

DISEÑO Y CONSTRUCCIÓN DE UN PROGRAMADOR DE MICROCONTROLADORES PIC Y ATMEL MEDIANTE EL PUERTO USB DEL PC

Jami Gómez Wilson Iván, Ing.
Acosta Herrera Gandhi Javier, Ing.
Chico Hidalgo Patricio, MSc.

Resumen:

Los microcontroladores son dispositivos muy versátiles, y que en la actualidad se encuentran en la mayoría de aparatos electrónicos. El proceso para su uso consiste en escribir una aplicación usando un lenguaje de alto o bajo nivel, para luego obtener el programa en lenguaje de máquina y por último descargarlo hacia el microcontrolador desde donde se ejecutará.

Normalmente el fabricante provee las herramientas para la utilización de sus microcontroladores, tales como son: programas compiladores, hojas de datos, programadores, etc.

En el presente trabajo, se presenta el diseño, construcción y resultados obtenidos de un programador de microcontroladores fabricados por las empresas MICROCHIP, y ATMEL que sean programables en forma serial. Específicamente el circuito desarrollado toma el archivo entregado por el programa ensamblador o compilador proporcionado por los fabricantes (extensión .HEX) y lo transfiere al microcontrolador.

El programador está conformado por una placa de circuito impreso con sus componentes, un programa desarrollado para la tarjeta programadora (firmware) y un programa desarrollado para que sea ejecutado en un computador personal.

El enlace entre la tarjeta de circuito impreso y el computador se realiza mediante comunicación por el puerto USB. Cabe indicar que la energía necesaria para realizar las tareas de programación en la tarjeta programadora se toma del puerto USB del computador personal, por lo que no es necesaria una fuente externa adicional.

En el computador personal la aplicación que ha sido desarrollada, permite al usuario de una forma amigable acceder a todas las tareas relacionadas con el proceso de programación.

La visión con la que ha sido concebido el presente trabajo, difiere significativamente de los programadores comerciales pues permite tener un mayor control sobre las tareas que éste realiza.

El programador construido, da al usuario la libertad de incorporar o eliminar de una forma sencilla elementos de la biblioteca de microcontroladores soportados, y no únicamente se limita a la incorporación de microcontroladores, sino, que también pueden incluirse memorias u otro tipo de dispositivos que sean programables en forma serial. Con esta idea el programador va creciendo y se va actualizando conforme a las necesidades del usuario.



FIGURA 1: HARDWARE "EPNprog"

Hardware:

Un esquema general del hardware construido se muestra en la figura 2 , y tiene las siguientes partes principales:

Microcontrolador USB PIC 16C745 que se encarga de las comunicaciones con el computador personal usando el puerto de comunicaciones USB, y de los algoritmos de programación a los dispositivos, además de activar y desactivar el circuito de potencia cuando sea así requerido.

Circuito de Potencia, pues muchos de los dispositivos a ser programados necesitan niveles de voltaje que pueden ser mayores a los disponibles en el puerto USB (5voltios), por lo que para conseguir tal fin se ha

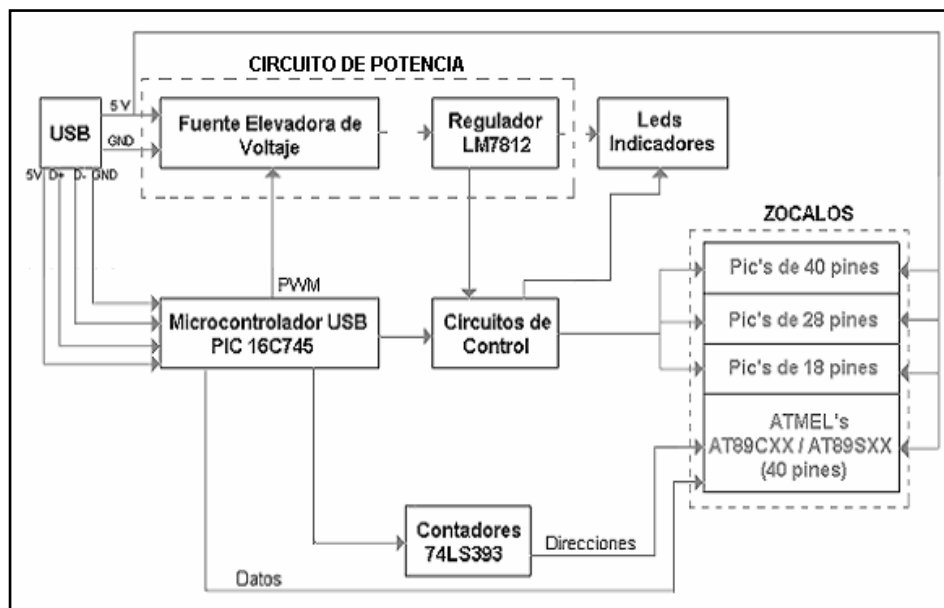


FIGURA 2: DIAGRAMA DEL HARDWARE CONSTRUIDO

diseñado un sistema elevador de voltaje para obtener los niveles requeridos. La energía para el funcionamiento de este bloque proviene del puerto USB del computador personal. En los dispositivos PIC y ATMEL, se eleva hasta 13 voltios

Circuito de Control que controla el flujo de las señales de programación a los respectivos pines del microcontrolador que se está programando.

Por último, se tienen indicadores (LED), que informan en todo momento el estado en que se encuentra la programación.

Software:

La aplicación que se ejecuta en el computador personal ha sido desarrollada pensando en el usuario, tratando siempre de que la manipulación del mismo se efectúe de la forma más amigable posible. Por lo que la ventana principal posee botones, lista de dispositivos, barras indicadoras del proceso de programación, y así todo lo inherente para la correcta manipulación.

La pantalla principal de la aplicación en el computador personal se muestra en la figura 3.

Entre las funciones principales que posee el software son:

- Permitir al usuario de una forma amigable programar un microcontrolador.
- Establecer y mantener la comunicación USB con el Hardware
- Mostrar en todo instante el estado del proceso de programación.
- Dar la oportunidad al usuario de introducir nuevos dispositivos en la lista de elementos que pueden ser programados por este programador.

Esta última característica que posee el software, se deriva de un análisis minucioso de los diferentes algoritmos encontrados en las hojas de datos proporcionadas por los fabricantes de los dispositivos soportados.

Las operaciones secuenciales que se deben ejecutar para programar estos dispositivos muestran una gran similitud entre todos ellos, y que consisten básicamente en datos binarios sincronizados con pulsos de reloj, y otras operaciones como retardos o condicionales que permiten desarrollarlos como una forma simple de secuencias lógicas.

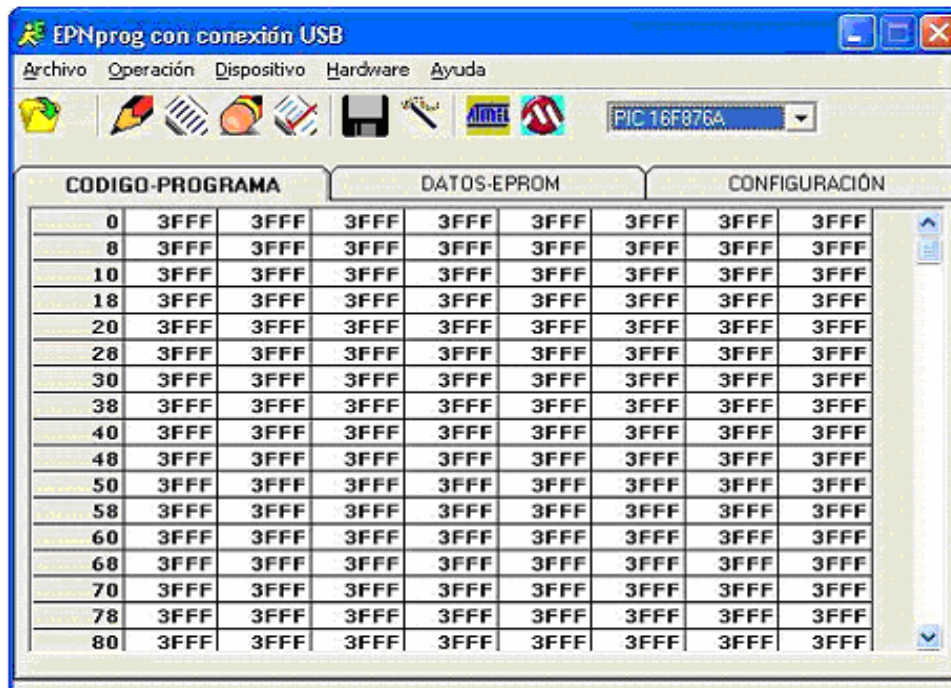


FIGURA 3: VENTANA PRINCIPAL DEL PROGRAMADOR

La construcción de secuencias lógicas simples se interpreta como la representación en un flujograma del proceso de programación que es proporcionado como un diagrama de tiempos o como una secuencia de eventos o como flujograma por el fabricante del dispositivo. Este flujograma será luego expresado como una secuencia de comandos básicos en la aplicación que corre en el computador personal, y que pasará a formar parte de la biblioteca de procedimientos para programar un determinado dispositivo.

En la programación de microcontroladores y otros dispositivos, se deben seguir de una forma ordenada las secuencias que especifica el fabricante. Tales secuencias, comprenden: la aplicación de niveles de voltaje en determinados terminales, así como también que por otros se transmitan o reciban datos binarios.

La aplicación que corre en el computador personal dispone de una herramienta denominada 'AGREGAR DISPOSITIVO' que de una forma amigable permite al usuario crear flujogramas para programar nuevos dispositivos.

Para la construcción de estos flujogramas el usuario dispone de cuatro tipos de eventos que se denominarán "unidades". Estas

unidades son suficientes para la construcción de los flujogramas de programación que cumplan con las secuencias de programación de los microcontroladores de los fabricantes mencionados y de muchos otros dispositivos.

Las unidades son de cuatro tipos, y se las denomina de la siguiente manera:

- Unidad "COMAND_DATO"
- Unidad "SALTO"
- Unidad "TIEMPO"
- Unidad "FIN"

El nombre que se ha dado a las unidades hace clara alusión a las funciones que de éste desempeña dentro del flujograma.

El usuario ubicará las unidades en el orden que más se ajuste a lo que indica el fabricante en las hojas de especificaciones para programar determinado espacio de memoria.

En la figura 4 se da un ejemplo simple de la manera como se pueden disponer de las unidades mencionadas para formar un flujograma.

Basados en los diagramas de tiempo y en las especificaciones de los fabricantes es posible desarrollar estos flujogramas únicamente con estos 4 tipos de unidades.

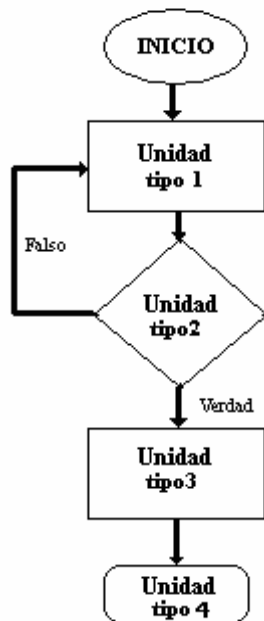


FIGURA 4: FLUJOGRAMA COMPUESTO DE UNIDADES

A continuación se detallan los cuatro tipos de unidades y su aplicación a este diagrama de flujo:

UNIDAD tipo 1: denominada **COMAND DATO** es la encargada de transmitir o recibir bits sincronizados con sus respectivos pulsos de reloj.

UNIDAD tipo 2: denominada **SALTO**, es la encargada de realizar una evaluación lógica, y en función del resultado que arroje (verdadero o falso), llevar la ejecución del flujograma por determinado camino.

UNIDAD tipo 3: denominada **TIEMPO**, producirá un retardo dentro de la ejecución del flujograma.

UNIDAD tipo 4: denominada **FIN**, dará por terminado la ejecución del flujograma.

Dentro de la herramienta “AGREGAR DISPOSITIVO” se encuentran estos cuatro tipos de unidades, y el usuario puede ocuparlas las veces que sean necesarias para construir flujogramas con excepción de la unidad FIN que debe ser única.

A modo de demostración se presenta en la figura 5 un flujograma incorporado con la herramienta mencionada. Este flujograma

permite borrar y luego programar el mapa de memoria FLASH de un PIC 16F877A.

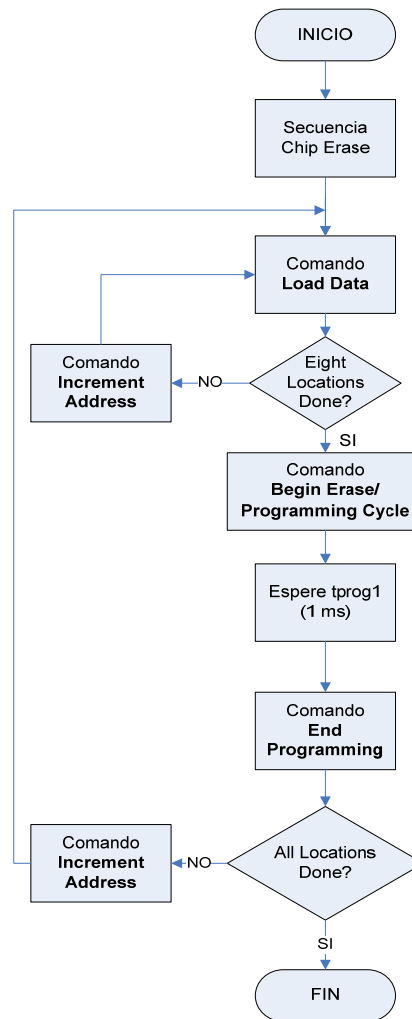


FIGURA 5. FLUJO PARA BORRAR Y PROGRAMAR UN PIC 16F877A

Este flujograma a su vez corresponde a una secuencia de eventos especificado por el mismo fabricante (tomado de PIC16F87xA FLASH Memory Programming Specification de MICROCHIP):

- :
1. Load a word at the current program memory address using the ‘Load Data’ command.
 2. Issue an ‘Increment Address’ command.
 3. Load a word at the current program memory address using the ‘Load Data’ command.
 4. Repeat Step 2 and Step 3 six times.
 5. Issue a ‘Begin Programming’ command to begin programming.
 6. Wait tprog (about 1 ms).
 7. Issue an ‘End Programming’ command.
 8. Increment to the next address.
 9. Repeat this sequence as required to write program and configuration memory.

El flujograma que se construya estará dedicado a realizar una función específica relativa a determinado espacio de memoria dentro del microcontrolador o dispositivo. Las posibles funciones del flujograma serán las de: Leer, Borrar y Programar. La función Verificar es el resultado de hacer una lectura y luego una comparación con un determinado archivo.

La guía que el usuario necesita para diseñar flujogramas las obtendrá de las hojas de especificaciones que proporciona el fabricante.

Cabe indicar que muchos dispositivos tienen algunas zonas de memoria no volátil que deben ser programadas utilizando diferentes algoritmos. Un ejemplo lo constituye el microcontrolador del ejemplo anterior que tiene memoria FLASH para el programa, una memoria no volátil de datos EEPROM y algunos registros de configuración especiales que son accesibles únicamente al tiempo de programación del dispositivo. Se deben crear por separado los flujogramas para todos estos algoritmos.

Se ha creado una codificación especial para cada uno de las unidades como se muestra a continuación:

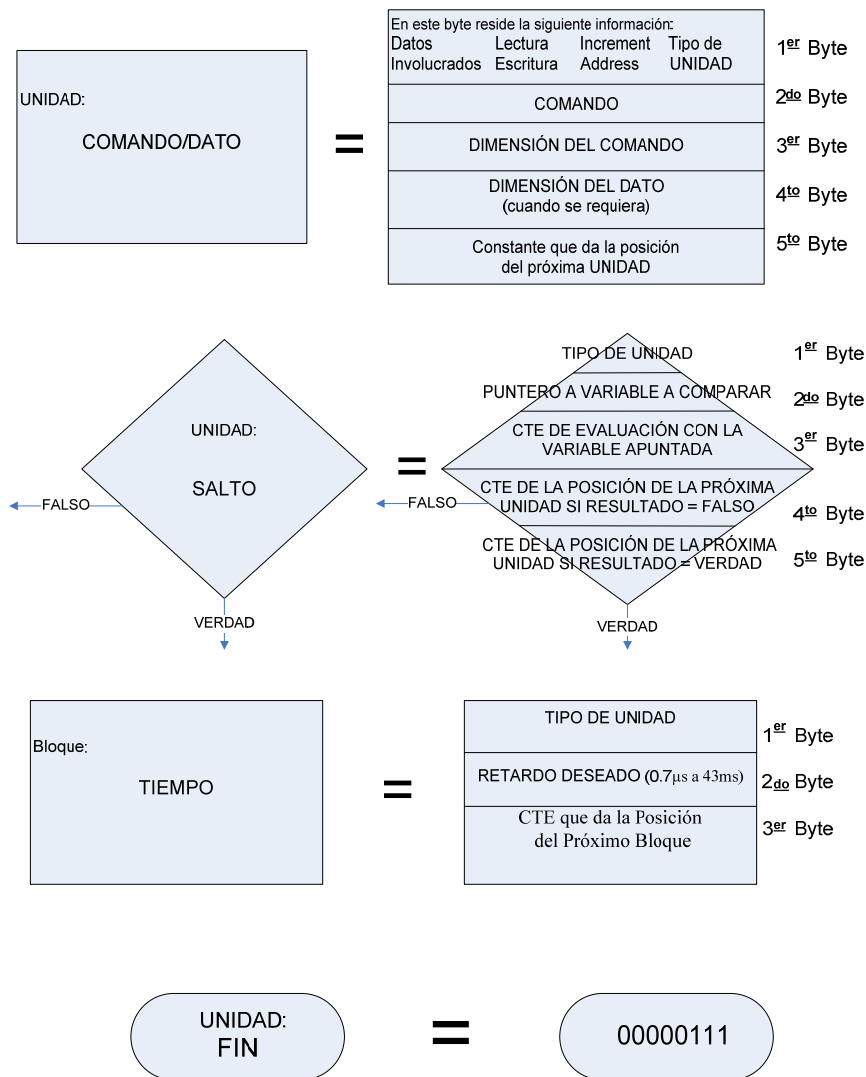


FIGURA 6: CODIFICACIÓN DE LAS DIFERENTES UNIDADES

De acuerdo al código que se ha establecido, y cuya descripción detallada no se va a tratar en esta publicación por motivos de espacio, el flujograma que se muestra en la figura 5 puede ser representado por la secuencia que se muestra en la figura 7.

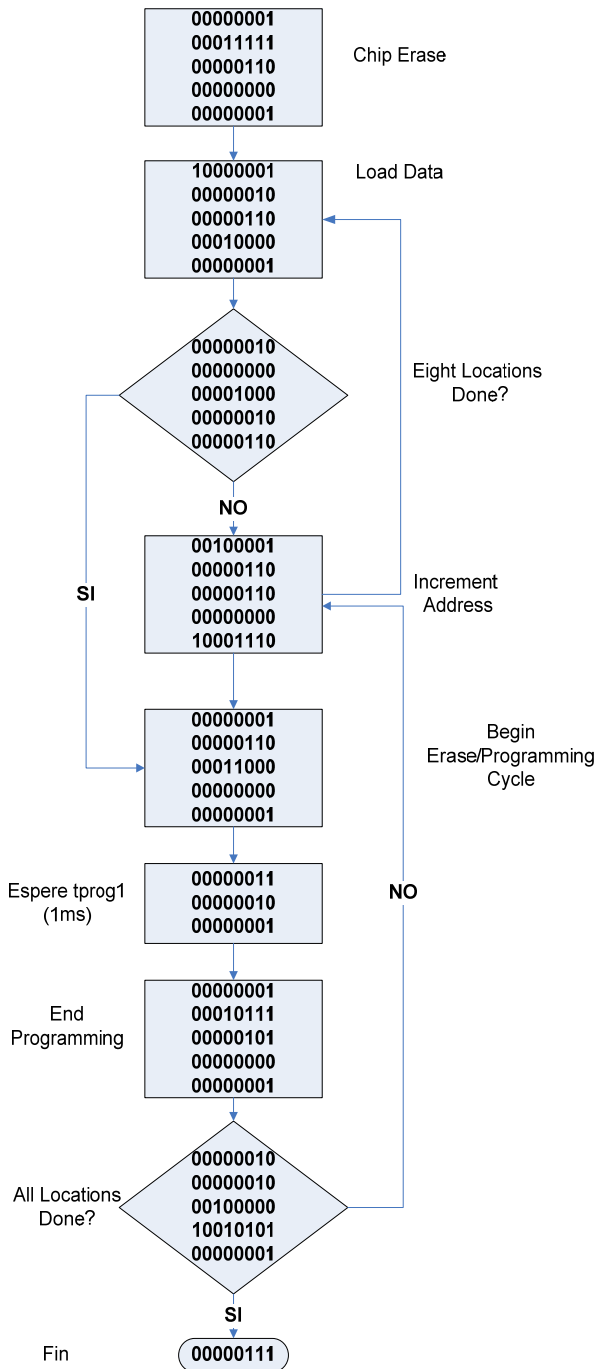


FIGURA 7: FLUJO CODIFICADO PARA BORRAR Y PROGRAMAR UN PIC 16F877A DE ACUERDO AL DIAGRAMA DE LA FIGURA 5

Las secuencias codificadas mostradas en la figura 7 deben ser generadas por el usuario e introducidas por una sola vez al programa que corre en el computador personal, el que lo almacenará y formarán parte de la biblioteca de procedimientos de los dispositivos soportados.

Cuando se desea hacer alguna operación sobre un dispositivo (programar, leer, borrar), el computador recupera estas secuencias de su biblioteca y la envía hacia la tarjeta programadora, donde el microcontrolador de la placa la interpreta y genera las señales adecuadas para una programación exitosa.

RESULTADOS

Luego de construido el programador, se procedió a incorporar un gran número de microcontroladores de programación serial de MICROCHIP y de ATMEL. Con esta biblioteca de elementos se procedió a realizar pruebas de programación, lectura, borrado y verificación de los dispositivos existentes en la biblioteca de dispositivos soportados.

El programador fue inicialmente concebido como una herramienta para programar microcontroladores de esos fabricantes, sobre todo algunos microcontroladores nuevos de los cuales los programadores comerciales disponibles no ofrecían soporte inmediato.

El programador soporta actualmente los siguientes microcontroladores:

Familias de Microchip: 16CXX, 16FXX, 16FXXA, 16F87X y 16F87XA.

Familias de ATMEL : AT89CXX y AT89SXX

Se han realizado pruebas con otros dispositivos seriales con éxito, y de estas pruebas y por el enfoque dado a la solución del problema se puede prever que una gran parte de los dispositivos de diversos fabricantes que se programen en forma serial, pueden ser programados con esta herramienta.

REFERENCIAS BIBLIOGRÁFICAS

JANN AXELSON, "USB Complete", Third Edition, Everything You Need to Develop Custom USB Peripherals, 2005

MICROCHIP, "Hojas de Especificaciones para Programación de los microcontroladores PIC 16F87XA, 16F87X, 16F8X Y 16CXX"

ATMEL, "Hojas de Especificaciones para Programación de los microcontroladores AT89C51/52/55 y AT89S51/52/55".

MICROCHIP, "Hojas de Datos del microcontrolador PIC 16C745"

CEBALLOS FRANCISCO JAVIER, "Curso de Programación Visual Basic 6.0. Alfaomega, 2000.

<http://pdf1.alldatasheet.com/datasheetpdf/view/51077/FAIRCHILD/DM74LS393M.html>

<http://www.alecmcnamara.freemove.co.uk/pc/calc/>

http://es.geocities.com/jnz_9zjn/

BIOGRAFÍAS DE LOS AUTORES:

ACOSTA HERRERA GANDHI JAVIER

Nace en Quito el 6 de mayo de 1980, realiza sus estudios primarios en la escuela EL CEBOLLAR LA SALLE (Quito), la secundaria en el colegio "SAN LUIS GONZAGA" (Quito), obteniendo el título de bachiller en la especialidad de "Físico Matemático", luego continua sus estudios superiores en la ESCUELA POLITÉCNICA NACIONAL obteniendo el título de **Ingeniero en Electrónica y Control**.

JAMI GÓMEZ WILSON IVÁN

Nace en Quito el 30 de agosto de 1979, realiza sus estudios primarios en la escuela OTTO AROSEMENA (Quito), la secundaria en el colegio técnico "MIGUEL DE SANTIAGO" (Quito), obteniendo el título de

bachiller técnico en la especialidad de "Electromecánica", luego continua sus estudios superiores en la ESCUELA POLITÉCNICA NACIONAL obteniendo el título de **Ingeniero en Electrónica y Control**

CHICO HIDALGO PATRICIO IVÁN

Ingeniero en Electrónica y Control. Escuela Politécnica Nacional 1987. **Master of Science in Electrical Engineering,** University of Texas at Arlington, 1994. Actualmente es profesor principal en la Escuela Politécnica Nacional en las áreas de Control Electrónico de Potencia y Microprocesadores.