



Ingeniare. Revista Chilena de Ingeniería

ISSN: 0718-3291

facing@uta.cl

Universidad de Tarapacá

Chile

Álvarez G, Carlos; Soto P., Andrés; Watkins O., Francisco

SIMULACIÓN DE CONTROLADORES DIGITALES

Ingeniare. Revista Chilena de Ingeniería, vol. 17, núm. 3, diciembre, 2009, pp. 309-316

Universidad de Tarapacá

Arica, Chile

Disponible en: <http://www.redalyc.org/articulo.oa?id=77212231004>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

SIMULACIÓN DE CONTROLADORES DIGITALES

SIMULATION OF DIGITAL CONTROLLERS

Carlos Álvarez G.¹ Andrés Soto P.¹ Francisco Watkins O.¹

Recibido 9 de marzo 2009, aceptado 9 de octubre de 2009

Received: March 9, 2009 Accepted: October 9, 2009

RESUMEN

El presente trabajo tiene como objetivo la implementación de controladores digitales en un entorno de simulación controlado, para esto se desarrolla una plataforma de hardware que permite ejecutar los programas en lenguaje C generados en una estación de trabajo. Estos programas corresponden al controlador y a la planta que son generados por un software que genera dichos programas a partir de sus parámetros de modelación aplicando teoría de control digital sobre procesos reales.

Palabras clave: Controladores digitales, simulación, microcontroladores, generadores de programas, control digital, control de procesos.

ABSTRACT

This paper describes an implementation of digital controllers in a simulation environment for including a hardware platform for running programs generated on a workstation. These programs for both the controller and the plant are generated by software based on parameters using digital control theory for real processes.

Keywords: Simulation, microcontroller, generators programs, digital control, process control.

INTRODUCCIÓN

En la actualidad el uso de herramientas de software de simulación para modelar controladores análogos y digitales es ampliamente utilizado en la mayoría de los laboratorios de educación superior y en organizaciones dedicadas al control automático. En general, dado un fenómeno físico que se desea controlar disponemos de un modelo matemático que representa dicho fenómeno, o bien se tiene una curva de respuesta del proceso, que mediante técnicas de identificación de procesos podemos obtener los parámetros de un modelo aproximado del fenómeno real, y de esta forma diseñar y obtener el controlador del proceso.

Existen variadas herramientas de software que permiten simular controladores, entornos robustos para la programación de rutinas en diferentes lenguajes y software orientado a la construcción de los circuitos electrónicos. Ante este gran abanico de soluciones, se detecta una carencia de herramientas que permitan generar programas que funcionen en un ambiente físico

utilizando microcontroladores a partir de los parámetros de controladores.

En este trabajo se considera el control de la planta en lazo cerrado o control “feedback” para un sistema SISO (Single Input, Single Output). Para esto la teoría de control nos entrega un marco teórico fundamental para desarrollar un controlador adecuado que satisfaga los requerimientos de diseño deseados.

La simulación nos permite realizar la sintonía fina en un esquema de prueba y error en forma simple y rápida. En la actualidad existe una brecha importante entre simular en un ambiente de software y simular en un ambiente físico. Nuestro interés es analizar, diseñar e implementar una solución que nos permita simular en forma física diferentes controladores digitales tales como: PID Discreto, Deadbeat, Dahlin y Kalman.

Este trabajo se enmarca en el desarrollo de plataformas de hardware y software con el objetivo de construir prototipos rápidos para el diseño e implementación de sistemas

¹ Departamento de Ingeniería Eléctrica. Universidad de Santiago de Chile. Av. Alameda 3363, Estación Central. Santiago, Chile.
E-mail: carlos.alvarez@technologies.cl; andres.sotop@usach.cl; francisco.watkins@usach.cl

embebidos [1-3], en particular nuestro interés se centra en el desarrollo de una plataforma cerrada de hardware y software, para la especificación de controladores digitales utilizando una interfaz de alto nivel que permita generar, compilar y traspasar programas al hardware específico de acuerdo a los parámetros.

En este esquema se dispondrá de un sistema en un PC que será capaz de realizar la generación y transferencia de dos programas para simular el controlador y la planta a partir de los parámetros de sintonía del proceso. Esto se lleva a cabo en dos microcontroladores, que constituyen el hardware de prueba. Para este fin los programas fuentes son generados en lenguaje C y compilados con una versión específica (CCS PCM Compiler, Versión 3.245) de la empresa Custom Computer service Inc., para microcontroladores PIC de Microchip. Para esto se seleccionó la serie 16f87x, debido a la alta disponibilidad en nuestro mercado y su versatilidad.

De este desarrollo se espera obtener respuestas de procesos con el menor offset posible ante un modelo ya implementado en entornos como: MatLab, Simulink y LabView, y de esta forma validar y probar la eficiencia del hardware, en control automático, como plataforma de desarrollo, bancos de prueba y simulación de procesos industriales.

METODOLOGÍA

La metodología de desarrollo para la obtención de controladores digitales, contempla las etapas que se muestran en la Figura 1.

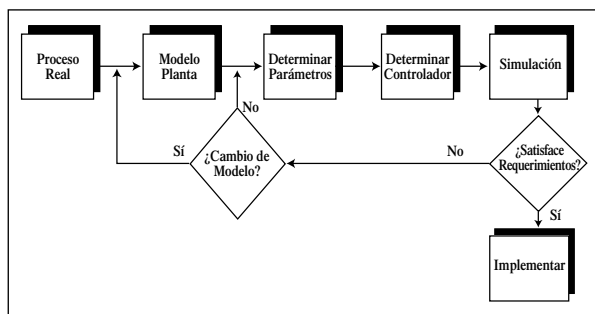


Figura 1. Etapas para la obtención de controladores.

A continuación, se describen estas etapas:

Proceso Real

Corresponde al proceso físico que se encuentra funcionando y que se desea simular su comportamiento de acuerdo al

controlador diseñado. Este proceso puede ser encontrado en la literatura [4, 9], o bien disponer de una curva de respuesta del proceso que puede ser identificado y estar acotado en el tiempo.

Modelo Planta

Se utilizan modelos de primer y segundo orden con y sin retardo [5], lo que se busca es definir el mejor modelo para el proceso real, cuyo objetivo principal es obtener una representación equivalente del proceso real lo más cercano posible.

Determinación de Parámetros

Una vez seleccionado el modelo de la planta, se buscan los parámetros: tiempo de retardo y tiempo de respuesta, estos parámetros pueden ser obtenidos mediante los siguientes criterios:

Plantas de primer orden sin retardo, se pueden obtener utilizando el método de sintetización de controladores de Dahlin [6].

Para las plantas de primer orden con retardo se pueden utilizar los métodos clásicos de Ziegler-Nichols, Miller, Analítico o utilizando métodos de minimización del error (IAE, ITAE, ISE) [7-8].

Para plantas de segundo orden con retardo se pueden utilizar los métodos de Smith-Cox y analítico [12]. Otros métodos menos conocidos pueden ser utilizados en esta etapa y que también presentan resultados exitosos [10].

Determinación del Controlador

Con los parámetros obtenidos podemos determinar un controlador P, PI o PID continuo utilizando el algoritmo de Ziegler-Nichols, Cohen-Coon o minimización del error (IAE, ITAE, ISE) [7], de esta forma se pueden obtener los valores: K_p (Ganancia Proporcional), T_i (Tiempo Integral) y T_d (Tiempo Derivativo) para utilizarlos en la obtención del controlador PID digital [11].

Para los controladores digitales Deadbeat, Kalman y Dahlin se pueden aplicar los parámetros en forma directa de acuerdo a cada algoritmo [12].

Simulación

Con el modelo de planta obtenido y el controlador diseñado podemos utilizar herramientas de simulación tales como: Matlab, Simulink y LabView, entre otras. Estas

herramientas nos facilitan la creación, sintonización y prueba de controladores obteniendo rápidamente el “mejor” controlador para el proceso tanto en tiempo continuo como en tiempo discreto, es importante modelar el mismo proceso utilizando modelos de primer y segundo orden y con diferentes parámetros, realizando las simulaciones de cada uno de ellos y de esta obtener el controlador más adecuado.

Implementación

En esta etapa se busca implementar el controlador en un entorno de hardware determinado. Este trabajo está basado en el uso de microcontroladores utilizando el lenguaje de programación C. El uso de este lenguaje ha sido un avance importante a la hora de implementar controladores, dado que el compilador C resuelve por nosotros ciertas complejidades que debemos manejar en forma muy precisa. La programación en C para el manejo de interrupciones, comunicación serial, convertidores A/D, D/A y otros, es más simple y directo en este lenguaje que utilizando assembler.

En todo este ciclo se observa que la etapa de implementación es la compleja, puesto que al momento de implementar los controladores obtenidos en la simulación habitualmente estos presentan problemas tales como: Saturación, Redondeos Numéricos y Presencia de ruido.

En la presente metodología es importante la iteración para probar diferentes modelos, parámetros y controladores, antes de llegar a la etapa de implementación.

EL PROBLEMA

Hasta la etapa de simulación la metodología es bastante clara, compacta y existen muchos métodos y herramientas conocidas, las cuales están ampliamente probadas y entregan un resultado satisfactorio la mayoría de las veces, pero al momento de querer implementar en un ambiente basado en hardware surgen varios problemas e interrogantes, dentro de los cuales encuentran: microcontrolador a utilizar, compilador, herramientas de desarrollo, entornos de simulación, ambiente de pruebas las que ponen una dificultad adicional.

La respuesta a estas interrogantes no es una respuesta simple y genera una complejidad importante a la hora de implementar controladores digitales. No obstante lo anterior, es posible encontrar implementaciones exitosas

de controladores digitales programados en assembler o lenguaje C para diferentes microcontroladores [13-14] y de diferentes fabricantes, los que son probados y embebidos en una plataforma de hardware que contiene toda la electrónica para su correcto funcionamiento y que muchas veces se utilizan como modelos.

En los ambientes de implementación se distinguen para el desarrollo y pruebas los siguientes esquemas:

Con entrenador

En este caso el microprocesador se monta en una tarjeta que posee una serie de prestaciones, conectadas a diferentes puertos, donde es posible enviar el programa y poder ejecutar el programa directamente y observar su desempeño.

Sin entrenador

En este caso el programa se carga directamente al microcontrolador y este es parte de un circuito en forma directa, usando habitualmente comunicación serial y hardware asociado.

Híbridos

En este caso el microcontrolador viene montado dentro de una plataforma de hardware que facilita la conexión de puertos y que puede ser acoplada a un circuito.

Lo que se requiere es disponer de herramientas que nos faciliten la simulación y prueba de nuestros controladores en un ambiente de hardware y cuya característica fundamental es que deben estar por sobre los fabricantes de microcontroladores, porque se observa que no existen prestaciones, estándares y herramientas únicas que estén estrechamente ligadas a las etapas anteriores del ciclo general para la obtención de controladores.

SOLUCIÓN PROPUESTA

La solución incorpora todas las etapas del ciclo general para la obtención de controladores digitales, esto significa, que esta debe acortar al máximo la brecha existente entre la obtención del controlador mediante la simulación por software y su implementación sobre el hardware. La arquitectura general de la solución se puede ver en la Figura 2.

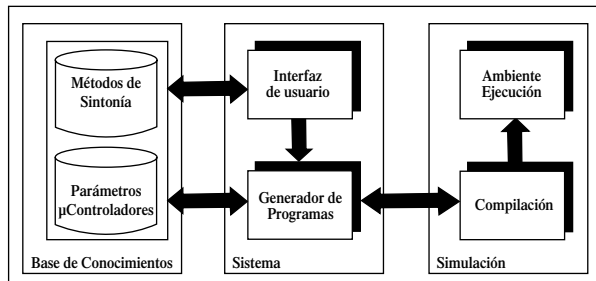


Figura 2. Arquitectura general de la solución.

La arquitectura contempla los siguientes elementos:

Interfaz de usuario

A través de esta interfaz se podrán definir distintas simulaciones y sus parámetros, agregar nuevos métodos de sintonía a los ya existentes y también nuevos controladores y su configuración, esto permitirá el ingreso de los parámetros de la planta para la construcción del programa del controlador.

Base de Conocimiento

Esto corresponde al repositorio de los métodos de sintonía y parámetros de controladores, aquí debemos definir y modelar la mejor forma de almacenar este conocimiento, este conocimiento será tomado por el generador de programas.

Generador de Programas

Este módulo será el encargado de generar un programa fuente, libre de errores, en algún lenguaje de programación que evidentemente funcione en los microcontroladores disponibles.

Entorno de Simulación

Este módulo podrá ejecutar el programa del controlador obtenido de la simulación, este programa puede ser fuente para compilar, o idealmente un binario que se pueda ejecutar en un entorno virtual, y lo relevante aquí es que deberá abarcar todos los conceptos involucrados en lazo de control de la Figura 3.

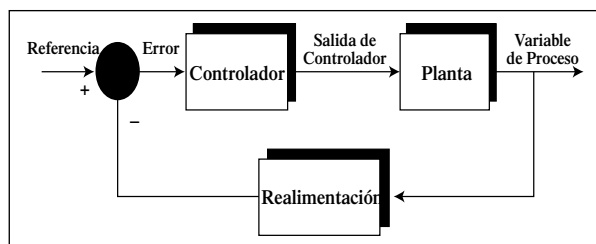


Figura 3. Lazo Cerrado de Control.

Este entorno puede ser 100% virtual o es posible diseñar una implementación 100% física o eventualmente híbrida, lo importante aquí es que resulte simple de utilizar y de configurar, el paso de la simulación a la implementación del programa generado, su prueba debe ser simple y rápida.

SIMULACIÓN

Nuestro interés se centra en implementar una solución que nos permita simular en forma física nuestro controlador, el esquema se muestra en la Figura 4.

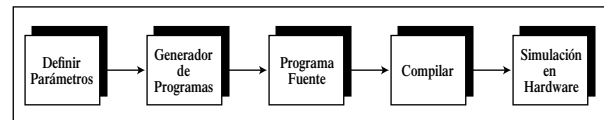


Figura 4. Esquema de Generación.

En este esquema se dispondrá de un sistema en un PC que será capaz de generar, a partir de los parámetros, los programas para el controlador y la planta.

Para simular un programa podemos tener una simulación virtual, por hardware o física e híbrida, en este trabajo se implementa una plataforma de hardware para hacer la simulación.

Simulación Virtual

En este caso el programa generado debe pasar a un entorno gráfico con herramientas del tipo drag and drop y debe permitir la construcción en forma gráfica el lazo de control mostrado en la Figura 3, disponiendo de todos los elementos gráficos que nos permitan ejecutar el programa en este entorno.

Para ello se debe tener acceso a la base de conocimientos de los controladores disponibles y a una base de datos de hardware virtual. Para establecer la referencia, se deben contemplar teclados, pulsadores y para la planta un procesador virtual que se comporte como planta y que permita generar el comportamiento mediante la generación de un programa.

Adicionalmente se deben tener LCD para mostrar la evolución de la variable de proceso, mostrar la referencia y eventualmente la salida del controlador, además de elementos conversores A/D y D/A, y comunicaciones seriales.

Para esta solución es posible desarrollar un tool box para matlab, pero sigue siendo una solución por software.

Simulación por Hardware

En este caso se debe construir una plataforma por hardware con todos los elementos para cargar y probar los programas generados, de esta forma simular el controlador y la planta por hardware, en la Figura 5 se muestra el esquema de esta alternativa:

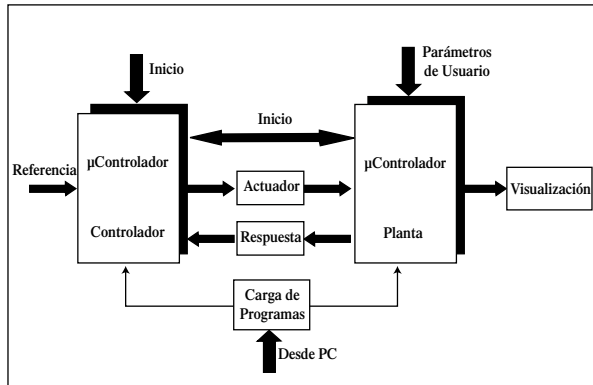


Figura 5. Plataforma de Hardware.

En este esquema se propone que existan dos microcontroladores uno para el controlador y el otro para la planta, donde el controlador hará de máster y la planta de esclavo, además debe tener un mecanismo simple de transferencia de programas compilados desde el PC a los microcontroladores y finalmente disponer todos los elementos de la simulación virtual para la conectividad del lazo cerrado de control y con esto lograr que funcionen los programas generados.

Esta solución implica tener que decidir la marca y fabricante de los microcontroladores y de todos los elementos presentes en la tarjeta.

Simulación Híbrida

En este caso la planta reside en un PC y la salida de control y variable de proceso son transferidas por la interfaz RS-323 del PC, también es posible que la planta sea simulada por un microcontrolador y el controlador funcione en el PC.

En resumen, todas estas alternativas resultan interesantes de implementar, sin embargo, nos atrae la idea de implementar la plataforma de hardware para la simulación física, fundamentalmente porque trabajar con el hardware

real permite ver los problemas que aquí se generan y que muchas veces no son menores y de esta forma desarrollar controladores simulando por software y hardware lo que es más completo.

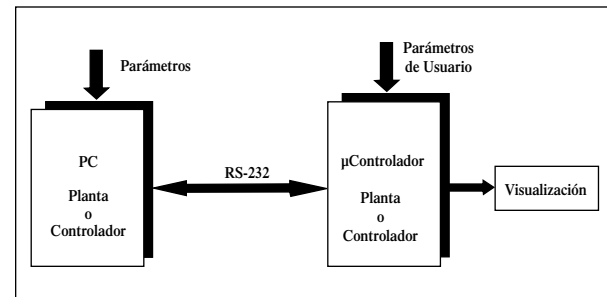


Figura 6. Solución Híbrida.

IMPLEMENTACIÓN

Para el desarrollo de la tarjeta se consideró la familia de microcontroladores PIC de la microchip y el lenguaje de programación en C, dentro de las características más importantes de diseño son:

- Permitir el cambio de microcontroladores.
- Carga de programas en forma directa.
- Ingreso de la referencia.
- Despliegue de la respuesta de la planta.

Los elementos de software y hardware se detallan a continuación.

Software

Tal como se muestra en la Figura 2, el software desarrollado tiene una interfaz con el usuario que le permite, definir y guardar distintas simulaciones, controladores, fórmulas matemáticas para el controlador y la planta, parámetros de configuración de los generadores de programas.

Los parámetros de configuración considerados para que el usuario defina a nivel de cada simulación se pueden ver en la Tabla 1.

Y en segundo término están los generadores de programas que son: el programa generador de programas en lenguaje C para el controlador y el segundo un generador de programa en lenguaje C para la simulación de plantas.

Tabla 1. Parámetros del Usuario.

Parámetro	Descripción
Planta	1er Orden sin Retardo 2do Orden con Retardo 1er Orden sin Retardo
Tiempo Respuesta	τ
Tiempo Retardo	Θ
Ganancia	K
Controlador Digital	PID, Deadbeat, Kalman Dahlin
Algoritmo para Controlador PID	Ziegler-Nichols, Cohen-Coon, IAE,ITAE, ISE.
μ Controladores Microchip	Familia 16f87x

El generador de programas para el controlador básicamente toma una secuencia de plantillas, previamente codificadas en lenguaje fuente, y reemplaza parte de este código de acuerdo a los parámetros definidos; estas plantillas resuelven diferentes aspectos tales como: declarativas en los programas generados, teclados para el ingreso de datos y resets, despliegue de datos en el lcd, comunicación serial entre los microcontroladores, conversores A/D y D/A para la recepción del feedback, manejo de puertos digitales e interrupciones, entre otros.

El generador de programas para la planta toma una secuencia de fórmulas matemáticas (ecuación de diferencias finitas), que residen en la base de datos y a partir de éstas genera el programa.

En la Figura 7 se puede ver la interfaz de usuario de la plataforma de software de simulación, en esta el usuario puede mantener una lista de simulaciones, donde a cada una de ellas le puede definir los parámetros según la Tabla 1, utilizando los distintos botones disponibles, una vez que estos son grabados, se puede invocar al generador de programas que a través de sus plantillas genera el código para el controlador y la planta, lo que posteriormente son compilados y transferidos a la plataforma de hardware para realizar la simulación por hardware.



Figura 7. Interfaz Software de Generación.

En la Tabla 2 se indican los diferentes aspectos técnicos utilizados para el desarrollo y pruebas del software generador de programas.

Tabla 2. Parámetros del Usuario.

Aspecto	Descripción
Metodología de Desarrollo	Incremental
Lenguaje de Programación	Java 2.0
Interfaz	Windows
Servidor de Aplicaciones	TOMCAT
Base de Datos	SQL/Server 2005 Express
Lenguaje Generado	C para microcontroladores de la Microchip
Compilador	C de la CSS (versión 3.245)
Entorno de Prueba de Programas	MPLab IDE 8.10
Entorno de Simulación de Programas y Circuitos	Proteus 7.1

Hardware

En esta etapa se implementó la plataforma que sustenta la simulación del controlador y de las plantas de prueba ante una entrada. Para ello se desarrolló un diseño en el cual se contemplan dos microcontroladores PIC.

En el Microcontrolador 1 se implementó el algoritmo de control digital: PID, Kalman, Deadbeat o Dahlin; y la etapa de comparación asociada a la realimentación de un sistema de control de lazo cerrado.

En el Microcontrolador 2 se desarrolló un programa que emula el funcionamiento de las plantas de prueba y su reacción ante una entrada previamente seteada, cuya salida es en el formato digital a través de 8 líneas.

Ambos microcontroladores están comunicados a través de su generador de baudios USART, por esto se utiliza la interrupción que esta contiene.

En el lazo de realimentación consta con hardware de apoyo externo, específicamente un conversor Digital / Análogo antes de ingresar nuevamente al microcontrolador 1, para su ciclo de comparación y algoritmo de control para la corrección de la respuesta de salida del proceso.

Para mayor facilidad en la operación de la plataforma, esta consta con una interfaz HIM del tipo LCD y teclado matricial, con los cuales se pueden setear los parámetros iniciales de la planta, con la amplitud del escalón deseado, reseteo de la plataforma y la evolución en forma numérica de la salida del proceso.

En forma adicional se interviene esta plataforma con una estación de ingeniería en la cual se pueden observar las gráficas de entrada y salida del sistema y el código generado por el generador de código desarrollado para ser implementado en el microcontrolador PIC 1, en el cual se implementa el algoritmo de control, como se muestra en el diagrama funcional de la Figura 8.

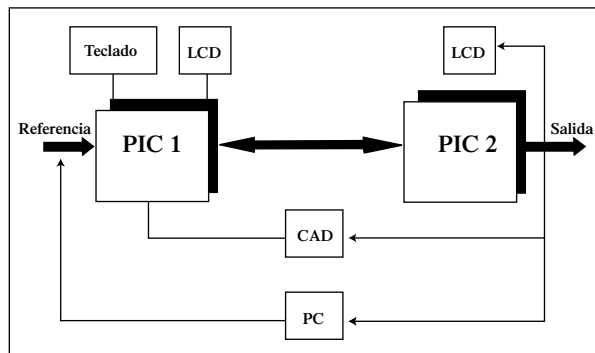


Figura 8. Diagrama funcional de Plataforma más estación de ingeniería.

En la Figura 9 se puede ver una vista 3D del esquemático proyectado.

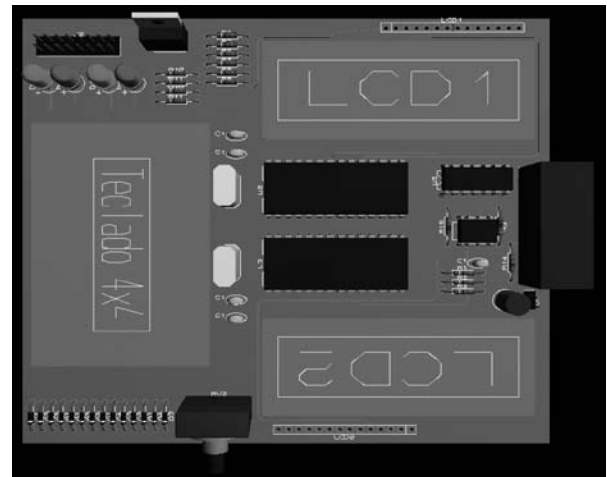


Figura 9. Diagrama Esquemático Funcional.

En la Figura 10 se muestra el prototipo de prueba utilizado en las simulaciones.

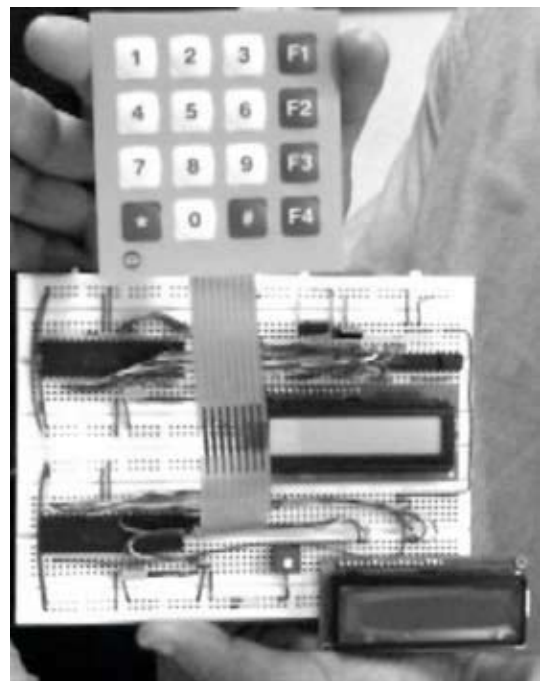


Figura 10. Prototipo de prueba.

CONCLUSIONES

Se diseñó y desarrolló con éxito una plataforma de hardware para simulación de controladores digitales bajo un esquema de generación de programas fuentes.

Con este esquema se pueden desarrollar plataformas para simular otras estrategias de control tales como: adaptativa, robusto, control difuso y otras técnicas.

Para la construcción de los generadores de programas se requiere tener conocimiento a fondo de los microcontroladores a utilizar, esto significa que la idea de un ambiente genérico para incluir todos los fabricantes se torna imposible, por ello definió un microcontrolador específico el cual cumple con los objetivos planteados para la simulación por hardware.

REFERENCIAS

- [1] M.B. Srivastava, T.I. Blumenau and R.W. Brodersen. "Design and implementation of a robot control system using a unified hardware-software rapid-prototyping framework". IEEE Conference Proceedings, International Conference on Computer Design: VLSI in Computers and Processors. ICCD '92. pp. 124-127. 1992.
- [2] M. Radetzki (Editor). "Languages for Embedded Systems and their Applications". Selected Contributions on Specification, Design, and Verification from FDL'08. Ed. Springer, Series: Lecture Notes in Electrical Engineering. Vol. 36. 2009.
- [3] G. de Micheli, R. Ernst. "Readings in Hardware/Software Co-Design (Systems on Silicon) (Paperback)". Ed. Academic Press, Morgan Kaufmann Publishers. 2002.
- [4] Richard C. Dorf y Robert H. Bishop. "Sistemas de Control Moderno". Ed. Pearson, Prentice Hall, 10ma Edición, pp. 37-310. 2005.
- [5] Michael A. Johnson and Mohammad H. Moradi. "PID Control, New Identification and Design Method". Ed. Springer-Verlag, 1ra Edición, pp. 47-107. 2005
- [6] Carlos A. Smith y Armando B. Corripio. "Control Automático de Procesos, Teoría y Práctica". Ed. LIMUSA, 1ra Edición, pp. 297-303. 1991.
- [7] Bela G. Liptak. "Instrument Engineers' Handbook, Fourth Edition: Process Control and Optimization". Ed. CRC, 4ta Edición, Vol. 2, pp. 414-431. 2006.
- [8] Katsuhiko Ogata. "Ingeniería de Control Moderna". Ed. Pearson Educación. 3ra Edición, pp. 669-683. 1997.
- [9] Alfred Roca Cusidó. "Control de Procesos". Ed. Alfaomefa. 2da Edición, pp. 51-151. 2005.
- [10] Aidan O'Dwyer. "Handbook of PI and PD Controller Tuning Rules". Ed. Imperial College Press. 2da Edición, pp. 154-460. 2006.
- [11] Katusiko Ogata. "Sistemas de Control el Tiempo Discreto". Ed. Pearson Educación. 2da Edición, pp. 114-121. 1995.
- [12] Cecil L. Smith. "Digital Computer Process Control". Ed. Intext Educational Publishers, pp. 136-179. 1972.
- [13] E.D. Bolat. K. Erkan and S. Postalcioglu. "Experimental Autotuning PID Control of Temperature Using Microcontroller". IEEE. The International Conference on Computer as a Tool, EUROCON 2005. Vol. 1, pp. 266-269. Noviembre 2005.
- [14] V. Chesaru, C. Dan y M. Bodea. "PID Algorithm for Controller Processors". IEEE, International Semiconductor Conference. Vol. 2, pp. 429-432. Septiembre 2006.