

# **Diseño e implementación de un sistema de adquisición de datos y generación de señales de control con monitoreo gráfico para una bancada de prueba de motores eléctricos**

## AUTORES

Edgar Cañar Vargas<sup>1</sup>, Luis Baque Realpe<sup>2</sup>, Alberto Larco Gómez<sup>3</sup>

<sup>1</sup>Ingeniero Eléctrico en Electrónica Industrial 1998

<sup>2</sup>Ingeniero Eléctrico en Electrónica Industrial 1998

<sup>3</sup>Director de Tesis, Ingeniero Eléctrico, Escuela Superior Politécnica del Litoral, 1985. Profesor ESPOL desde 1.981.

## RESUMEN

En nuestro medio existen diferentes tipos de motores, el motor que se trata en esta tesis es el motor eléctrico. En los tipos de motores eléctricos tenemos el motor de corriente alterna (AC) y motor de corriente directa (DC).

Este proyecto de tesis consiste en crear un módulo supervisor y semicontrolador de un motor DC de 1.2 Kw. El motor es manejado por un computador standard que a su vez monitorea cuatro señales características del mismo (voltaje de armadura, corriente, velocidad y torque de carga). Estas señales son mostradas en el monitor del computador, todas a la vez o una a la vez; dependiendo del tipo de análisis que se quiera efectuar.

Para realizar las pruebas antes mencionadas nos apoyamos en varios equipos del Laboratorio de Electrónica de Potencia y el módulo de interface de adquisición y envío de datos, que es el equipo a diseñar y construir. La parte más importante de las tesis consiste en comunicar la parte digital de las señales codificadas al puerto paralelo, pues debe existir una compatibilidad tanto en la parte eléctrica, electrónica y a nivel de software, controlando la recepción y envío de datos.

## INTRODUCCION

Cuando se realiza una práctica en el laboratorio de Electrónica de Potencia, se dispone de algunos instrumentos de medición, de los cuales sobresale el osciloscopio. La mayoría de ellos tiene dos canales de medición de señales, pero ofrece toda una gama de utilidades para el monitoreo de un motor por ejemplo.

La investigación del tema se inicia por la necesidad de contar en el laboratorio de electrónica con un instrumento virtual, el mismo que monitoree las señales y a la vez produzca variaciones en los parámetros del motor. Entonces se trabaja en una idea adicional, esto es, que su control deba hacerlo un computador, pudiendo modificarse el programa controlador según las necesidades de análisis. El equipo en si es solo un prototipo de lo que se puede crear según la necesidad, pudiendo crearse por ejemplo no 4 señales de entrada para monitorear, sino 8, 16, 128 o las que se deseen. Así mismo el control está sujeto a muchas posibilidades de salida.

El programa controlador es el que administra el trafico de señales y establece su control, sujeto por supuesto a la arquitectura del diseño del hardware.

## CONTENIDO

### Adquisición y envío de datos

Hoy en día la mayoría de los fenómenos físicos o análogos pueden ser convertidos en señales codificadas digitales, por lo que existen muchos textos que hablan acerca de este proceso (5). En todo proceso de adquisición de datos existen sensores y transductores. Para el voltaje utilizamos un circuito divisor de voltaje, que reduce el voltaje de armadura del motor de 220V a 5V (4); para la forma de onda de la corriente un circuito electrónico, que se escala a través de la caída de voltaje en un resistor; para la velocidad un tacómetro DC; para el torque un medidor de deformación "strain gauge".

En el proyecto se toman las cuatro señales del motor: voltaje, corriente, velocidad y torque, las cuales son llevadas al computador a través del módulo de interface, que esta dividido en dos tarjetas, una tarjeta de fuerza, que es la encargada de recibir la señal del motor, amplificarla para que luego la tarjeta digital pueda procesarla en valores lógicos para su ingreso al computador.

La tarjeta digital que es la encargada de recibir las señales analógicas y convertirlas a códigos binarios, es en si una tarjeta de adquisición de datos. La conversión se realiza a través de los integrados (cuatro) ADC08161 de la National Semiconductor, estos cuentan con un reloj propio por lo que no hace falta crear un reloj para este. Los valores límites en los que recibe las señales son de 0V a 5 V, con una resolución binaria de 8 bits.

Estos 8 bits son transportados al puerto paralelo, siendo el computador el que administra su ingreso. En la Fig. 1 se presentan los diagramas de bloques del sistema.

### El puerto paralelo

Este es el dispositivo por el cual vamos a intercomunicar el módulo de adquisición y envío de datos con el computador. La tabla 1 muestra las señales en el conector del puerto paralelo; estas señales siguen mas o menos la interface de impresora popularizada por *Centronics*, a pesar de que, el conector de 25 pines no usa las 36 líneas de la interface original. Aunque cada señal tiene un nombre que sugiere una función particular, no necesariamente deben usarse para el mismo propósito, como es el trabajo de este proyecto.

### Modo de operación nibble del puerto paralelo

El modo nibble es el modo más común o también llamado modo standard. Todos los puertos paralelos standard están provistos de 5 líneas desde el periférico hacia el computador. Esta información es llevada a través del registro de status en secuencias de 8 bits (en lotes de 4 bits). Generalmente la quinta línea es utilizada para indicar cual de los dos lotes está ingresando. Este es el registro por el cual está ingresando la información de las señales codificadas.

### Descripción de la interacción de los componentes del sistema

Un diagrama de bloques de todo el sistema empleado en nuestro proyecto se presenta en la Fig. 2

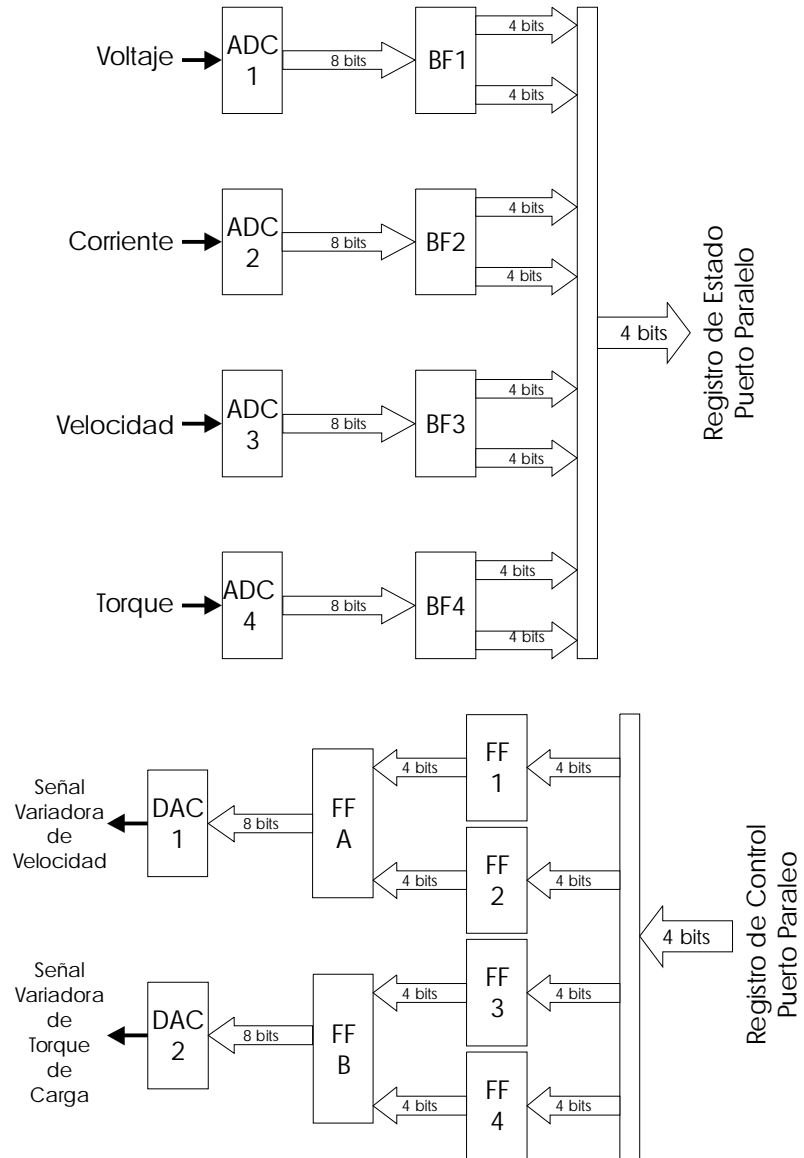


Fig. 1 Diagrama de bloques del sistema de adquisición y envío de datos

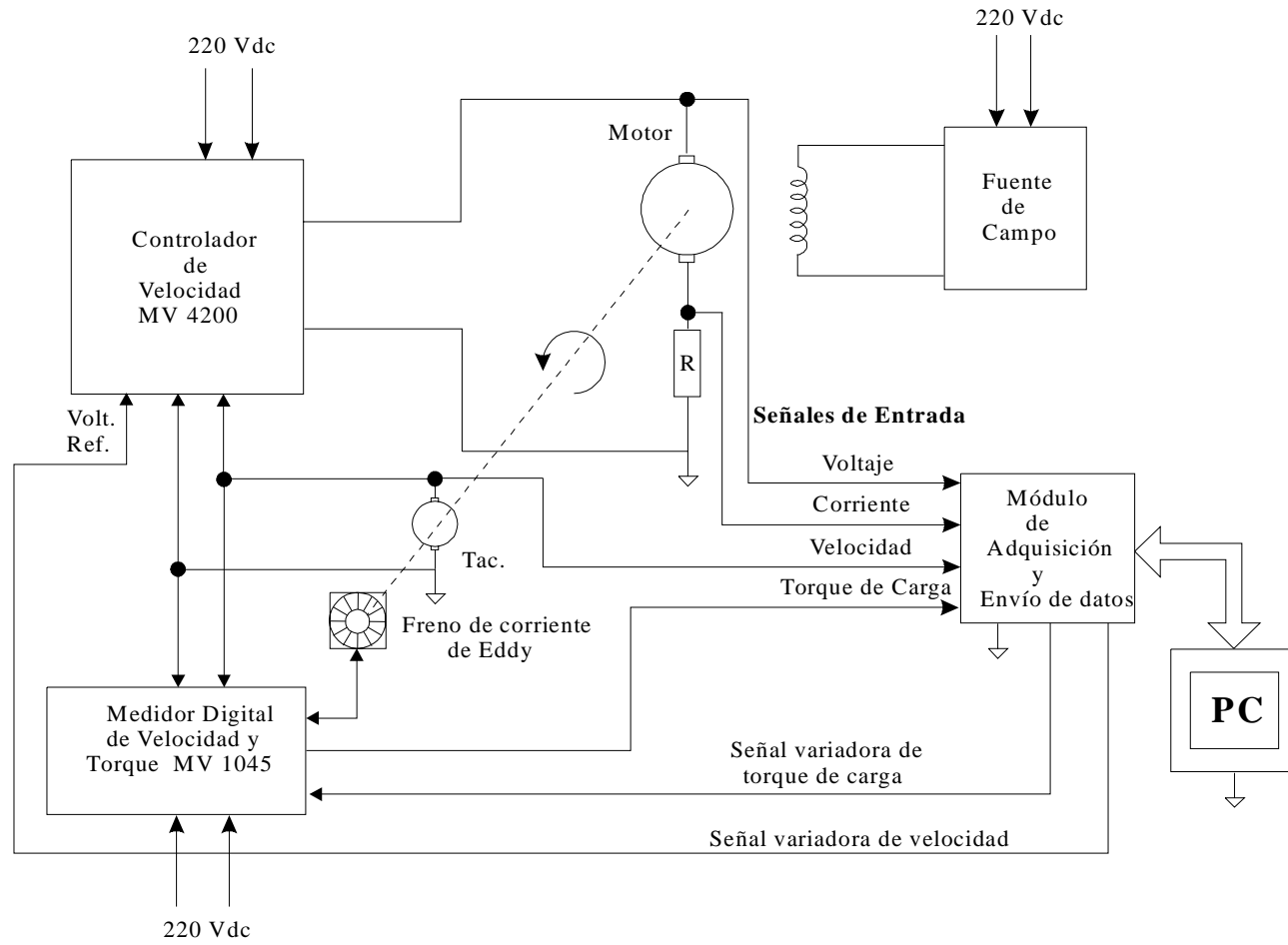


Fig. 2 Diagrama de bloques de la conexión de los componentes del sistema

Además todos los componentes que interactúan con el motor son interconectados para regular la velocidad por tacómetro, tal como indica el catálogo del controlador de velocidad para motores de corriente continua (2).

Los circuitos empleados para sensar y obtener datos sirven para que el computador adquiera las señales en forma de bits luego de haber pasado por la sección de conversión analógica/digital (ADC).

El sistema utilizado para captar las señales del motor fue escogido de tal manera que dichas señales fueran sensadas tratando de que exista la menor cantidad de pérdidas.

La señal de voltaje es medida directamente de las terminales del motor y reducida por medio de un divisor de voltaje, siendo una señal que no sufre ninguna deformación lo cual se demuestra en la gráfica obtenida por el computador.

La señal de corriente es obtenida a través de un resistor conectado en la misma línea del motor, lo que garantiza una señal prácticamente real en su forma, luego la señal pasa hacia el ADC, antes de su ingreso hacia el computador.

Los datos de velocidad son obtenidos por medio de un tacómetro DC ubicado en el eje motor, y son escalados por medio de un divisor de voltaje, de forma semejante al de la señal de voltaje, luego se emplea una circuitería de acondicionamiento hacia el ADC, antes de su ingreso al computador.

Los datos de torque son dados por el medidor de torque de carga y velocidad ( MV 1045 ), a través de una salida de corriente en una de sus terminales de salida para interfaces externas y la circuitería en el módulo de adquisición y envío de datos permite el paso hacia el computador.

Los circuitos de envío de datos son los que sirven para adaptar las señales analógicas hacia el controlador de velocidad y el medidor de torque y velocidad, luego que los bits han pasado a señales analógicas por medio de los convertidores digital/analógicos (DAC).

#### Programa de supervisión e interface con el usuario

El computador para comunicarse con la interface que controla el motor utiliza dos herramientas que son el puerto paralelo físico, el puerto paralelo virtual, que es el programa o software encargado de cambiar el estado del puerto físico, por medio de señales emitidas a conveniencia de los requerimientos del programa escrito en lenguaje C++.

El programa de supervisión es creado en lenguaje C++ V 3.0 bajo DOS. Para la presentación, este programa utiliza opciones de modo gráfico, que permite crear controles que hagan más fácil su maniobrabilidad. El programa tiene como finalidad dar al usuario las opciones necesarias para manipular la lectura y escritura del puerto paralelo, así como también procesar estos datos leídos para presentar a partir de los mismos, reportes que permitan analizar el funcionamiento del motor en tiempo real.

El programa fuente consta de 6 archivos PRINCIPA.CPP, GRAFICA.CPP, ACCION.CPP, OBJETO.CPP, CARGA.CPP, VELOCIDAD.CPP, los cuales cumplen su función que serán explicados en el transcurso de este capítulo; todos estos archivos se enlazan para formar un solo archivo ejecutable el cual tiene el nombre de (motor.exe). La pantalla inicial del programa se muestra en la Fig. 3

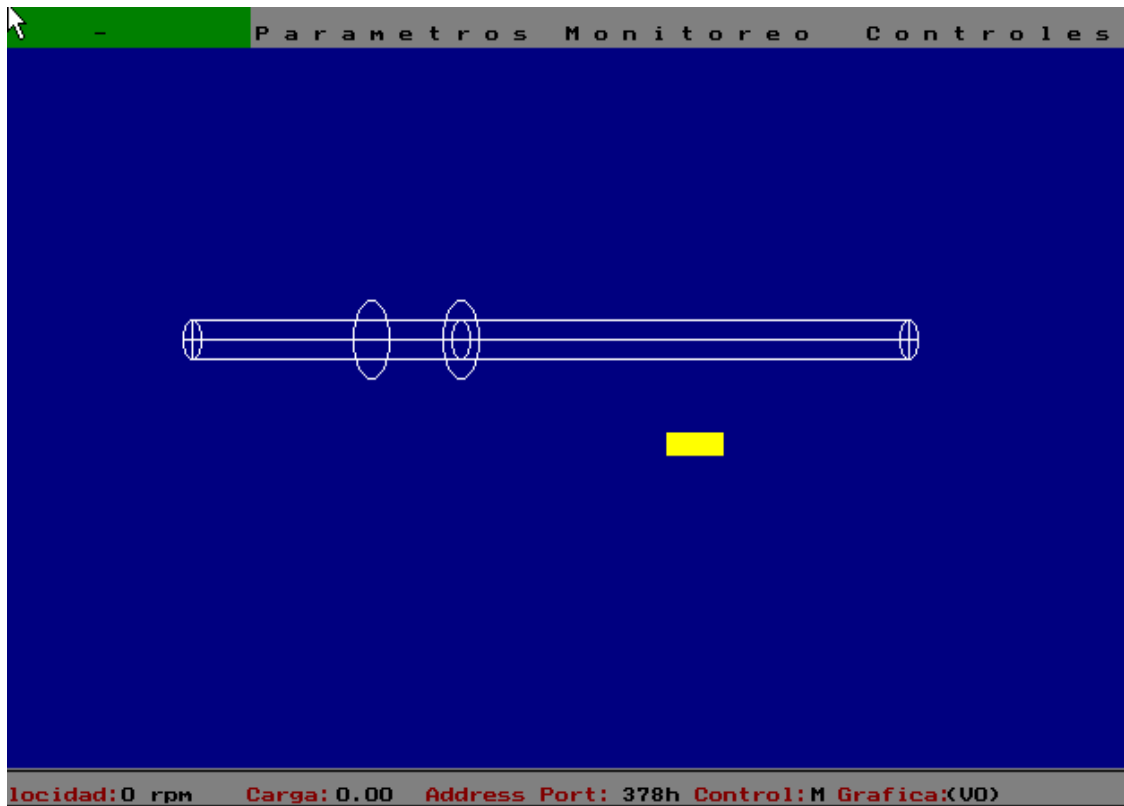


Fig. 3 Pantalla inicial del sistema controlador

#### Generación de datos para controlar el motor

El programa recibe dos tipos de datos desde el teclado, los cuales son velocidad y carga. La generación de los datos para controlar el motor, consiste en filtrar y enmascarar los valores de velocidad y carga que el usuario desea enviar por el puerto paralelo. Los datos generados para controlar el motor son ingresados por el usuario en RPM en caso de la velocidad y en newton metros en caso de la carga,

El computador transforma estos datos ingresados a su equivalente en BYTES (número del 0 al 255)

Este número en byte es el que recibe el puerto paralelo.

La comunicación del programa en lenguaje "C" con el puerto paralelo se realiza a través de los comandos **inport** (para leer el puerto paralelo) y **outport** (para escribir al puerto paralelo) (3).

Manual del usuario

Procesador	Pentium	100mhz
Monitor	Color	VGA
Memoria	16M O MÁS	
Puerto paralelo	Standard Parallel Port	
Sistema Operativo	DOS 6.0 o Superior	

Tabla 2. Requerimiento del sistema:

Descripción de las interfaces del Programa

Guía De Instalación

Para la instalación de este programa se siguen los siguientes pasos:

1. Crear un directorio con el nombre de motor
2. Copiar en el directorio creado el programa **motor.exe**
3. Copiar en este directorio el archivo **EGAVGA.BGI**

Menú De Parámetros

**Dirección de Puertos:**

Esta opción del menú contiene las direcciones de los puertos más usados en la PC, los cuales son (378h,278h,3BCh)

Por omisión la dirección de puerto elegido es 378h,

El puerto escogido se puede visualizar en la barra de estados.

**Arranque**

Esta opción del menú permite escoger una velocidad de inicio, que es el tiempo de aceleración que le toma al motor para vencer su inercia, se colocan diferentes velocidades como opciones (50 rpm, 10 rpm, 150 rpm)

Menú de Monitoreo

**Gráfico**

Esta opción del menú permite escoger el tipo de gráfica a visualizar las cuales son Velocidad, Corriente, Voltaje, Torque.

Los tipos de ondas escogidas se verán en la barra de estado (vo=voltaje, co=corriente, ve=velocidad, to=torque)

Menú de Controles

### **Automático**

Esta opción del menú presenta una ventana en la cual se puede ingresar un conjunto de velocidades y carga, los cuales van a ser enviadas automáticamente controlada por un intervalo de tiempo que el usuario también ingresará.

El tiempo  $t$  se refiere al número de despliegues de ondas que debe transcurrir para que se realice el cambio de velocidad y de carga, así por ejemplo si se coloca  $t=5$ , significa que se darán cinco pantallazos hasta ver el siguiente cambio. En la Fig. 4 se muestra un ejemplo a cerca del manejo de esta opción.

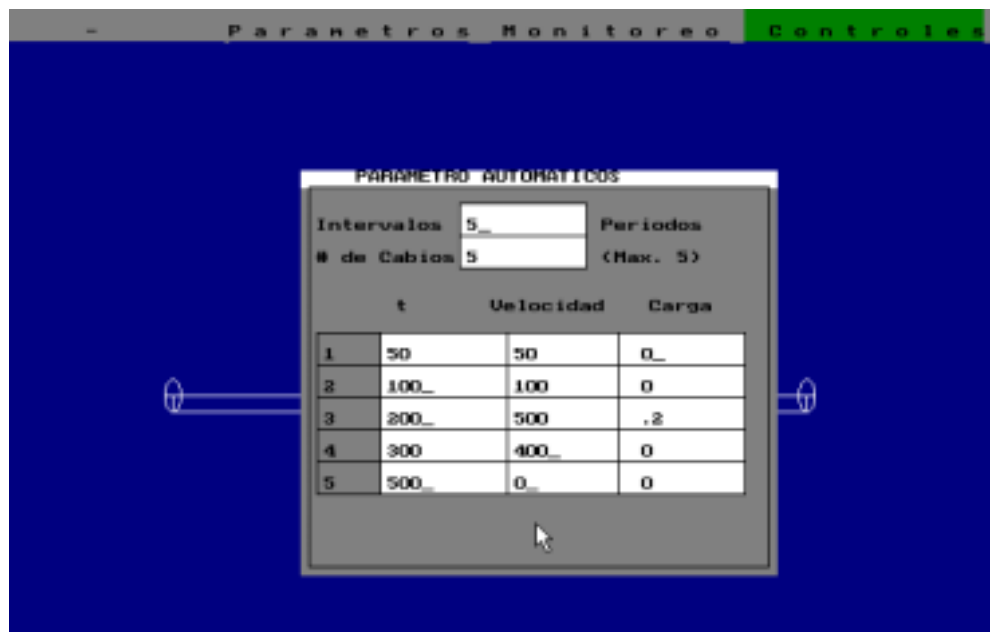


Fig. 4 Ejemplo para la Opción Automatico del sistema

### **Manual**

Esta opción del menú permite anular el modo automático del programa, permitiendo que el usuario tenga control del envío de órdenes al motor

### **Controlar**

Con esta opción el usuario ingresa a una interface interactiva, en la cual visualiza, las ondas de corriente, voltaje, velocidad, torque que emite el motor, además, el usuario podrá controlar el movimiento del motor enviándole variaciones de velocidad o carga.

Esta pantalla se divide en 4 partes:

1. Una porción de la pantalla que permite visualizar las ondas de voltaje, corriente, velocidad, torque.
2. La franja de controles que permite manipular el motor y observar la respuesta del mismo.



3. La franja de parámetros que permite saber la velocidad con que inicio el motor, la carga con que inicio el motor, la dirección del puerto paralelo por donde se envían los datos al motor, el modo de control (M>manual, A=automático), y el tipo de gráfico que se presenta en la pantalla.
4. La franja o barra de mensajes que da a conocer las principales teclas con las cuales manejar el programa.

#### Conclusiones Y Recomendaciones

Una vez concluido el proyecto y observado los resultados se observa, que trabajar con el puerto paralelo no es tan complicado. Si agregamos a ello, los conocimientos en la programación del lenguaje C++ bajo DOS; se podría desarrollar no solo un controlador para un motor sino, algún otro tipo de controlador con diferente propósito; por ejemplo, aplicable a la seguridad en el hogar o maquinaria en alguna industria.

El software del proyecto estuvo planificado realizarlo en Windows 95, bajo el lenguaje Visual C++. Lo complicado es la programación, la cual demanda un mayor conocimiento para los estudiantes de electrónica industrial. Esta complicación condujo a emplear Visual Basic 5.0. Todo marchaba bien, pero en una de las pruebas se observó que Windows 95 en su sistema interno, realiza algún tipo de inspección sobre los periféricos conectados al computador y desconfigura completamente al puerto paralelo en un momento determinado, tiempo que, no permitía mantener los cambios en los parámetros del motor. Por este motivo se decidió regresar al lenguaje C++ 3.0 bajo DOS.

En las investigaciones realizadas, conocimos que a más del puerto SPP (Puerto Paralelo Standard), existen otros puertos diseñados bajo la norma IEEE 1284, que crea un puerto paralelo real de dos vías, esto es, los 8 bits del registro de DATA son bidireccionales. Esto sirve para simplificar el tipo de hardware que se requiera diseñar. La programación era igualmente complicada como el Visual C++, por lo que se decidió trabajar con el puerto SPP.

Lo interesante del proyecto es el trabajo con el puerto paralelo, realizando control sobre algún dispositivo externo de tipo electrónico.

Una de las recomendaciones sería el trabajar con ADC de varios canales, y realizar un controlador externo para la parte digital del sistema, de esta manera se hace más flexible o universal la aplicación del módulo supervisor, pudiéndose crear con mayor facilidad otros programas con propósitos diferentes.

Así mismo, de las investigaciones realizadas, existen softwares para este propósito, diseñados en Windows 95 que superan el problema de la configuración del puerto, por lo cual, se recomienda investigar las técnicas que hacen esto posible. En la mayoría de los softwares mencionan el uso de una tarjeta de adquisición en el bus de datos, que manejan de manera más rápida y confiable la transferencia de información a través del puerto.

#### Referencias

1. Edgar Cañar y Luis Baque, "Diseño e implementación de un sistema de adquisición de datos y generación de señales de control con monitoreo gráfico para una bancada de prueba de motores eléctricos DC" (Tesis, Facultad de Ingeniería Eléctrica, Escuela Superior Politécnica del Litoral, 1998)
2. TERCO, Thyristor Unit 4200, Theory and Laboratory Exercises, Manual del Laboratorio ESPOL, Guayaquil 1990, pp. 4
3. Javier Cevallos Sierra, Enciclopedia del Lenguaje C++, Ra-ma, U.S.A. 1993. pp 384
4. Harry Fox, Master OP-AMP Applications Handbook, Folleto ESPOL, Guayaquil, 1988. pp102

5. Ronald J. Tocci, *Sistemas Digitales*, Prentice Hall Hispanoamericana S.A, México, 1987, pp. 452-463