que llevan desde el desarrollo de los prototipos más avanzados dentro de la electrónica de propósito industrial, robótica etc. hasta otras muchas aplicaciones en las que se puede destacar el campo de la educación.

#### **- FPGA Gestora del bus.**

Deberá de encargarse de gestionar el espacio reservado al sistema dentro del banco de puertos del PC (tarea realizada por la etapa de gestión y decodificación) y asignarlo como recurso de comunicación entre el PC y la FPGA Cliente o la interface de usuario cliente. A la FPGA Gestora, también se le delega la tarea de incorporar internamente el hardware necesario para realizar el proceso de configuración de la FPGA Cliente. La especificación que inicialmente se propone al sistema de tener un carácter abierto, se consigue en parte, gracias a este bloque debido a la capacidad que tiene de ser reconfigurado en cualquier momento y con un adecuado programa de gestión del bus (que puede ser más o menos complejo). Este, es un punto clave e importante a destacar ya que con la funcionalidad y complejidad con la que se dote al programa que gestiona el bus, determinará precisamente, la eficacia y potencia del sistema para comunicarse con el hardware cliente (considerando como hardware cliente a la FPGA cliente o al hardware exterior conectado con la interface de usuario cliente). Consecuentemente, la capacidad de comunicación también afectará evidentemente a la posibilidad del sistema para adaptarse a los distintos campos de aplicación para el cual se quiera dar uso a la plataforma de desarrollo. En este trabajo se propone seguir una filosofía de interconexión implementada mediante recursos de almacenamiento compartido como pueden ser las memorias de doble puerto que nos ofrecen la máxima potencia de comunicación entre el sistema y el hardware cliente.

En cuanto a la plataforma software cabe destacar el papel del driver que se ha construido en forma de librería de enlace dinámico (DLL). Este driver contiene funciones de bajo nivel que permiten manejar directamente los accesos a hardware. Este driver realiza básicamente dos funciones:

- Configuración de las FPGAs Gestora y Cliente.
- Comunicación con el hardware.

## **4 SOFTWARE DE APLICACIÓN.**

La aplicación software de propósito general propuesta para el desarrollo del sistema no tiene sentido sino se utiliza conjuntamente con el driver

propuesto en apartados anteriores, el objetivo fundamental de esta aplicación no obstante, es servir como herramienta de ayuda para manejar y administrar de forma cómoda los archivos utilizados para la configuración de las dos FPGAs que forman el sistema.

Junto con la tarea de administración de archivos de configuración, la aplicación propuesta, también permite configurar independientemente cada una de las FPGAs haciendo uso para ello del driver comentado anteriormente.

Por último, en la aplicación también se ha propuesto una herramienta de evaluación y emulación que permite al usuario interactuar de manera directa y gráficamente, sobre el estados de los pines de las FPGAs permitiendo su lectura y escritura.

Las principales características del software por tanto son:

- Administración de archivos de configuración de la FPGA Gestora.
- Administración de archivos de configuración de la FPGA Cliente.
- Configuración de la FPGA Gestora.
- Configuración de la FPGA Cliente.
- Configuración de los pines de la FPGA Gestora/FPGA Cliente.
- Interface interactiva de evaluación y monitorización de la FPGA Cliente.
- Información del estado del proceso en tiempo real y mediante histórico.
- Hace uso del driver del sistema.

La interface gráfica del software de aplicación muestra su aspecto completo en la figura 2. En ella, se puede ver que la aplicación esta formada principalmente por tres ventanas.

- Ventana de Administración y Configuración.
- Ventana de evaluación y emulación de la FPGA Cliente.
- Ventana de proceso.

Cada una de estas tres ventanas, representan y se usan para desempeñar las tres funciones principales del software.

- Administración y configuración de archivos y pines.
- Evaluación y emulación de la FPGA Cliente.
- Información del estado del proceso.

En la figura 2 se muestra una imagen del programa de aplicación.

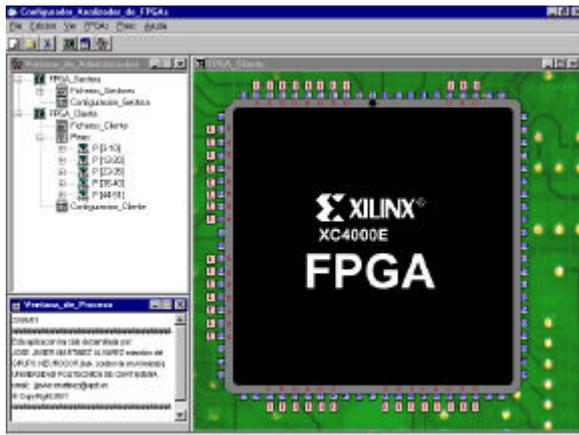


Figura 2: Programa de aplicación.

## 5 IMPLEMENTACIÓN DE UN HARDWARE DE UN CONTROLADOR MEDIANTE ESTA HERRAMIENTA.

A continuación se muestran los detalles de la implementación de un sistema de control discreto para el posicionamiento en tiempo real de un servomotor de CC. Si es necesario implementar varios controladores hardware, esto se hace de forma sumamente sencilla (cortar y pegar), siempre y cuando los recursos necesarios no desborden la capacidad de la FPGA. La principal repercusión en cuanto al control, que se obtiene del hecho de que el sistema se haya implementado dentro de una FPGA, radica esencialmente, en la ventaja que supone poder ejecutar en un solo chip y de forma concurrente, uno o más algoritmos de control con periodos de muestreo suficientemente pequeños e independientemente de su cantidad y complejidad (en este ejemplo en concreto, solo se va a mostrar un algoritmo y con un periodo de muestreo relativamente alto 125ns).

Otro aspecto interesante a destacar en este desarrollo, es el hecho de que la FPGA, además de implementar el algoritmo de control del sistema, también incluye, las interfaces necesarias para poder recoger la información de consigna y realimentación del servomotor, así, como la interface necesaria, para poder generar al exterior; tanto la señal requerida por la etapa de potencia para hacer girar al motor (PWM, Sentido de giro) como la información necesaria para comunicar la posición real del motor, al equipo monitor. Todo esto permite obtener en un solo chip un sistema completo y autónomo que permite al usuario a través de un bus único (genérico), hacer control de uno o varios sistemas servomotor.

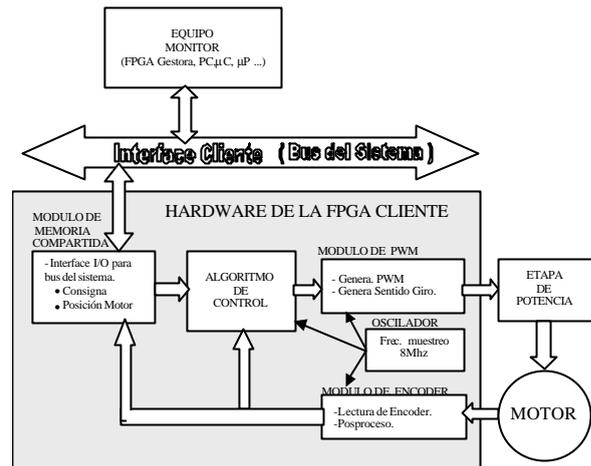


Figura 3: Diagrama de bloques del controlador implementado sobre FPGA Cliente.

En la figura 3 se muestra un diagrama de bloques del controlador que se ha implementado sobre FPGA. El algoritmo constituye tan solo uno de los módulos que debe integrar el controlador hardware, ya que el sistema implementado además, debe realizar otras tareas como la de manejar la electrónica de potencia, o comunicarse con otros niveles de más alto nivel. A continuación se muestran las características del controlador que se ha implementado en hardware:

- Implanta y ejecuta concurrentemente en un solo chip, uno o varios algoritmos de control (la cantidad y complejidad de los algoritmos será función de la capacidad de la FPGA y de las necesidades del usuario).
- Establece un medio de comunicación entre la FPGA y el equipo monitor a través de recursos de almacenamiento compartido, permitiendo la comunicación a alta velocidad y de forma bidireccional a través del bus del sistema.
- Permite generar las señales apropiadas demandadas por la etapa de potencia (PWM y Sentido de giro), para hacer actuar al motor.
- Permite reconocer y procesar las señales procedentes del encoder, con el objetivo de conseguir la posición real del servomotor.
- Discretiza al sistema de control, con periodos de muestreos suficientemente pequeños e independientemente de la complejidad del sistema o número de algoritmos (en FPGAs de alta velocidad se pueden conseguir periodos de muestreos de hasta 5ns)

En este sentido, el hardware definitivo inferido dentro de la FPGA, se puede dividir cuatro bloques principales.

- Modulo de recursos de memoria compartido (Interface I/O).  
Permiten establecer la comunicación bidireccional entre el equipo monitor y el resto del hardware de la FPGA (por medio del bus del sistema). Su objetivo, es enviar al equipo monitor, la posición real del motor en tiempo real, y establecer en el otro sentido de la comunicación, la señal de consigna necesaria para el control (en este caso enviada por el equipo monitor).
- Modulo de lectura y procesado de encoder.  
Este modulo hardware, tiene como tarea, recoger las señales que proporciona el encoder del servomotor, para posteriormente procesarlas y obtener de ellas en tiempo real, la posición absoluta del motor necesaria para el calculo del algoritmo de control. Este modulo además de suministrar esta información al algoritmo de control, también comparte recursos de memoria con el equipo monitor, con el objetivo de informarle a el también de la posición real del motor.
- Modulo de generación de PWM.  
Este otro modulo tiene como función, traducir el resultado obtenido en el calculo del algoritmo de control, en dos señales físicas (PWM, Sentido de giro) que una vez amplificadas, son capaces de hacer actuar al motor de acuerdo a lo previsto por el algoritmo.
- Oscilador.  
El oscilador es el elemento que sincroniza y determina el periodo de muestreo del sistema, incluida la discretización en el calculo del algoritmo de control.
- Algoritmo de control.  
El algoritmo de control del servomotor, que en cuestión se ha implementado. Consiste en un control de posición de carácter no lineal cuyo diagrama de flujo se representa en la figura 4.

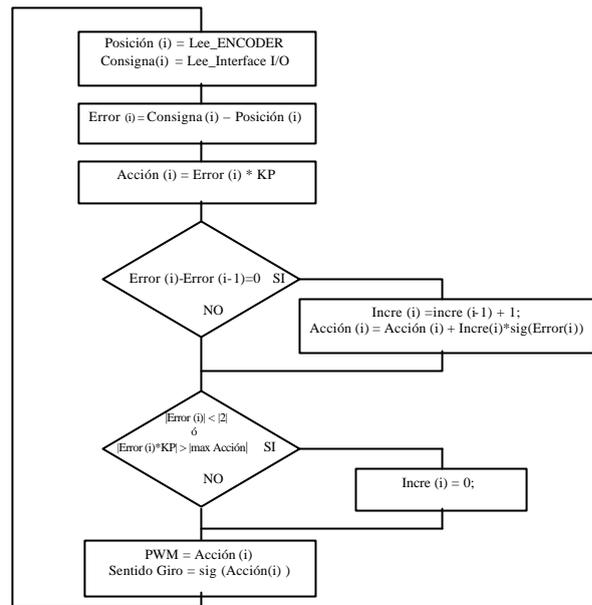


Figura 4: Algoritmo implementado.

Hay que hacer notar que aunque el algoritmo se haya representado en la figura, siguiendo un esquema ciertamente secuencial evidentemente esto no es a sí, ya que la ejecución del algoritmo de control se hace de forma concurrente debido a la naturaleza intrínseca de la FPGA.

## 6 CONCLUSIONES.

En este artículo se ha mostrado una herramienta que se ha desarrollado en el grupo de Neurotecnología, Control y Robótica de la UPCT destinada a facilitar la implementación de controladores en hardware. En este artículo se han mostrado los detalles de su estructura hardware y software. Como se ha visto la herramienta consta de dos FPGAs, una gestora y otra cliente y esto a su vez le confiera un carácter más abierto. Finalmente se han mostrado las prestaciones de la herramienta en un ejemplo en el que se ha implementado un controlador para un motor de corriente continua. Esta herramienta se ha ideado con la intención de implementar controladores de inspiración biológica del tipo VITE, AVITE y FLETE integrados en plataformas mecatrónicas. Finalmente podemos decir que esta herramienta ha mostrado prestaciones suficientes para abordar una tarea de esta magnitud.

## **Agradecimientos**

Deseamos agradecerle al departamento de electrónica, tecnología de computadoras y proyectos su apoyo técnico en la realización de este proyecto. Además deseamos agradecer al CEDETEL su apoyo económico.

## **Referencias**

- [1] Manual de referencia “XC4000E and XC4000x Series Field programmable Gate Arrays” de Xilinx 1999.
- [2] Aracil Santoja, R., Jiménez Avello, A., (1991). “Sistema discretos de control”, *Sección de publicaciones ETSIIM*.
- [3] Ruiz Merino, R.J., Garrigos Guerrero, J., Doménech Asensi, G., Zapata Perez, J.R., López Alcantud, J.A., Hinojosa Jiménez, J., (2000). “Diseño de sistemas electrónicos digitales”, *Ediciones Diego Marín Colecciones texto.(Universidad de Murcia)*.