



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA

30  
24.

Conexión de una tarjeta ProControl PA-CP12 con el  
software LabVIEW y su uso en la producción de programas  
de cómputo con fines docentes

TESIS

QUE PARA OBTENER EL TÍTULO DE:

Ingeniero en Computación

PRESENTAN:

Isabel Georgina Gallardo Valadez

Aline Loza Carbajal

Asesor de Tesis Dr. Rafael Fernández Flores

MÉXICO, D.F.

1998

TESIS CON  
FALLA DE ORIGEN

265233



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## AGRADECIMIENTOS

GRACIAS A NUESTRA UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO POR DARNOS LA OPORTUNIDAD DE SER PERSONAS ÚTILES AL PAÍS Y A LA SOCIEDAD.

GRACIAS A LA FACULTAD DE INGENIERÍA POR ALOJARNOS EN SUS AULAS.

GRACIAS AL DR. RAFAEL FERNÁNDEZ FLORES, POR CREER EN NOSOTRAS Y BRINDARNOS LA MARAVILLOSA OPORTUNIDAD DE TRABAJAR A SU LADO DIRIGIENDO EXTRAORDINARIAMENTE ESTA TESIS, POR SU APOYO EN TODO MOMENTO, SU ENORME PACIENCIA Y POR TODO EL TIEMPO QUE NOS REGALO PARA CUMPLIR JUNTO A ÉL ESTA META.

GRACIAS A TODOS NUESTROS MAESTROS POR DEDICAR PARTE DE SU VIDA EN TRANSMITIRNOS SUS CONOCIMIENTOS Y EXPERIENCIAS.

GRACIAS A TODOS AQUELLOS COMPAÑEROS QUE COMO NOSOTRAS TIENEN LA ILUSIÓN DE CONCLUIR UNA PROFESIÓN.

GRACIAS A TODOS LOS INTEGRANTES DEL LABORATORIO DE ENSEÑANZA Y APRENDIZAJE ASISTIDO POR COMPUTADORA, POR APOYARNOS Y COLABORAR CON NOSOTRAS EN LA REALIZACIÓN DE ESTA TESIS, ESPECIALMENTE A DANIEL MARTINEZ PLASCENCIA POR SU INVALUABLE PARTICIPACIÓN Y AMISTAD.

GRACIAS A LA LIC. LIBIA BARAJAS MARISCAL, POR APOYARNOS INCONDICIONALMENTE CON TODOS LOS RECURSOS DEL LEAAC.

GRACIAS A LOS PROFESORES QUE COLABORARON EN ESTE TRABAJO, EN ESPECIAL A: LUPITA, YOLANDA Y TOÑITA, POR LAS FACILIDADES PROPORCIONADAS EN SUS LABORATORIOS, A CUPERTINO RUBIO POR SU DISPONIBILIDAD, A JUAN ANTONIO FLORES Y SUSANA RAMIREZ POR SU APOYO Y AMISTAD.

GRACIAS AL ING. DIEGO MEJIA, POR COLABORAR DESINTERESADAMENTE EN LA PARTE FINAL, SIENDO UN AMIGO CUANDO LA DESESPERACIÓN NOS INVADÍA.

GRACIAS A NUESTRO AMIGO ESTEBAN QUIEN EN SU CORTA EXISTENCIA NOS DEMOSTRÓ LA ENTEREZA E ILUSIONES DE LA JUVENTUD.

GRACIAS A TODOS LOS QUE SE TOMEN LA MOLESTIA DE LEER ESTAS LÍNEAS Y CONTINUEN CON LA LABOR QUE NOSOTRAS EMPRENDIMOS, APOYANDO LA EDUCACIÓN MEDIANTE EL CÓMPUTO.

## *DEDICATORIA*

DEDICO CON TODO MI CORAZÓN Y AMOR, EL ESFUERZO, SACRIFICIOS, SATISFACCIONES Y EL ÉXITO QUE PUEDA TRAER, ESTE TRABAJO, CON EL QUE CONCLUYO UNA DE LAS ETAPAS MÁS IMPORTANTES DE MI VIDA, COMPROMETIENDO CONTIGO Y CONMIGO A HONRAR Y LLEVAR MUY EN ALTO MI PROFESIÓN POR TODOS LOS DÍAS QUE ME RESTEN DE VIDA, SIENDO CADA DÍA UN POCO MEJOR QUE AYER Y MENOS QUE MAÑANA.

GRACIAS A DIOS, POR DARME LA MARAVILLOSA OPORTUNIDAD DE EXISTIR Y DE VIVIR LA VIDA QUE ME TOCO Y SER COMO SOY "A MI MANERA".

MIL GRACIAS, A MI VIEJITO DEL ALMA, QUE ME ENSEÑO EL VALOR DE VIVIR, QUE CON SU EJEMPLO SEMBRÓ EN MI LA SEMILLA DE LA SUPERACIÓN, LA HONRADEZ Y EL AMOR AL TRABAJO, A TI ABUELITO MÁS QUE A NADIE LE OFREZCO ESTE TRABAJO, CON EL QUE CUMPLO MI PROMESA DE SER LO QUE HOY SOY, CONSERVANDO LA ETERNA ESPERANZA DE REUNIRNOS AL FINAL. A TI MI SER MÁS AMADO.

AGRADEZCO CON TODO MI AMOR, A MI MADRE, POR DARME LA VIDA, POR OBLIGARME A IR A LA ESCUELA Y ESTAR CONMIGO SIEMPRE PARA LLEGAR JUNTAS HASTA AL FINAL.

AGRADEZCO CON TODO MI RESPETO, A MI PADRE, POR SIEMPRE SER EXIGENTE CONMIGO Y ENSEÑARME EL VALOR DE LA RESPONSABILIDAD, VIEJO, NO TE FALLE.

AGRADEZCO CON TODA MI ALMA, A MI ABUELITA ANGELITA, POR CUIDARME DESDE QUE NACÍ HASTA HOY Y DARME TODO EL AMOR DE UNA MADRE.

AGRADEZCO A MI ABUELITO ISABEL POR SIEMPRE CREER EN MI, AUNQUE HOY YA NO ESTAS AQUÍ, SÉ QUE ESTAS ORGULLOSO DE MI Y A MI ABUELITA CRUZ, POR ESA FORTALEZA.

AGRADEZCO A MIS HERMANOS JESÚS, LUZ MARÍA Y VERÓNICA, POR RECORRER JUNTOS ESTE CAMINO, POR ESE APOYO, POR ESA CONFIANZA, COMPARTO ESTE TRIUNFO DEL CUAL GRAN PARTE LES PERTENECE.

UN AGRADECIMIENTO ESPECIAL, A LA FAMILIA LOZA-CARBAJAL-NAVARRETE POR ADOPTARME COMO UN MIEMBRO MÁS DE SU HERMOSA FAMILIA, A TI MI ADORADA ALINE POR SER MÁS QUE COMPAÑERA Y AMIGA, POR SER MI HERMANA Y ESTAR LLEGAR JUNTAS AL FIN COMO SE QUE SIEMPRE ESTAREMOS.

GRACIAS, A MI DEMÁS FAMILIA, EN PARTICULAR A BETTY Y LUPITA POR CREER EN MI Y OFRECERME LAS PUERTAS DE SU CASA Y DE SU CORAZÓN.

GRACIAS MIL A TODOS MIS AMIGOS, POR ESTAR SIEMPRE APOYÁNDOME Y COMPARTIENDO CONMIGO SU VIDA, AMISTAD Y CARIÑO, A OSCAR, HORACIO, IRAM, JOSUE, TOÑO, BAZAN, EMMA, AYMEÉ, VERO, AHMED, CYNTHIA, MARCOS, GABY, EDNA, ANTONIO, CARLOS, RAMÓN, GUILLERMO, ELOHIM, DANIEL, KARINA, IVETTE, OSVALDO, MARIANA, CLAUDIA, GABRIELA, ANA GLORIA, SUSANA, CARMEN, A MIS AMIGOS DE PEMEX POR SUS ENSEÑANZAS, Y A TODOS LOS QUE SE ME ESCAPAN DE LA MEMORIA PERO NUNCA DEL ALMA.

FINALMENTE A MI FLACO, POR CREER EN MI, POR APOYARME, POR HACERME FELIZ, POR HACERME SUFRIR, POR DARMER LA OPORTUNIDAD DE AMAR Y SER AMADA, POR DEJARME SER COMO QUIERO SER Y POR ESOS MARAVILLOSOS TRES AÑOS Y MEDIO DE MI VIDA QUE PERDÍ A SU LADO.

Y LUCHO PARA QUE AL FINAL DEL CAMINO CUANDO YO YA NO EXISTA DIGAN “QUE MI LEGADO FUE TRATAR DE DEJAR UN MUNDO MEJOR QUE EL QUE ENCONTRÉ” Y EN EL OCASO DEL PARPADEO ENTRE DOS ETERNIDADES LLAMADO VIDA PODER DECIR CON ORGULLO “VIDA NADA ME DEBES, VIDA NADA TE DEBO, VIDA POR FIN ESTAMOS EN PAZ”.

ISABEL GEORGINA GALLARDO VALADEZ

## DEDICATORIAS

A Dios por darme el Don maravilloso de la vida.

A mis Padres

Lic. Rubén Loza Manjarrez

Profra. Myrna Carbajal Labrada

Que me enseñaron el valor de la vida y el de ser feliz. Gracias por su confianza y su apoyo, y su gran amor que me fortalece cada momento de mi vida.

A mi Abuelita Soco: Por toda su dedicación y cariño. Por ser la mejor consejera y amiga.

A mis Hermanos Malena, Cecilia, Jorge, Alejandro, Alexandra, Gabriela y Diego: Por ser los mejores hermanos y por todo lo que hemos compartido.

A mis Sobrinos: Por ser la alegría que me impulsa a seguir.

A mi Familia: Por estar conmigo en todo momento y por creer siempre en mí.

A mis Amigos: Por los momentos que hemos vivido juntos y por su ayuda incondicional. A Isa por el aguante durante este trabajo.

Dedico esta tesis a las personas que hoy no están conmigo, pero que en vida fueron importantes para mí.

Aline Loza Carbajal

# INDICE

Introducción	
Presentación	2
Planteamiento del problema	3
Antecedentes	7
<b>Capítulo I. Captura automática de datos</b>	<b>15</b>
I.1 Tarjetas de adquisición de datos	16
I.2 Software de adquisición de datos	19
<b>Capítulo II. Tarjeta ProControl PA-CP12</b>	<b>39</b>
II.1 Convertidor analógico/digital	45
II.2 Convertidor digital/analógico	52
II.3 Entradas y salidas digitales	55
<b>Capítulo III. Software LabVIEW</b>	<b>56</b>
III.1 Lectura de la tarjeta	62
III.2 Compilador de LabVIEW	79
III.3 Interacción LabVIEW - Windows 95	87
<b>Capítulo IV. Conexión ProControl - LabVIEW</b>	<b>89</b>
IV.1 Creación de programas (CINs)	92
<b>Capítulo V. Ejemplos de desarrollo de programas de cómputo para la enseñanza experimental mediante LabVIEW</b>	<b>101</b>
V.1 Características de los sensores	103
V.2 Programas	105
V.3 Propuesta de prácticas experimentales	131
V.4 Ejemplos de reportes de las actividades experimentales	139
Conclusiones	143
Glosario	144
Bibliografía	147
Páginas web consultadas	149

## PRESENTACIÓN

En el amplio campo de la computación, una rama particularmente útil en el cómputo educativo, la cual pretende introducir la computadora como una herramienta versátil en el apoyo a la enseñanza-aprendizaje.

En esta tesis se propone una opción para la enseñanza experimental, que consiste en la automatización de laboratorios, mediante equipo de cómputo. Como parte fundamental de este trabajo se realizó la comunicación entre diversos componentes. Esto se realiza para proveer una solución modular en la adquisición automática de datos, reuniendo elementos de distintas compañías.

En este trabajo el lector encontrará algunas palabras escritas en forma cursiva, de las cuales se proporciona su descripción en el apartado denominado glosario, para una mejor comprensión de su significado.

Se describe a continuación un esquema global del contenido de este trabajo.

Capítulo I.- Captura automática de datos. En este capítulo se muestra un análisis comparativo entre una variedad de tipos de hardware y software para la adquisición de datos, con el propósito de presentar diferentes opciones como guía en la elección de ambos.

Capítulo II.- Tarjeta ProControl. Este capítulo se enfoca a la descripción de la tarjeta ProControl PA-CP12, utilizada en este trabajo, se muestra el funcionamiento de sus elementos.

Capítulo III.- Software LabVIEW. En este capítulo se explican las características del software LabVIEW, para la lectura de la tarjeta y la creación de programas ejecutables.

Capítulo IV.- Conexión ProControl - LabVIEW. Este capítulo se enfoca al diseño del programa capaz de hacer la conexión entre el Hardware y Software usados.

Capítulo V.- Ejemplo de desarrollos de programas de cómputo para la enseñanza experimental, mediante LabVIEW. En este capítulo se proponen diversas actividades experimentales realizando la captura de datos mediante programas ejecutables.

Este trabajo se realizó sobre la plataforma Windows 95 por ser el sistema operativo comercializado en mayor proporción, en el momento de desarrollar el tema. Sin embargo, los programas pueden ser ejecutados en Windows 98.

## PLANTEAMIENTO DEL PROBLEMA

A lo largo de la historia de la civilización, la forma en que se colecta y guarda la información ha variado significativamente, hasta llegar al caso de procesos automáticos. En el caso específico del cómputo la revolución tecnológica no se ha quedado atrás ya que la velocidad con la que la computación se desarrolla es realmente impresionante, porque esta herramienta tiene una alta prioridad en todos los ámbitos de nuestra vida. Por lo que este impacto de desarrollo se está dejando sentir en el campo de la educación en variadas formas.

Siendo la computadora un instrumento valioso y poderoso en el terreno industrial, financiero y prácticamente necesario en todas las actividades, el ámbito educativo en nuestro país no está exento de una serie de cambios positivos con la introducción de estas nuevas tecnologías, para apoyar e incrementar la enseñanza desde la educación primaria hasta niveles profesionales.

La tarea de llevar las computadoras a las aulas es un trabajo arduo pero finalmente gratificante en muchos aspectos. Es importante seguir y adquirir la nueva tecnología para fortalecer la educación. Día a día se necesita gente mejor preparada, capaz de competir a nivel nacional e internacional, para lo que se requiere una enseñanza que cumpla con las exigencias de nuestro presente. Una de las herramientas necesaria e indispensable es, indiscutiblemente, la computadora.

Como podemos ver en la vida cotidiana, gran parte de nuestras actividades se manejan por medio de sistemas de cómputo, desde el supermercado hasta las transacciones financieras.

Razón por la cual los estudiantes necesitan tener bases y herramientas computacionales que les permitan enfrentar, comprender y luchar en un mundo cada vez más sistematizado.

No es suficiente hoy en día, poseer información específica del conocimiento. Es de vital importancia que los estudiantes cuenten con los recursos informáticos que les permitan desarrollar y fortalecer los conocimientos adquiridos en la disciplina particular en la que pretenden desarrollarse.

Otra de las razones por las que la computación es importante en las aulas de enseñanza, se debe al impresionante poder de atracción que tiene en los estudiantes, además de ser un apoyo invaluable para facilitar la comprensión, especialmente en las materias experimentales (Física, Química y Biología por citar algunas), en las cuales los estudiantes siempre presentan rechazo, desconcierto, miedo y en algunas ocasiones llegar hasta el punto de no seguir con sus estudios, por no acreditar estas materias.

La educación apoyada en recursos computacionales está obteniendo un apoyo para el mejoramiento de la enseñanza, así como el incremento de interés, tratando con ello de mejorar la enseñanza-aprendizaje, en disciplinas difíciles de entender y aceptar por los estudiantes.

Algunas de las ventajas que se pueden visualizar fácilmente, basándose en lo anterior, son *cooperar en la disminución del índice de reprobación en materias experimentales*, ya que con el fortalecimiento del aprendizaje y el interés de los estudiantes se les complicará menos comprender las teorías y conceptos que son difíciles de asimilar.

El resultado concreto de la enseñanza computarizada se presenta en un incremento de estudiantes que opten por el estudio de carreras del área de ciencias e ingeniería, en las que no se tiene una demanda abundante como la que se presenta en áreas económico-administrativas y/o sociales; por supuesto no queremos que se malinterprete o desprecie la riqueza de las mismas, es sólo, que nuestro país necesita contar con personas que deseen involucrarse de manera profesional en el campo científico.

### *AUTOMATIZACIÓN DE LABORATORIOS.*

En el área de materias experimentales es importante realizar actividades de laboratorio que puedan reforzar los conocimientos teóricos de la asignatura, para lo cual un sistema de cómputo que nos proporcione ayuda significativa en la experimentación es de gran valor.

El Laboratorio de Enseñanza y Aprendizaje Asistidos por Computadora (LEAAC) perteneciente a la Dirección General de Servicios de Cómputo Académico (DGSCA) de la UNAM, se ha dedicado a dar una solución al problema, particularmente la que se refiere a lograr una comunicación eficiente entre la computadora y diversos instrumentos de medición, como *sensores* de luz, temperatura, presión, oxígeno, entre algunos otros, para ser utilizados como apoyo en la impartición de los laboratorios de materias experimentales.

La idea de *automatizar* un laboratorio se basa principalmente en realizar la adquisición de datos representativos del fenómeno en análisis en forma automática, es decir, en algunos experimentos el tiempo de desarrollo y supervisión varía desde algunas horas hasta días completos, lo cual resulta difícil para los investigadores, profesores y alumnos.

Teniendo el equipo de cómputo para realizar la automatización, este esfuerzo disminuye notablemente ya que una computadora es capaz de recolectar datos en un tiempo mayor sin necesidad de que el experimentador esté todo el tiempo observando el fenómeno.

La precisión que una computadora nos puede proporcionar es mayor que la que pudiese dar una persona, porque la computadora no se cansa, ni tiene actividades que la puedan distraer del experimento, por lo que es capaz de realizar las mediciones exactamente en el tiempo necesario y de manera eficaz.

En la búsqueda de soluciones para llevar a cabo la automatización de laboratorios, se han encontrado en el mercado una variedad de ellas, algunas se presentan de manera integral, un mismo proveedor proporciona "la solución" completa, es decir, proporciona todos los elementos necesarios que se requieren para llevar a cabo la adquisición automática de datos.

Otra alternativa es la solución modular, la cual consiste en la posibilidad de adquirir de forma independiente cada uno de los elementos que conforman un sistema de adquisición automática de datos, o el desarrollar por otro lado algunas de estas partes adecuándolas a las necesidades y requerimientos necesarios para un caso particular.

Esta última opción es la que se desarrolla en este trabajo de tesis, en donde uno de los problemas a resolver es lograr la compatibilidad entre los distintos módulos. Una de las razones fundamentales por las que se decidió optar por la solución modular, es el costo relativamente elevado que algunos componentes tienen, particularmente las *tarjetas de adquisición de datos* que se acoplan con herramientas de despliegue gráfico, lo cual significa una opción poco conveniente cuando se requieren equipos en gran número de laboratorios.

De lo anterior surge la posibilidad de usar tarjetas más económicas que, para los fines de la enseñanza experimental, cumplan con las características básicas para la adquisición automática de datos.

Analizando los diversos sistemas operativos que normalmente se utilizan en el proceso de enseñanza-aprendizaje, y teniendo presente las tendencias actuales del mercado hacia el uso de *interfaces* gráficas, el problema esencial es resolver la comunicación eficiente entre las tarjetas de adquisición de datos susceptibles de ser operadas desde interfaces gráficas como es el caso de la tarjeta ProControl PA-CP12, que cuenta con los requerimientos necesarios de comunicación y el *software* LabVIEW de National Instruments para la realización de Instrumentos Virtuales capaces de adquirir datos y controlar variables desde la computadora.

Partiendo de un experimento real, en el que se necesitan instrumentos que sean capaces de medir las variables que intervienen en el fenómeno, así como las modificaciones que estas variables sufren.

En un laboratorio normal se pueden encontrar desde simples termómetros hasta otros instrumentos más complejos, pero con ello se corre el riesgo ineludible de tomar alguna medida errónea, por ejemplo, citaremos el caso de un termómetro, donde el obtener el dato depende del ángulo de observación con el que se toma la lectura, el que puede ser diferente si ese mismo ángulo se varía, y tendremos una lectura con un grado de error, el cual depende directamente del mismo ángulo con que dicha lectura fue

tomada. Con elementos sensores como los que se proponen a continuación se puede evitar o disminuir este problema. Si en lugar de un termómetro común se coloca un sensor que tenga la capacidad de medir variaciones de temperatura, eliminando el error ya explicado, podemos asegurar un conocimiento más exacto del comportamiento y las variables del fenómeno en estudio.

Un sistema de adquisición de datos basado en un equipo de cómputo consiste en que al tener un experimento real, necesitamos instrumentos que sean capaces de medir las variables que intervienen en el fenómeno, así como las mediciones que éstas sufren; el elemento capaz de transportar los datos a la computadora y el programa que permita analizar estos datos.

Llamamos transductores (sensores) a los componentes electrónicos capaces de simular un instrumento de medición, pero cabe mencionar que al ser un circuito electrónico el involucrado en la medición, sólo es capaz de proporcionar una señal de voltaje, así como las variaciones que registre, por lo cual se requiere algún otro elemento que permita introducir esta señal a la computadora.

El siguiente elemento es una interface<sup>1</sup>, la cual permite acondicionar la señal obtenida por el sensor e introducirla a la computadora, pero para introducir esta señal es necesario contar con un medio propio de la computadora que permita llevar esta señal, para esto se cuenta con varias opciones. La introducción de la señal puede ser a través de un puerto (serial o paralelo) incluidos en la misma por omisión, como medio de entrada/salida o por una tarjeta de adquisición de datos que es un elemento de *hardware* el cual se introduce en la computadora.

Las tarjetas de adquisición de datos son el camino que seleccionamos para introducir la señal desde el experimento. Estas tienen como entrada señales analógicas provistas por los sensores, las cuales deben ser transformadas a señales digitales, por medio de un convertidor.

Finalmente, al tener la señal del sensor acondicionada e introducida por la tarjeta en la computadora, ahora se pasa al manejo y procesamiento de esta señal, lo cual se realiza por medio de la otra parte importante de la computadora: el software. Es necesario contar con un software específico para la adquisición de datos, el cual sea capaz de utilizar esta señal y manejarla para los fines específicos de cada usuario.

En este caso particular, el software que se utilizó fue elegido en función directa de nuestros recursos y necesidades.

---

<sup>1</sup> Este elemento interface es un desarrollo elaborado en el Laboratorio de Enseñanza y Aprendizaje Asistidos por Computadora, para mayor información refiérase al documento "Construcción de interface para sensores Vernier y tarjeta ProControl PA-CP12" elaborado por Armando Vargas González, Daniel Martínez Plascencia y Roberto Lafont Perales.

## ANTECEDENTES

Actualmente el uso de la computadora es imprescindible para realizar infinidad de procesos. La computadora es un instrumento de apoyo en diferentes actividades, desde procesos industriales como también en el campo educativo. Dentro del campo educativo existen diversas aplicaciones, la computadora como apoyo en un salón de clases hasta su uso como herramienta en experimentos dentro de los laboratorios de ciencias experimentales.

La automatización de laboratorios ha tomado importancia dentro de los diferentes niveles de la educación. El uso de las computadoras en los desarrollos de experimentos ayuda a que el alumno aprecie más el fenómeno, es decir, dedica más tiempo a observar las reacciones y efectos dentro del experimento que a tomar los datos.

La adquisición automática de datos es un proceso por el que la computadora almacena valores, que son capturados por medio de una interface que conecta a la máquina con una variedad de sensores. Dentro de este proceso se requieren diversos elementos para obtener resultados del experimento, desde el dispositivo que se encarga de capturar los datos del fenómeno, hasta el programa que nos permita visualizar esos datos en la computadora.

El primer paso de la adquisición automática de datos es definir las variables que se van medir dentro del experimento a realizar. Estos valores pueden ser medidos con diferentes instrumentos de medición como son, por ejemplo, los termómetros, las balanzas, los medidores de presión, espectrómetros, por citar algunos. Sin embargo, existen métodos indirectos como son el uso de sensores, los cuales son conectados a computadoras, donde la captura y las operaciones estadísticas son realizadas por la máquina, dando así mas exactitud y menos errores en las pruebas. Cabe aclarar que tanto los instrumentos de medición como los sensores pueden estar mal calibrados y por lo mismo obtenerse datos erróneos.

Algunas de las ventajas en el uso de métodos indirectos es la frecuencia con la que se capturan los datos, una tarjeta de adquisición de datos puede tomar muestras cada 200ms y con una mayor precisión; en el uso de instrumentos el espectador tendría que poner mayor atención a la toma de datos que al experimento en sí, sin llegar a tomar una gran cantidad de datos en poco tiempo y no obtendría la precisión que se da con un sensor.

El uso de la computadora en el proceso de la adquisición de datos ofrece grandes ventajas, como lo es el despliegue en diferentes formatos de la información obtenida y el tratamiento de los datos, tareas que pueden ser realizadas en tiempo real o en análisis posteriores, almacenamiento de grandes cantidades de datos; también según el software utilizado es posible emular una gran cantidad de dispositivos de medición, inclusive varios instrumentos pueden ser activados al mismo tiempo.

En este trabajo hablaremos de los sensores, los cuales son aparatos de medición, que son capaces de adquirir cada una de las variaciones que presente el fenómeno, la forma en que estos aparatos adquieren los datos se realiza a través de señales analógicas, por ser este aparato un conjunto de circuitos electrónicos.

Existe una variedad de sensores, cada uno de ellos utilizados para medir variables específicas desde variables básicas como la temperatura o el oxígeno disuelto en un experimento.<sup>2</sup>

Al ser la computadora un aparato digital, es capaz de recibir señales digitales, las cuales consisten en niveles de voltaje discretos con representación lógica (bajo=0, alto=1). Por esto las señales que se introduzcan a la computadora deben de pasar por algunos procesos de conversión y de acondicionamiento para que la computadora los capture adecuadamente.

Un paso necesario en la adquisición de datos es el tratamiento de la señal para que no contenga ruido o señales de interferencia. Este proceso, que se conoce generalmente como acondicionamiento de la señal, incluye funciones tales como amplificación de la señal, filtrado, aislamiento eléctrico, y multiplexaje. A continuación se explican brevemente algunas de estas funciones.

Amplificación: El tipo más común de acondicionamiento es la amplificación. Señales de bajo nivel, por ejemplo, pueden ser amplificadas para incrementar la resolución y reducir el ruido. Para una mayor exactitud, la señal puede ser amplificada al máximo rango de voltaje del acondicionamiento de señal, equivalente al rango de entradas del convertidor analógico/digital. La amplificación en los sistemas de acondicionamiento de la señal aplican ganancia fuera de la computadora y cerca de la fuente de señal, este factor de amplificación puede aumentar la resolución medida y reduce efectivamente los efectos del ruido.

Sin embargo, el rendimiento en voltaje de muchos sensores es sobre milivolts o microvolts. Amplificar estas señales analógicas directamente sobre la tarjeta de adquisición de datos puede amplificar cualquier ruido que interfiera desde los cables principales de la señal o desde dentro del *chasis* de la computadora. Un método simple para reducir los efectos de ruido del sistema sobre la señal, está en amplificar la señal cerca de la fuente, lo más posible que se pueda, esto impulsa a que la señal analógica tenga un nivel arriba del nivel de ruido, antes de que el ruido generado en el *chasis* de la computadora o los cables, pueda afectar la señal.

Aislamiento: Un problema dentro de los sistemas de adquisición de datos es la conexión inadecuada de la tarjeta de adquisición de datos, ya que afecta en las mediciones de los datos y puede dañar la tarjeta. Otra aplicación común de los acondicionadores de señal es aislar las señales de los sensores de la computadora para diferentes propósitos. El sistema monitorea las señales que contienen un alto voltaje, que pudieran afectar a la computadora, esto es, aísla las señales que pudieran dañar las tarjetas.

---

<sup>2</sup> Refiérase a la bibliografía del libro "Métodos experimentales para ingenieros".

Los métodos comunes para el aislamiento, son el aislamiento capacitivo, óptico o magnético. Los aislamientos magnético y capacitivo modulan la señal para convertirla de voltaje a frecuencia. La frecuencia puede entonces transmitirse a través de un transformador o el condensador sin una conexión física directa, antes de volver a convertir al valor de voltaje.

El aislamiento también provee una función importante de seguridad, protege contra oleajes de alto-voltaje que pueden dañar el equipo; esto se hace quebrando la conexión galvánica, los acondicionadores de señal producen una barrera entre el tarjeta de adquisición de datos y el alto-voltaje.

Filtrado: La filtración remueve las señales no deseadas, de las requeridas para las mediciones. Un filtro de ruido es usado en las señales de CD (Corriente Directa, Direct Current, DC) para atenuar las señales de frecuencias altas que pudieran reducir la exactitud de las mediciones.

Las señales de CA (corriente Alterna, Altern Current, AC) tales como la vibración requieren de un tipo diferente de filtro conocido como antialiasamiento (Antialiasing). Como filtro de ruido este filtro es un pasabajas, sin embargo, este corta excesivamente el rango, por lo cual casi remueve completamente todos los componentes de las frecuencias más altas que el ancho de banda de la entrada de la tarjeta. Si las señales no son removidas, pueden aparecer erróneamente señales dentro del ancho de banda de la tarjeta.

Aunque muchos acondicionadores de señal incluyen filtros para quitar el ruido indeseable, se puede usar un software para quitar ruido adicional. El software es una técnica efectiva y simple que filtra digitalmente las lecturas adquiridas para cada dato que se necesita. El sistema de adquisición de datos toma y promedia muchas lecturas de voltaje.

Linealización: Otra función de los acondicionadores de señal es la linealización. Muchos sensores, tales como los termopares, tienen una respuesta no lineal que cambian cuando empiezan las mediciones.

En muchos casos, la relación que existe entre la variable de entrada al sensor y su señal, no es lineal. Así, las no linealidades reales causan errores en los datos medidos. Para reducir estos errores, la salida del sensor se puede linealizar como parte del proceso de acondicionamiento de señal. Para corregir las señales de salida del sensor se emplean varias técnicas de linealización, incluyendo la modificación de los circuitos del sensor o el procesamiento analógico de la señal del sensor.

Al tener nuestros datos capturados por un sensor y acondicionados requerimos el elemento capaz de introducir datos en la computadora. Existen diferentes posibilidades como son las tarjetas de adquisición de datos, entradas por puertos paralelo o serie.

Tarjetas de adquisición de datos: Este dispositivo se conecta en una *ranura (slot)* del *bus* interno de la computadora, por el cual van a pasar las señales para que la computadora las pueda procesar. Algunas de estas tarjetas utilizan diferentes *manejadores (drivers)*. Existe una amplia gama de tarjetas, entre ellas se puede seleccionar la que mejor se adecúe a la aplicación.

Las tarjetas pueden catalogarse en tres ramas generales:

- Conectar y ejecutar (Plug and Play): Estas tarjetas están listas para conectarse y empezar a trabajar. Son utilizadas comúnmente en plataformas que utilizan sistemas como Windows 95, en el cual el sistema detecta automáticamente el dispositivo que es conectado.

- Conectar (Plug In): Este tipo de tarjeta se inserta en el bus fácilmente y para su uso se necesitan manejadores especiales (programas proporcionados por el fabricante) de cada dispositivo para poder controlarlas.

- No detectable (no Plug and Play): Estas tarjetas son las que se conectan igualmente que las anteriores en un bus, pero la maquina no detecta que hay un dispositivo conectado por lo cual el usuario debe crear un programa para que se pueda controlar.

En una tarjeta son varios los factores que influyen para su buen funcionamiento, como son precisión, resolución, número de *canales*, número de entradas y salidas etc., estas características son las que nos van a dar una mejor muestra.

Puerto serial: Cuando se transmiten datos por medio del puerto serial los *bits* se van pasando uno por uno, un estándar en esta forma de transmitir los datos es RS-232.

En este caso se transmiten pulsos de voltaje a la velocidad seleccionada de *baud*, que se usan para representar los datos digitales El "1" lógico se representa por un nivel de voltaje transmitido en el rango de -3V a -15 V, mientras que el "0" lógico se representa mediante un nivel de voltaje transmitido de +3V a +15 V.

Puerto paralelo: La conexión a través del puerto paralelo es un método general, en el cual la computadora recibe varios bits al mismo tiempo. Un ejemplo de esto es la barra de distribución IEEE-488 la cual define una interface asíncrona de 8 bits en paralelo y 1 bit en serie. La estructura del bus IEEE-488<sup>3</sup> comprende dieciséis líneas de señal. Ocho de esas líneas se asignan para transportar datos, tres líneas son para la función de reconocimiento y cinco líneas son para funciones de administración del bus que efectúan un flujo ordenado de mensajes a través de la interface.

---

<sup>3</sup> En algunos casos se hace referencia al bus de interface de propósito general (*GP/IB*) y al bus de interface de Hewlett-Packard, sin embargo, son otros nombres del bus IEEE-488.

Como se ha visto existen diferentes medios de capturar datos, aunque cabe mencionar que el que trataremos en este caso serán las tarjetas de adquisición de datos, esto es por las ventajas que conlleva usarlas, como es el tiempo de muestreo y la precisión en los datos. Podemos resaltar primeramente que existe un mercado muy amplio en el cual se puede seleccionar la que mejor se adapte a nuestro desarrollo, se pueden conectar varios sensores a la vez en una misma tarjeta, es decir, capturar varios factores a la vez dentro de la misma aplicación.

Una tarjeta de adquisición de datos contiene una diversidad de dispositivos necesarios para comunicar una computadora con el mundo exterior. Generalmente, puede contener: convertidores analógicos/digitales y convertidores digitales/analógicos, entradas y salidas digitales/analógicas, *relojes* internos, *interruptores (jumpers)*<sup>4</sup> y otra serie de dispositivos necesarios para su correcto funcionamiento.

Explicaremos cada uno de estos componentes capaces de transformar la señal del sensor para que pueda introducirse en la computadora.

### Convertidor analógico-digital (CAD) (ADC Analogic Digital Convert)

Convierte las señales analógicas a señales digitales, para que puedan acceder a la computadora. Las principales características de funcionamiento de los convertidores analógico/digital son:

**Señal de entrada:** Es el máximo margen de tensión de entrada analógica permisible y puede ser unipolar 0 a 10 V o bipolar, es decir, de +5V a +-10V como ejemplos.

**Tiempo de conversión:** Es el tiempo requerido para completar las mediciones del convertidor analógico/digital. Este tiempo no incluye el tiempo de adquisición, el tiempo de inicialización del multiplexor u otros elementos para completar la conversión, depende del tipo de convertidor.

**Formato de salida:** Es la forma en que se expresará la salida. Se dispone de una diversidad de formatos, incluyendo el binario unipolar, binario offset, complemento a 1, complemento a 2 y varios códigos estándar.

**Precisión:** La precisión incluye errores provenientes de ambas partes analógicas y digitales del sistema. El error digital es debido a la cuantificación y el error de cuantificación resultante es usualmente ( $\frac{1}{2}$  Bit menos significativo). La principal fuente de error analógico es el comparador. La precisión necesaria y el número de bits deben ser compatibles.

**Bit menos significativo (LSB Least Significant Bit):** En el *sistema binario* este es el bit que carga el valor o medida más pequeño.

---

<sup>4</sup> En este caso la palabra "jumper" se traduce como "interruptor", para no crear conflictos con la palabra "conector" usada en los siguientes capítulos.

**Bit mas significativo (MSB More Significant Bit):** En el sistema binario es el bit que tiene el valor mas alto.

**Error de offset:** Es la diferencia entre el voltaje ideal de entrada y el actual voltaje de entrada.

**Resolución:** Líneas de código digital para representar una señal analógica en una señal discreta. Para los convertidores la resolución comúnmente esta especificada en bits, cuando el número de códigos digitales es equivalente a  $2^n$ . Como ejemplo, en un convertidor con resolución de 12 bits, la señal analógica es igual a 4096 códigos digitales.

**Especificaciones estáticas:** Son las especificaciones del convertidor analógico digital pertinentes a la entrada de la señal de corriente directa. Estas incluyen error de ganancia, error de offset y errores diferenciales.

### Convertidor digital-analógico (CDA) (DAC Digital Analogic Convert)

Convierte las señales digitales en señales analógicas, para que se puedan controlar aplicaciones mediante señales analógicas. Las principales características de funcionamiento del convertidor digital/analógico son:

**Resolución:** La resolución de un CDA es el recíproco de número de escalones discretos de la salida. Este depende del número de bits de entrada. El número total de escalones discretos es igual a  $2^n - 1$ , donde n es el número de bits. La resolución también puede expresarse como el número de bits que se convierten.

**Precisión:** La precisión es una comparación entre la salida real de un CDA y la salida esperada.

**Linealidad:** Un error lineal es una desviación de la salida ideal (una línea recta) del CDA. Un caso especial es el error de offset, que es la tensión de salida cuando los bits de entrada son todos cero.

**Monotonicidad:** Un CDA es monótonico si no produce escalones inversos cuando se le aplica secuencialmente su rango completo de bits de entrada.

**Tiempo de establecimiento:** Normalmente el tiempo de establecimiento se define como el tiempo que tarda un CDA en quedar dentro del valor final cuando se produce un cambio en el código de entrada.

**No monotonicidad:** Un funcionamiento no monótonico es una forma de no-linealidad. Este error se presenta cuando un cortocircuito hace que la línea de entrada del bit permanezca a nivel bajo (valor constante 0).

No-linealidad diferencial: La no-linealidad diferencial se observa cuando la amplitud del escalón es menor de lo que debería ser para ciertos códigos de entrada. Esta salida en concreto podría ser producida por el peso insuficiente de un bit, debido a un fallo de la resistencia de entrada. También podría haber escalones con amplitudes mayores que la normal, si un peso en particular fuera mayor de lo que debería ser.

Baja y alta ganancia: En el caso de baja ganancia, todas las amplitudes de los escalones son menores que la ideal. En el caso de alta ganancia, todas las amplitudes de los escalones son mayores que la ideal. Esta situación puede deberse a un fallo de la resistencia de realimentación en el circuito del amplificador operacional.

Error de offset: Este error se observa cuando una entrada binaria por ejemplo es un cero, este error puede ser por un fallo del amplificador operacional.

Puertos de entrada y salida digitales: Sirven para sensar y/o manejar en forma apropiada y segura, señales del tipo discreto provenientes de botones, leds (focos), etc.

Contadores y temporizadores: Sirven para medir frecuencias y contar eventos.

Interruptores: Nos sirven para configurar varios aspectos en la tarjeta de adquisición de datos, como por ejemplo los tipos de conversión, rangos de voltaje, selección de canales, etc.

Lo anterior nos da una idea general de los elementos necesarios para la captura de datos de un experimento, ahora lo que se necesita es el elemento capaz de almacenar estos datos y realizar análisis y operaciones estadísticas con ellos. Así pues la computadora es nuestro instrumento de adquisición de datos y nuestra herramienta de análisis. El análisis de los datos se realiza con el software a utilizar, donde se debe tener en cuenta en qué plataforma se trabajará, es decir, que tipo de hardware ocuparemos y sobre qué sistema operativo se desarrollará la aplicación.

## MS - DOS

Es un sistema operativo que básicamente utiliza un solo procesador (principalmente en los procesadores Intel). Este sistema operativo utiliza un conjunto de comandos llamados BIOS (Basic Input/Output System), el cual contiene los manejadores de los dispositivos estándar de entrada y salida, localizadas en la memoria ROM (Read Only Memory).

Los programas básicos que componen el sistema operativo DOS son:

-Command.com: Es un sistema de archivos, llamadas al sistema programas de utilerías y otras características de arranque.

-Autoexec.bat: Es un archivo que MS-DOS ejecuta cada vez que inicia, este archivo contiene los comandos que se deseen ejecutar al inicializar el sistema.

-Config.sys: Es un archivo de texto que contienen comandos que configuran los componentes de hardware de la máquina.

Para utilizar MS-DOS solo hay que encender la computadora. A diferencia de UNIX no hay un procedimiento de entrada y contraseñas, puesto que este sistema operativo está pensado para utilizarse en una computadora personal (Personal Computer, PC); sustentando la hipótesis de que la máquina sólo será utilizada por una persona, razón por la cual los archivos y directorios no tiene propietario y no existen bits de protección, permitiendo que el usuario regularmente realice lo que desee.

## WINDOWS 95

Es un sistema operativo de 32 bits que funciona en ambiente gráfico. En este sistema se puede instalar hardware con tecnología conectar y ejecutar, con lo que el uso de diferentes dispositivos conectados a la computadora se hace mas fácilmente.

La interface con el usuario es totalmente gráfica, donde el uso del *ratón* se hace indispensable. Al igual que MS-DOS los archivos y directorios no tienen propietario y no existen bits de protección, eso sí, algunos paquetes si manejan código de protección en documentos.

Existen otros sistemas operativos como son UNIX y System X.X para Macintosh, pero en el caso de este trabajo solo se citan.

En los tipos de computadoras existentes, las arquitecturas son diferentes, por lo cual el hardware difiere entre ellas. En nuestro caso la tarjeta de adquisición de datos se conecta a una ranura de un bus, este al ser parte del hardware, tiene diferencias entre las diferentes plataformas. Por ejemplo los buses de una PC son diferentes a los de una Macintosh, en el caso de una PC puede usar buses ISA (Industry Standard Architecture), que es un bus de 16 bits, usado en muchas máquinas y costo económico o EISA (Extended Industry Standard Architecture), es un bus de 32 bits, rápido y un costo elevado y en el caso de una Macintosh los buses se conocen como NuBus, PCI (Peripheral Computer Interconnect) o PDS (Procesor Direct Slot) usados en los diferentes modelos de Macintosh.

En este trabajo de tesis se seleccionó el uso de una computadora personal con procesadores Intel o similares, ya que estas computadoras están más comercializadas con respecto a las otras.

**CAPÍTULO I**  
**Captura Automática de Datos**

## **CAPÍTULO I CAPTURA AUTOMÁTICA DE DATOS**

La parte inicial de este trabajo de tesis se basa en la captura de datos de un experimento a través de un equipo de cómputo, el cual nos permite realizar la adquisición de forma automática, es decir, sin necesidad de que el experimentador deba vigilar todo el tiempo dicho experimento.

Para realizar una captura de datos automáticamente se requiere una computadora con el hardware y el software necesarios, para llevar a cabo dicha tarea.

En este capítulo se dará una explicación detallada de los principales elementos que se utilizan para este fin. En la parte de hardware se usan tarjetas de adquisición de datos y en lo relacionado al software se utiliza uno que cumpla con los requisitos para poder manejar, procesar y analizar los datos adquiridos por la tarjeta.

La distribución de la información en este capítulo se divide en:

- Tarjetas de adquisición de datos.
- Software de adquisición de datos.

Cabe mencionar que todas las opciones que a continuación se mencionan se encuentran actualmente disponibles en el mercado.

### **I.1 TARJETAS DE ADQUISICIÓN DE DATOS**

En la parte introductoria de este trabajo proporcionamos la información acerca de lo que se requiere para una correcta adquisición de datos. Seleccionamos el uso de tarjetas como medio de tomar las mediciones, esto es por las ventajas que ofrecen sobre otros sistemas, las cuales son :

- La instalación de las tarjetas en muchos casos es muy fácil ya que solo se conectan y se pueden usar. Existen las que usan programas proporcionados por el fabricante para su uso, y las que el usuario tiene que programar un manejador para utilizarlas.
- Casi todas las tarjetas existentes son compatibles con diferentes tipos de software, así el usuario no está obligado a comprar tarjetas de la misma compañía que el software a utilizar, ya que esto, a veces, es de un costo elevado. Para disminuir costos una solución es elegir entre varias compañías los elementos necesarios para nuestra actividad, constatando que puedan ser compatibles entre ellos.

- Existen en el mercado una variedad de tarjetas de adquisición de datos. De entre ellas el usuario puede seleccionar la que mejor cubra sus necesidades.

Como el elemento a utilizar son las tarjetas de adquisición de datos, previamente se realizó un análisis comparativo de diferentes compañías que se encargan de comercializar tarjetas. A continuación se muestran varias opciones.

## NATIONAL INSTRUMENTS

### *TARJETA PC-LPM16PnP*

Esta tarjeta cuenta con 16 canales analógicos de *entrada de una sola terminal* de 12 bits de resolución, la velocidad de muestreo es de 50KHz. Cuenta con 8 canales de entrada/salida digital.

Esta tarjeta está realizada para usar la tecnología de conectar y ejecutar, la cual es 100% compatible con Windows 95, ya que el sistema reconoce la tarjeta automáticamente.

## PROCONTROL

### *PROCONTROL PA-CP12*

Esta tarjeta cuenta con 16 canales analógicos de entrada de 12 bits, y 16 canales de entrada/salida digital. El tiempo de conversión es de 200 microsegundos (5KHz). Esta tarjeta cuenta con diferentes métodos de conversión analógica/digital, ya sea por medio de software o por medio del reloj interno (8254) de la tarjeta. Cuenta con dos convertidores de resolución de 12 bits.

El manejo de esta tarjeta se hace por medio de lenguajes de programación como C o Basic. El fabricante no distribuye manejadores, por lo que el usuario debe hacer un programa capaz de leer la tarjeta. El intervalo de direcciones que puede acceder esta tarjeta es amplio, por lo cual no interfiere con programas que se estén ejecutando sobre Windows.

La adquisición de esta tarjeta es de un costo económico, lo que la hace una buena elección, cuando el costo es un factor determinante.

## VERNIER

### *TARJETA DE INTERFACE PARA IBM Y COMPATIBLES*

Esta tarjeta se usa con el programa MPLI (MultiPurpose Lab Interface), en el cual la tarjeta viene incluida en la compra. El costo no es tan elevado con respecto a otras tarjetas ya que sólo cuenta con 3 canales de entrada.

Es una tarjeta de resolución de 12 bits y 3 entradas analógicas, en esto difiere con las demás que por lo menos tienen 8 entradas. Cuenta con un convertidor analógico/digital. El número máximo de muestreo es de 75,000 muestras por segundo.

Esta tarjeta forma parte de una solución integral, la compañía provee también el software, la interface y los sensores.

### **OMEGA ENGINEERING**

#### *TARJETA DE ADQUISICIÓN DE DATOS DAQ 1100*

Esta tarjeta cuenta con 16 canales analógicos de una sola terminal y 8 diferenciales. La resolución que maneja es de 12 bits y la velocidad de muestreo es 330KHz. Contiene una entrada/salida digital de 32 bits y tres contadores de 16 bits (82C54).

El software que soporta esta tarjeta es llamado DaqEZ, es una aplicación para Windows, el cual despliega los datos en tiempo real. Este software soporta varios lenguajes de programación tales como C/C++, Turbo Pascal, QuickBasic.

Las conexiones de entradas/salidas analógicas, entradas/salidas digitales de 8 bits y el control de líneas, son accedidos por medio de un conector D-37. El costo de la tarjeta es un poco elevado.

### **SDS PRODUCTS**

#### *TARJETA DAS1700ST-DA*

Esta tarjeta cuenta con una serie de canales analógicos y digitales. Tiene 16 entradas analógicas de una sola terminal y 8 diferenciales, 4 salidas digitales/analógicas, 4 entradas digitales y 4 salidas digitales, las entradas analógicas son expandibles a 256.

La resolución de la tarjeta es de 12 bits y la velocidad máxima de muestreo es de 160KHz. Esta tarjeta es de un costo elevado, por la capacidad de canales digitales que contiene y su uso es para actividades de mayor capacidad.

### **ANALOG DEVICES**

#### *TARJETA DAQ16*

Esta tarjeta tiene resolución de 16 bits, contiene 8 entradas diferenciales analógicas y 2 canales de salida analógica de 12 bits, la velocidad de muestreo es de 100KHz.

Soporta lenguajes como C, Pascal, Fortran, Basic. Además de que esta tarjeta contiene manejadores los cuales nos permiten seleccionar diferentes cantidades de ganancias. El precio de esta tarjeta es elevado, por las avanzadas características que posee.

**BSOFT SOFTWARE, INC.****TARJETA ANA100**

Esta tarjeta es de las más económicas, (la cuarta parte de las anteriores) del mercado ya que sólo contiene 8 canales de 8 bits de resolución. Es compatible con los diferentes tipos de bus PC, XT, AT ISA. La tarjeta contiene interruptores con los cuales se configuran los diferentes rangos de voltaje. Incluye manejadores de software para entrada/salida en Windows 3.xx o 95.

**ADAC CORPORATION****TARJETA 5508LF**

Esta tarjeta es de precio económico por sus características. Con respecto a otras, ésta cuenta con pocos canales y una resolución baja, a pesar de que puede usarse con Windows. Contiene 8 entradas analógicas de 12 bits de resolución, la velocidad de muestreo es de 100KHz.

Utiliza el software DIRECT VIEW que viene incluido con la tarjeta, el cual permite utilizarla sin complicaciones. Las ganancias son programables mediante este software.

**LR INFORMÁTICA INDUSTRIAL****TARJETA CIO-DAS1600/16**

Esta tarjeta contiene 16 entradas analógicas, además de 2 canales de salida analógica. Respecto a la parte digital tiene 4 canales de entrada digital y 2 canales de salida digital. La velocidad de muestreo es de 100KHz. Contiene dos convertidores, el convertidor A/D es de 16 bits de resolución y el convertidor D/A es de 12 bits de resolución.

Tiene manejadores que pueden usarse con Windows. El precio es elevado con respecto a las otras por el número de canales que contiene.

**I. 2. SOFTWARE DE ADQUISICIÓN DE DATOS**

El software es una parte muy importante en el desarrollo de adquisición de datos automático, así como en los sistemas de control para instrumentos. El software es quien proporciona el acceso a todas las herramientas de análisis disponibles y la forma de presentar los datos adquiridos de manera comprensible, útil y fácil de manejar; ya que no sólo es importante coleccionar datos, sino también es necesario darles un proceso analítico.

La calidad y flexibilidad del software empleado en las aplicaciones de los sistemas de instrumentación se reflejan en el desempeño total del sistema de adquisición de datos.

Algunas de las características más importantes que podemos tomar en cuenta al seleccionar un software de adquisición de datos para que se ajuste mejor al sistema deben ser: funcionalidad, desempeño, plataforma de trabajo, tipo de sistema operativo que utiliza, versión requerida, frecuencia y tiempo de muestreo, tipo de análisis (matemático, estadístico, gráfico, etc.), conectividad, compatibilidad con otros lenguajes de programación y el uso de otro software para análisis posterior, especificaciones de software con el que funciona para obtener un adecuado funcionamiento del sistema, costo, pertinencia, oportunidad en el mercado, entre otras muchas características relevantes que cada software en particular posee.

Al realizar un análisis de tipos de software para adquisición de datos disponibles en el mercado, decidimos efectuar una división de éste en dos categorías, basándonos en las diferentes características de cada uno. Estas categorías se denominan: software con compilador propio y software sin compilador

El software con compilador propio, es un lenguaje de programación que cuenta con un compilador propio, por lo que se pueden realizar programas para soluciones específicas que sean capaces de cubrir todas las necesidades que un problema de adquisición de datos necesite. Es importante mencionar que consideramos que este software resultó ser la mejor opción para el desarrollo de este trabajo de tesis, ya que cada programa se ajusta a los requerimientos de las actividades experimentales, evitando tener que ajustarse solamente a las opciones invariables que el software sin compilador proporciona.

La otra categoría, se presenta cuando el software sólo realiza las opciones que ya tiene definidas sin opción de modificarlas, es decir, no se cuenta con la facilidad de ajustarlo a las necesidades de cada actividad, al contrario, el usuario debe ajustarse únicamente a las opciones que se le proporcionan.

A continuación se mencionan las características generales que se encuentran en cada categoría de software.

#### **SOFTWARE CON COMPILADOR PROPIO:**

- Elaboración de programas específicos.
- Realizar llamadas a códigos externos desarrollados en otros lenguajes de programación textual como C o C++, entre algunos otros.
- Exportación de datos a otros programas de análisis estadístico, matemático, hoja de cálculo y procesador de textos.
- Integración a complejo sistemas de aplicaciones múltiples mediante la herramienta Active X, además de integración remota a través de hojas web.
- Adquisición de datos por medio de tarjetas o puertos.
- Control de instrumentos desde el programa.
- Funciones de análisis estadísticos de datos como: promedio, máximo y mínimo, varianza, desviación estándar, media, sumatoria, entre algunas más.
- Funciones de análisis matemático, como: cálculo de área bajo la curva, derivación, interpolación, extrapolación, ecuaciones.

- Análisis de procesamiento digital de señales: transformadas de Fourier, filtros y conversiones.
- Funciones para manejo de archivos como guardar, leer y escribir los datos adquiridos
- Despliegue gráfico de datos, con diversas opciones de graficación.
- Presentación de los datos mediante controles, indicadores, tanques, termómetros, etc.

#### ***SOFTWARE SIN COMPILADOR:***

- Adquisición de datos.
- Presentación gráfica de los resultados.
- Análisis estadístico de datos, tales como: promedio, mínimo, máximo, desviación estándar, media, sumatoria.
- Análisis matemático de datos, como: integración, derivación, interpolación, extrapolación, entre otras.
- Funciones para manejo de archivos como guardar, leer y escribir datos adquiridos.
- Exportación de datos a otros programas de análisis estadístico y matemático, así como a hojas de cálculo y procesadores de texto.

En la sección denominada Antecedentes hacemos un análisis de las diferentes plataformas, así como una explicación de las principales características para facilitar la elección del software. Básicamente se realiza una clasificación de los diferentes tipos de software disponibles en el mercado en:

#### **Plataformas de software para adquisición de datos**

- ◆ Microsoft DOS y Windows con procesadores Intel
- ◆ UNIX
- ◆ Macintosh

En el caso particular de este trabajo de tesis, utilizamos un sistema de adquisición de datos basado en computadoras personales y procesadores Intel.

Primero se presentan diferentes opciones de software que trabajan en computadoras personales y utilizan sistema operativo MS-DOS y Windows en cualquiera de las versiones existentes y después se muestran a modo de referencia, algunas características de tipos de software que trabajan con otros sistemas operativos.

Se aclara que, en cada uno estos, ya no se mencionan todas las características anteriormente definidas para ambas categorías – software con compilador y sin compilador, asumiendo que cada software lo contiene.

#### ***SOFTWARE CON COMPILADOR PROPIO***

##### **LabVIEW**

LabVIEW es un lenguaje de programación gráfica integral, diseñado para construir sistemas de aplicaciones con interfaces gráficas. Cuenta con bibliotecas de manejadores, los cuales simplifican los esfuerzos de programación de alto nivel y proporcionan control sobre dispositivos o tarjetas interface en la adquisición de datos.

LabVIEW se presenta en tres modalidades que se ajustan a las necesidades particulares de cada sistema

- TEST SUITE, el cual contiene las herramientas necesarias para desarrollar un sistema de adquisición de datos completamente automatizado, con análisis de procesos estadísticos, control y manejo de bases de datos (para almacenamiento de datos), varios manejadores para control de instrumentos entre algunas otras.
- FULL DEVELOPMENT SYSTEM (FDS). El FDS incluye el VISA, GPIB, RS-232 y bibliotecas de adquisición de datos, las bibliotecas de manejadores para instrumentación de LabVIEW, las bibliotecas de análisis avanzado. El FDS también incluye las funciones necesarias para acceso directo a bibliotecas de ligas dinámicas (*Dynamic Link Library, DLL*) y otros códigos externos
- PROFESIONAL DEVELOPMENT SYSTEM, es diseñado para desarrolladores de grandes aplicaciones, y/o quienes adhieren estándares de calidad tal como ISO9000. Además, incluye bibliotecas avanzadas para análisis de instrumentos virtuales, tiene la habilidad de construir ejecutables y utilerías para control de código.

En el caso particular de LabVIEW, solo se mencionan aquí las características principales, ya que este software será objeto de un mayor análisis en el Capítulo III.

### SOFTWARE HP VEE 4.0

HP VEE versión 4.0 es un lenguaje de programación visual que maneja el control de instrumentos, adquisición de datos, procesamiento de mediciones y reportes de pruebas. Configura instrumentos automáticamente, acelera la creación de interfaces operadoras, realiza modernas pruebas secuencialmente, con una distribución simplificada de aplicaciones; además, provee flexibilidad y funcionalidad sin sacrificar simplicidad de programación.

Cuenta con un manejador de instrumentos que automáticamente configura los instrumentos adicionados. Este manejador revisa el bus para dispositivos y es necesario construir algunas cadenas de direcciones. Carga los manejadores, y realiza el manejo de servicios de entrada/salida.

Contiene un programa explorador que ayuda a navegar por medio de programas complejos, durante la depuración y después de ser ejecutado. Usando ventanas de documentos se editan fácilmente las funciones HP VEE 4.0 y realiza búsquedas para objetos por nombre, texto, tipo, descripción, color, lugar y otros valores de propiedad.

HP VEE 4.0 provee herramientas que ayudan a optimizar programas, permitiendo inspeccionar la ejecución y el flujo de datos. Se pueden visualizar errores con llamadas a *stacks* que muestran la secuencia de funciones ejecutadas, colocando puntos de ruptura y corriendo el programa por pasos para detenerlo en cualquier punto. Además, proporciona una rápida ejecución usando la tecnología de compilador incremental. Los programas desarrollados pueden ser compilados y corridos. El compilador incluido substancialmente incrementa el funcionamiento mientras mantiene todas las ventajas de un intérprete. Durante el desarrollo de programas de prueba, se hace una compilación

inicial y entonces sólo recompila las funciones editadas. Después de que un programa es desarrollado, la compilación ocurre sólo una vez sobre la primera ejecución.

### *HP VEE EN LA EDUCACIÓN*

Automatiza fácilmente laboratorios de Ingeniería. Es una herramienta de enseñanza basada en un lenguaje de programación visual, fácil de aprender y de usar, que habilita a los estudiantes a enfocarse en el aprendizaje de la materia que cursa, no al lenguaje de programación. Los programas HP VEE pueden ser usados para simulación, modelado y análisis. Para un control de instrumentos modernizado, se proporcionan manejadores para el manejo de cualquier instrumento y de cualquier proveedor, si este es GPIB, RS-232, VXIbus.

### **LABWINDOWS/CVI**

Es una colección de herramientas para ejecución y análisis de resultados de sistemas automáticos de adquisición de datos. Es un ambiente de programación visual integrado con un compilador ANSI-C de 32 bits, ligador, depurador, editor fuente un editor para la interface con el usuario diseñado para construir pruebas.

### **SOFTWARE DRIVER NI-DAQ PARA ADQUISICIÓN DE DATOS**

NI-DAQ es un software que proporciona el control de los dispositivos de adquisición de datos. En la programación los componentes específicos de la computadora, tal como los controladores de *acceso directo de memoria (DMA)*, controladores de interrupciones como interconexión de componentes periféricos (PCI) PC/XT/ISA/EISA y arquitectura NuBus Macintosh, intercomunicación de una variedad de sistemas operativos y manejo de los datos adquiridos. El software NI-DAQ utiliza todas las ventajas del hardware y las características del sistema operativo. NI-DAQ incrementa las capacidades del hardware de adquisición de datos y el sistema operativo con el que se pueden desarrollar poderosas soluciones para instrumentos virtuales.

Los programas basados sobre NI-DAQ son totalmente escalables, como el hardware de adquisición de datos, los buses y sistemas operativos. Contiene un conjunto de manejadores específicos separados en sistemas operativos de 16 y 32 bits. NI-DAQ transfiere datos dentro del sistema de memoria usando un DMA para maximizar el desempeño en la adquisición de datos.

Las funciones de entrada/salida de adquisición de datos tienen una variedad de funciones estándar analógicas, digitales y regulación de tiempo, ambos con multiplexaje y *disparo externo*.

### **GEOTEST**

#### *ATEasy, AMBIENTE DE DESARROLLO ATE (AUTOMATIC TEST EQUIPMENT)*

ATEasy es un lenguaje de programación en un ambiente de desarrollo para aplicaciones.

Basado en Windows, el sistema está orientado a una estructura hecha por ATEasy que es totalmente adaptable al introducir cualquier requerimiento de hardware para sistemas automáticos de pruebas y aplicaciones de medición. Cuenta con soporte para cualquier instrumento, ATEasy combina todas las herramientas necesarias al desarrollar una aplicación.

- Ambiente de desarrollo de aplicación rápido para ATE(equipo automático de prueba, Automatic Test Equipment). Incluye programa editor, editor driver, pruebas ejecutivas y más.
- Programas de prueba con estructura similar a la estructura del documento de requerimientos.
- El Editor Driver soporta instrumentos VXI, GPIB, RS-232 y basados en PC.
- Instrumentos manejadores incluidos para programas y unidades bajo prueba (UUT).
- Construcción en editor diálogo para interfaces de usuario o instrumentos virtuales.

ATEasy contiene características de construcción proporcionadas para desarrollar aplicaciones. Permite a los usuarios crear instrumentos manejadores, ventanas de interface de usuarios, programas y resultados de prueba conectados a la computadora. Este lenguaje de programación puede ser usado por programadores y no programadores, siendo semejante a los lenguajes tal como C o Pascal. Las declaraciones de programas pueden ser escritas por selección de menú o escribirse a texto directamente. Reduce ciclos de desarrollo tal como: edición, compilación, ligado, prueba y depuración. Posee extensas características de depuración para programas, tal como: detener por falla, prueba por prueba, descripciones, pantalla de comunicación par ver las variables en tiempo real. También posee opciones de depuración, tal como: punto de quiebra, ejecución paso a paso, traza. Tiene una arquitectura de sistema abierto que provee acceso a DLL, DDE y Windows.

## *ESTRUCTURA*

La estructura del software ATEasy tiene cinco módulos mayores:

- El editor de manejador – es usado para desarrollar manejadores de instrumentos, los cuales incluyen mensajes basados en instrumentos tales como: VXI, GPIB, RS-232.
- El editor de sistema – es usado al configurar la existencia de los manejadores dentro del ambiente ATEasy y adicionar sintaxis a nivel sistema acomodando las aplicaciones específicas.
- El editor de programa - usado al editar y desarrollar programas de prueba.
- El editor de diálogo - usado para crear instrumentos gráficamente con paneles frontales (paneles virtuales) y paneles interactivos para el usuario.
- Pruebas ejecutivas - usado al ejecutar el programa de prueba.

Todos los módulos ATEasy incluyen un rango completo de herramientas para depuración e integración sin el cual el ambiente de desarrollo de programas no sería programas de prueba ya existentes en las unidades bajo prueba.

## WINDMILL LABIML, SOFTWARE EN EL LABORATORIO

Usando el manejador universal Windmill LabIML, se pueden integrar diferentes tipos de instrumentos de laboratorio dentro del sistema.

Realiza lectura y envío de datos a instrumentos de laboratorios remotos. Trabaja con instrumentos que tienen puerto RS-232. Manejan instrumentos que tienen salida de datos continuamente, enviando automáticamente mensajes de comandos. Se pueden conectar varios instrumentos de diversas compañías.

### SOFTWARE SIN COMPILADOR

#### VERNIER SOFTWARE

Es un ambiente de desarrollo integral para la adquisición de datos utilizando instrumentos sensores que permiten adquirir las muestras obtenidas desde experimentos reales, las cuales deben de pasar a través de una interface que acopla la señal obtenida para introducirla a la computadora mediante una tarjeta insertable de adquisición de datos o utilizando el puerto serial.

El software que se utiliza para el análisis y control de la adquisición se encuentra disponible en dos plataformas: Macintosh y compatibles con IBM. Cada una de estas opciones posee diversos tipos de interfaces, programas específicos para cada plataforma, uso de sensores y manuales de experimentos basados en cada uno de los sensores disponibles.

El software se encuentra para ambiente MS-DOS Y Windows. Consta de varios programas individuales para la adquisición de datos desde sensores específicos. El programa Interface Multipropósito para Laboratorio (MPLI, MultiPurpose Lab Interface), puede capturar muestras de la mayoría de los sensores, con opciones de despliegue gráfico, tabular, graficación en tiempo real, cálculo de integración y regresión lineal por intervalos, definición del periodo de muestreo y número de muestras a adquirir en periodo, manejo de escalas, incluyendo la opción de osciloscopio de tres trazas; entre otras muchas opciones.

El programa *monitor de ritmo cardiaco (HRMX -Heart Rate Monitor)*, se usa para la adquisición de datos desde el sensor de ritmo cardiaco, este programa y sensor proporcionan básicamente el número de latidos del corazón por minuto, contando con algunas de las opciones del anterior programa.

El programa *cronometro de precisión (PT, Precision Timer)*, se utiliza para la adquisición de datos en sensores de movimiento, es decir, para el cálculo de velocidades y aceleraciones de cuerpos en movimiento, contiene las mismas opciones de despliegue y análisis que el programa MPLI.

El programa *analizador gráfico (GA, Graphical Analysis)*, es una herramienta adicional a los anteriores programas en los que se pueden realizar análisis de los datos muestreados con algunas opciones más específicas y poderosas. Este programa acepta todos los archivos generados en los otros programas con la única condición de que el archivo tenga la extensión *dat*, para especificar que es un archivo de datos.

El programa *espectrofotómetro (Spectrophotometer)*, utiliza el sensor llamado espectrofotómetro, para el análisis de absorbancia y transmitancia de ondas de luz, con opciones de análisis y graficación semejantes a los demás programas.

El software Vernier tiene una velocidad de muestreo de hasta 10,000 muestras por segundo, con opción de muestrear dos, tres o cuatro señales simultáneamente, lo cual depende de la interface que se utilice, con una precisión de 12 bits,

Vernier, además provee material curricular, el cual es adicional a las interfaces de pruebas, demuestra como el equipo puede ser usado en el salón de clases en Biología, Química y Física.

El software Vernier actualmente es utilizado en el Laboratorio de Enseñanza y Aprendizaje Asistidos por Computadora (LEAAC), así como en algunos planteles de la Escuela Nacional Preparatoria y del Colegio de Ciencias y Humanidades, de la UNAM. Utilizado en la captura automática de datos para prácticas experimentales. La compañía Vernier proporciona soluciones globales, es decir, proporciona lo necesario para la adquisición de datos como son: el software, la interface, la tarjeta de adquisición de datos y los sensores.

## VIRTUAL BENCH

Virtual Bench puede cargar y guardar los datos de formas de onda a disco en extensiones de programas populares y procesadores de palabras. Es importante mencionar que este software al igual que LabVIEW pertenece a la misma compañía, pero a diferencia del primero, éste no es un lenguaje de programación.

Se proporciona un fácil uso de instrumentos con el panel de control y el hardware de adquisición de datos de la compañía productora; usando soluciones portátiles PCMCIA, tarjetas de adquisición de datos entre 12 o 16 bits de resolución, soluciones de bajo costo usando puertos paralelos, o algunas otras soluciones de canal amplio usando SCXI. Virtual Bench incluye las especificaciones basadas en el hardware de adquisición de datos para máximo desempeño y flexibilidad.

Este software contiene varias opciones de programas que son:

- *VIRTUAL BENCH-SCOPE. ADQUISICIÓN DE SEÑALES DE ALTA PRECISIÓN.* Ofrece un rango de muestreo superior a 1.2MHz. Anchos de banda superior a 1.6MHz, cuatro canales con un rango de sensibilidad desde 1mV/div a 10V/div.

- **OSCILOSCOPIO VIRTUAL STAND-ALONE PARA WINDOWS 95/3.1.** El osciloscopio tiene un pequeño ancho de banda, alta resolución, digitalización para aplicaciones y mediciones a baja frecuencia.
- **ANALIZADOR DE SEÑALES VIRTUALES.** Virtual Bench-DSA es un analizador de señales dinámico que usa técnicas de muestreo digital y transformada rápida de Fourier en el dominio del tiempo, espectro y redes de análisis de señales.
- **GENERADOR DE ONDA VIRTUAL ARBITRARIA.** Virtual Bench-ARB es un editor de formas de onda multicanal e instrumento generador.
- **GENERADOR DE FUNCIONES Y MULTIMETRO VIRTUAL.** Virtual Bench-FG es un generador de funciones de baja frecuencia para la generación y sintetización de señales estimulantes.
- **SEPARADOR DE DATOS VIRTUAL.** Virtual Bench Logger es un separador de datos multicanal para grabar y desplegar señales de baja frecuencia. Adquiere más de 384 canales de datos y despliega más de 16 canales.

### WINDMIL SOFTWARE LTD

Esta compañía posee algunos programas que se explican a continuación.

#### **STREAMER 3.0: SOFTWARE DE CAPTURA PARA FORMAS DE ONDA**

El software Streamer posee una alta velocidad de monitoreo y grabación de formas de onda, graficar datos en tiempo real, puede guardar bibliotecas de archivos y examinar las gráficas de los datos después de la colección. La última versión del Streamer provee un sistema distribuido portátil de captura de formas de onda. Es capaz de manejar un amplio rango de sensores e instrumentos para un sistema de adquisición con un alto desempeño y flexibilidad. Corre bajo Windows NT, 95 o 3.1. Contiene la opción de iniciar la toma de datos cuando un disparo externo se recibe o cuando una entrada seleccionada atraviesa un umbral.

Tiene una máxima velocidad de 100,000 muestras por segundo utilizando una tarjeta para bus ISA ( un canal a 100 KHz, 2 a 50 KHz, etc.), puede muestrear 40,000 muestras por segundo con instrumentos GPIB, 35,000 muestras por segundo con instrumentos *Ethernet*, 700 muestras por segundo con instrumentos RS-232, acepta más de 200 canales (de diferentes sensores), además de enviar flujo de datos analógicos directamente a disco.

Realiza una linealización para ocho tipos de termopares y sensores. Se pueden grabar archivos de datos de cualquier tamaño limitado sólo por el del disco duro, permite manejar datos por medio de DDE a otros programas de análisis y a la vez enviar comandos al control Streamer. No requiere programación.

### **WINDMILL 4.3, SOFTWARE DE MEDICIÓN Y CONTROL**

El software Windmill proporciona versatilidad y facilidad a los sistemas de adquisición de datos para ser expandibles cuando los requerimientos aumentan. Este software contiene cientos de aplicaciones para sistemas de Investigación y desarrollo, pruebas y mediciones y aseguramiento de calidad. Corre bajo Windows NT, 95 o 3.1.

Cada programa Windmill se concentra en un trabajo sencillo y el usuario simplemente ejecuta el número de programas que necesite, además, se pueden correr distintas reproducciones de cada programa, lo cual significa que un sistema Windmill no tiene un límite fijo de tamaño y se pueden incluir en los programas anotaciones, trazas, simulación de procesos, alarmas de control PID (integrador proporcional derivativo) y control de secuencia, además posee alarmas indicadoras en todas las entradas.

Para proyectos complicados, se pueden construir módulos especializados, usando un lenguaje de programación igual a Visual Basic o LabVIEW. Windmill permite hacerse cargo de la configuración, control de hardware y flujo a disco. Soporta redes DDE así los datos pueden ser inmediatamente analizados por otros programas, en otras computadoras, utilizando red. No requiere programación, todos los programas son leídos al ejecutarse.

Windmill maneja varios equipos de interfaces como Ethernet, GPIB, bus ISA, LanWork, RS232, RS485, ModBus, y se puede transferir datos a través del teléfono. Posee entradas y salidas analógicas y digitales, incluyendo: contadores, temporizadores. Tiene una velocidad de muestreo máxima de 200 muestras por segundo.

### ***WINDMILL ELS: ENTERPRISE LOGGING SOFTWARE***

Es un software para interconectar empresas por medio de redes, con una distribución de datos flexible y confiable, puede ser usado en cualquier equipo por su fácil uso e implantación. Como el nombre lo sugiere, el software es diseñado para un amplio rango de aplicaciones en la empresa, incluyendo: monitoreo ambiental, disminución en el consumo de energía, pruebas de equipo para monitoreo, entre algunas otras.

Windmill ELS corre bajo Windows NT y 95 sobre una estructura Ethernet. Los datos pueden ser desplegados en tiempo real cuando se colectan y regularmente se duplica la información en un archivo central, siendo un sistema distribuido, ya que no es necesaria una base de datos central para procesar la información.

Cuenta con ocho tipos de opciones de activación: mediciones analógicas, barras en movimiento, *iconos* de cambio (cerrar y abrir válvulas), entrada de valores con alarmas codificados en colores, etiquetas de tiempo y datos, botones de control digital, botones de control de programa

## WINDMILL SOFTWARE LTD. IML TOOLKIT

### REALIZACIÓN DE APLICACIONES PROPIAS EN WINDOWS

Las pruebas y mediciones de ingeniería ofrecen la posibilidad de escribir las aplicaciones para tareas específicas. El Toolkit IML da velocidad al proceso, proporcionando:

- *DLLs* para lenguajes de programación en Windows.
- Ejemplos en LabVIEW y Visual Basic que se pueden incorporar a los programas.
- Cuatro aplicaciones Windmill incluyendo el servidor y panel de intercambio dinámico de datos.

Los *DLLs* incrementan las capacidades del lenguaje de programación, por ligas a rutinas que accesan al hardware de pruebas y mediciones. Las rutinas dan la configuración al hardware, toman una lectura desde un canal de entrada, escriben un valor en un canal de salida y coleccionan los datos a alta velocidad desde el canal de entrada habilitado.

## MICROLINK

Es una compañía en la cual realizan una gama de productos para el control y adquisición de datos, los desarrollos se enfilan a la captura de datos

### FAMOS, SOFTWARE DE ANÁLISIS DE DATOS

Este software de análisis de datos, corre bajo Windows, contiene más de 100 características y funciones, incluyendo funciones de filtros, de tiempo, estas funciones pueden actuar en distintas formas de onda inmediatamente, pueden ser desplegadas más de 150 formas de onda en una ventana, puede premuestrear formas de onda a diferentes velocidades.

Contiene un editor de secuencia con 22 opciones para despliegue de formas de onda, tal como leyendas y flechas movibles y duplicación de ventanas para distintas vistas de una forma de onda, tiene deslizamiento y ejecución de ejes y sistemas coordinados entre ventanas y guarda cualquier número de forma de onda y notas en un archivo.

Famos analiza formas de onda que han sido colectadas a altas velocidades (Streamer o Wavedisk). Los archivos de datos pueden estar en formato binario, ASCII o Famos.

Las funciones de Famos son agrupadas en conjuntos de botones. Todos los comandos, constantes y constantes predefinidas pueden ser asignadas a botones de funciones. Algunas funciones de Famos son correlación, ajustes exponenciales, cálculo de Histograma; interpolación, regresión y normalización lineal, integración Sliding, interpolación Spline y función Smith-Trigger; reducción y expansión de conjuntos de datos. También posee alternativas para filtros como filtros digitales por simulación de

filtros de Bessel, Butterworth y Chebyzhev, transformada rápida de Fourier y transformada inversa, análisis de octava/tercera/octava, espectro en potencias y análisis waterfall.

## DATAQ INSTRUMENTS

### *WINDAQ/200, SOFTWARE DE ADQUISICIÓN DE DATOS MULTITAREAS*

Windaq ofrece despliegue en tiempo real y flujo a disco, utilizando todas las características multitareas de Windows, es decir, mientras se guardan datos a disco en segundo plano, se puede correr otro programa al mismo tiempo. El despliegue en tiempo real puede operar en dos modos, uno de ellos funciona mostrando los datos en pantalla y el otro en modo de barrido por disparo, siendo escalables dentro de cualquier unidad de medición. Se pueden realizar comentarios dentro de los resultados permitiendo anotar información descriptiva en la sesión de adquisición, todas las veces que se guarde a disco.

Puede guardar más de 240 canales a disco y despliega más de 32 canales al mismo tiempo, realiza retardo en el eje par en tiempo real; activa a otras 80,000 muestras por segundo mientras graba a disco las formas de onda. Tiene control de velocidad de trazo independiente del rango de muestreo, correlación de datos en forma de onda con tiempo y la adquisición de datos con un segundo de resolución. Crea archivos de datos limitados en extensión solo por el tamaño del disco, permitiendo escalas en tiempo real y cada una es desplegada con su propia unidad de medida.

Windaq tiene un modo de grabación con disparo en un nivel seleccionable y tiempos de pre y post disparo. Adquiere datos tal como se necesiten antes y/o después del punto de disparo.

### *WINDAQ/LT LIBRE*

Esta es una versión para un solo canal simple del WINDAQ/200, tiene las características de software de análisis y playback para el ambiente Windows

### *WINDAQ PLAYBACK Y SOFTWARE DE ANÁLISIS*

Este software despliega formas de onda con varias opciones. Tiene 29 canales de despliegue, puede realizar compresión de formas de onda: permitiendo una vista comprimida de despliegue de formas de onda incluyendo factores de compresión y cualquier factor de despliegue requerido para desplegar completo el archivo de forma de onda en la pantalla. El despliegue de las formas de onda puede ser y contra  $t$ , o frecuencia contra amplitud.

Para realizar la medición de las formas de onda Windaq ofrece varias alternativas:

Mediciones y contra  $t$ : Mediciones de amplitud por canal en unidades calibradas; tiempo transmitido, tiempo de fecha en cursor, mediciones de tiempo en un mismo o diferente canal: diferencia de valor-Y, Dos-puntos de pendiente ( $d/dt$ ), número de

muestras, Hz, ciclos por minuto. Mediciones orientados a cursor (frecuencia contra amplitud): frecuencia contra db; frecuencia contra magnitud (unidades ingenieriles).

Cálculo de transformada de Fourier: 32 a 16,384 puntos (FFT, transformada rápida de Fourier) de 2 a 8,191 puntos (DFT, transformada directa de Fourier), transformada inversa de Fourier: de 2 a 16,384 puntos, con dominio de Tiempo de las formas de onda que son insertadas dentro de una ventana de despliegue como canales y ventanas de transformada rápida de Fourier seleccionables. Edición gráfica de espectro de poder para paso-altas, paso-bajas, paso-banda y supresor de banda.

## **SABLE SYSTEMS INTERNATIONAL**

### *EL DATACAN V SISTEMA DE ADQUISICIÓN DE DATOS Y ANÁLISIS*

El paquete de análisis y adquisición de datos DATACAN V de Sable Systems tiene características de una interface de usuario intuitivo, macros poderosas, multitud de transformaciones útiles para adquisición y análisis de datos, proporcionando total control digital de los experimentos, biblioteca de adquisición de datos.

DATACAN V ofrece alta resolución de adquisición de datos (16 bits) con despliegue de unidades reales en tiempo real y capacidades de control total sobre cualquier computadora 286 o superior que corra bajo el sistema operativo DOS pudiéndose utilizar en PC o con Laptop para llevarse incluso al campo. Tiene la capacidad de muestrear cuatro canales (expandible a 8) con conectores estándar en todas las interfaces, compatible con conectores banana o BNC (con adaptador), compatible con cualquier instrumento con salida analógica. Tiene una resolución: 16 bits (una parte en 65,536), una resolución total de 16 tiempos de interface en 12 bits. Con una calibración versátil de instrumentos lo que facilita la realización de los mismos a dos puntos, multi-punto lineal, multi-punto no lineal, los cuales son rápidos y fáciles. El intervalo de muestreo: es de 2000 a 0.001 Hz (dependiendo del límite más alto de la máquina).

Para el programa de análisis de datos también proporciona opciones específicas, como el uso de menús y manipulación de gráficas de datos interactivas. Transformaciones versátiles de cualquier canal o combinación de canales como lineal, exponencial, logarítmico, polinomial, diferenciación (pendientes), integración (áreas), corrección de pila, corrección Base Line y transformada Z. Las transformaciones pueden ser automáticas con macros generados en programa de cualquier longitud. Las macros son totalmente compatibles con el programa DataFile de la misma compañía.

Las operaciones analíticas son totalmente graficables, seleccionando cualquier parte de un registro y se pueden obtener pendientes lineales e intercepción de cualquier canal y ajuste polinomial contra segundos, minutos, horas u otros canales. Ejecución de transformada rápida de Fourier, para separar mezclas complejas de frecuencias. La interpolación de datos se utiliza al recuperar desde puntos de ruido y malfuncionamiento experimental, se puede borrar, sumar o substraer desde cualquier sección deseada en cualquier canal y guardar las secciones seleccionadas como un archivo compatible en ASCII, Lotus 1-2-3 para exportar a paquetes gráficos o estadísticos.

### *EL PROGRAMA DATA FILE*

Este software es un camino de organización, procesamiento y visualización de los datos. Cuando se tienen series largas de experimentos o ejercicios en el laboratorio de enseñanza, y hay una gran producción de archivos es necesario contar con rutinas de manipulación de datos y transformaciones para aplicar a los archivos de datos. Sable Systems Macro Language (SSML), es un lenguaje creado para este propósito.

El programa DataFile resuelve los problemas que se presenten para realizar copias a disco duro y macros, también puede cargarlos, uno por uno, cualquiera de los archivos ya existentes. Determina automáticamente cada formato (incluyendo ASCII). Entonces proporciona generación de gráficas de alta resolución totalmente escalables, además, las gráficas son impresas junto con anotaciones, información en intervalos de muestreo y el promedio, valor máximo, mínimo, rango y desviación estándar de cada canal. Es posible utilizar macros desde el programa de análisis de datos o editar en el DataFile los datos

### *SISTEMA DE LABORATORIO DE ENSEÑANZA*

Es un software con alta velocidad para un sistema de adquisición y análisis de datos específicamente diseñados para enseñanza y aprendizaje.

El sistema de laboratorio de enseñanza (TLS, Teaching Laboratory System) es un paquete basado en PC de bajo costo, proporcionando funciones de análisis de datos.

Tiene un modo de almacenamiento de osciloscopio con frecuencia de muestreo de 100 Hz a 13 KHz, control de sensibilidad de despliegue total, cuadrulado X y Y, disparo versátil, incremento de (1 a 1000), ajuste del nivel de corriente directa (manual o automático). Requiere una PC con despliegue CGA mínimo y al menos un slot estándar ISA de 16 bits y 512K de RAM.

## **PICOTECH**

### *PICOSCOPE*

Los osciloscopios basados en PC ofrecen todas las opciones de los osciloscopios convencionales. El software PicoScope permite usar la computadora como un osciloscopio digital de almacenamiento, analizador de espectro, multímetro y en este caso un generador de señales. Existe el software PicoScope para Windows y para DOS, manejadores para DOS y Windows y una macro que permite coleccionar datos directamente dentro de hojas de cálculo.

PicoScope proporciona características generales como múltiples vistas, vistas simultáneas de amplitud, espectro y medidas, permitiendo capturar mucha información acerca de la señal. Se pueden observar señales de entrada, con cuadrícula para observar voltajes y tiempos, reglas para aproximación de mediciones, puede salvar datos a disco por medio de un disparo y captura eventos intermitentes sobre períodos largos

con estampa de tiempo y fecha, realiza comparaciones entre trazas de referencia con señales reales.

El analizador de espectro realiza transformada rápida de Fourier permitiendo capturar uno de los eventos, calcula el promedio de las señales para reducción de ruido, posee reglas de medición para amplitud y frecuencia, tiene siete tipos de ventanas, realiza un despliegue lineal o logarítmico ambos de frecuencia y amplitud. El medidor tiene un voltímetro real en *RMS*, un medidor de frecuencia, mide en *dB* (*decibeles*), realiza autoalineación y proporciona un despliegue simultáneo de múltiples parámetros.

## SOFTWARE PICOLOG

Es un osciloscopio de canal sencillo. Simplemente se inserta la unidad de hardware a utilizar en el puerto paralelo de la PC y se activa este software, es útil para señales de baja frecuencia, obteniendo todas las funciones de un osciloscopio convencional. Este software es capaz de realizar otras funciones las cuales dependen de la unidad de hardware que utilice, ya que estas son quienes finalmente capturan la señal muestreada, además, que este software utiliza hardware especial que provee la misma compañía, no ahondaremos en esto, solo se hace la aclaración.

PicoLog posee las mismas características de análisis que el software PicoScope.

## PICOSCOPE PARA SOFTWARE BAJO PLATAFORMA DOS

PicoScope proporciona a la computadora cuatro instrumentos virtuales: osciloscopio de almacenamiento digital, analizador de espectro, voltímetro y medidor de frecuencia. Múltiples señales pueden ser guardadas en disco, al presionar una tecla se puede mostrar una pantalla proporcionando el voltaje y la frecuencia de una señal desplegando en el osciloscopio la traza, con una segunda tecla presionada se muestra el espectro. Las trazas de espectro y amplitud pueden ser salvadas en disco e impresas.

El Scope tiene todas las funciones de un osciloscopio, el tiempo base puede ponerse entre 1ms/div y 5s/div (100 microsegundos/div a 5s/div), con traza dual y en modo XY. Las opciones de disparo son auto, simple o con repetición, también posee las funciones de pre y post disparo. Ambos ejes tienen funciones múltiples y de regulación del nivel de corriente directa. Las funciones de ajustes movibles para regulación de tiempo en eventos, y la habilidad de realizar anotaciones en las gráficas y en los ejes antes de imprimir, son algunas ventajas adicionales que se proporcionan por encima de los osciloscopios convencionales.

El analizador de espectros tiene un despliegue en tiempo real de frecuencia contra amplitud, ésta amplitud será desplegada en forma lineal o logarítmica. Una regla móvil tiene una lectura de salida ambas de frecuencia y amplitud, ambas frecuencias máxima y mínima son desplegadas y pueden ser seleccionados los promedios de señales para reducir el ruido.

PicoScope puede desplegar arriba de 12 medidores en pantalla; un medidor puede desplegar cualquiera de los dos voltajes, corriente directa o alterna, *dB* o frecuencia. Esto hace posible desplegar ambos voltajes y frecuencias de una señal.

PicoScope y PicoLog para DOS requiere una computadora IBM PC o AT compatible corriendo en MS-DOS versión 3.0 o superior. Despliegue compatible con monitores Hercules, VGA, CGA y EGA.

## CASSY

Leybold Didactic ofrece sistemas compatibles de enseñanza y aprendizaje para las áreas relevantes especializadas de procesos tecnológicos, Química, Física, Termodinámica, Dinámica de fluidos, Ingeniería eléctrica, Tecnología de medición y control, Ciencias computacionales, Biotecnología.

Una gran variedad de experimentos en procesos tecnológicos son disponibles desde un modelo de reacción simple de proceso tecnológico hasta configuraciones experimentales complejas sobre un laboratorio adaptado el cual también proporciona los requerimientos de las Universidades.

Leybold Didactic ofrece un sistema de conceptos claramente preparado e instructivo. Permite demostraciones irrestringidas, en sistemas que son consistentemente transparentes y preparados para un nivel sencillo. Las grabaciones de valores de medición con el programa de Windows titulado "Adquisición de datos universal", estos sistemas pueden ser fácilmente controlados desde una PC. La mayoría de los aparatos tienen un diseño modular.

### *PROGRAMA UNIVERSAL DE MEDICIÓN Y EVALUACIÓN*

Cassy cuenta cinco programas universales con un programa de extraordinarias opciones de evaluación total para mayor procesamiento de mediciones de datos registrados.

Tres programas de medición para voltajes de procesamiento o cantidades convertidas en voltajes:

- Multímetro, canal sencillo o dual.
- Osciloscopio de almacenamiento de canal sencillo o dual.
- Grabadora XY.

Dos programas de medición para procesamiento de trenes de pulso:

- Reloj de paro electrónico.
- Medidor de frecuencia y velocidad

*LH - PASCAL CASSY.* Extensión del lenguaje de programación para problemas de libre desarrollo de programas en Turbo Pascal versiones 4.0..7.0 para CASSY; procedimientos y unidades compiladas para controlar varias entradas y salidas de la interface.

Además, Cassy cuenta con diversos programas de adquisición de datos para varias disciplinas como Física, Química y Biología.

### *SOFTWARE ESPECIALIZADO DE FÍSICA*

- Pulsaciones acústicas.
- Péndulos acoplados.
- Movimiento.
- Movimiento (transductor de movimiento).
- Experimentos de impacto (vallas de luz).
- Distribución de Poisson (radiación radiactiva).
- Grabadora transitoria con transformada rápida de Fourier.
- Modelos direccionales de antenas y micrófonos.
- Control, para uso en circuitos de control real.

### *SOFTWARE ESPECIALIZADO DE QUÍMICA*

- Medición del valor de ph.
- Temperatura.
- Conductividad.

### *SOFTWARE ESPECIALIZADO DE BIOLOGÍA*

- Pulso.
- Resistencia de la piel.
- ECG(Electroencefalograma).
- EMG(Electromiograma).
- Presión de sangre.
- Reacción.

Cassy Leybold tiene otros muchos programas, solo citamos los más representativos de cada área, por considerarse información más que suficiente para darse una idea de lo que esta compañía produce.

### **SOFTWARE PARA OTRAS PLATAFORMAS**

Hasta el momento se han analizado diversas opciones de software de adquisición de datos basados en MS-DOS y Windows en cualquiera de sus versiones, así como el uso de procesadores Intel, una de las razones por las cuales este análisis se centro recae en la importancia y mayoría que tienen las aplicaciones basadas en tecnología Intel y por ende en sistemas operativos de Microsoft ya que el dominio de las computadoras personales en el mercado de cómputo es considerable. Pero esto no quiere decir que no existan sistemas de adquisición de datos basados en plataformas Macintosh y UNIX, por lo cual a continuación se mencionan algunos productos de software que trabajan sobre plataformas Macintosh y UNIX.

En lo concerniente a la plataforma Macintosh, de la cual ya se explicaron sus características principales en la introducción de este trabajo de tesis, se explican las siguientes opciones.

### VERNIER

Para las computadoras Macintosh es importante mencionar que la adquisición de datos la realiza mediante la conexión de la interface al puerto serial y se utiliza la interface ULI (Universal Lab Interface), lo que en plataforma MS-DOS sería la interface MPLI, los programas son similares a los descritos anteriormente. Los siguientes son solo programas para esta plataforma.

El programa *Mass Plotter*, permite grabar y graficar datos de una balanza utilizando la computadora.

El programa *Logger Pro* para Power Macintosh, este programa integra algunas de las principales características de los programas basados en DOS en un programa en Windows, puede ser usado con varios sensores como el detector de movimiento, de sonido, de campo magnético, pH y conductividad, entre otros.

También cuenta con el software para utilizar el sensor espectrofotómetro y de dióxido de carbono, así como programas de simulación de movimiento de proyectiles, suma de vectores y movimiento orbital y varios más

### PASCO

#### *SOFTWARE SCIENCE WORKSHOP (TALLER DE CIENCIA)*

El programa Science Workshop funciona en computadoras Macintosh, en las cuales se utilizan ventanas modificables fácilmente, se interactúa con cajas de diálogo, utiliza iconos que se pueden tomar y deslizar, así como el desarrollo de otros pasos necesarios para la instalación de los experimentos, teniendo como resultado un aprendizaje rápido y divertido.

La adquisición de los datos se realiza desde sensores analógicos o de movimiento, también se cuenta con la opción para adquirir datos de sensores digitales, con la opción de definir el tiempo de muestreo, ya que el programa utiliza los mismos procedimientos para ambos tipos de adquisición.

Al utilizar equipo Macintosh se obtienen varias ventajas de funcionalidad como la versatilidad, ya que este software se puede usar como una herramienta veloz de adquisición de datos logrando ser tan eficiente como un osciloscopio de tres trazas, un generador de funciones o una fuente de poder de corriente directa. La opción de graficación, permite la visualización de variables múltiples, es decir, se pueden generar gráficas de una o varias variables contra tiempo o contra otra variable, además, el programa proporciona diversos estilos para generar las gráficas como lo pueden ser *barras de errores*, *línea continua de puntos* y algunas otras opciones más.

También cuenta con despliegue de los datos en forma tabular, utilizando despliegues múltiples en ventanas para observar los datos en formato gráfico o en tablas simultáneamente.

Una parte fundamental de la adquisición de datos reales por medio de sensores es la calibración de los mismos, ya que se requiere proporcionar los archivos que tengan la información del sensor que se va a utilizar en particular, proporcionando con ello una lectura correcta de los datos muestreado, por lo que el archivo debe contener la información de calibración para todos los sensores y la opción de calibrar el sensor de forma particular por el usuario.

Realiza análisis espectral de Fourier, transformada rápida de Fourier para entrada y salida de datos. Integración con la que calcula el área bajo la curva, así como el histograma el cual muestra cálculos y límites de intervalos. Además posee un "cursor inteligente" que muestra las coordenadas de un punto en la gráfica y también una región de interés seleccionando un área de la gráfica para un análisis más exacto.

El programa Science Workshop colecta, despliega y analiza datos desde tres canales analógicos y cuatro canales digitales simultáneamente con un intervalo de muestreo para canal analógico de más de 100,000 muestras por segundo y 10,000 muestras por segundo para tres canales analógicos y cuatro digitales simultáneamente.

Tiene la capacidad de adquirir datos de entradas analógicas y digitales, al mismo tiempo que puede controlar señales de salidas digitales y analógicas, contando con la opción de iniciar o detener la captura o generación de señal mediante un disparo o activación por software. Tiene un generador de señales que produce una salida de corriente directa similar a la de un generador de señales con una frecuencia mayor a 5 KHz.

### UNIX

En el caso de los sistemas de adquisición de datos basados en la plataforma UNIX, se encontró un software que esta desarrollado para funcionar con este tipo de sistema operativo, del cual solo se van a describir los aspectos más importantes.

## **SISTEMA DE ADQUISICIÓN DE DATOS RD13**

El software de adquisición de datos RD13 se basa en diagramas de objetos que se usan al definir el programa interface para acceder a los objetos de hardware y software para la adquisición de datos, por medio de software para el acceso a bibliotecas de bases de datos. Las bases de datos pueden ser usadas para almacenar el orden y los parámetros del sistema, las funciones y propiedades desde los componentes de adquisición de datos, así como la inicialización de las aplicaciones durante el tiempo de corrida, llevando un registro actual y total de las variables del sistema de adquisición de datos.

El sistema RD13 está básicamente desarrollado para funcionar con bases de datos escalables, así como su uso en una red de computadoras. El corazón del sistema es la construcción de una estructura de adquisición de datos que satisface los requerimientos necesarios para ser escalable, en el número de origen de datos y ejecución, teniendo un amplio poder de procesamiento y ancho de banda; este sistema proporciona una gran modularidad particionándose en unidades funcionales; siendo un sistema abierto para una integración fluida de nuevos componentes y características fáciles de implementar.

La actividad fundamental es el uso del tiempo real de un sistema operativo UNIX con el uso de procesadores frontend basados en la tecnología RISC, logrando el mayor aprovechamiento de todas estas características combinadas, convergiendo hacia un estándar de sistemas operativos y alcanzar una plataforma de operación totalmente independiente, confiable y funcional.

Para usar el sistema en tiempo real se han aplicado técnicas de ingeniería de software al desarrollar un protocolo de flujo de datos para poder utilizar un gran número de productos para la ejecución de un sistema de control con aplicaciones en bases de datos y el uso de interfaces. Las aplicaciones de ingeniería de software incluyen la introducción de una tecnología orientada a objetos la cual incrementa la funcionalidad de la adquisición de datos actual y garantiza la independencia de la plataforma por evaluación de nuevas estaciones de trabajo y procesadores.

La descripción de un software de adquisición de datos orientado a objetos está basado sobre un concepto simple: un objeto de software se considera como tres entidades conectadas:

- **SW\_OBJECT:** describe un objeto de software desde el punto de vista lógico, proporciona el nombre del objeto.
- **PROGRAM:** se usa para describir la parte física estática de un objeto de software.
- **PROCESS:** es usado al describir la parte física dinámica de un objeto de software, tal como un control.

Los datos referentes a la computadora en donde el objeto de software es ejecutado pueden ser definidos antes de arrancar el proceso como lo es el vínculo estático entre procesos y objetos de hardware, pero también pueden ser definidos después de arrancar el proceso utilizando un vínculo dinámico entre los procesos y los objetos de hardware.

### *EMPLEO DE BASES DE DATOS PARA LA ADQUISICIÓN DE DATOS*

Un sistema de adquisición de datos necesita un gran número de parámetros para describir todos los componentes de hardware y software a utilizar en la ejecución del sistema. En el caso específico de RD13 se tienen cuatro diferentes bases de datos, una base de datos que almacene la configuración del hardware en la que se describe la disposición del software en términos de procesos, se proporcionan servicios y conexiones; otra base de datos en la que se almacenan los parámetros de corrida, número de corrida, grabación del dispositivo, el nivel en que se encuentra el estado de disparo.

## **CAPÍTULO II**

### **Tarjeta ProControl**

## CAPÍTULO II TARJETA PROCONTROL

En el capítulo anterior hicimos un análisis de los principales elementos de hardware y software necesarios para la realización de adquisición de datos.

En lo relacionado con el hardware, la parte fundamental a tratar son las tarjetas para la adquisición de datos, tema que se desarrolla en el presente capítulo. Analizando las características básicas de la tarjeta ProControl PA-CP12, se seleccionó para el desarrollo de este trabajo de tesis, la cual cumple con todos los requisitos para realizar la adquisición de datos en prácticas experimentales enfocados a la docencia, además de proporcionar la alternativa de un costo menor.

El análisis de los elementos que constituyen la tarjeta ProControl PA-CP12, se divide en cuatro secciones principales, que explican el funcionamiento básico de la misma, y son:

- *Especificaciones generales de la tarjeta.* Donde se mencionan cuales son las características generales de ésta, así como sus conexiones.
- *Conversión Analógica/Digital.* Donde se describe el proceso de conversión que la tarjeta usa para procesar las señales analógicas provenientes de los sensores.
- *Conversión Digital/Analógica.* Donde se describe el proceso de conversión que permite controlar sistemas que tomen valores digitales y representar estos como niveles de voltaje analógico.
- *Entradas y salidas digitales.* Donde se describe cómo proporcionan capacidades de control desde la computadora para utilizarse en aplicaciones que requieran un control digital de encendido o apagado.

### ESPECIFICACIONES GENERALES

Resolución:	12 bits.
Número de canales de entrada:	16 canales.
Rango de voltaje de entrada:	-5V a + 5V, -10 V a +10 V 0V a +5V, 0V a +10 V
Precisión:	2 bits.
Tiempo de conversión	200ms, 5KHz.
Sobrevoltaje:	35 V.
Método de conversión:	Disparo por software. Disparo por reloj externo. Disparo por temporizador interno 8254.
Conector:	Conector macho de 37 pines.

**CONVERTIDOR DIGITAL/ANALÓGICO**

Resolución:	12 bits.
Canal:	1
Rango de voltaje de entrada:	-5V a + 5V, -10 V a +10 V 0V a +5V, 0V a +10 V
Precisión:	2 bits.
Tiempo de conversión	20ms, 50KHz.
Conector:	Conector macho de 37 pines (compartido con el convertidor A/D).

**ENTRADAS Y SALIDAS DIGITALES**

Número de entradas digitales	8 bits.
Tipo de entrada:	74LS244.
Voltaje alto/bajo de entrada:	2.0V/0.8V
Número de salidas digitales:	8 bits.
Tipo de entrada:	74LS244.
Voltaje alto/bajo de entrada:	2.4V/0.5V.
Conector:	Conector de 20 pines macho.

**TEMPORIZADOR PROGRAMABLE**

Temporizador:	8254.
Contadores:	3 de 16 bits. 2 para la conversión A/D. 1 para uso interno de la conversión A/D.
Fuente del reloj:	4 MHz.
Tipo de entrada/salida:	TTL.

**CARACTERÍSTICAS DE LA INTERFACE**

Compatible con:	Bus IBM PC y compatibles.
Tipo de interface:	12 bits E/S mapeadas.
Rango de puertos de dirección:	100H a FF0H.
Interrupciones:	IRQ2, 3, 5, 7, 10, 11, 12, 14, 15.
Destino de la interrupción:	Final de la conversión A/D.
Requerimientos de alimentación:	+5V, +12V, -12V.

**AMBIENTE FÍSICO**

Temperatura de operación:	5°C a + 50°C.
Temperatura de almacenamiento:	-25° a + 85°.
Humedad de operación:	0 a 90%, no condensada.
Humedad de almacenamiento:	0 a 90%, no condensada.
Peso:	170g.
Dimensiones:	19 x 13.2 x 2.5 cm.

Es importante mencionar que esta tarjeta no está diseñada para funcionar bajo la tecnología conectar y ejecutar, solamente se conecta en una ranura de la computadora. Como se mencionó anteriormente las tarjetas de adquisición de datos necesitan un manejador para que la computadora pueda reconocerlas. En este caso la tarjeta que se seleccionó no cuenta con manejadores proporcionados por el fabricante; por lo cual un paso necesario para que la computadora reconozca la tarjeta es la realización de un programa (por ejemplo en lenguaje C), el cual sea capaz de capturar y transmitir datos a las direcciones de memoria establecidas por la tarjeta.

Los pasos iniciales para la utilización de la tarjeta ProControl PA-CP12 son la instalación, la configuración de los interruptores y conectores y el ajuste de los rangos de conversión. A continuación se mencionan los conectores que contiene la tarjeta ProControl PA-CP12.

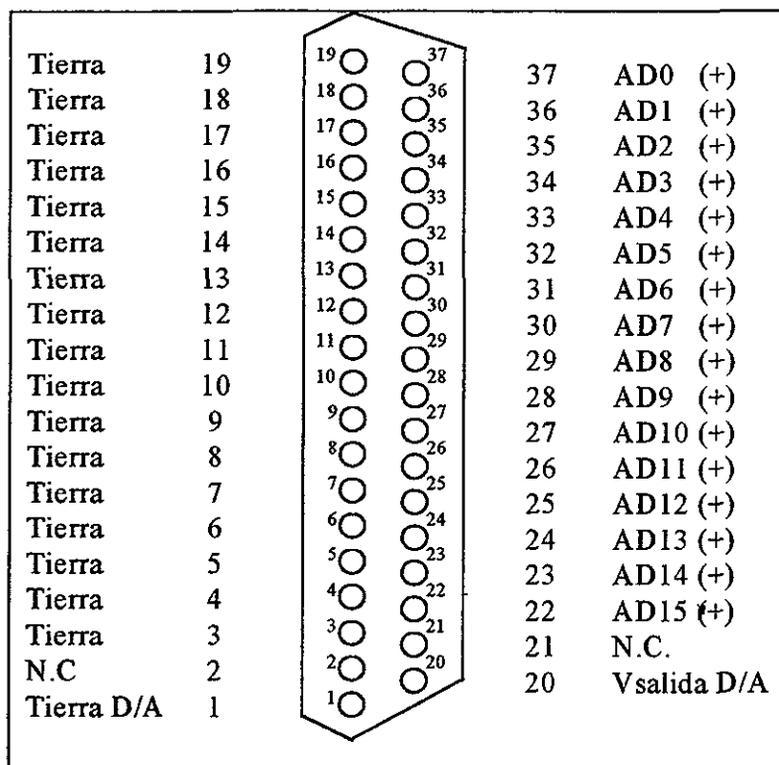


Fig. II.1 Diagrama del conector para entrada de señal analógica.

### INTERRUPTORES Y CONECTORES DE LA TARJETA ProControl PA-CP12<sup>1</sup>

- JP1 Rango de voltaje de entrada A/D
- JP2 Rango de voltaje de salida D/A
- JP3 Selección de canales por medio de interrupciones
- JP4 Selección de direcciones de entrada/salida

<sup>1</sup> Para mayor detalle refiérase a la hoja de configuraciones del manual de la tarjeta ProControl PA-CP12.

- J1 Interruptor A/D y D/A
- J4 Fuente del reloj externo
- CON1 Conector de salida digital
- CON2 Conector de entrada digital

El ajuste de los rangos de conversión se realiza por medio de *potenciómetros* localizados en la tarjeta, los cuales son:

- VR30 Ajusta el rango del convertidor D/A
- VR31 Ajusta el offset del convertidor D/A
- VR32 Ajusta el rango del convertidor A/D
- VR33 Ajusta el offset del convertidor A/D

**CALIBRACIÓN**

La calibración de la parte D/A de la tarjeta ProControl PA-CP12 consta de dos pasos. El primero es el ajuste de offset, y el segundo es el ajuste total de escala. La calibración se realiza una vez, ajustando los valores a usarse en la tarjeta ProControl PA-CP12.

El offset se ajusta primero. Al ajustar el offset, se escribe el valor cero (000H para unipolar, 800H para bipolar) al puerto de salida deseado, midiendo la salida con un voltímetro, se ajusta el potenciómetro offset VR31 hasta que el voltaje es cero.

Después se necesita ajustar el rango por lo cual se escribe el valor digital total de la escala al puerto de salida (en la tabla II.1 se muestran los valores para ajustar la tarjeta a los rangos de voltaje requeridos), y midiendo con un voltímetro, el usuario debe ajustar el potenciómetro VR30 de ganancia/rango hasta que la salida sea igual al valor máximo del rango seleccionado.

Rangos de salida		Resultados de datos D/A (Bajo + Alto)		
Bajo	Alto	000 H	800H	FFFH
0 V	+ 10 V	0 V	+ 5.000 V	+ 9.9988 V
0 V	+ 5 V	0 V	+ 2.500 V	+ 4.9995 V
-10 V	(+ 10V) 9.999 V	- 10.000 V	0 V	+ 9.9975 V
-5 V	(+ 5V) 4.999 V	- 5.000 V	0 V	+ 4.9988 V

Tabla II.1 Valores de voltaje D/A.

Esta tarjeta maneja la información por medio de registros y direcciones. El intervalo de direcciones va de la dirección 100H a la FF0H. Para el uso de la tarjeta se necesitan solo un rango de 15 valores por lo que se seleccionó el intervalo de 02E0H a 02EFH.

Realizamos un análisis de la memoria que utiliza Windows 95, confirmando que el intervalo de memoria usado por la tarjeta ProControl PA-CP12 no interfiere con los programas y aplicaciones utilizados por este sistema.

Es importante referir que la base inferior de memoria es la dirección inicial 02E0H, cuando se menciona un aumento en cada base, quiere decir que se aumenta un número específico a la dirección inicial. Por ejemplo, si se menciona la base+5 se está haciendo referencia a la dirección de memoria 02E5H.

La siguiente tabla de registros, contiene la información de cada base, su función y dirección de memoria.

PUERTO	DEFAULT	DIRECCION	FUNCION
Base +0	02E0H	Lectura entrada/salida	Dato Bit Menos Significativo A/D
		Escritura entrada/salida	Inicialización por Software
Base +1	02E1H	Lectura entrada/salida	Dato Bit Mas Significativo A/D
		Escritura entrada/salida	Ninguna
Base +2	02E2H	Lectura entrada/salida	Selección de canal
		Escritura entrada/salida	
Base +3	02E3H	Lectura entrada/salida	Estado A/D
		Escritura entrada/salida	Comando A/D
Base +4	02E4H	Lectura entrada/salida	---
		Escritura entrada/salida	Dato Bit Menos Significativo D/A
Base +5	02E5H	Lectura entrada/salida	---
		Escritura entrada/salida	Dato Bit Más Significativo D/A
Base +6	02E6H	Lectura entrada/salida	No se usa
		Escritura entrada/salida	No se usa
Base +7	02E7H	Lectura entrada/salida	No se usa
		Escritura entrada/salida	No se usa
Base +8	02E8H	Lectura entrada/salida	Escritura Digital (CON1)
		Escritura entrada/salida	Lectura Digital (CON2)
Base +9	02E9H	Lectura entrada/salida	No se usa
		Escritura entrada/salida	No se usa
Base +10	02EAH	Lectura entrada/salida	No se usa
		Escritura entrada/salida	No se usa
Base +11	02EBH	Lectura entrada/salida	No se usa
		Escritura entrada/salida	No se usa
Base +12	02ECH	Lectura entrada/salida	Contador 8254 Dato 0
		Escritura entrada/salida	(Para uso interno)
Base +13	02EDH	Lectura entrada/salida	Contador 8254 Dato 1
		Escritura entrada/salida	( Para tiempo de conversión)
Base +14	02EEH	Lectura entrada/salida	Contador 8254 Dato 2
		Escritura entrada/salida	( Para tiempo de conversión)
Base +15	02EFH	Lectura entrada/salida	Estado de lectura del 8254
		Escritura entrada/salida	Comando 8254

Tabla II.2 Información de los registros.

Parte importante de la captura automática de datos, es el proceso de conversión, como se mencionó anteriormente. La señal analógica debe de pasar por una serie de procedimientos para poder ser leída correctamente por la computadora. De igual manera, el proceso inverso en el que se requieran transformar señales de niveles digitales desde la computadora a señales analógicas. Haciendo referencia a las especificaciones citadas con anterioridad sobre los convertidores Analógicos/Digitales y Digitales/Analógicos, presentamos un análisis más detallado sobre la teoría de operación de estos convertidores en la tarjeta ProControl PA-CP12.

## II. 1 CONVERSIÓN ANALÓGICO/DIGITAL

La conversión analógica/digital en la tarjeta ProControl PA-CP12 consiste de 4 bloques, los cuales son:

Bloque A: Entrada de la señal y selección de canal.

Bloque B: Conversión analógico/digital y salida de registros de datos.

Bloque C: Control de la conversión analógico/digital.

Bloque D: Registros de estado.

En la figura II.2, se presenta el diagrama de bloques del proceso de conversión analógico/digital.

### ***BLOQUE A: Entrada de la señal y selección de canal.***

El conector para la señal es un conector macho tipo D de 37 pines. El convertidor A/D y D/A comparten el mismo conector.

### ***SELECCIÓN DE CANAL.***

La tarjeta PA-CP12 puede ser configurada para 16 entradas de una sola terminal solamente, el canal se selecciona por medio de software, en el caso de este trabajo se hace mediante un programa en lenguaje C. El puerto usado para la selección del canal es base+2, por ejemplo, si se quiere seleccionar el canal 3, se debe escribir en el puerto el número del canal. La tarjeta no proporciona un circuito de muestreo y retención, es decir, por medio del software a utilizar, se selecciona el tiempo de muestreo y se almacenan los datos.

### ***BLOQUE B: Conversión analógico/digital.***

Cuando la conversión analógico/digital se realiza (puede ser por un disparador de software, por un disparador de reloj externo, etc.), la línea EOC (final de la conversión, End Of Conversion ) funciona en alto, es decir, el bit que corresponde a esta operación es 1. En la tabla II.4 se muestra el estado de los registros necesarios para la conversión analógica/digital. Cuando inicia la conversión la señal es recibida. La conversión de voltaje del canal seleccionado es inicializada. Una vez que la conversión es inicializada otro comando no podrá ser aceptado hasta que se complete la conversión que se está ejecutando.

# CONVERSIÓN ANALÓGICA DIGITAL

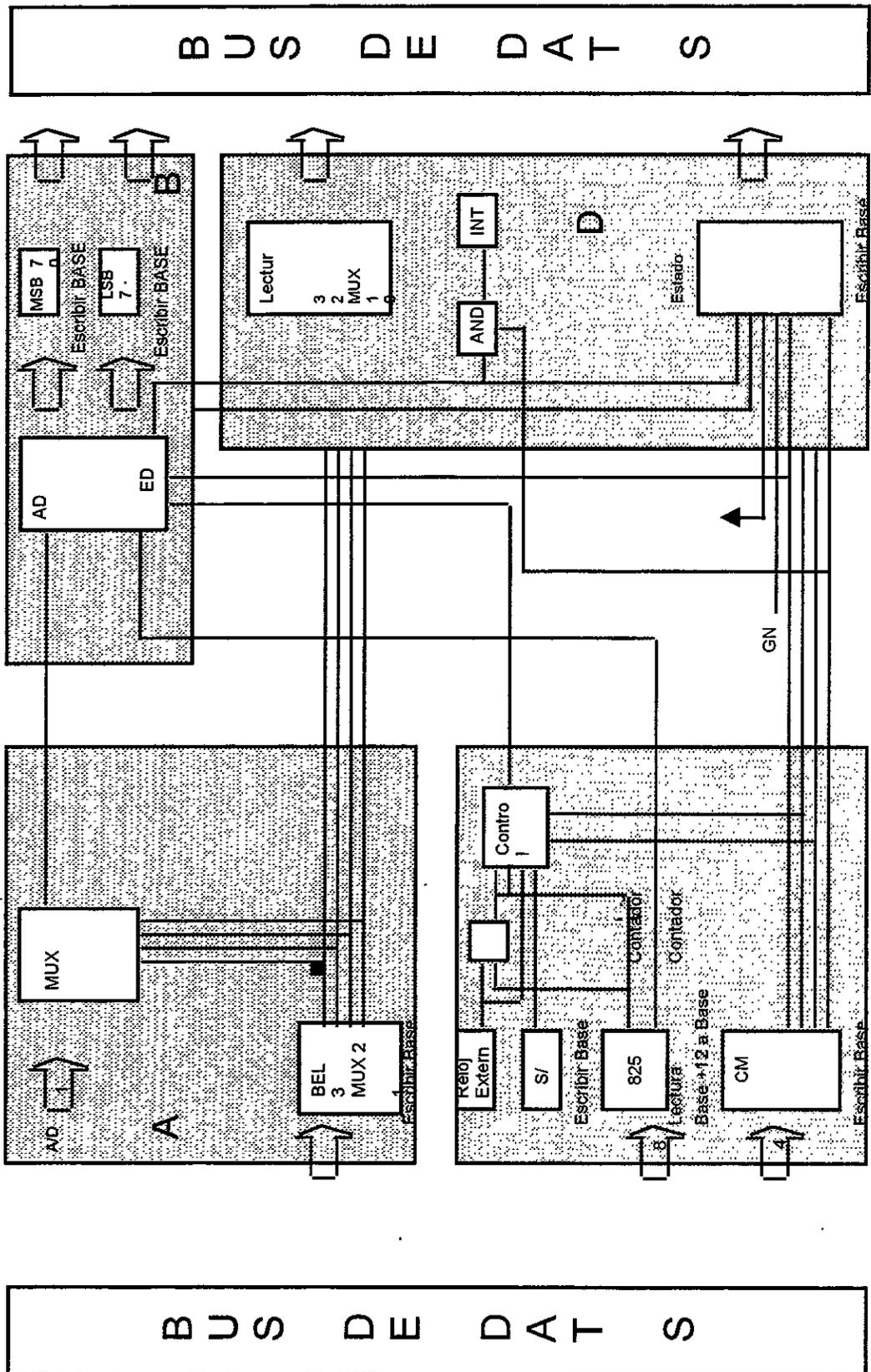


Fig. 11.2 Diagrama de bloques del proceso de conversión

Los 8 bits del puerto base+1 son los 8 bits más significativos del resultado de la conversión analógico/digital, y los 4 bits más significativos del puerto base+0 son los 4 bits menos significativos del resultado de la conversión analógico/digital, teniendo así los 12 bits que nos da el convertidor.

### ***BLOQUE C: Control de la conversión A/D.***

La conversión A/D puede ser inicializada de tres maneras distintas. Cada una de ellas tiene ciertas ventajas con respecto a las otras. Estas tres maneras son:

- Inicialización de la conversión por medio de software
- Inicialización de la conversión por medio de un reloj externo
- Inicialización de la conversión por medio de un temporizador 8254

La figura II.3, nos muestra el diagrama del control de conversión Analógico/Digital.

La conversión es seleccionada por medio de la escritura en el puerto de comando base+3; el bit 3 decide la resolución de la conversión A/D, el bit 2 y 1 deciden la fuente del disparo para la conversión (es decir, como se inicializara); el bit 0 habilita o deshabilita la interrupción. [ ver tabla II.4 ]

### ***INICIALIZACIÓN DE LA CONVERSIÓN POR MEDIO DE SOFTWARE***

Este método se implementa fácilmente porque el software es el que controla directamente las conversiones.

### ***INICIALIZACIÓN DE LA CONVERSIÓN POR MEDIO DE UN RELOJ EXTERNO***

Consiste en usar un reloj externo o un pulso, más largo de 200 nanosegundos, para disparar la conversión A/D. La señal viene desde un interruptor (J4), el cual se encuentra en la tarjeta. Este método da una gran flexibilidad en la señal de disparo permitiendo la sincronización del convertidor A/D con otros dispositivos.

### ***INICIALIZACIÓN DE LA CONVERSIÓN POR MEDIO DE UN TEMPORIZADOR 8254***

Este método es similar al disparador externo. Esta conversión se da en segundo plano, es decir, deja al CPU realizar tareas más importantes. Sin embargo, el temporizador 8254 se encuentra en la tarjeta y tiene tres relojes configurables para decidir la forma de señal de disparo.

Otra opción que se puede realizar con este temporizador es enlazar la señal del temporizador y la señal externa para dar un control externo sobre las conversiones con el temporizador 8254.

# CONTROL DE LA CONVERSIÓN ANALÓGICO/DIGITAL

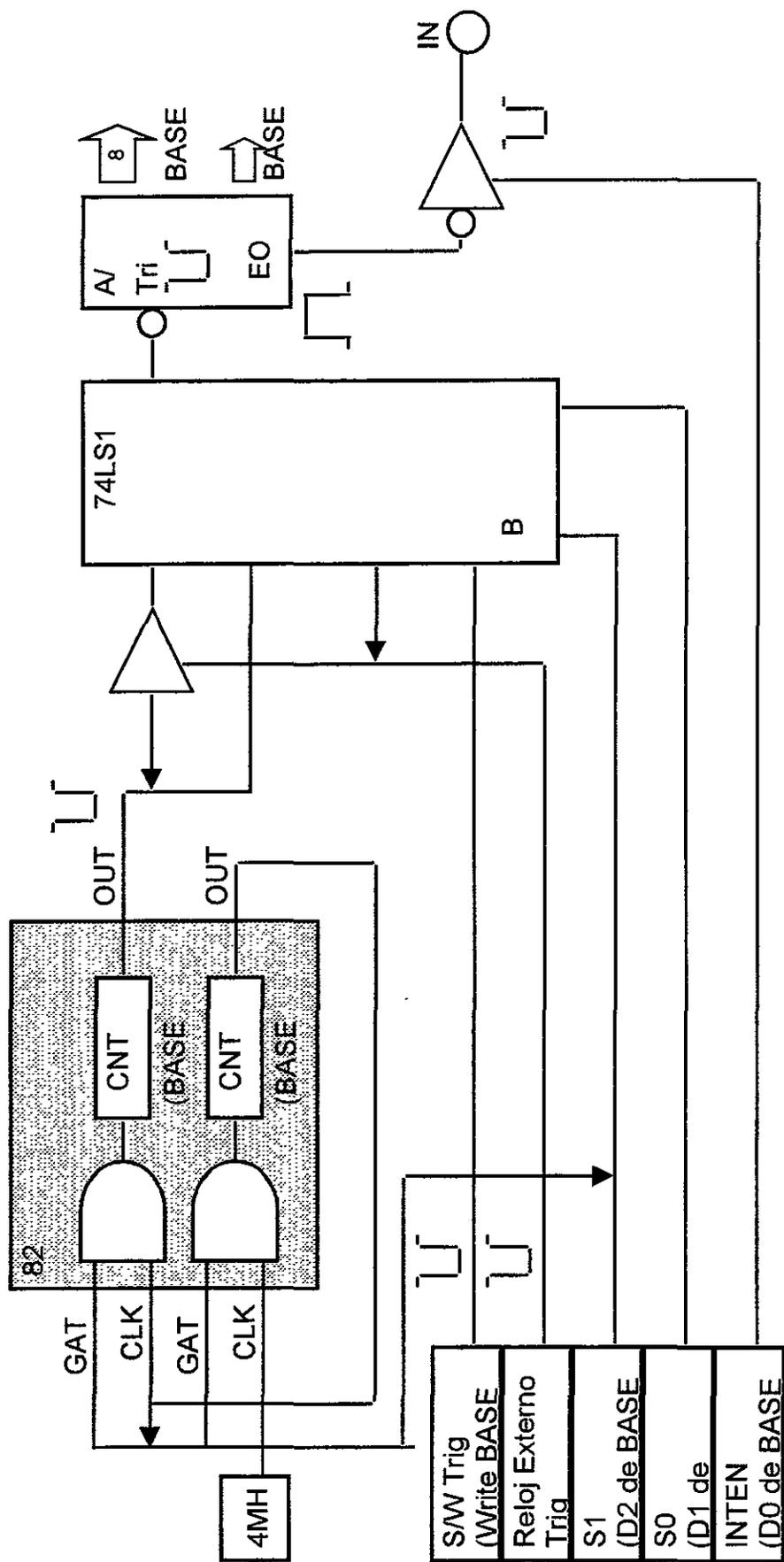


Fig. II.3 Control de conversión

La tarjeta PA-CP12 ofrece gran variedad de configuraciones para la conversión A/D. En la tabla II.3 se muestran los métodos de conversión.

Modo de conversión	Iniciar por software	Iniciar por reloj externo	Iniciar por reloj 8254
Comando a escribir	x0H	x3H	X5H o x7H
Inicio 8254	No	No	Si
Cuando empieza la conversión	Escribir en puerto base+0 para generar el pulso	El interruptor J4 en bajo	8254 salida 1 y 2
Durante la conversión	El estado EOC se pone en alto, otro comando de inicialización no puede ser activado.		
Como saber que la conversión finalizó	EOC del puerto de estado en bajo	La interrupción ocurrirá	La interrupción ocurrirá
La frecuencia de la conversión será decidida por	Frecuencia de inicio por software S/W	Frecuencia del reloj externo	8254 contador 0 y 1
Proceso de la conversión a	Foreground	Background	Background
Beneficios	Fácil programación	Sincronización con otro dispositivo	Tiempo de conversión
Lenguaje de programación	Ensamblador, C, Basic	Ensamblador por kernel	Ensamblador por kernel

Tabla II.3 Métodos de conversión Analógico/Digital.

Después de analizar cada una de las opciones para el control de la conversión, decidimos utilizar la opción de control por medio de disparo por software, porque ofrece diversas ventajas sobre los otros medios.

El lenguaje de programación que se utiliza, es un lenguaje de programación relativamente sencillo (como es Basic o C), por lo cual, el trabajo se ve reducido en tiempo de desarrollo, con esto se pueden controlar los canales por medio de software y con esto hacer más fácil el uso de la tarjeta ProControl PA-CP12.

#### ***BLOQUE D: Registros de estado (Estado de lectura)***

El estado del convertidor A/D puede leerse desde dos puertos; base+2 y base+3. El puerto base+3 contiene la información importante para la instalación y las condiciones del convertidor, mientras que el puerto base+2 contiene la información del canal seleccionado.

##### ***Base+2: Estado del canal***

Los bits 3-0 representan el canal analógico/digital seleccionado.

*Base+3: Estado del sistema*

El bit 7 representa el final de la conversión; el bit 6 representa la configuración de las entradas unipolares y bipolares; el bit 5 representa la configuración de entrada diferencial o de una sola terminal, porque la tarjeta sólo puede configurarse para entradas de una sola terminal, este bit se encuentra siempre en 1; el bit 4 siempre está en cero, el bit 3 representa la configuración de la resolución de la conversión A/D ya sea de 12 o 6 bits; los bits 2 y 1 representan el tipo de control de la conversión y el bit 0 nos dice si las interrupciones son habilitadas o deshabilitadas. La tabla II.4 muestra detalladamente cada bit del puerto base+3, correspondiente al estado de la conversión A/D.

Bit	Símbolo	Función
7	EOC	Final de la conversión A/D 1 Conversión A/D en proceso 0 Conversión A/D finalizada
6	SELR	Modo entrada de voltaje A/D 0 Modo bipolar 1 Modo unipolar
5	S1-S0	Tipo de entrada A/D 1 Debe de mantenerse en uno este bit, porque es entrada de una sola terminal
4	INTE	0 Siempre en cero, por especificaciones del proveedor.
3	SELR	Selección de la resolución 0 Conversión de 12 bits 1 Conversión de 6 bits.
2-1	S1-S0	Control de la conversión A/D S1 S0 Señal de la conversión 0 0 Disparo por software 0 1 Disparo por reloj externo 1 0 Disparo por reloj 8254 1 1 Disparo por reloj externo
0	INTE	Habilitar interrupciones 0 Interrupción deshabilitada 1 Interrupción habilitada

Tabla II.4 Estados de los registros del puerto base+3

## PROCESOS DE CONVERSIÓN

En esta sección se describen los pasos necesarios para la conversión analógica/digital.

### *INICIALIZACIÓN DE RELOJ INTERNO (8254 CONTADOR 0)*

La tarjeta ProControl PA-CP12 usa un chip 8254 contador 0 para generar un reloj de 74KHz de uso interno. Sin embargo, no es importante el modo de conversión que se elija, el 8254 contador 0 debe ser inicializado primero. Este procedimiento necesita ser realizado una sola vez.

Como paso inicial para la conversión se debe inicializar el contador 8254 con los siguientes pasos:

1. - Escribir 34H en el puerto base+15 (Puerto de comando 8254). Este puerto se refiere al temporizador 8254; 34H es el comando necesario para inicializar el reloj.
2. - Escribir 36H en el puerto base+12 (Puerto de datos del contador 8254). Este puerto es de uso interno de la tarjeta, se le asigna 36H como dato de inicio.
3. - Escribir 00H en el puerto base +12 (Puerto de datos del contador 8254). Al escribir 000H en la tarjeta se inicializa por completo el temporizador 8254 y lo inicializa en cero.

### *CONVERSIÓN POR DISPARO DE SOFTWARE*

Esta conversión empieza cuando se escribe 0000x000B en el puerto base+3. Esto inicializa el puerto (si d3=0 la conversión es de 12 bits y si d3=1 entonces la conversión es de 6 bits). Para seleccionar el canal se escribe el número de canal en el puerto base+2. Se puede verificar el estado de la conversión leyendo base+3. El bit que marca el final de la conversión tiene que estar en cero.

### *CONVERSIÓN DE FUENTE DE DISPARO DE RELOJ EXTERNO*

Existen tres partes básicas en este método, las cuales son: inicializar, rutina de interrupción y finalizar.

**INICIALIZAR:** Primeramente se debe definir el *buffer* de datos, el número de canal y el vector de interrupción. Siguiendo con la inicialización del comando A/D. Después, el controlador de interrupciones 8259 debe ser programado para habilitar la interrupción. Cuando esto este completo, la rutina de interrupción puede ser detenida.

**RUTINA DE INTERRUPCIÓN:** Cuando la conversión esta completa, la interrupción ocurre y se pueden leer los datos de ambos registros de datos ( base+0 y base+1). Los datos adquiridos serán guardados en el buffer de datos, el canal puede cambiarse si se necesita, y al final de la señal de interrupción se pueden mandar al 8259 escribiendo 20H en el puerto base+3.

**FINALIZAR:** El procedimiento que se debe seguir para terminar, es deshabilitar la interrupción del canal del controlador de interrupciones 8259. Después el vector de interrupciones puede recobrar los valores previos.

***CONVERSIÓN POR DISPARO DEL TEMPORIZADOR CON EL 8254***

La conversión por disparo del temporizador usando la salida del reloj 8254, es muy similar a emplear el disparo externo. La diferencia es que cuando se utiliza la salida del temporizador 8254 la frecuencia de conversión que es determinada por los contadores 1 y 2.

Inicializar el 8254 se realiza usando los puertos base+13 a través del puerto base+15. El resto de la programación es de la misma manera que la conversión por disparador externo. Los ejemplos de datos para inicializar el 8254 se especifican en la tabla II.5.

Frecuencia de la conversión A/D		Datos para la inicialización del 8254					
		8254 contador 2			8254 contador 1		
		Base+15	Base+14	Base+14	Base+15	Base+13	Base+13
5KHz	200us	B4H	4	0	74H	200	0
4KHz	250us	B4H	4	0	74H	250	0
2KHz	500us	B4H	4	0	74H	244	1
1KHz	1ms	B4H	4	0	74H	232	3
500Hz	2ms	B4H	4	0	74H	208	7
400Hz	2.5ms	B4H	4	0	74H	196	9
200Hz	5ms	B4H	4	0	74H	136	19
100Hz	10ms	B4H	4	0	74H	16	39
50Hz	20ms	B4H	4	0	74H	32	78
40Hz	25ms	B4H	4	0	74H	168	97
20Hz	50ms	B4H	4	0	74H	80	195
10Hz	100ms	B4H	40	0	74H	16	39
5Hz	200ms	B4H	40	0	74H	32	78
4Hz	250ms	B4H	40	0	74H	169	97
2Hz	500ms	B4H	40	0	74H	80	195
1Hz	1s	B4H	144	1	74H	16	39

Tabla II.5 Datos para la inicialización del 8254.

**II. 2 CONVERSIÓN DIGITAL/ANALÓGICA**

La tarjeta ProControl PA-CP12 cuenta con un convertidor digital/análogo el cual sirve para controlar un dispositivo desde la computadora, que use señales digitales las cuales requieren ser transformadas a señales analógicas.

Es un convertidor de 12 bits, aunque el bus de datos es de 8 bits de longitud, sin embargo la tarjeta ProControl PA-CP12 cuenta con un registro temporal para poder hacer la conversión de 12 bits, es decir, el registro del bit menos significativo funciona como un registro temporal, después, cuando el bit más significativo está en el bus, se puede enviar la palabra completa de 12 bits al convertidor Digital/Analógico simplemente por escritura del bit más significativo al puerto de salida.

El siguiente diagrama de bloques corresponde a la conversión digital/analógica.

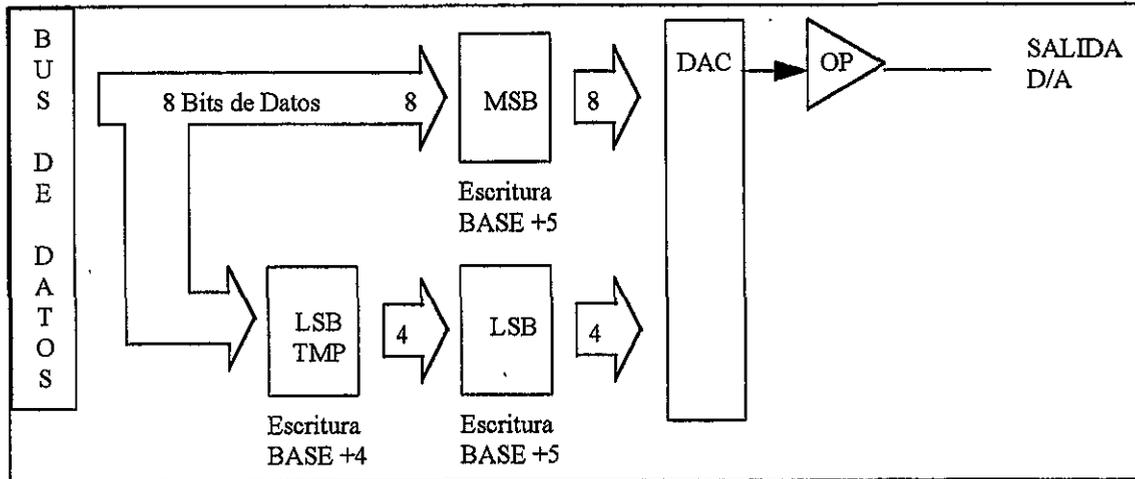


Fig. II.4. Diagrama de bloques de conversión digital/analógica

La salida desde el convertidor Digital/Analógico puede ser configurado para diferentes rangos de voltaje. Podemos decidir entre cuatro diferentes rangos de voltaje; estos son: -10V a +10V, -5V a +5V, 0 a +10V, y 0 a 5V. Un interruptor controla la salida que puede ser unipolar o bipolar, los otros dos interruptores controlan la salida que puede ser de 5V o 10V máximo.

El conector utilizado es el mismo conector tipo D de 37 pines empleado para el convertidor Analógico/Digital. Referente a las direcciones de memoria que usa el convertidor son, base+4 que es temporalmente el registro de retención para el bit menos significativo en el Digital/Analógico y base+5 es el registro de salida para el bit más significativo. A continuación se muestran las tablas de registros.

### PROCEDIMIENTO DE CONVERSIÓN

La programación del convertidor D/A para 12 bits se realiza de una forma sencilla. Se escriben los 4 bits menos significativos al registro temporal D7-D4 (base+4). En seguida se escriben los 8 bits más significativos al puerto de salida (base+5), para así tener los 12 bits que se necesitan. Si se requiere hacer otra conversión D/A, se debe observar si el puerto base+4 no ha cambiado, es decir se conservan los 4 bits menos significativos en el registro temporal de igual manera que la primera vez, entonces lo único que se hace es tomar nuevamente los 8 bits más significativos del puerto de salida base + 5 para la conversión del siguiente número.

REGISTROS DEL CANAL D/A

Base +4		Escritura		Canal de registro temporal del bit menos significativo D/A			
7	6	5	4	3	2	1	0
D03	D02	D01	D00	x	X	x	x
Bit	Símbolo		Función				
7 - 4	D03 - D00		Dato bit menos significativo D/A				
3 - 0	X		No importa				

Base +5		Escritura		Canal de registro temporal bit mas significativo D/A			
7	6	5	4	3	2	1	0
D11	D10	D09	D08	D07	D06	D05	D04
Bit	Símbolo		Función				
7 - 0	D11 - D04		dato bit mas significativo d/a				

Puerto	Base +5							Base +4				
Bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4
D/A	11	10	9	8	7	6	5	4	3	2	1	0

Tabla II.6 Registros de los canales Digital/Analógico..

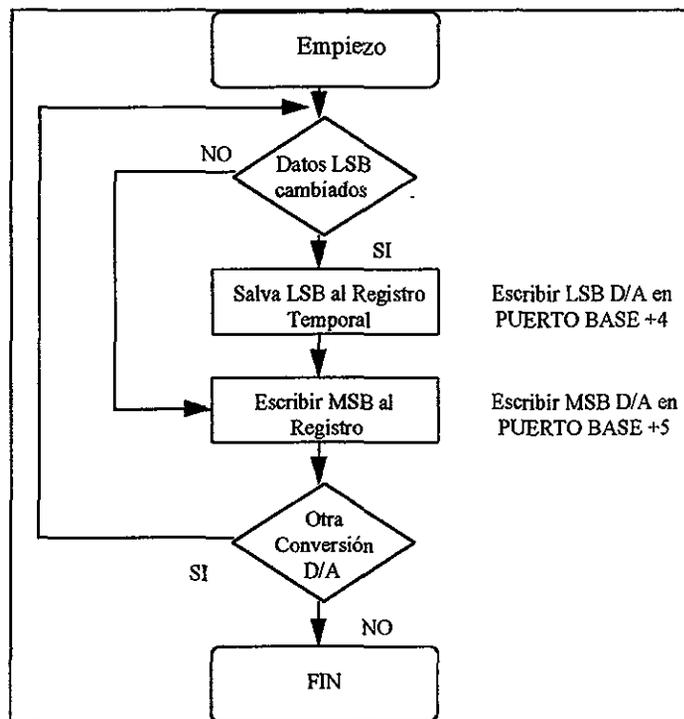


Fig. II.5. Diagrama de procedimiento de programación D/A.

### III. 3 ENTRADAS Y SALIDAS DIGITALES

La tarjeta ProControl PA-CP12 contiene 16 canales de entradas/salidas digitales; entradas digitales de 8 bits y salidas digitales de 8 bits.

#### CONECTORES

Tiene dos conectores para la entrada y salida digital. Cada uno consiste de un conector de 20 pines tipo caja cabecera, el conector 1 (CON1) es salida y el conector 2 (CON2) es entrada.

El diagrama de los conectores se muestra en la figura II.6.

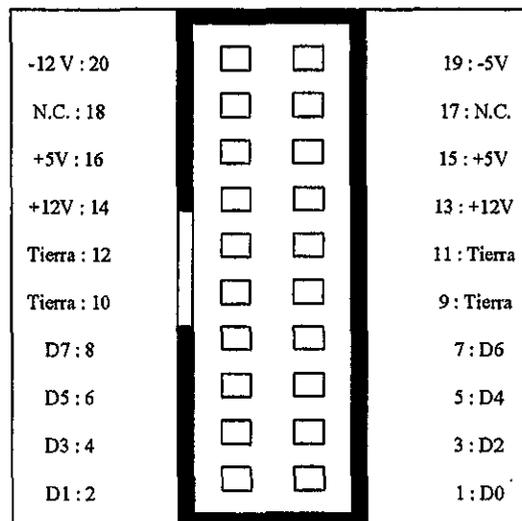


Fig. II.6 Conector para entrada y salida digital.

#### SALIDAS DIGITALES

Las salidas digitales consisten en 8 bits dedicados para la ruta de salida. Para escribir datos hacia fuera de esta ruta, simplemente se debe escribir a los puertos de E/S; base +8 que es la salida para el puerto de CON1.

#### ENTRADAS DIGITALES

Las entradas digitales consisten en 8 bits dedicados para la ruta de entrada. Para leer datos de esta ruta, simplemente se debe leer de los puertos de E/S; base +8 que es la entrada para el puerto de CON2.

En el caso de este trabajo empleamos un canal digital para el control de encendido/apagado de un foco y una parrilla, el programa se explica detalladamente en el capítulo V.

**CAPÍTULO III**  
**Software LabVIEW**

## **CAPÍTULO III**

### **SOFTWARE LABVIEW**

#### **Laboratory Virtual Instrument Engineering Workbench**

En este capítulo se efectúa una descripción detallada del software que se utiliza para el desarrollo de los programas de este trabajo de tesis. El software elegido pertenece a la compañía National Instruments y se llama LabVIEW

LabVIEW es una aplicación para desarrollo de programas bajo la plataforma Windows, enfocado a la adquisición de datos, el proceso y el almacenamiento de los mismos. Estos datos o muestras pueden ser tomadas de sistemas físicos en tiempo real a través de sensores.

LabVIEW en términos computacionales es comparable con distintos lenguajes de programación tan poderosos como Visual Basic, Pascal, Visual C o con el LabWindows de la misma compañía que el seleccionado; teniendo con los anteriores una diferencia esencial. Estos lenguajes son basados en texto, es decir, la programación se basa en crear líneas de código, en tanto que LabVIEW ha evolucionado más que los lenguajes ya mencionados siendo un lenguaje de programación gráfica, llamado lenguaje G, la realización de sus programas se hace en forma de diagrama de bloques siendo un lenguaje de tipo gráfico estructurado. Este lenguaje consta de todos los elementos que constituyen a un lenguaje, como: secuencia de instrucciones, estructuras de repetición, entrada y salida de datos en tiempo de ejecución, llamada a subrutinas, bifurcaciones y toma de decisiones.

LabVIEW puede ser usado por programadores que no posean grandes conocimientos y experiencia. Usa terminologías, iconos e ideas familiares a científicos e ingenieros y lo realiza con símbolos gráficos, resultando mejor que un lenguaje textual al describir acciones de programación.

LabVIEW tiene extensas bibliotecas de funciones y subrutinas para mejorar las tareas de programación, como por ejemplo funciones para diferentes estructuras (secuencia, decisiones, ciclos for y while), numéricas (suma, resta, multiplicación incrementos, redondeo, etc.), *booleanas*, cadenas de caracteres, arreglos, comparaciones, archivos e instrumentos de entrada y salida, comunicación, adquisición de datos, análisis y muchas más.

También contiene bibliotecas de aplicación específica para adquisición de datos, instrumentos de control serial, VXI y GPIB, análisis de datos, presentación de datos y almacenamiento de los mismos. Con LabVIEW es posible colocar puntos de ruptura, animar la ejecución para observar cómo los datos fluyen a través del programa y, por medio de pasos sencillos, éste realiza la depuración y desarrolla las instrucciones más fácilmente.

LabVIEW es capaz de ejecutar códigos programados en lenguaje C, esto es mediante un proceso de compilación que se lleva a cabo por el compilador Watcon o un programa realizado en Visual C++.

LabVIEW es una herramienta para desarrollar aplicaciones que adquieran datos externos a la computadora, sin embargo para que la información sea recabada es necesario que la computadora que ejecute esta acción cuente con una tarjeta de adquisición de datos conectada a la misma. En principio esta tarjeta debe ser de la misma compañía que el software, sin embargo, nosotros utilizamos una tarjeta diferente que pertenece a otra compañía (la cual se describió ampliamente en el capítulo anterior); además de requerir un sensor el cual contenga un circuito electrónico que proporciona las mediciones del medio externo (sean cuales fueren el tipo de estas mediciones) y sea conectado a la tarjeta que realiza la captura de los datos.

### COMO TRABAJA LabVIEW

LabVIEW incluye bibliotecas de funciones y herramientas de desarrollo diseñadas específicamente para adquisición de datos e instrumentos de control. Los programas realizados en LabVIEW son llamados "instrumentos virtuales" (IVs) porque su apariencia y operación imitan a los instrumentos reales, los cuales son análogos a funciones realizadas desde programas en lenguaje convencional. Los IVs tienen una interface interactiva con el usuario y un código fuente equivalente que aceptan parámetros desde otros IVs de nivel más alto.

#### Componentes esenciales de los IVs

- La interface interactiva con el usuario de un IV es llamado el **panel frontal** porque este simula el tablero de un instrumento físico. El panel frontal contiene perillas, botones o teclas de presión, gráficas así como otros controles e indicadores diseñados por el usuario. Los datos provienen de un sensor o bien se pueden introducir usando el *ratón*, el *teclado*, y así ver los resultados en la pantalla de la computadora.

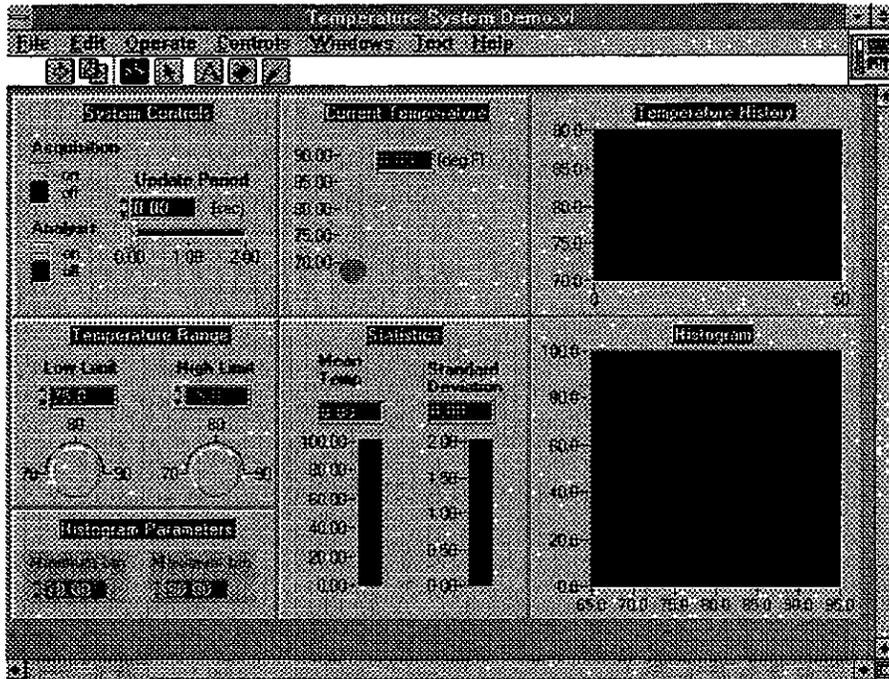


Fig. III.1 Ejemplo de un panel de control de un instrumento virtual, el cual simula un sistema de control de temperatura.

- El IV recibe instrucciones desde un **diagrama de bloques**, el cual se construye en lenguaje G. El diagrama de bloques es una solución pictórica a un problema de programación, y éste es también el código fuente para el IV.

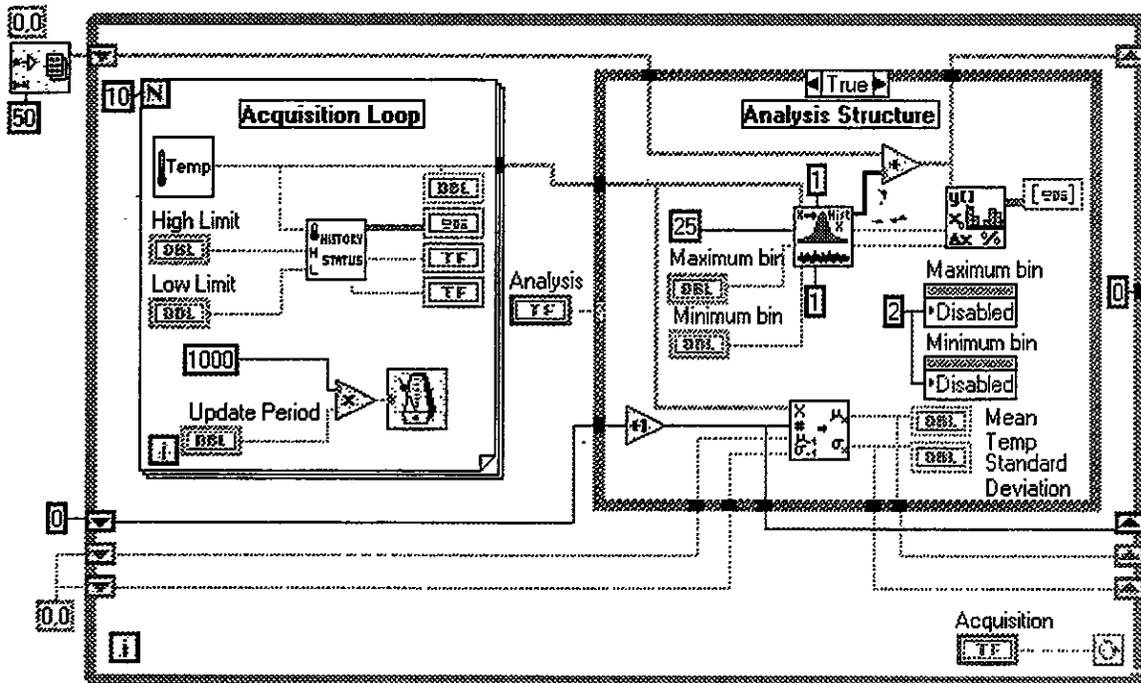
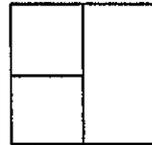


Fig. III.2 Ejemplo de un diagrama de bloques de un instrumento virtual.

- Un diagrama de bloques contiene elementos de entrada/salida, y componentes computacionales, los cuales representan nodos del programa tales como ciclos for, estructuras case, funciones varias y Sub Instrumentos Virtuales (SubIVs) que son representados por iconos y por líneas interconectadas directamente al flujo de los datos. Los componentes entrada/salida se comunican directamente con tarjetas de adquisición de datos o GPIB con instrumentos físicos externos. Los componentes desempeñan funciones aritméticas, así como otras funciones. Los SubIV llaman otros IVs, pasando datos a través de sus iconos/conectores.
- El icono/conector se utiliza para convertir un IV en un objeto que pueda usarse en el diagrama de bloques de otros IVs como subrutinas o funciones. El icono y conector se localizan en la esquina superior derecha del panel del IV. El icono representa gráficamente el IV en el diagrama de bloques de otros IVs. Las *terminales* del conector determinan donde deben alambrarse las entradas y las salidas del icono, éstas son análogas a los parámetros de una subrutina o función y corresponden a los controles o indicadores en el panel frontal del IV.



ICONO



CONECTOR

- Los IVs son jerárquicos y modulares. Estos se pueden usar como programas de nivel superior o como subprogramas con otros programas. A un IV dentro de otro IV se le llama un SubIV. El icono y el conector de un IV trabaja como una lista de parámetros gráficos de tal forma que otros IVs pueden pasar datos a este como un SubIV.

Con estas características, LabVIEW promueve y adhiere al concepto de **programación modular**. Se puede dividir una aplicación en una serie de tareas, que se pueden volver a dividir una y otra vez hasta que una aplicación complicada comience una serie de subtareas simples. Se construye un IV al completar cada subtarea y entonces combinar estos IVs en otro diagrama de bloques para terminar la tarea más grande. Finalmente, el IV de nivel superior contiene una colección de subIVs que representan funciones de la aplicación.

Algunas de las razones por las que hemos decidido usar LabVIEW son:

- Utiliza todos los beneficios que proporciona el ambiente Windows.
- Es interactivo en tiempo de ejecución, al adquirir y presentar datos.
- Permite el almacenamiento de información en archivos para ser compartidos y transportados a hojas de cálculo y procesadores de palabras.
- Facilita el ajuste y medición de los parámetros.
- Nos permite la implantación de equipos de medición reales, eliminando así las tareas de mantenimiento y reparación de los instrumentos de medición electrónicos.

- Permite la construcción sólo de los elementos necesarios para un instrumento de medición, eliminando el hecho de tener que ajustarse a las condiciones de funcionamiento de un instrumento de medición electrónica real.
- Se puede realizar la depuración y mejora de los instrumentos virtuales con el fin de actualizarlos y tener resultados más confiables.

LabVIEW puede ser utilizado por ingenieros, científicos y especialistas que se dediquen al desarrollo de la instrumentación y medición en laboratorios experimentales, utilizando la computadora (hardware y software) como una herramienta que les permita capturar datos desde un sistema físico real con la ventaja de velocidad, precisión y duración del evento (el tiempo de captura puede ser cualquiera) sin la necesidad de estar día y noche supervisando el experimento ya que la computadora puede realizar estas tareas sin ningún problema. Además de que se pueden construir los programas justo a la medida de las especificaciones del usuario dependiendo de las necesidades y recursos particulares.

### *REQUERIMIENTOS PARA USAR LabVIEW*

En el caso particular de la adquisición de datos reales:

- Se necesita una tarjeta de adquisición de datos, de cualquier número de canales de entrada. Con la restricción de que sea compatible con el hardware de la computadora y pueda ser accesada desde LabVIEW (para esto LabVIEW proporciona una lista de las tarjetas manejadas por LabVIEW). En otro caso, si la tarjeta no es reconocida por LabVIEW puede adaptarse mediante una interface desarrollada en lenguaje C++, como es el caso que se presenta en este trabajo de Tesis.
- En caso de una simulación de eventos se pueden utilizar datos aleatorios. Sólo se necesita realizar el programa y utilizar alguno de los iconos de LabVIEW para generar los datos (señales analógicas o digitales)

### *PROGRAMACIÓN EN LENGUAJE G.*

G es el lenguaje de programación de flujo de datos gráfico fácil de usar en el cual LabVIEW está basado. G simplifica la computación científica, proporciona monitoreo y control de los procesos para aplicaciones en pruebas y mediciones, además de una extensa variedad de aplicaciones.

Estos son los conceptos básicos de G.

- ◆ Realiza instrumentos virtuales los cuales se componen de tres partes principales: panel frontal, diagrama de bloques e icono/conector (ya explicados anteriormente).
- ◆ Ciclos y cartas. G tiene dos estructuras de ejecución repetitiva de un subdiagrama, el ciclo While y el ciclo For, ambas estructuras son cajas de tamaño dimensionable. Se coloca el subdiagrama a ser repetido dentro del borde de la estructura del ciclo. La estructura While se ejecuta tantas veces como el valor de la terminal condicional sea verdadero. El ciclo For ejecuta un conjunto el números de veces especificado. Las cartas son usadas al desplegar la dirección de la información en tiempo real al operador.

- ◆ Estructuras Case y secuencia. La estructura Case es una estructura condicional ramificada de control, la cual ejecuta un subdiagrama basándose en una entrada específica. La estructura secuencia, es una estructura de control ramificada condicional, la cual ejecuta un subdiagrama basado en ciertas entradas. Una estructura secuencia, es una estructura de control del programa que ejecuta estos subdiagramas en orden numérico.
- ◆ Nodos atributos, son nodos especiales del diagrama de bloques que pueden usarse al controlar la apariencia y funcionalidad de características de controles e indicadores.
- ◆ Arreglos, Cluster y gráficas. Un arreglo es una colección de elementos de datos del mismo tipo de un tamaño ajustable. Un Cluster es una colección de longitud estática de elementos de datos del mismo o de diferente tipo. Las gráficas son comúnmente usadas al desplegar datos.

### III . 1 LECTURA DE LA TARJETA

En esta parte se analizan las diferentes opciones que LabVIEW proporciona para realizar la adquisición de datos a través de una tarjeta especial para este fin. En este trabajo de tesis no se utiliza una tarjeta de la misma compañía que el software, razón por la cual el software no es capaz de reconocerla automáticamente; pero para proporcionar una información completa de las capacidades que el software posee, a continuación se realiza un análisis detallado de cada una de las opciones que LabVIEW tiene para la adquisición de datos.

El primer paso para la adquisición de datos a través de una tarjeta de adquisición de datos y el software LabVIEW es la configuración de la tarjeta, para lo cual LabVIEW nos proporciona un archivo con las utilerías de configuración para el establecimiento de todos los parámetros necesarios en la adquisición de los datos. Esta utilería llamada WDAQCONF que salva los parámetros de configuración en un archivo llamado WDAQCONF.CFG. Al usar WDAQCONF se asigna el dispositivo, el número del dispositivo y el conjunto de la base de direcciones, el nivel de interrupción y el canal DMA entre otros datos. En nuestro caso al no ocupar una tarjeta de la compañía se realiza un programa de lectura de la tarjeta ProControl PA-CP12 y luego se compila para que LabVIEW pueda acceder a ella, cabe aclarar que el software no la reconoce como dispositivo sino que la lectura y la escritura a la tarjeta se hace por medio de direcciones de memoria.

El programa SETUP copia WDAQCONF al directorio donde se instala LabVIEW. Lo cual se puede hacer desde Windows, ejecutando WDAQCONF con doble click en este icono.

Es importante definir algunos conceptos de los datos utilizados en la configuración de las tarjetas, como lo son:

- ◆ Dispositivo (Device). Es la tarjeta insertable de adquisición de datos, la que puede contener canales múltiples y dispositivos de conversión.

- ◆ Número de dispositivo (Device number). Este es el número de la ranura o el número ID de la tarjeta asignada al realizar la configuración.
- ◆ Canal DMA. Es el canal por el cual se tiene acceso directo a memoria.
- ◆ Interrupción de nivel (Interrupt). Es el nivel de voltaje de la señal la cual indica que la unidad de procesamiento central suspenderá la tarea actual para servir una actividad designada.

Una vez que se ha configurado la tarjeta se procede a salvar los datos de configuración utilizando la opción guardar (Save) del menú archivo (File).

LabVIEW también proporciona la posibilidad de controlar los instrumentos por medio de la tarjeta de adquisición de datos, para lo cual cuenta con archivos manejadores específicos para esta tarea, como lo es el manejador NI-DAQ<sup>6</sup>, el cual es una biblioteca de unión dinámica (*DLL*, Dynamic Link Library) con una interface de alto nivel de hardware para acondicionamiento de señal o dispositivo de adquisición de datos.

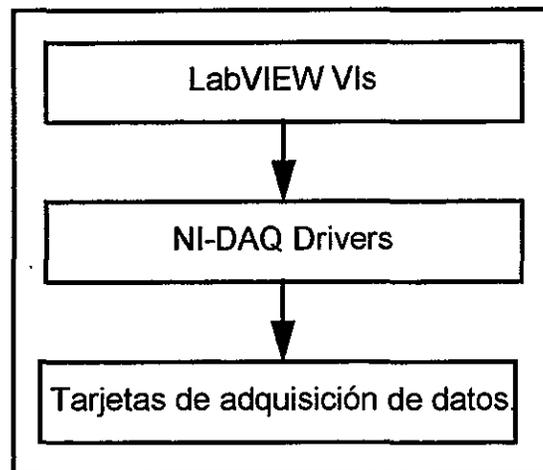


Fig. III.3 Relación de LabVIEW a NI-DAQ y el dispositivo de adquisición de datos.

El manejador NI-DAQ es llamado NIDAQ.DLL. El archivo de configuración (Setup) copia el NIDAQ.DLL en el directorio del sistema Windows.

Después que se ha instalado y configurado el hardware, ya se pueden leer y correr los programas IV de adquisición de datos los cuales tienen varias áreas básicas funcionales de como utilizar los IVs dependiendo del tipo de entrada y salida de la tarjeta que se utilice para la adquisición de datos.

### **ORGANIZACIÓN DE LOS IVs DE ADQUISICIÓN DE DATOS**

La organización de las diversas formas en que LabVIEW provee las opciones para realizar la adquisición de datos se divide en cuatro clases que son:

<sup>6</sup> NI-DAQ: National Instruments Data Acquisition.

- Easy I/O IVs. (Instrumentos virtuales básicos de entrada/salida)
- Intermediate IVs. (Instrumentos virtuales intermediarios)
- Utility IVs. (Instrumentos virtuales de utilerías)
- Advanced Ivs ((Instrumentos virtuales avanzados)

Además de estos cuatro tipos, se cuenta con una serie de utilerías para la adquisición de datos. Cabe mencionar que nosotros no utilizamos ninguno de estos iconos, esto es porque al ser ProControl una tarjeta no reconocida por el software, nosotros creamos un programa y un icono para la lectura de ésta.

### *EASY I/O IVS - ENTRADA/SALIDA BÁSICA DE INSTRUMENTOS VIRTUALES.*

Las opciones de entradas y salidas básicas para instrumentos virtuales ejecutan las más comunes funciones de entrada y salida simples. Estas opciones son muy directas y fáciles de usar y ejecutan automáticamente la verificación de errores, pero tienen capacidades limitadas. Estos IVs tienen muchos usos, en particular son especialmente eficientes para la presentación y demostración rápida de prototipos y pruebas de hardware.

Los " Easy I/O " se componen de IVs intermediarios, los cuales son a su vez componentes de IVs avanzados. Los "Easy I/O IVs" proporcionan una interface básica, utilizando solo las entradas y salidas más comunes. Dentro de esta clase de IVs para la adquisición de datos se cuenta con una diversidad de iconos que realizan funciones específicas, siendo éstas catalogadas en entradas analógicas, salidas analógicas y entradas/salidas digitales, las que se describen a continuación.

También LabVIEW cuenta con opciones de IVs para programación con contadores, exclusivo para tarjetas que contengan el chip 8254. Se mencionan estas opciones como otra alternativa a usarse en la tarjeta ProControl PA-CP12; estos iconos no se utilizan en este caso ya que se realizó un programa en C++ capaz de inicializar el reloj 8254, en el capítulo V se hace referencia a este punto.

### *OPCIONES PARA ENTRADAS ANALÓGICAS (ANALOGIC INPUT, AI)*



**AI Acquire Waveform** (Adquisición de formas de onda de una entrada analógica). Adquiere un número específico de muestras a una velocidad de muestreo especificada desde un solo canal de entrada y muestra los datos adquiridos. Se ejecutan las mediciones de una forma de onda (lecturas de voltaje múltiple en una velocidad de muestreo especificada) con una duración limitada.



**AI Acquire Waveforms** (Adquisición de formas de onda de una entrada analógica). Adquiere datos desde varios canales específicos y con una velocidad de muestreo de registros requerida, ejecuta mediciones de duración limitada de forma de onda desde varios canales de entrada analógica. El número de canales depende de los que contenga la tarjeta.



**AI Sample Channel** (Canal de muestreo de entrada analógica). Mide la señal del canal especificado y regresa el voltaje medido. Este icono ejecuta una medición de duración limitada de un solo canal sencillo.



**AI Sample Channels** (Canales de muestreo de entrada analógica). Ejecuta lecturas de voltaje sencillo desde cada uno de los canales especificados. Este icono mide un voltaje sencillo desde cada uno de los canales de entrada analógica especificados.

### OPCIONES PARA SALIDAS ANALÓGICAS (ANALOGIC OUTPUT, AO)



**AO Generate Waveform** (Generación de forma de onda de salida analógica). Genera un voltaje de forma de onda en un canal de salida analógica a una velocidad de actualización especificada. Este icono genera una forma de onda de voltaje multipunto en un canal de salida analógica específico.



**AO Generate Waveforms** (Generación de forma de onda de salida analógica). Genera voltajes de formas de onda en canales de salida analógica a una velocidad de actualización especificada. Este icono genera formas de onda de voltaje multipunto en canales de salidas analógicas específicos.



**AO Update Channel** (Canal de actualización de información de salida analógica). Escribe un valor de voltaje específico a un canal de salida analógico. Este icono escribe una actualización de información sencilla a un canal de salida analógico.



**AO Update Channels** (Canales de actualización de información de salida analógica). Escribe valores de voltaje a cada uno de los canales de salidas analógicas. Actualiza canales de salidas analógicas con valores de voltaje sencillo.

### OPCIONES PARA ENTRADAS Y SALIDA DIGITAL (DIGITAL INPUT/OUTPUT, DIO)



**Read from Digital Port** (Lectura desde un puerto digital). Lee desde un canal digital que se configure.



**Write to Digital Line** (Escribir a una línea digital). Fija el estado lógico de una línea digital en un canal que se especifique.



**Read from Digital Line** (Lectura desde una línea digital). Lee los estados lógicos de una línea digital de un canal digital se configure.



**Write to Digital Port** (Escribir a un puerto digital). Escribe datos digitales en el puerto digital que se especifique.

### ***INTERMEDIATE IVs - INSTRUMENTOS VIRTUALES INTERMEDIARIOS.***

LabVIEW nos proporciona otros tipos de instrumentos virtuales que son los intermediarios, estos SubIV intermediarios son grupos de IVs avanzados que se usan en algunos de los más poderosos y flexibles IVs avanzados, pero usando menos parámetros. Estos IVs intermediarios están disponibles para las opciones de entradas analógicas, salidas analógicas, contadores, entrada y salida digital, calibración y configuración.

**ENTRADAS ANALÓGICAS.** Esta opción permite adquirir datos desde la tarjeta con canales de entrada analógicos. Los IVs de LabVIEW tienen controles para la entrada e indicadores para salida. Cuando se conectan SubIVs entre sí, se crea una aplicación; los nombres del control son los parámetros de entrada que se pueden ver en la ventana de ayuda y los indicadores son los parámetros de salida.

Dejando el control de canales puesto en 0, significa canal de entrada analógico 0 en la tarjeta de adquisición de datos.

Para entender mejor como se realiza la adquisición de datos es necesario conocer algunos conceptos claves, como son:

- \* Un registro (scan), es una adquisición o lectura desde cada canal en un grupo de entradas analógicas.
- \* El número de registros a adquirir (number of scans), es el número de adquisiciones o lecturas de datos desde cada canal.
- \* La velocidad de registros (scan rate), determina como LabVIEW adquiere datos en muchos tiempos por segundo desde canales, lo cual significa que LabVIEW muestrea cada canal a la velocidad de registro que se cambie.
- \* El grupo (Group), es una colección de canales de entrada, salida o puertos a definir, el cual solo puede contener un tipo de canal ya sea entrada, salida digital o analógica. Se utiliza un número *TaskID* (identificador de tarea) al referirse a un grupo después de crear este.
- \* Task ID. Es el número generado por LabVIEW, el cual codifica el número de dispositivo y el número de grupo después de configurar un grupo, este se usa para especificar las tarjetas o canales en el cual el IV o IVs operan.
- \* Una actualización (update) es una escritura a cada canal o puerto en un grupo de salida.

Dentro del submenú de entradas analógicas LabVIEW cuenta con varias opciones a utilizar para las terminales de entrada del IV que se mencionan a continuación.

### OPCIONES PARA ENTRADAS ANALÓGICAS (ANALOGIC INPUT, AI)



**AI Config** (Configura entrada analógica). Este icono permite seleccionar el dispositivo (tarjeta de adquisición de datos) y define un grupo para este dispositivo que contiene el canal elegido en el arreglo de *canales*. Estos controles del panel frontal son alambrados a las correspondientes terminales de entrada a AI Config, produce una salida *taskID* y un error cluster. Todos los otros IVs de entrada analógica toma el *taskID* como entrada al identificar el dispositivo y el grupo de canales en el cual opera. Ya que el *taskID* es una entrada y una salida a los otros IVs, este parámetro también forma una dependencia de datos que controla la ejecución del diagrama de flujo.

El error cluster contiene la información del estado de error. El parámetro del número de registros a adquirir es conectado a la terminal del tamaño de buffer de AI Config. Este icono asigna un buffer de adquisición del tamaño dado a ocupar las muestras de datos. El parámetro del tamaño del buffer esta en unidades de registros.



**AI Start** (Inicio de entrada analógico). Este icono inicia la adquisición de datos desde la tarjeta con una velocidad de registro específica. Por omisión, el AI Start IV adquiere los datos suficientes para llenar el buffer de adquisición del AI Config IV asignado, el cual fue previamente especificado. Después de que el buffer de adquisición de la tarjeta ha sido llenado, esta adquisición de datos se detiene.



**AI Read** (Lectura de entrada analógica). La única terminal de entrada que se conecta es el *taskID*. Por omisión, el AI Read recupera todos los datos que el AI Start IV adquiere. El AI Read IV espera hasta que el dato esta disponible, copia este desde el buffer de adquisición, escala los datos a volts, y regresa un arreglo de dos dimensiones de voltajes al diagrama de LabVIEW. La primera dimensión del arreglo indexa los registros, y la segunda dimensión indexa los canales con cada registro. Se tiene que transformar el arreglo al poner la dimensión en el formato correcto para una gráfica deseada en el AI Read IV.



**AI Clear** (Limpiar entrada analógica). Este icono designa el buffer de adquisición y libera cualquier fuente del sistema usado por la adquisición, tal como los canales DMA y niveles IRQ (interrupciones). Para detener una adquisición asociada con el *taskID* se usa el AI Control.



**AI Single Scan** (Lectura de un registro sencillo). Regresa un registro de datos desde un grupo de canales previamente configurados.

### OPCIONES PARA SALIDA ANALÓGICA (ANALOGIC OUTPUT, AO)

Para poder hacer uso de las opciones de salida analógica que proporciona el software LabVIEW es necesario que inicialmente se asegure que la tarjeta de adquisición de datos cuente con un módulo de salida analógica. Posteriormente se debe asegurar que

el número del dispositivo corresponda a una tarjeta de adquisición de datos configurada para salida analógica, y así poder iniciar la captura de los datos.

Los IV de salida analógica generan una onda senoidal de 10Hz con un rango de pico a pico de -1.0 V a +1.0 V en el canal de salida analógica 0. El indicador de ciclo finalizado se incrementara en un valor al mismo tiempo que el IV corra.

A continuación se describen brevemente las opciones que LabVIEW proporciona para salidas analógicas.



**AO Config** (Configuración de salida analógica). Selecciona el dispositivo a usar, el canal de salida y usa la entrada con el tamaño del buffer para asignar un buffer al introducir la forma de onda. La función Build Array (construir arreglo) convierte una cadena de canal en un arreglo. Este arreglo de canales es utilizado mientras se asigna un límite de salida diferente a cada canal. Los límites son los voltajes máximo y mínimo que el canal puede producir. El buffer asignado por el AO Config se ocupa con el dato de forma de onda que va a ser generado. Se llama a este buffer *interno* porque el usuario no tiene acceso directo a esté.

El AO Config produce como salida un taskID y un error cluster. El taskID es un número de referencia que identifica el dispositivo y estos contadores al IV funcionando después en el flujo de datos.

El error cluster describe el estado de error del IV. Este contiene un estado *Booleano* (TRUE para un error), un código de error entero de 32 bits y una cadena que identifica donde el error ha ocurrido.



**AO Write** (Escritura a Salida analógica). Copia del buffer la onda senoidal al buffer interno asignado por el AO Config. Este icono requiere un buffer de dos dimensiones para el manejo de formas de onda multicanal si es el caso. Modificando este IV al generar una forma de onda en el canal 1 además del canal 0 después.



**AO Start** (Iniciar una salida analógica). Pone el número de iteraciones en buffer, pone la velocidad actual y comienza la forma de onda. Cuando LabVIEW genera el dato en el buffer una sola vez, este ejecuta una iteración en el buffer. El hardware ejecuta un proceso en dos pasos cuando se escriba un voltaje a un convertidor digital/analógico. Primero, este describe un modelo digital al CAD. Entonces el hardware envía un pulso al CAD, el cual propicia la ejecución en la conversión digital/analógica y produce el voltaje



**AO Clear** (Limpia salida analógica). Se ejecuta por ejemplo al finalizar un ciclo While. Este detiene la generación de formas de onda (si esté esta corriendo totalmente), desasigna el grupo, y libera el buffer de forma de onda interna creada por el AO Config. Pero en su ejecución esencial el AO Clear, limpia la tarea de salida analógica asociada con el taskID.



**AO Wait** (Espera salida analógica). Se pone en estado de espera hasta que las tareas de generación de forma de onda estén completas después regresa. Este icono se usa al esperar un almacenamiento a buffer, la generación de onda de forma finita verifica antes de la llamada del AO Clear. El AO Wait verifica el estado de la tarea en intervalos regulares por llamado al AO Write y verificando la generación de salida completa. El AO Wait espera asincrónicamente entre intervalos libres del procesador para otras operaciones. El IV calcula el intervalo de espera por división de la entrada **check every N updates** (verificando cada N actualización) por la velocidad de actualización. No se debe de usar el AO Wait cuando se generen datos continuamente, porque la generación nunca finaliza.

### OPCIONES PARA ENTRADAS Y SALIDAS DIGITALES

Para utilizar estas opciones es necesario contar con una tarjeta insertable que contenga entrada y salida digital. LabVIEW proporciona diversos iconos para realizar la captura digital de datos en diversas formas. Todas las tarjetas que soporten entrada/salida digital pueden ejecutar entrada/salida digital inmediata, pero solo ciertas tarjetas pueden desempeñar entrada/salida digital con reconocimiento de protocolo de comunicación.

A continuación se describirán brevemente algunos iconos que LabVIEW provee para este tipo de captura de datos.



**DIO Clear** (Limpia entrada/salida digital). Llama al Digital Group Buffer Control IV (Control de buffer del grupo digital del IV) al detener una transferencia y limpiar el grupo.



**DIO Config** (Configura entrada/salida digital). Llama al Digital Group Config (Configuración del grupo digital) avanzado del IV al asignar una lista de puertos al grupo, estableciendo la dirección del grupo y producir el taskID. El IV entonces llama al modo de configuración digital del IV para establecer los parámetros con reconocimiento de protocolos de comunicación el cual solo afecta la operación de las tarjetas tipo DIO-32F (la cual es una tarjeta de National Instruments). Finalmente, el IV llama al Buffer Digital Config IV a asignar un buffer y detener los registros que son leídos o las actualizaciones que van a ser escritas.



**DIO Read** (Lectura entrada/salida digital). Llama al Digital Buffer Read IV para leer datos desde el buffer de transferencia interna y regresa el dato leído en el modelo.



**DIO Single Read/Write** (Lectura/escritura sencilla de entrada/salida digital). Lecturas o escrituras de datos digitales a los puertos especificados en la lista de puerto. Este sencillo IV configura y transfiere datos.



**DIO Start.** (Inicio de entrada/salida digital). Inicia la operación de entrada/salida digital almacenado a buffer. Este IV llama el Digital Clock Config IV al poner la velocidad del reloj si el reloj interno produce la señal con reconocimiento de protocolo de comunicación, y entonces empieza la transferencia de datos por llamado del Digital Buffer Control IV.



**DIO Wait** (Espera de entrada/salida digital). Espera hasta que la entrada digital almacenado a buffer o la operación de salida se complete antes de regresar. Para entrada, el IV detecta la terminación cuando el indicador de generación completa (generation complete) del DIO Write IV es verdadera.



**DIO Write** (Escritura de entrada/salida digital). Llama el Digital Buffer Write IV para escribir al buffer de transferencia interna.

#### OPCIONES DE CONTADORES.



**ICRT Control.** Este IV controla contadores en tarjetas que contengan el chip 8254.

#### UTILITY IVs - INSTRUMENTOS VIRTUALES DE UTILERIAS.

Estas opciones de instrumentos virtuales se proporcionan para la solución de problemas comunes encontrados en la adquisición de entradas y salidas analógicas.

#### OPCIONES PARA ENTRADAS ANALÓGICAS.

Estas opciones son instrumentos virtuales proporcionados para solucionar problemas comunes en la entrada analógica, estos IVs son útiles, pero poco flexibles, son construidos desde IVs de Entrada Analógica Intermediaria.



**AI Continuous Scan** (Registro continuo de entrada analógica). Realiza continuamente mediciones en tiempo de muestreo de un grupo de canales, almacena los datos en buffer circular, y regresa un número especificado de mediciones de registros en cada llamada. Esta opción trabaja indefinidamente, por lo cual se debe conectar la iteración del IV a la terminal de iteración de un ciclo While.



**AI Read One Scan** (Lectura de un registro de entrada analógica). Mide las señales en los canales especificados y regresa la medición en un arreglo de valores binarios o voltajes. Ejecuta inmediatamente la medición de un grupo de uno o más canales.



**AI Waveform Scan** (Registro de forma de onda de entrada analógica). Adquiere el número de registros especificados a una velocidad de registro específica y regrese todos los datos adquiridos. También se puede iniciar la adquisición por medio de un disparo.

### *OPCIONES PARA SALIDAS ANALÓGICAS.*

Los IVs de utilerías para salidas analógicas son instrumentos virtuales generados para solucionar problemas comunes de salidas analógicas, estos son muy útiles, pero tienen poca flexibilidad, y son construidos desde IVs intermediarios de salidas analógicas.



**AO Continuos Gen** (Generación de forma de onda continua de salida analógica). Genera una forma de onda continua, de duración limitada, tiene un buffer circular para los canales de salida asignados en una velocidad de actualización especificada. El IV actualiza el buffer de salida continuamente así como este genera los datos



**AO Waveform Gen** (Generación de formas de onda de salida analógica). Genera una forma de onda de duración limitada con almacenamiento simple, o almacenado a buffer circular para los canales de salida dados en la velocidad de actualización especificada. A menos que se ejecute una generación infinita, el IV regresa el control a el diagrama de LabVIEW solo cuando la generación se ha completado.



**AO Write One Update** (Actualización de escritura a salida analógica). Escribe un valor de voltaje sencillo a cada uno de los canales de salida analógicos especificados.

### **ADVANCED IVs - INSTRUMENTOS VIRTUALES AVANZADOS.**

Los IVs avanzados forman la construcción de bloques para todos los otros IVs de adquisición de datos. Los IVs avanzados son más flexibles y extremadamente poderosos, son tan fáciles de usar como los IVs intermediarios. Si se construye una aplicación usando IVs avanzados en lugar de IVs intermediarios, se deberán utilizar algunos IVs más, así como algunas entradas adicionales, pero el concepto fundamental es el mismo.

El parámetro de velocidad de lecturas o registros habilita el intervalo de muestreo en tarjetas que soportan un intervalo de muestreo (el intervalo más grande entre registros que incluyen un registro entre canales individuales). El parámetro de velocidad de reloj del canal define el tiempo entre la adquisición de muestras consecutivas.

### *OPCIONES PARA ENTRADAS ANALÓGICAS.*

Los IVs son la interface del software de adquisición de datos NI-DAQ y son los fundamentos de los IVs de Easy Input/Output y los intermediarios. En seguida se describen brevemente los instrumentos virtuales para entrada analógica avanzada.



**AI Buffer Config** (Configuración de buffer de entrada analógica). Asigna memoria para LabVIEW al almacenar datos de entrada analógica hasta que el AI Buffer Read pueda liberar estos



**AI Buffer Read** (Lectura de buffer de entrada analógica). Regresa un dato de entrada analógica desde el buffer interno de datos.



**AI Clock Config** (Configuración de reloj de entrada analógica). Pone el canal y registra la velocidad del reloj.



**AI Control** (Control de entrada analógica). Controla las tareas de entrada analógica y especifica la cantidad de datos a adquirir.



**AI Group Config** (Configuración de grupo de entrada analógica). Define que canales corresponden a un grupo y asignados a ellos.



**AI Hardware Config** (Configuración de hardware de entrada analógica). Configura cualquiera de los límites de entrada más alto y más bajo o el rango, polaridad y ganancia. También configura el acoplamiento, modo de entrada y número de tarjeta. La utilería de configuración determina el marco por omisión para los parámetros de este IV. Se pueden definir diferentes límites para cada canal en particular del mismo grupo, se puede ejecutar el muestreo de uno a uno de los canales.



**AI Single Scan** (Registro sencillo de entrada analógica). Regresa un registro de datos. Si se inicializa una adquisición de datos con el AI Control, este IV lee un registro de los datos desde el buffer interno, o desde la tarjeta *FIFO*, si la adquisición no se almacena a buffer. La configuración del grupo determina los canales de las muestras del IV.



**AI Trigger Config** (Configuración de disparo de entrada analógica). Configura las condiciones de disparo para inicializar el registro, el reloj de canal y el contador de registros.



**AI Parameter** (Parámetros de entrada analógica). Configura y recibe una miscelánea de parámetros asociados con entradas analógicas de cualquier operación de un dispositivo que no es cubierta con otro IV de entrada analógica.

#### OPCIONES AVANZADAS PARA SALIDAS ANALÓGICAS.



**AO Group Config** (Configuración de grupo de salida analógica). Asigna una lista de canales de salida analógica al un número de un grupo y produce el taskID que todos los demás IVs de salida analógica utilizan.



**AO Single Update** (Actualización sencilla de salida analógica). Ejecuta una actualización inmediata de los canales en el grupo.



**AO Clock Config** (Configuración de reloj de salida analógica). Configura una actualización o intervalo de reloj para salida analógica.



**AO Hardware Config** (Configuración de hardware de salida analógica). Configura el nivel de voltaje de referencia, polaridad de salida y la unidad de medición para el dato de un canal dado (volts o miliamperes). Este IV siempre regresa la referencia original para todos los canales en el grupo.



**AO Buffer Config** (Configuración de buffer de salida analógica). Asigna memoria para un buffer de salida analógica. Si se están usando interrupciones, se pueden asignar una serie de buffers de salida analógica y se asignan a un grupo por llamada del AO Buffer Config IV en tiempos múltiples. Cada buffer puede tener su propio tamaño. Si se está usando DMA, se asignará solo un buffer.



**AO Control** (Control de salida analógica). Inicia, pone en pausa, reanuda y limpia las tareas de salida analógica.



**AO Buffer Write** (Escribir al buffer de salida analógica). Escribe los datos de la salida analógica a los buffers creados por el AO Buffer Config IV. También empieza la escritura en la actualización en donde los puntos marcan la escritura. Después de una escritura, los puntos de marca en la actualización, siguen la última actualización escrita.



**AO Trigger and Gate Config** (Configuración de disparo y puerta de salida analógica). Configura las condiciones de disparo y puerta para operaciones de salida analógica en dispositivos series E (específico para tarjetas National Instruments).



**AO Parameter** (Parámetros de salida analógica). Coloca miscelánea de parámetros asociados con la operación de salida analógica de los dispositivos que no son cubiertos con otros IVs de salida analógica.

### OPCIONES AVANZADAS PARA ENTRADA Y SALIDA DIGITAL

El uso de los puertos digitales de los IVs se utilizan para lecturas inmediatas y escrituras a líneas y puertos digitales. Se pueden usar el grupo digital de IVs para entrada y salida inmediata, con reconocimiento de protocolo de comunicación y cronometrado para puertos múltiples. Estos IVs son la interface al software de adquisición de datos NI-DAQ y los fundamentos de los IVs Fáciles e Intermediarios de entrada/salida.



**DIO Port Config** (Configuración de puerto de entrada/salida digital). Establece la configuración de un puerto. Se puede usar el taskID que este IV regresa solo en IV de puertos digitales.



**DIO Port Read** (Lectura de puerto de entrada/salida digital). Lee el puerto identificado por el taskID y regresa la lectura modelo en la variable pattern.



**DIO Port Write** (Escritura de puerto de entrada/salida digital). Escribe el valor en la variable `pattern` al puerto identificado por `taskID`.

### OPCIONES PARA ENTRADA Y SALIDA DE GRUPO DIGITAL.

Los IVs de grupo digital ejecutan entradas/salidas inmediatas con reconocimiento de protocolo de comunicación o con cronometro.



**Digital Buffer Config** (Configuración del buffer digital). Asigna memoria para una entrada digital o buffer de salida.



**Digital Buffer Control** (Control del buffer digital). Inicia una operación de entrada o salida.



**Digital Buffer Read** (Lectura del buffer digital). Escribe datos de salida digital al buffer creado por el Digital Buffer Config IV. La escritura siempre comienza en el marco de escritura. Después de una escritura, los puntos de marco de escritura siguen la última actualización escrita.



**Digital Clock Config** (Configura el reloj digital). Configura una tarjeta al producir señales con reconocimiento de protocolo de comunicación en la salida de un reloj para entrada/salida digital de duración limitada.



**Digital Group Config** (Configuración de grupo digital). Define un grupo de entrada o salida digital. Se puede usar el `taskID` que este IV regresa solo en el grupo digital de IVs.



**Digital Mode Config** (Modo de configuración digital). Configura las características de reconocimiento de protocolos de comunicación para las tarjetas que tienen esta capacidad.



**Digital Single Read** (Lectura simple digital). Lee el puerto que corresponde a el grupo identificado por el `taskID` y regresa el modelo de lectura.



**Digital Single Write** (Escritura simple digital). Escribe el dato en el `pattern array` (arreglo de modelos) al puerto que corresponde en el grupo identificado por el `taskID`.



**Digital Buffer Write** (Escritura digital a buffer). Escribe datos de salida digital al buffer creado por el mismo icono.



**Digital Buffer Config** (Configuración digital del buffer). La escritura siempre se inicia en la marca de escritura, después del punto de marca se coloca la última actualización de escritura.



**DIO Parameter** (Parámetros de entrada/salida digital). Configura y recibe una miscelánea de parámetros asociados con entrada y salida digital que no son configurados por otros IVs de entrada/salida digital.



**Digital Trigger Config** (Configuración de disparo digital). Configura la condición de disparo para iniciar y/o detener la operación de generación de modelos digitales. Este IV solo es valido cuando el Digital Clock Config IV tiene este parámetro en el origen de reconocimiento de protocolo activado en los valores de 1 o 4.

#### OPCIONES PARA CONTADOR.



**ICRT Control.** Controla contadores en tarjetas con dispositivos 8254.

### TIPOS DE ENTRADA Y SALIDA POR SOFTWARE

Dentro de la adquisición de datos mediante el uso del software LabVIEW se cuenta con tres tipos de entrada/salida que son: almacenamiento inmediato sin uso del buffer (Immediate nonbuffered), almacenamiento de duración limitada sin uso del buffer (timed nonbuffered) y almacenamiento al buffer con tiempo limitado (timed buffered).

#### ALMACENAMIENTO INMEDIATO SIN BUFFER (IMMEDIATE NONBUFFERED) DE ENTRADA/SALIDA

Es una operación inmediata de entrada/salida de lectura de un registro simple o escritura de una actualización sencilla a un grupo de canales cuando se ejecuta un IV de registro sencillo o actualización sencilla. Una operación de entrada/salida inmediata no usa un reloj de registro o actualización. El reloj del canal para operaciones de entrada analógica provee un hardware de tiempo limitado para retardo intercanal en los canales con un registro. Si se necesita controlar la velocidad de registro, se usan las funciones Wait (Espera) o Wait Until Next Interval (espera hasta el siguiente intervalo) o se puede usar una operación de entrada/salida de tiempo limitado. Las funciones anteriores no son tan precisas como entrada/salida de hardware de tiempo limitado

### **ALMACENAMIENTO DE TIEMPO LIMITADO SIN BUFFER (TIMED NONBUFFERED) DE ENTRADA/SALIDA.**

Para entrada/salida con almacenamiento a buffer de tiempo limitado, LabVIEW corre una operación de registro o actualización en segundo plano a la velocidad requerida, pero usa solo el FIFO u otra memoria para el buffer de datos en la tarjeta de adquisición de datos.

Entonces, cuando se ejecuta un registro sencillo o actualización, LabVIEW lee uno de los dos datos, el dato más nuevo o el más viejo, desde el FIFO. Se puede usar este tipo de operación cuando se necesite precisión, baja velocidad de muestreo y no se desee una elevada operación de buffer extra. Un uso típico para este tipo de adquisición es un proceso en ciclo de control o monitoreo. El hardware opera sobre una base de regulación de tiempo precisa, así el IV no tiene un ciclo de tiempo

### **ALMACENAMIENTO DE TIEMPO LIMITADO CON BUFFER (TIMED BUFFERED) DE ENTRADA/SALIDA.**

Esta opción se utiliza si se requiere precisión de hardware con duración limitada para operaciones de entrada/salida, o si la velocidad de actualización o de registro son también rápidas para el IV al leer o escribir un registro o actualización en un tiempo, se deberá usar el almacenamiento de entrada/salida a buffer. Con almacenamiento de entrada/salida a buffer, LabVIEW transfiere el dato en intervalos de tiempo o desde un buffer de memoria al dispositivo de conversión para salida. Para adquisición almacenada a buffer de duración limitada, se especifican las condiciones de reloj y de disparo. Hay dos tipos de almacenamiento de entrada/salida a buffer, que son: almacenamiento de entrada/salida con buffer simple y almacenamiento de entrada/salida con buffer circular.

El almacenamiento de entrada/salida con buffer simple utiliza un buffer sencillo con la memoria suficiente para todos los datos. LabVIEW transfiere datos dentro o fuera de este buffer a la velocidad especificada, comenzando en el inicio del buffer y parando al fin del buffer. Se utiliza almacenamiento de entrada/salida con buffer sencillo al adquirir pequeñas cantidades de datos relativos a los límites de memoria. Para operaciones de almacenamiento de entrada con buffer sencillo, se pueden leer todos los datos de una sola vez después que LabVIEW adquiere el dato, o se pueden leer pequeñas porciones de datos mientras la adquisición esta en progreso. Estos datos pueden ser leídos de acuerdo a las necesidades, porque el dato permanece en el buffer hasta que se limpia la tarea de adquisición. Se pueden especificar las condiciones de disparo al iniciar la adquisición almacenando a un buffer simple.

Para operaciones de salida almacenado en buffers simples, se pueden escribir todos los datos en el buffer antes de comenzar la generación, o escribir pequeñas porciones de los datos mientras la generación esta en progreso. De cualquier forma, el buffer se llena más rápido de lo que LabVIEW transfiere la salida de los datos.

En la figura III.4 se representa una operación de almacenamiento de entrada/salida con un buffer simple. N es el número de registros o actualizaciones que el buffer puede soportar, T es la ocurrencia de disparo si se especifica un disparo, o el inicio de la operación si no existe un disparo.

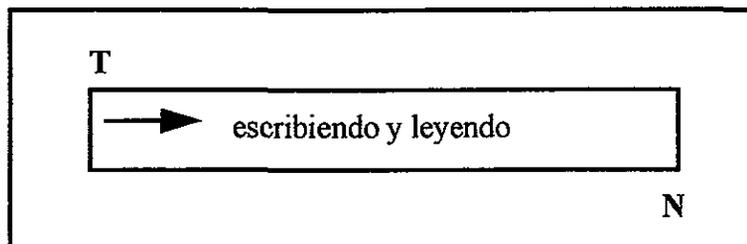


Fig. III.4 Buffereo simple de entrada/salida.

El almacenamiento de entrada/salida con buffer circular lee o escribe más datos que puedan llenar a un buffer sencillo. Esto es, LabVIEW usa el buffer en forma circular. Cuando LabVIEW llega al fin del buffer, este regresa al inicio del buffer y continua la transferencia de datos. Si el número de registros que se requieren adquirir esta fijo o excede el tamaño del buffer, se debe especificar la operación continua y entonces detener la adquisición hasta que se reciba la cantidad especifica de registros. Para el almacenamiento de entrada/salida con buffer circular se debe leer o escribir el dato mientras la operación esta en progreso.

De otro modo, una operación de entrada sobreescribirá el dato en el buffer mientras el segundo ciclo inicia, y una operación de salida transfiere el dato viejo desde el buffer durante este segundo ciclo. Se pueden especificar condiciones de disparo al iniciar la adquisición almacenando en un buffer circular. La figura III.5 representa una operación de entrada/salida almacenado a un buffer circular.

En LabVIEW, el control del flujo de los datos va en el orden de la ejecución del IV. Cuando se ejecutan los IVs de adquisición de datos ellos reciben el taskID y el valor de error.

### TASK CONTROL IV

El control de los instrumentos son IVs analógicos avanzados e intermediarios que arrancan, pausan, reanudan y limpian la operación de entrada/salida para una tarea dada. Un controlador de tareas (task) es un reloj guía en una operación de adquisición o una generación que comienza cuando se inicializa el reloj. Al iniciar una operación el AI Control IV arma la adquisición o generación, la cual entonces espera para la condición de disparo, si alguna, ocurre antes de inicializar. Si no se especifica un disparo, o si se salva el dato a predisparo, la operación comienza inmediatamente. Al poner en pausa una operación, el Control del IV detiene la adquisición o generación hasta que se reanude está. Al limpiar la operación, el control IV para la operación y libera el buffer de memoria.

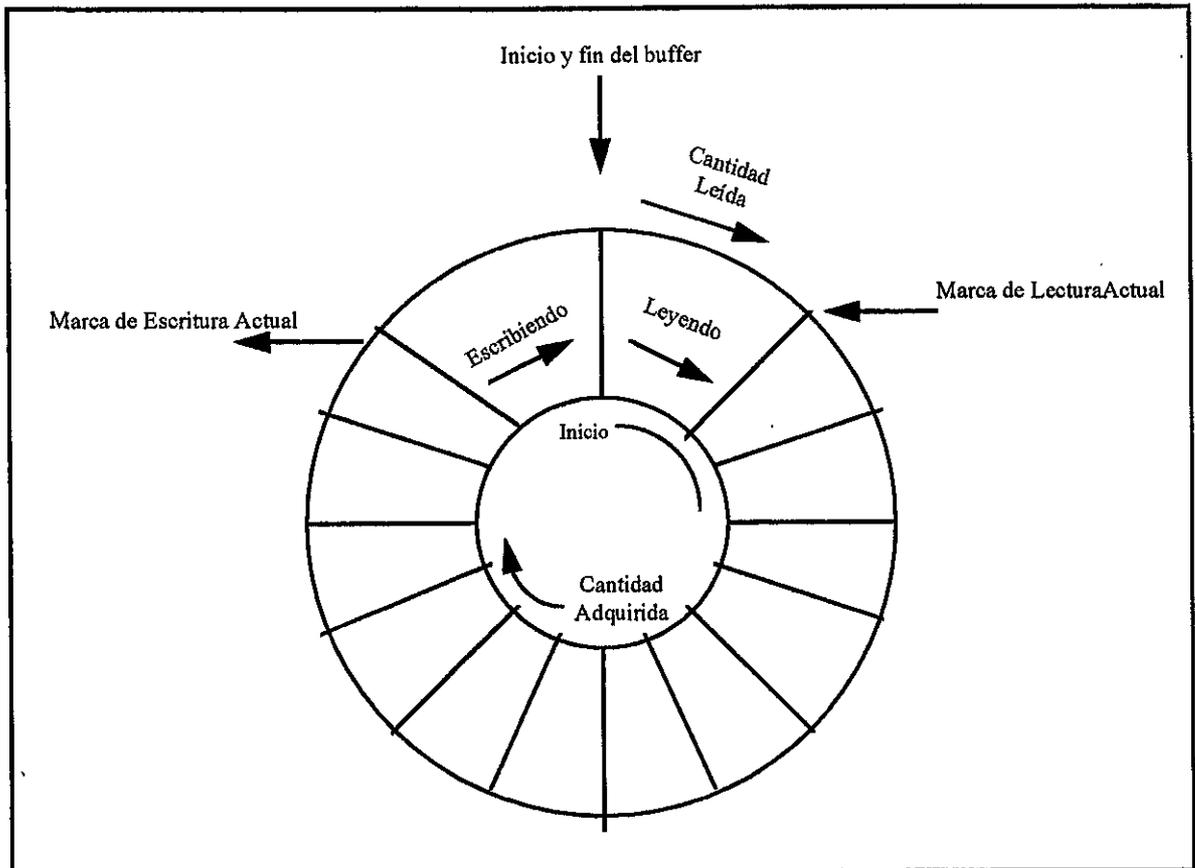


Fig. III.5 Búfer circular de entrada/salida.

Para operaciones de entrada almacenando al búfer, hay que asignar el búfer con una configuración IV, arrancar la adquisición con un control IV, leer el dato desde el búfer con una lectura IV, limpiar el task y liberar el búfer con un control IV. Para operaciones de salida almacenando a búfer, se deberá asignar el búfer con una configuración IV, escribir el dato al búfer con un IV de escritura, arrancar la generación con un control IV. Es posible restaurar una operación de Entrada/salida muchas veces, usando el mismo búfer de memoria y configuración, antes que finalmente este se limpie. Usando la misma configuración se evita la saturación de asignación del búfer repetidamente.

En este trabajo no se utiliza ninguno de los iconos mostrados para la lectura de la tarjeta, pero se proporciona la información anterior como otra alternativa que el propio software dispone para el manejo de datos con tarjetas de la misma compañía.

Para la lectura analógica se creó un icono que inicializa el reloj 8254 y lee los canales analógicos de la tarjeta ProControl PA-CP12, esto con la finalidad de ocupar un solo icono que pudiera hacer estos dos procesos. Para la escritura digital se creó un icono capaz de escribir datos al canal de escritura digital de la tarjeta.

### III . 2 COMPILADOR DE LABVIEW

LabVIEW es un software de programación gráfica orientado al control de instrumentos. En términos computacionales se puede comparar con lenguajes de programación, tales como Visual Basic, o Visual C; con una diferencia muy importante, LabVIEW no es lenguaje de programación basado en texto, es decir, el código del programa no está escrito textualmente, sino que su programación se basa en iconos y enlaces entre ellos. Este lenguaje es llamado lenguaje G.

Al ser LabVIEW un lenguaje de programación necesita de un proceso de compilación en los programas para realizar la depuración y supresión de posibles errores. LabVIEW compila automáticamente los IV antes de correrlos por primera vez y después de cualquier alteración que se produzca en el mismo.

Para realizar la ejecución de los programas en LabVIEW, éste cuenta con diversas barras de opciones, las cuales varían dependiendo de las herramientas que posean.

LabVIEW, tiene dos elementos básicos, el panel frontal y el diagrama de bloques, en ambos se cuentan con diversas herramientas para realizar los programas que a continuación se muestran.

La barra de herramientas desde el panel frontal, tiene las opciones de ejecutar un IV una sola vez o continuamente, es decir, si se selecciona la primera flecha que se muestra en la figura III.6 (de izquierda a derecha), el programa se ejecuta completo solo una vez, pero si se requiere realizar ejecuciones continuas es necesario oprimir el segundo icono que tiene dos flechas en un ciclo. También se cuenta con la opción de detener la ejecución o ponerla en pausa momentánea.



Fig. III.6 Barra de herramientas del panel frontal.

La barra de herramientas desde el diagrama de bloques, permite ejecutar el programa con la opción de visualizar la forma en que los datos pasan a través del programa, encendiendo el icono con un foco.



Fig. III.7 Barra de herramientas del diagrama de bloques.

**ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA**

Cuando se ejecuta un IV ya sea desde el panel frontal o el diagrama de bloques la barra de herramientas se modifica respectivamente de la siguiente manera.



Fig. III.8 Barras de herramientas cuando el IV esta en ejecución.

Si un IV no se encuentra listo para ejecutarse, se dice que esta "fuera de servicio" (break), lo cual se puede observar cuando la flecha para correr el programa esta rota, como se muestra en la figura III.9. Al dar un click en la flecha rota aparece una ventana llamada lista de errores "Error list".



Fig. III.9 Barra de herramientas cuando el IV tiene algún error.

En la figura III.10 se muestra la ventana de errores, donde se muestran los errores que se encontraron en el programa y la posible solución a estos.

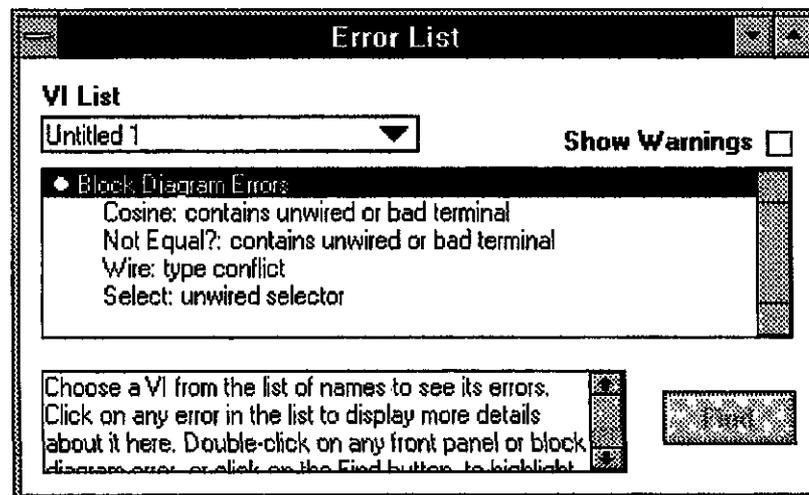


Fig. III.10 Lista de errores cuando un IV tiene algún error.

### COMPILACIÓN DE BIBLIOTECAS

Una biblioteca es un conjunto de IVs comprimidos (esto es solo para ahorro de espacio en disco), es un directorio que contiene diversos archivos, de los cuales solo LabVIEW puede ver su contenido. Por ejemplo, se puede crear una biblioteca de gráficas dentro de las cuales se tiene una variedad de IVs con diferentes tipos de gráficas, o tal vez una biblioteca llamada "Filtros" en donde existen IVs que simulen una gran variedad de filtros.

Para crear bibliotecas, primeramente se crea el IV, después se guarda seleccionando la opción editar biblioteca de un IV (Edit VI Library) del menú archivo (File), donde se puede tener una gran cantidad de IVs que ocupan un espacio menor, comparado con el que ocuparían si se salvan cada uno por separado.

Dentro de la compilación, también se pueden compilar bibliotecas completas, esto se realiza con la opción Mass Compile del menú archivo, la cual compila todos los IV que se encuentran en una biblioteca.

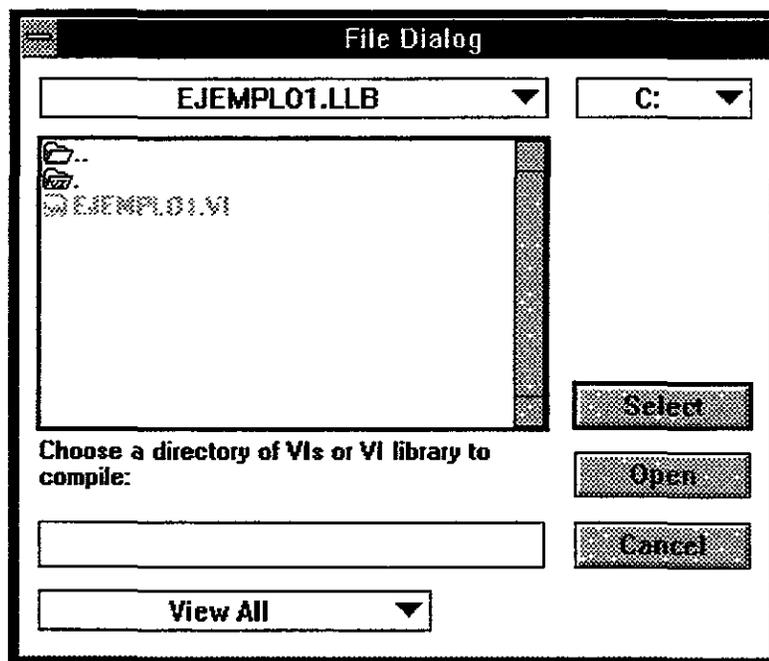


Fig. III.11 Pantalla de la opción *Mass Compile*

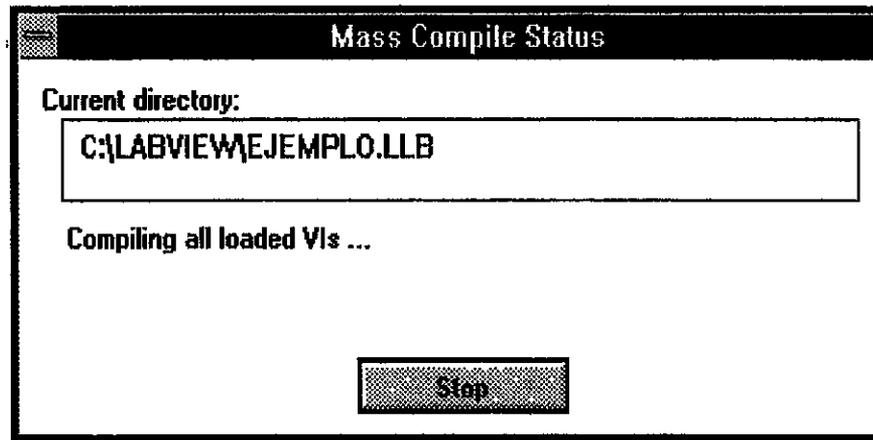


Fig. III.12 Pantalla del proceso de compilación de bibliotecas

## CREACIÓN DE PROGRAMAS EJECUTABLES

Dentro de las utilerías del compilador de LabVIEW existe la opción llamada Build Application (Construir aplicación). Esta opción se adhiere al instalar el paquete complementario LabVIEW Application Builder (constructor de aplicaciones LabVIEW), para darle a LabVIEW la capacidad de crear programas ejecutables e independientes del mismo ambiente. Cuando se crea un programa ejecutable se puede distribuir el programa sin necesidad que el usuario tenga el software LabVIEW.

Los requerimientos del sistema para usar la construcción de aplicaciones, son las mismas que para el software LabVIEW. Lo único que podría cambiar sería la memoria necesaria, pero esto depende del tamaño de la aplicación. Existen dos tipos de bibliotecas de aplicaciones, una para Windows 3.1, 3.11 o 95 y otra para Windows NT. Si se necesitan correr las aplicaciones en ambos sistemas, se deben construir dos aplicaciones diferentes.

Las aplicaciones de LabVIEW pueden correr en una computadora 386 con coprocesador, pero se recomienda una computadora 486. Las aplicaciones elaboradas para Windows NT pueden correr sin coprocesador, aunque la ejecución puede ser de bajo rendimiento. Se recomienda que la computadora cuente por lo menos con 8 megabytes de RAM.

Al instalar este software dentro del menú Project aparecerá la opción Build Application, y dentro del directorio de LabVIEW que contiene los archivos necesarios para poder realizar programas ejecutables.

Cuando se anexa un IV, no se necesita insertar cada uno de los IVs. Se puede usar un control de IVs que dinámicamente llama a los instrumentos que se encuentran incrustados en la aplicación.

La principal característica de una aplicación ejecutable se basa en que es una interface que permite solamente la operación del IV. Los menús no contienen opciones de edición. La opción de guardar, funciones y controles no están presentes. Los usuarios no pueden modificar los IVs o ver los diagramas de edición. Los menús que se despliegan solamente son los relacionados con la operación del IV. Las opciones disponibles al usuario son las siguientes:



Fig. III.13 Barra de herramientas para un programa ejecutable

- Operar los controles y cambiar los valores
- Interacción con la tira de caracteres y los indicadores de gráfica.
- Cambiar los límites de escala.
- Fijar los controles, indicadores, y arreglos de elementos con valores por omisión.
- Usar las opciones de cortar, copiar y pegar datos desde un indicador de control a otro control.
- Se puede ver los menús de los controles e indicadores para ver la descripción del control, así como el funcionamiento adicional de las operaciones, tales como el despliegue de la barra de control de las gráficas.
- Impresión del panel principal.
- Obtener información acerca del instrumento virtual (Get Info)
- Usar la ventana de ayuda para ver la descripción de controles e indicadores.

### *CÓMO CONSTRUIR UNA APLICACIÓN EJECUTABLE*

Para la construcción de la aplicación se deben tener en cuenta el IV como los SubIV que éste contiene, así como subrutinas a las que hace llamadas. Existen dos formas crear una aplicación.

Una es haciendo el IV ejecutable, es decir sin incluirlo en una biblioteca en este caso al correr el IV, se preguntará qué IV deseamos correr, a esta forma de ejecución se le llama Run Time.

La otra forma y la usada en este trabajo es crear una biblioteca donde se tengan todos los IVs, subIVs y subrutinas necesarias para que al crear la aplicación se corra automáticamente sin la necesidad de que el programa pregunte cual se desea correr, así solamente se tendrá un programa ejecutable donde el único IV que se puede observar en pantalla es el panel frontal del IV principal.

Los siguientes pasos describen cómo se realiza un programa ejecutable de un IV. Como ejemplo tomamos el IV "micrófono".

1. Se construye el IV que se desea grabar como ejecutable (donde anteriormente se crearon los subIVs que éste ocupará), el cual se salva dentro de una biblioteca con el mismo nombre. Seleccionando la opción guardar como (Save As...) del menú archivo,

en la cual se escoge la opción nueva biblioteca de IVs, esto se hace para crear una biblioteca con el mismo nombre que el IV. Nuestro IV se llama micrófono.vi y la biblioteca se llama práctica micrófono. Esta biblioteca contiene todos los SubIVs necesarios para correr la aplicación.

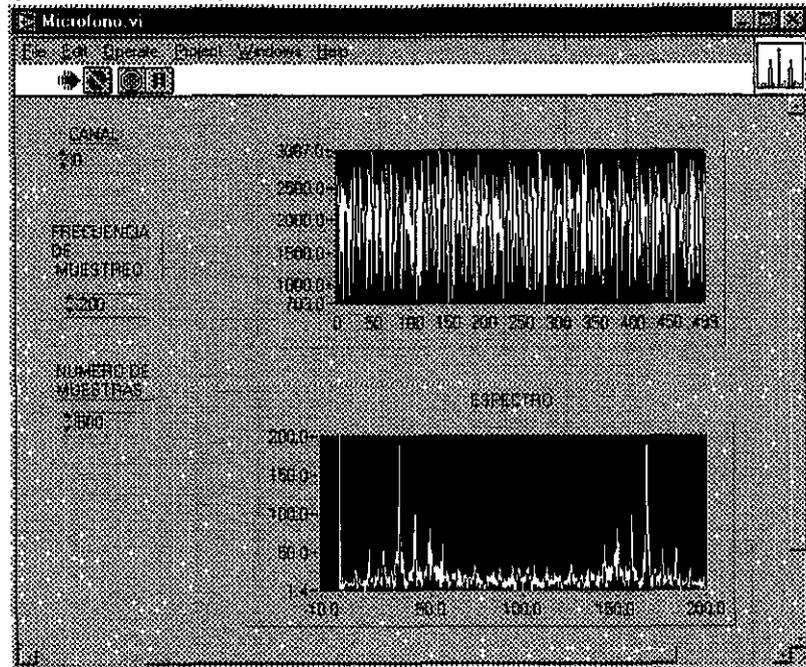


Fig. III.14 Panel de control del IV micrófono.

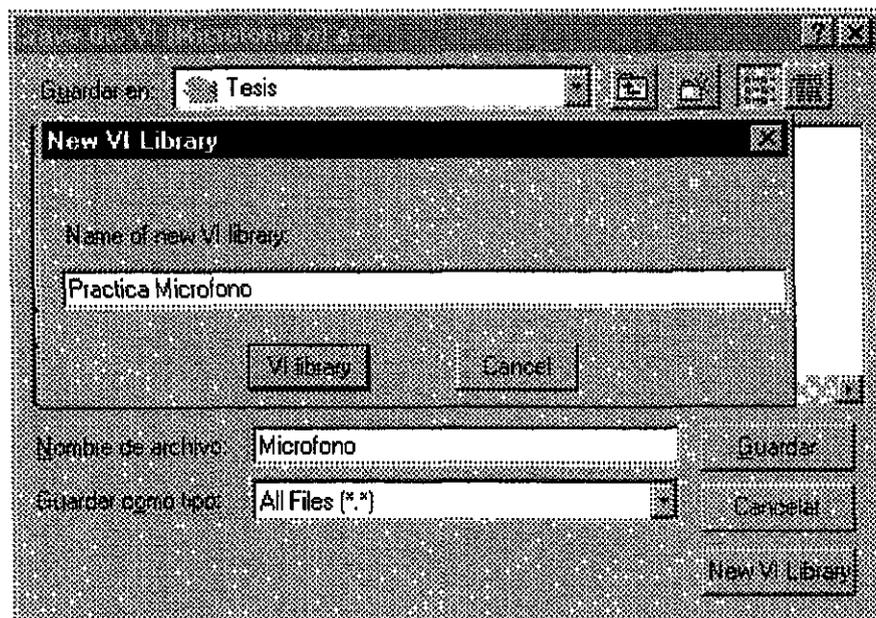


Fig. III.15 Pantalla correspondiente a la selección de nombre de biblioteca

2. Dentro del menú archivo se selecciona la opción guardar con opciones (Save With Options) y en ella seleccionamos el modo de guardar como distribución de la aplicación (Application Distribution), esta opción guarda todos los IVs, y rutinas externas. Se salva como una librería donde el mismo LabVIEW remueve los diagramas de bloques de todos los IVs.

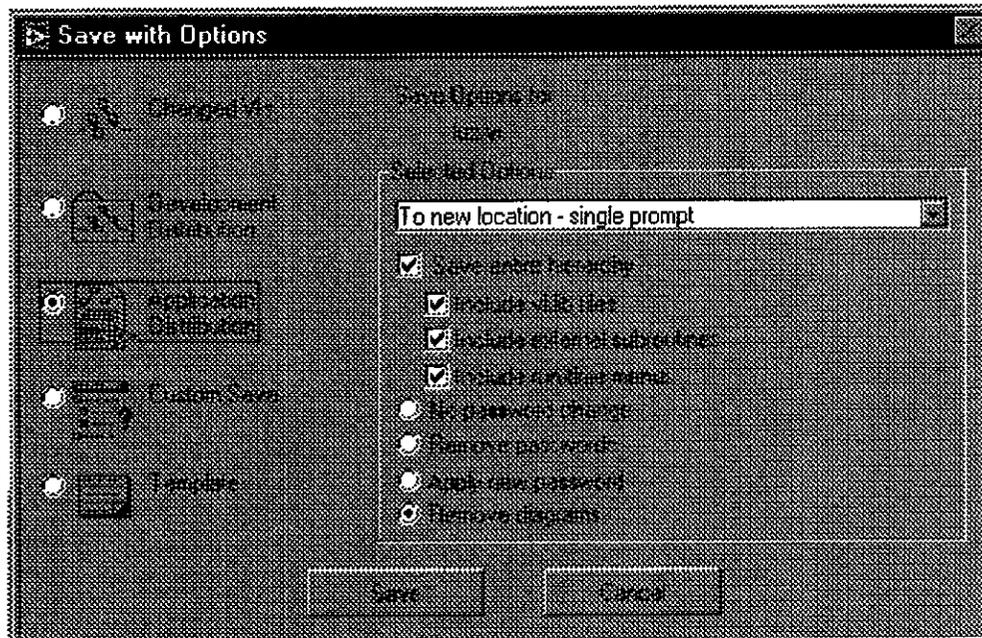


Fig. III.16 Opción para salvar con opciones

3. Como paso opcional se puede crear un archivo "Acerca del IV". Muchas aplicaciones contienen una caja de diálogo llamada "Acerca de." (About) en la cual se despliega información acerca de la aplicación. Se puede crear una caja "Acerca de." que LabVIEW ejecuta cuando el usuario selecciona About del menú de ayuda. Pero se puede tener solamente un "Acerca de." por cada aplicación, y tener solo uno si se inserta a una biblioteca.

Para crear el propio "Acerca de.", se crea un IV y se salva en la biblioteca del IV. El Acerca de... debe contener mensajes que indiquen que la aplicación ha sido creada usando LabVIEW, así como una leyenda que especifique el nombre de este archivo, el cual debe empezar con la palabra About seguida del nombre del IV. Cuando la aplicación se pone en operación se muestra el IV empezando con la palabra About, autores del programa y algunas especificaciones más.

4. Finalmente para crear un programa ejecutable, se selecciona la opción Crear Aplicación (Build Application) del menú Project. Es necesario insertar una biblioteca (la que contiene el IV, los SubIV y el Acerca de...), para esto se debe seleccionar la opción empotrar (embed).

A continuación se muestra una caja de dialogo en la cual se selecciona la biblioteca que se desea usar en la construcción de la aplicación. Se selecciona la biblioteca que se creó en el primer paso.

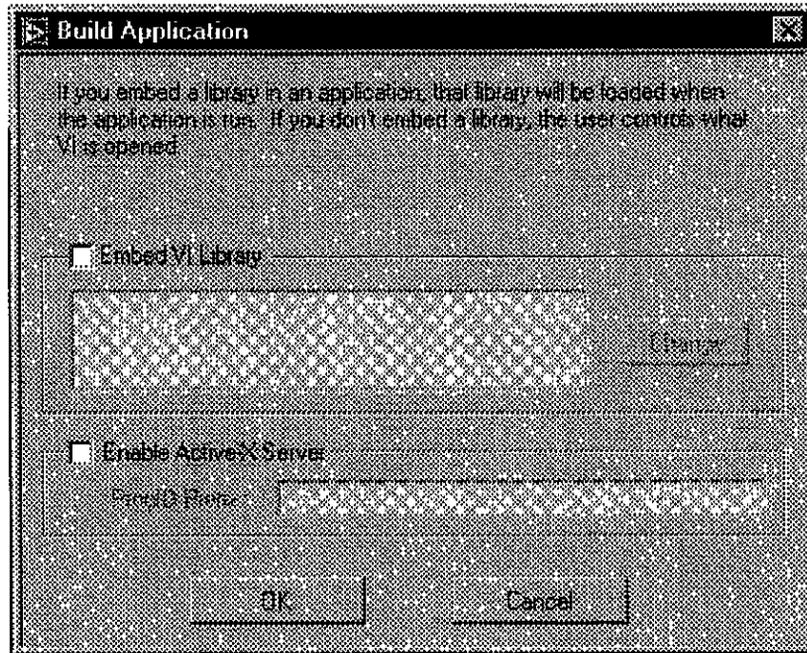


Fig. III.17 Opción para construir aplicaciones.

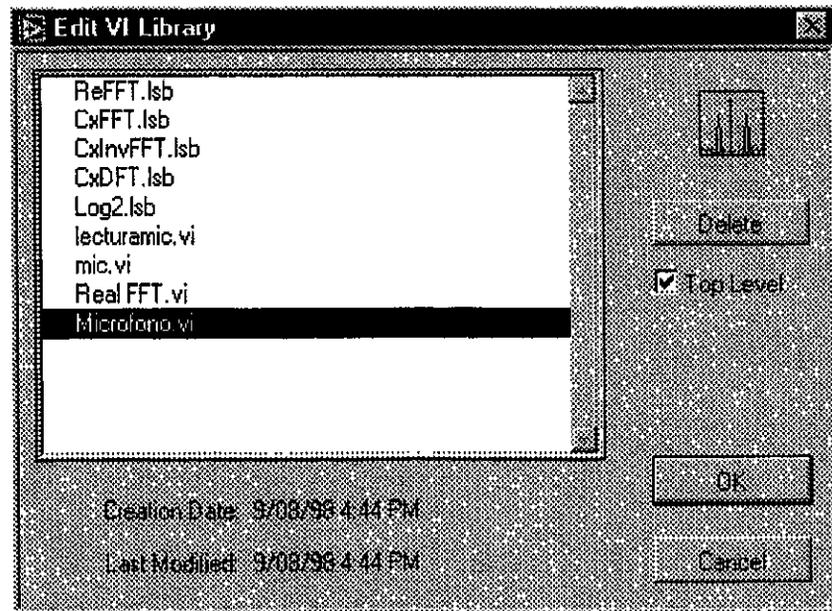


Fig. III.18 Selección de IV para alto nivel.

Cuando se han acabado de hacer las selecciones se oprime OK. Si se insertó una biblioteca y no hay IVs en ésta, se marca con la opción Autoload, LabVIEW despliega la caja de diálogo para editar una biblioteca IV (edit VI Library), así el usuario puede seleccionar cual IV se abrirá a la hora de la ejecución. Al seleccionar el IV deseado para que aparezca cuando se inicia la aplicación, se activa la opción Nivel Alto(Top-Level).

Finalmente, aparece el cuadro donde se debe teclear el nombre del IV con extensión EXE, que será el que se podrá ejecutar en cualquier máquina sin tener LabVIEW. Es importante tener en cuenta que para la ejecución de cualquier programa realizado en LabVIEW se requiere contar con las bibliotecas que se especifiquen de acuerdo al sistema de adquisición de datos a utilizar.

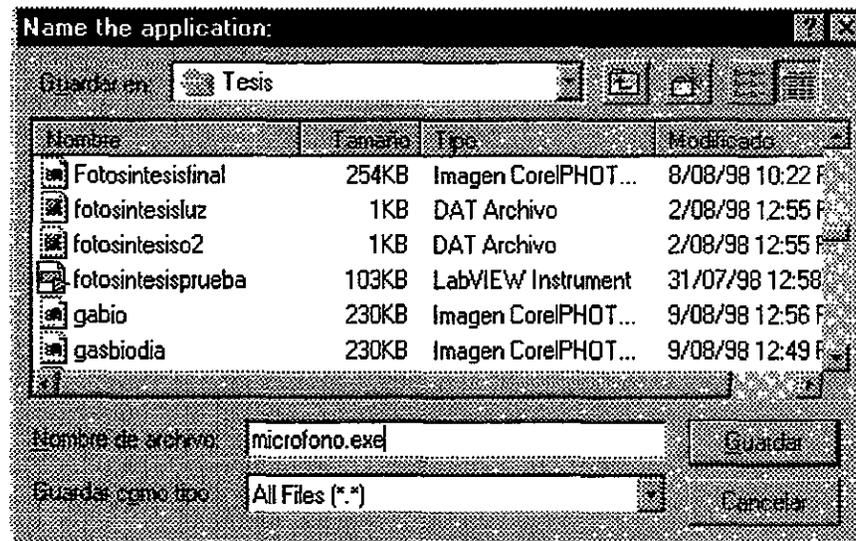


Fig. III.19 Pantalla para asignar nombre de la aplicación.

### III.3 INTERACCIÓN LabVIEW-WINDOWS95

Una de las razones por las cuales se hace un análisis de la conexión entre LabVIEW y Windows, se fundamenta en el despliegue gráfico del que LabVIEW hace uso para la representación de sus procesos.

LabVIEW es un lenguaje de programación gráfica, por lo cual, requiere de un sistema operativo con las características gráficas que permitan el aprovechamiento y correcto funcionamiento de todas las herramientas que posee LabVIEW.

LabVIEW es capaz de operar en cualquier versión de Windows, pero para el caso particular de este trabajo de tesis, nos basamos en la versión de Windows95, por ser uno de los sistemas operativos más utilizados hasta el momento en el mercado actual de las PCs con plataforma basada en Microsoft y procesadores Intel.

La versión de LabVIEW a utilizarse en este trabajo (versión 5.0) hace uso de los avances de Windows 95 de los cuales pueden considerarse varios de ellos.

**MULTIHILOS.** La función multihilos que ejecuta LabVIEW básicamente se enfoca a realizar una división de una tarea en varias tareas pequeñas las cuales corran en paralelo. Este término se refiere a trabajar con un sistema multitareas en una aplicación simple; esto es, cada una de las partes del programa será un hilo de la aplicación, los cuales son desarrollados como una parte del mismo programa, logrando que el programador tenga control en la ejecución de ellos. Existen dos formas para utilizar el sistema como multihilos, cooperativamente y preferentemente. El primero se explicó anteriormente y en el preferente, el sistema operativo maneja los programas de forma preferente, resultando una mejor distribución del procesador al activar los hilos del programa.

**MULTITAREAS.** Básicamente el sistema Windows95 es multitareas. Primordialmente usan una forma de multitareas llamada cooperativa, en la cual cada programa es ejecutado periódicamente. La ejecución multitareas preferentes realiza las ejecuciones con una calendarización previa de los eventos.

**EJECUCIÓN DE UN SISTEMA G.** Este sistema de Ejecución G usa multihilos preferente. En Windows95, la ejecución del sistema preferentemente multitareas es usado con hilos en los instrumentos virtuales. Pero el número de hilos está limitado, para aplicaciones en paralelo se usa multitareas cooperativa. También el sistema operativo toma cuidado del multitareas preferentemente entre la aplicación y otras tareas.

**ACTIVE X.** Esta herramienta es nativa de Windows95. Con automatización ActiveX, se puede acceder a propiedades y métodos específicos, como abrir y cerrar, compilar y ejecutar instrumentos virtuales desde ActiveX, entre otros, los cuales son usualmente agrupados dentro de objetos de una aplicación Windows y usados en otras aplicaciones de Windows. Una aplicación soporta automatización como cliente o como servidor. Las aplicaciones que exponen objetos y proveen métodos para operación sobre estos objetos son servidores de automatización ActiveX. Las aplicaciones que usan los métodos expuestos por otras aplicaciones son clientes de automatización ActiveX.

LabVIEW puede funcionar de ambas formas como un servidor ActiveX y como un cliente ActiveX. LabVIEW puede también desplegar un objeto ActiveX en el panel frontal usando la caja ActiveX. Por ejemplo, se puede acceder a un instrumento virtual desde una página web, situando como servidor la computadora en donde esté localizada la aplicación en LabVIEW, así como enviar datos desde LabVIEW hacia otras aplicaciones como Excel.

Se realiza un aprovechamiento del procesador, cuando se esta ejecutando un instrumento virtual que realiza adquisición de datos, LabVIEW centra la atención del procesador en la adquisición y monitorea los demás eventos del sistema, además de poder ejecutar varios programas con la opción multitareas de Windows95.

Al ser Windows95 un sistema operativo de 32 bits, y LabVIEW un lenguaje de programación que necesita para su correcta ejecución un sistema operativo de 32 bits, la ejecución de los sistemas de adquisición de datos basados en esta plataforma se realiza de manera eficaz.

## **CAPÍTULO IV**

### **Conexión ProControl - LabVIEW**

## CAPITULO IV CONEXIÓN PROCONTROL - LABVIEW

En el presente capítulo, se resuelve el problema esencial de la adquisición de datos, el cual radica en cómo se debe realizar la conexión entre el mundo exterior y la computadora.

El problema que surge al realizar la conexión entre la tarjeta ProControl PA-CP12 y LabVIEW se basa fundamentalmente en que la tarjeta no pertenece a la misma compañía, por lo cual, el software LabVIEW no posee los manejadores necesarios para reconocerla de forma automática, pero el software proporciona otra opción para cuando se presenta este tipo de casos. La solución se basa en emplear una interface de código denominada CIN.

Esta interface es capaz de poder introducir las lecturas leídas desde la tarjeta al instrumento virtual programado en LabVIEW para realizar el procesamiento de los datos adquiridos. La interface se basa en un código realizado en otro lenguaje de programación como C, el cual debe ser compilado para obtener el formato correcto que LabVIEW acepte, para esto se utiliza Watcom C.

Descrito el problema, así como su solución, se realizará una explicación más detallada del procedimiento y de los elementos utilizados para realizar la conexión entre la tarjeta ProControl PA-CP12 y LabVIEW.

Un CIN como lo dicen sus siglas Code Interface Node (nodo de interface por código), es un nodo que se coloca en el diagrama de bloques, el cual esta asociado con un código fuente escrito en un lenguaje de programación convencional, para obtener un CIN es necesario compilar primero el código fuente y después ligarlo a un código ejecutable. LabVIEW hace una llamada al código ejecutable cuando el nodo asociado a este es ejecutado, pasando los datos de entrada desde el diagrama de bloques al código ejecutable y regresando datos desde el código ejecutable al diagrama de bloques.

La forma de ejecución de los CINs es síncrona, esto significa que mientras el código del CIN es ejecutado, LabVIEW no puede ejecutar otro proceso. Cuando un IV se ejecuta, LabVIEW monitorea a los menús y al teclado permitiendo que otras aplicaciones se ejecuten así como ejecutar varios IVs simultáneamente. De cualquier forma, cuando el código objeto del CIN se ejecuta, este toma el control del procesador, así que el CPU ignora los eventos del teclado, ratón y otros. Los CINs que corren bajo Windows, también evitan que otras aplicaciones se ejecuten. Se pueden crear IVs que usen CINs y funcionar en una forma asíncrona, lo anterior es importante de considerarse ya que si se necesita un CIN que ejecute una tarea larga se requiere que LabVIEW posea la capacidad de ser multitarea.

Como LabVIEW puede generar un código lo suficientemente rápido para la realización de tareas de programación, permite usar los CIN o subrutinas en diversas aplicaciones. Los CINs se pueden usar para complementar algunas tareas que ejecutadas en un lenguaje convencional serian más lentas o difíciles.

También se pueden usar los CINs para desarrollar acciones adicionales en un momento específico de la ejecución del IV; por ejemplo, la inicialización de estructuras de datos en el momento de cargar o liberar estas estructuras privadas cuando se cierra el IV que contiene el CIN. Al igual en tareas que son críticas en tiempo o requieren una gran distribución para la eficiente manipulación de datos, para estas situaciones se escriben las rutinas que LabVIEW llama en momentos predefinidos o cuando el nodo se ejecuta, generalmente se usan estas rutinas en CINs que desempeñan una acción en marcha, como lo pueden ser en acumulación de resultados de llamada a llamada, en donde se pueden asignar, inicializar y desasignar recursos en el momento correcto.

Los CINs también se pueden utilizar para realizar tareas que no se desarrollen directamente desde el diagrama, tal como llamadas a rutinas específicas del sistema para lo cual LabVIEW no tiene funciones equivalentes, como es el caso de la tarjeta ProControl; aunque para que se puedan utilizar se deberá modificar el código de manera que este use los mismos tipos de datos que LabVIEW para garantizar datos correctos.

LabVIEW cuenta con la capacidad de soportar diversos tipos de código tal como el de los CINs o aceptar subrutinas externas generadas por un código externo. Cabe aclarar que existe una diferencia entre ambos tipos de código, el CIN es un nodo localizado en el diagrama de bloques asociado con código fuente y la subrutina externa es una sección de código que se puede llamar desde otro código externo. Se pueden escribir múltiples CINs que hagan llamadas a la misma subrutina, realizando una división de la subrutina externa, el código de una subrutina externa se encuentra en un archivo separado; cuando LabVIEW carga una sección del código externo que se refiere a una subrutina externa, cargándose la subrutina externa en memoria. Al usarse subrutinas externas se hacen llamadas por cada sección a un código muy pequeño, ya que la subrutina no requiere código extra. También se pueden hacer modificaciones a la subrutina realizando cambios una sola vez.

La interface para CINs y subrutinas externas soportan una variedad de compiladores, pero no todos los compiladores pueden crear un código ejecutable en el formato correcto, es decir, que tenga la extensión .lsb, que se requiere para el caso concreto de utilizar Windows, como es el de este trabajo.

El código externo se compila en un formato ejecutable, debiendo ser este el apropiado para cada plataforma en específico. El código es relocalizable, porque LabVIEW carga el código externo dentro del mismo espacio de memoria como una aplicación principal.

LabVIEW es una aplicación que tiene un modelo de memoria fijo de 32 bits, por lo cual requiere un compilador que posea un modelo de memoria de 32 bits. Watcom C++ es un compilador soportado por LabVIEW, que puede crear un código de 32 bit en el formato correcto. Existe también la posibilidad de crear un CIN realizando un programa en Visual C++.

## IV.1 CREACIÓN DE PROGRAMAS (CINs)

En este trabajo de tesis, se necesitó realizar un par de CINs, uno para adquirir datos desde un canal analógico, y otro para proporcionar un control utilizando los canales digitales de salida.

A continuación explicaremos el procedimiento para realizar un CIN basándonos en el ejemplo de la lectura en las entradas analógicas, y posteriormente solo citaremos el código fuente del CIN que realiza la escritura a un canal digital de la tarjeta ProControl PA-CP12, así como las diferencias que se presentan entre ellos.

1. - Dentro de LabVIEW se programa un IV en el cual se crea el CIN con una entrada y una salida. Se selecciona la función nodo de interface por código (Code Interface Node) desde la paleta avanzados (Advanced) del menú de funciones (Functions).

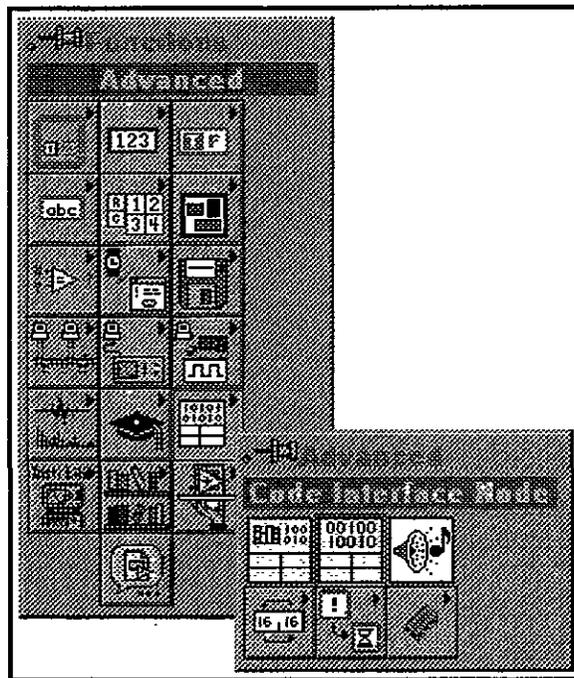


Fig. IV.1 Menú donde se encuentra la función nodo interface de código.

Un CIN tiene terminales con las cuales se pueden indicar que dato pasa hacia y desde un CIN. Inicialmente, el CIN tiene un conjunto de terminales, y se pueden pasar valores sencillos hacia y desde el CIN. Es posible agregar terminales adicionales realizando modificaciones de tamaño en el nodo (se posiciona el cursor en una de las esquinas inferiores del CIN, oprimiendo el botón del ratón y arrastrándolo hacia abajo sin dejar de presionar hasta tener el tamaño deseado) o por selección de la opción adicionar parámetros (Add Parameter) desde el menú que se despliega al presionar el botón derecho del ratón de la terminal del CIN.

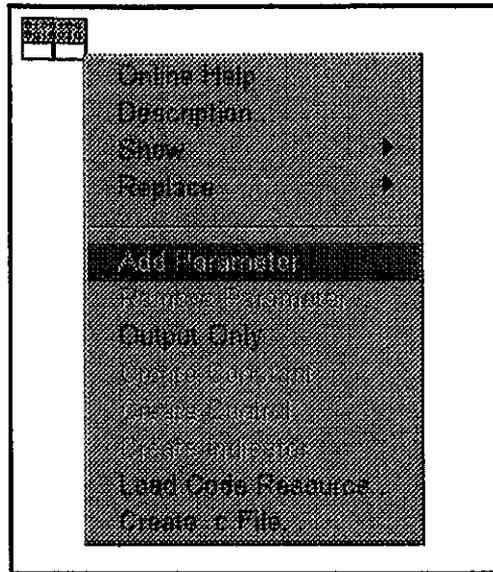


Fig. IV.2 Muestra como adicionar parámetros a un CIN.

En este caso particular, se tiene una entrada que será el canal a seleccionar, esta variable es de tipo entero y se alambra al lado derecho del CIN, como se muestra en la figura IV.3. En el lado izquierdo se alambra una variable de tipo flotante porque es el número digital leído por la tarjeta. Es importante aclarar que cada una de las dos variables se alambra en diferente renglón del CIN, porque se requiere tener en la salida un indicador que muestre el valor que el CIN regresa.

El IV queda de la siguiente forma:

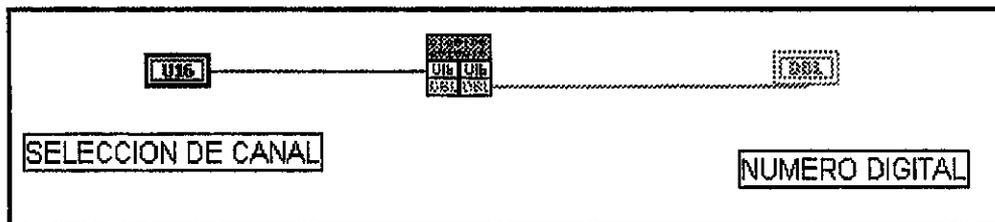


Fig. IV.3 CIN para lectura de entrada analógica.

2. - El siguiente paso es posicionar el ratón sobre el icono del CIN, y con un clic en el botón derecho del ratón aparece un submenú, en este se selecciona la opción crear un archivo.c (Create .c file).

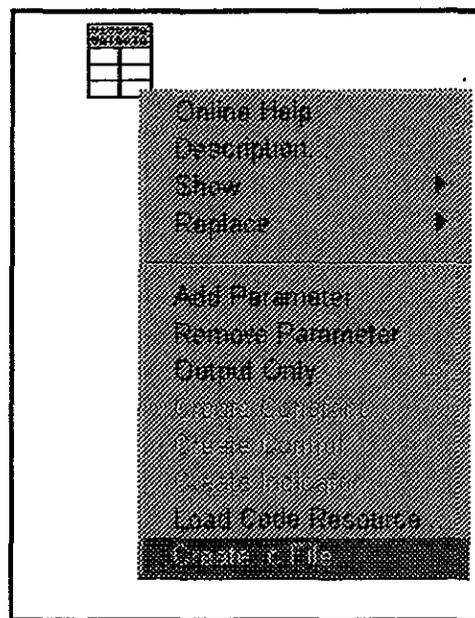


Fig. IV.4 Opción para crear un archivo con extensión c.

Esta opción permite crear un archivo .c que muestra las rutinas que se deben escribir y el tipo de datos de los parámetros que pasan al CIN, así como un espacio en donde se especifican las instrucciones necesarias para el funcionamiento de éste. A continuación se muestra como aparece el archivo creado por LabVIEW.

```

/* CIN source file */
#include "extcode.h"
/* stubs for advanced CIN functions */

UseDefaultCINInit
UseDefaultCINDispose
UseDefaultCINAbort
UseDefaultCINLoad
UseDefaultCINUnload
UseDefaultCINSave

CIN MgErr CINRun(int32 *num_in, int32 *num_out);

CIN MgErr CINRun(int32 *num_in, int32 *num_out) {
    /* ENTER YOUR CODE HERE */
    return noErr;
}

```

El extracto anterior de código es el archivo principal.c para el nodo. Este especifica siete rutinas (inicialmente vacías de cualquier código) que deberán ser escritas para el CIN. Las macros UseDefault se ponen por omisión para seis de las siete rutinas, la rutina CINRun es usada comúnmente, esta contiene la estructura general de la función dentro de la cual se puede introducir el código.

Las seis macros tienen #defines en extcode.h, las cuales se expanden en rutinas vacías; la rutina CINRun es definida por el usuario dependiendo de las variables necesarias.

Dentro del archivo creado se encuentra "extcode.h" el cual se incluye automáticamente; este es un archivo que define los tipos de datos básicos y el número de rutinas que pueden ser usadas por los CINs y las subrutinas externas, "extcode.h" define algunas constantes y tipos cuyas definiciones pueden crear conflictos con las definiciones de los archivos cabecera del sistema. El directorio cintools de LabVIEW también contiene un archivo, hosttype.h, que resuelve estas posibles diferencias. Este archivo cabecera también incluye varios de los archivos más comúnmente utilizados para la plataforma.

Usando el editor de Watcom C++, se escribe el código en C, el cual contiene las instrucciones necesarias para realizar la lectura de la tarjeta ProControl PA-CP12 como se muestra enseguida:

```

/*
 * CIN source file
 */
#include "c:\Labview5\cintools\extcode.h"
#include <stdio.h>
#include <dos.h>
#include "c:\watcom\h\math.h"
#include <conio.h>

//Base de la tarjeta Procontrol PA-CP12

#define base 0X02E0

/*
UseDefaultCINInit
UseDefaultCINDispose
UseDefaultCINAbort
UseDefaultCINLoad
UseDefaultCINUnload
UseDefaultCINSave
*/

CIN MgErr CINRun(uint16 *SELECCION_DE_CANAL, float64 *NUMERO_DIGITAL);

CIN MgErr CINRun(uint16 *SELECCION_DE_CANAL, float64 *NUMERO_DIGITAL) {

//ADT= Tiempo que tomara las lecturas.
static int bandera=1,ADT=0;

// Valor bit menos significativo (parte_baj), bit mas significativo (parte_alt).

int parte__alt,parte__baj;

```

Las seis macros tienen #defines en extcode.h, las cuales se expanden en rutinas vacías; la rutina CINRun es definida por el usuario dependiendo de las variables necesarias.

Dentro del archivo creado se encuentra "extcode.h" el cual se incluye automáticamente; este es un archivo que define los tipos de datos básicos y el número de rutinas que pueden ser usadas por los CINs y las subrutinas externas, "extcode.h" define algunas constantes y tipos cuyas definiciones pueden crear conflictos con las definiciones de los archivos cabecera del sistema. El directorio cintools de LabVIEW también contiene un archivo, hosttype.h, que resuelve estas posibles diferencias. Este archivo cabecera también incluye varios de los archivos más comúnmente utilizados para la plataforma.

Usando el editor de Watcom C++, se escribe el código en C, el cual contiene las instrucciones necesarias para realizar la lectura de la tarjeta ProControl PA-CP12 como se muestra enseguida:

```

/*
 * CIN source file
 */
#include "c:\Labview5\cintools\extcode.h"
#include <stdio.h>
#include <dos.h>
#include "c:\watcom\h\math.h"
#include <conio.h>

//Base de la tarjeta Procontrol PA-CP12

#define base 0X02E0

/*
UseDefaultCINInit
UseDefaultCINDispose
UseDefaultCINAbort
UseDefaultCINLoad
UseDefaultCINUnload
UseDefaultCINSave
*/

CIN MgErr CINRun(uint16 *SELECCION_DE_CANAL, float64 *NUMERO_DIGITAL);

CIN MgErr CINRun(uint16 *SELECCION_DE_CANAL, float64 *NUMERO_DIGITAL) {

//ADT= Tiempo que tomara las lecturas.
static int bandera=1,ADT=0;

// Valor bit menos significativo (parte_baj), bit mas significativo (parte_alt).

int parte_alt,parte_baj;

```

```

//Inicio de toma de datos y conversión

//Este For funciona como un retardo
for(y=0;y<=ADT;y++);

// Selección de canal a leer.
outp(base+2,*SELECCION_DE_CANAL);

//Empieza la conversión.
outp(base+0,0);

//Este ciclo esta checando el status de la conversión
while(inp(base+3) & 0x80);

//Lectura del bit menos significativo y el bit mas significativo
parte_baj = inp(base+0);
parte_alt = inp(base+1);

//Conversión a formato de 12 bits

*NUMERO_DIGITAL = (parte_alt*16)+(parte_baj%16);

return noErr;
}

```

3. - El siguiente paso es crear un archivo en un editor de texto con extensión <nombre de archivo>.lvm que contenga lo siguiente.

**name = nombre** Es el nombre para el código; indica el nombre base que se desea poner para el CIN. El código fuente para el CIN deberá ser <nombre>.c.

**type = tipo** Es el tipo de código externo que se desea crear. Para los CINs, se deberá usar un tipo específico.

**codedir = codeDir** Es el nombre de la ruta parcial o completa al directorio que contiene el archivo.c usado par el CIN.

**wcDir = wcDir** Es el nombre de la ruta parcial o completa al directorio que contiene el compilador Watcom.

**cinToolsDir = cinToolsDir** Es el nombre de la ruta parcial o completa al directorio cintools LabVIEW, el cual es localizado en el directorio de LabVIEW. Este directorio contiene los archivos cabecera para que se pueda crear el CIN, y las herramientas que la utileria wmake usa al crear el CIN.

**inclDir = inclDir(opcional)** Es el nombre de la ruta parcial o completa al directorio que contiene cualquier archivo.h adicional.

`objFiles = objFiles(opcional)` Es la lista de archivos objeto adicionales (archivos con una extensión.obj) que tu código necesita para compilar. Los nombres de los archivos se separan con espacios.

`subrNames = subrNames(opcional)` Es la lista de subrutinas externas que el CIN llama. Se necesita `subrNames` solo si el CIN llama subrutinas externas. Los nombres de las subrutinas se separan con espacios.

Por omisión se pone la siguiente instrucción

```
! include $(cinToolsDir) \ generic.mak
```

El archivo que nosotros editamos para este trabajo queda de la siguiente forma:

```
name=lectura
type=CIN
codeDir=c:\watcom\binb
cinToolsDir=c:\labview5\cintools
wcDir=c:\watcom
!include $(cinToolsDir)\generic.mak
```

4. - Con los pasos anteriormente realizados inicializamos la máquina en modo MS-DOS. En este paso compilaremos el programa en C que se creó con anterioridad, en Windows los CIN's son compilados bajo el compilador Watcom C++, compilador de 32 bits. En este compilador la opción utilizada es **WMAKE**, la cual convierte un archivo, en este caso en lenguaje c, a un archivo **lsb**.

Esta opción, especifica que el nombre que le siga será el nombre del archivo a procesar.

```
wmake /f <nombre>.lvm
```

La utilidad `wmake` compila el código y entonces transforma este en un formato que LabVIEW puede usar. El código resultante es almacenado en un archivo **nombre.lsb**, el cual puede ser leído por LabVIEW.

5. Se carga el código a utilizar accedendo a la opción **Load Code Resouce** del menú que aparece al presionar el botón derecho del ratón, se selecciona el archivo.lsb que ha sido creado en el paso 4.

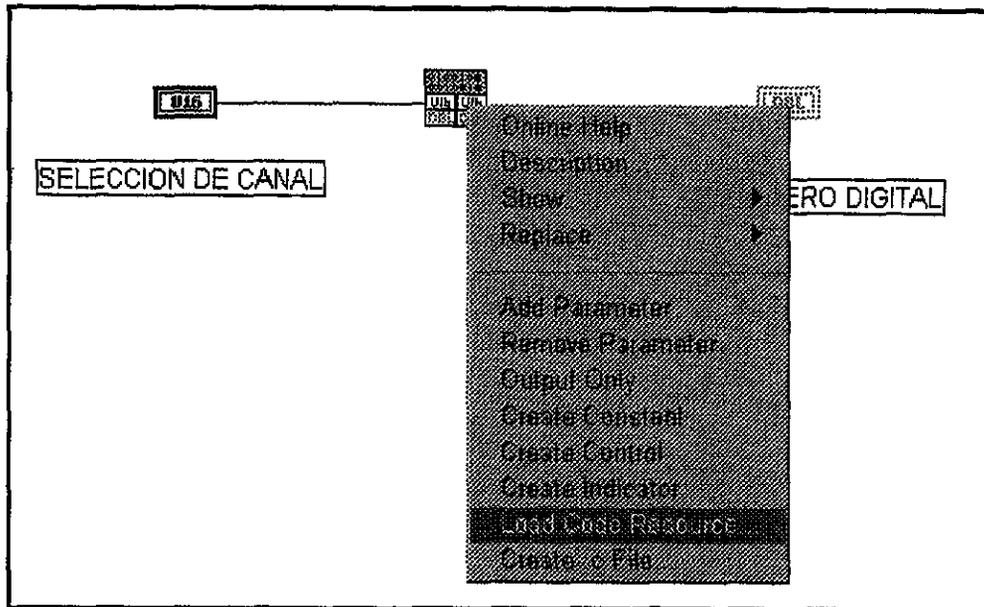


Fig. IV.5 Opción para cargar el código al CIN.

Este comando carga el código objeto en memoria y liga el código actual al panel frontal/diagrama de bloques (front panel/block diagram), después se salva el IV, el archivo contiene ya el código objeto, el cual no es necesario volver a cargar puesto que este se encuentra residente en la computadora al ejecutar el IV de LabVIEW.

En seguida se muestra el icono que resulta del CIN que permite realizar la adquisición de datos desde un canal de lectura analógica.



Fig. IV.6 Icono del CIN de lectura analógica.

A continuación se presenta el código en C, así como su icono, los cuales se utilizan para el CIN que realiza la escritura digital, cabe mencionar que este maneja algunas otras variables e instrucciones diferentes al anterior.

El CIN de escritura digital, escribe un valor digital 0 ó 1 en un canal digital de la tarjeta ProControl PA-CP12, este se utiliza para realizar un control de encendido/apagado de una variable dependiente de las prácticas experimentales a realizar.

```
/*  
 * CIN source file  
 */  
  
#include "extcode.h"  
#include <conio.h>  
  
// Define la base de la tarjeta Procontrol.  
  
#define base 0x2e0  
  
CIN MgErr CINRun(int8 *CONTROL, int8 *CONTROL);  
  
CIN MgErr CINRun(int8 *CONTROL, int8 *CONTROL) {  
  
    // Escribe en el puerto base+8 un uno o un cero, entregados por el control de encendido y  
    // apagado del IV "control.vi".  
  
        outp(base+8,*control);  
  
    return noErr;  
}
```



Fig. IV.7 Icono del CIN de escritura digital

## **CAPITULO V**

### **EJEMPLO DE DESARROLLOS DE PROGRAMAS DE CÓMPUTO PARA LA ENSEÑANZA EXPERIMENTAL, MEDIANTE LabVIEW**

## CAPITULO V

### EJEMPLO DE DESARROLLOS DE PROGRAMAS DE CÓMPUTO PARA LA ENSEÑANZA EXPERIMENTAL, MEDIANTE LabVIEW

En este capítulo se realizan programas en LabVIEW que adquieren datos a través de la tarjeta Procontrol PA-CP12, así como la opción de control de alguna variable independiente del experimento realizado.

Los programas que se realizan están enfocados a las áreas experimentales de la enseñanza como son: Química, Física y Biología, para lo cual se han realizado los experimentos siguientes.

- Fotosíntesis
- Simulación de fotosíntesis
- Transferencia de calor en un sistema acuático
- Análisis de sonido

A continuación se presenta la información de los programas realizados para las prácticas experimentales ya citadas, en el siguiente orden.

- V.1 Características de los sensores
- V. 2 Programas
- V. 3 Propuesta de práctica experimentales
- V. 4 Ejemplos de reportes de las actividades experimentales

## V.1 CARACTERÍSTICAS DE LOS SENSORES

En este tema se proporciona de manera general la información de los sensores utilizados en este trabajo, tal como intervalos medición, funcionamiento y uso.

### *SENSOR DE GAS DE DIÓXIDO DE CARBONO*

El sensor de CO<sub>2</sub> es usado para monitorear niveles de dióxido de carbono gaseoso en una variedad de experimentos de biología y química. Éste puede medir pequeños cambios en la concentración de dióxido de carbono en experimentos de fotosíntesis y respiración. El rango que mide es de 0 a 5000 ppm (partes por millón), registrando la cantidad de radiación infrarroja absorbida por moléculas de dióxido de carbono

El sensor utiliza un filamento de metal caliente como una fuente. Al incrementarse la concentración de gas absorbente en el tubo de muestreo, es menor la radiación de la fuente a través del tubo, el incremento de la temperatura en el sensor infrarrojo produce un voltaje que es amplificado para leerse.

### *SENSOR DE PRESIÓN DE GAS BIOLÓGICO*

El sensor de presión de gas biológico se utiliza en actividades biológicas que implican medir la producción o el consumo de un gas, dando por resultado un cambio en la presión de los gases en el experimento.

El sensor funciona mediante una membrana que se flexiona con los cambios de presión. Este sensor mide presión absoluta. Un lado de la membrana esta al vacío, mientras el otro lado esta abierto a la atmósfera. Produciendo un voltaje de salida el cual varia en forma lineal con la presión absoluta.

El intervalo de medición de la presión de 0.75 a 1.54 atmósferas(atm). Se usa para monitorear la producción de O<sub>2</sub> durante la fotosíntesis de una planta acuática, como lo es la Elodea.

### *SENSOR DE LUZ*

Este sensor emplea un fotodiodo de silicio tipo Hamamatsu S1133, el cual produce un voltaje que es lineal con la intensidad luminosa. La respuesta espectral responde aproximadamente a la respuesta del ojo humano. El interruptor en la caja se utiliza para seleccionar el rango, que se encuentra dentro de los siguientes límites.

- El rango de 0-600 luxes, es útil para niveles bajos de iluminación. El voltaje del sensor cambia un volt por cada 200 luxes.
- El rango de 0-6,000 luxes, es un buen rango de propósito general para niveles de iluminación en interiores. El voltaje cambia un volt por cada 2219 luxes.
- El rango de 0-150,000 luxes, se emplea principalmente para tomar mediciones en exteriores con luz del sol. El voltaje cambia un volt por cada 50,000 luxes.

### *SENSOR DE OXÍGENO DISUELTO*

El sensor de oxígeno disuelto se usa para medir la concentración de oxígeno disuelto en muestras de agua. Ya que el oxígeno disuelto es uno de los indicadores primarios de la calidad de un medio acuático, este sensor se puede usar para realizar una amplia variedad de pruebas o experimentos para determinar los cambios en niveles de oxígeno disuelto en líquidos.

Algunas de las principales actividades donde se puede utilizar es para determinar la relación entre la concentración de oxígeno disuelto y temperatura de una muestra de agua; también se puede monitorear la proporción en la cual el oxígeno se disuelve para diferentes niveles de turbulencia.

El sensor es un electrodo polarográfico tipo Clark que sensa la concentración de oxígeno. Un cátodo de platino y un ánodo de referencia cloruro de plata/plata en un electrolito KCl, los cuales son separados de la muestra por una membrana plástica permeable al gas. Un voltaje fijo es aplicado al electrodo de platino y la oxidación del oxígeno toma como referencia el ánodo. La corriente que fluye es proporcional al rango de difusión del oxígeno

El intervalo de medición es de 0 a 15 mg/L (o ppm, partes por millón). Intervalo de temperatura en el que trabaja es de 5 a 35°C.

### *SENSOR DE TEMPERATURA*

El sensor de temperatura se utiliza para medir variaciones de temperatura en aire y líquidos, este sistema tiene una caja acondicionadora de señal y una sonda, que es un tubo recubierto de teflón con un sensor de temperatura en la punta. El sensor de temperatura utiliza un transductor de temperatura AD590JH, el cual produce un voltaje que es lineal con la temperatura.

El intervalo de temperatura del sensor va de -50°C hasta +150°C.

## V. 2 PROGRAMAS

### *EXPERIMENTO DE LA FOTOSÍNTESIS*

El programa adquiere los datos de la fotosíntesis, colecta datos para las variables luz, dióxido de carbono y presión de oxígeno.

El programa principal muestra en la pantalla un instrumento virtual en donde se muestra la gráfica de las tres variables (fig. V.1), así como el promedio de los valores adquiridos, valor máximo y mínimo de cada variable, y la selección del canal del cual se va a muestrear cada sensor.

Es importante mencionar que el programa principal es denominado en LabVIEW instrumento virtual (IV). Este contiene varios pequeños programas llamados subinstrumentos virtuales (subIV), los cuales realizan procesos específicos dentro del programa principal. (fig. V.2).

A continuación se presentan los paneles frontales y los respectivos diagramas de bloques de cada uno de los instrumentos virtuales y subinstrumentos virtuales, proporcionando una explicación detallada de sus funciones.

#### *PROGRAMA: FOTOSÍNTESIS.VI*

El IV llamado fotosíntesis, realiza la adquisición de datos del experimento de fotosíntesis, dentro del cual contiene los subIV, como luz.vi, co2.vi, o2.vi, que realizan la adquisición de datos de cada variable, cabe mencionar que cada programa es diferente por contener los valores para la calibración propia de cada sensor (esto se explica con detalle posteriormente). Además de los subIV guardar.vi que graba los datos a un archivo, y lectura.vi; que realiza la lectura del canal analógico.

Este IV presenta los valores adquiridos en forma de gráficas, realiza un análisis estadístico proporcionando promedio, valor máximo y mínimo. Además el programa guarda los datos en el archivo que se le indique.

En la figura V.1 se muestra el panel frontal del Instrumento Virtual correspondiente a la fotosíntesis, y el diagrama de bloques esta en la figura V.3. En la figura V.2 se proporciona la jerarquía del programa fotosíntesis, en donde se pueden apreciar todos los subIV e iconos utilizados en la programación de éste.

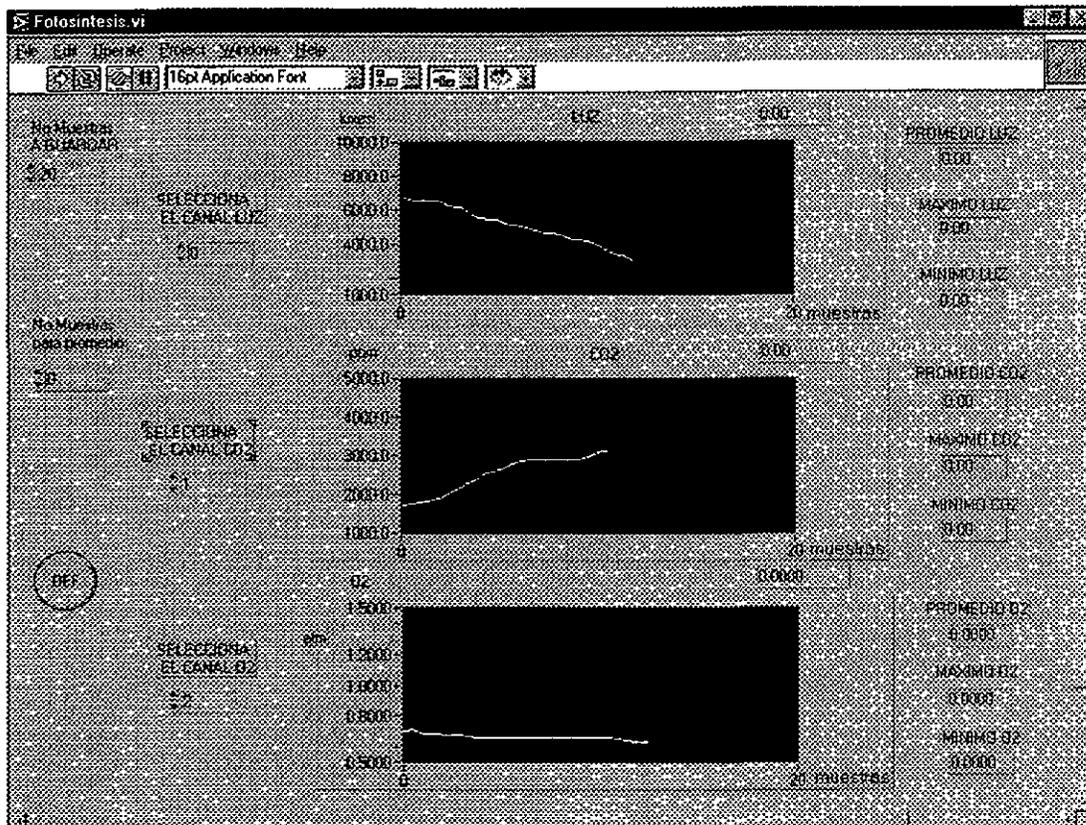


Fig. V.1 Panel frontal del instrumento virtual del programa fotosíntesis

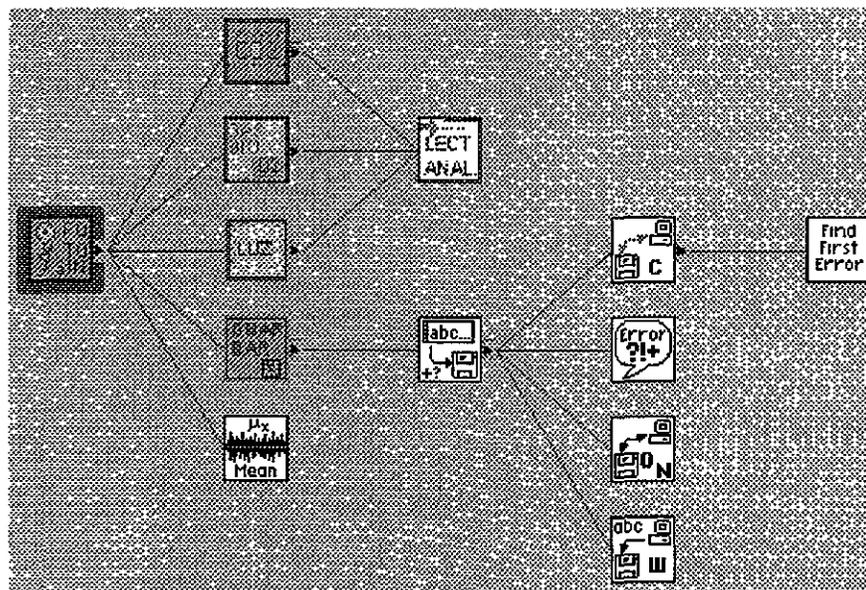


Fig. V.2 Jerarquía del programa fotosíntesis.

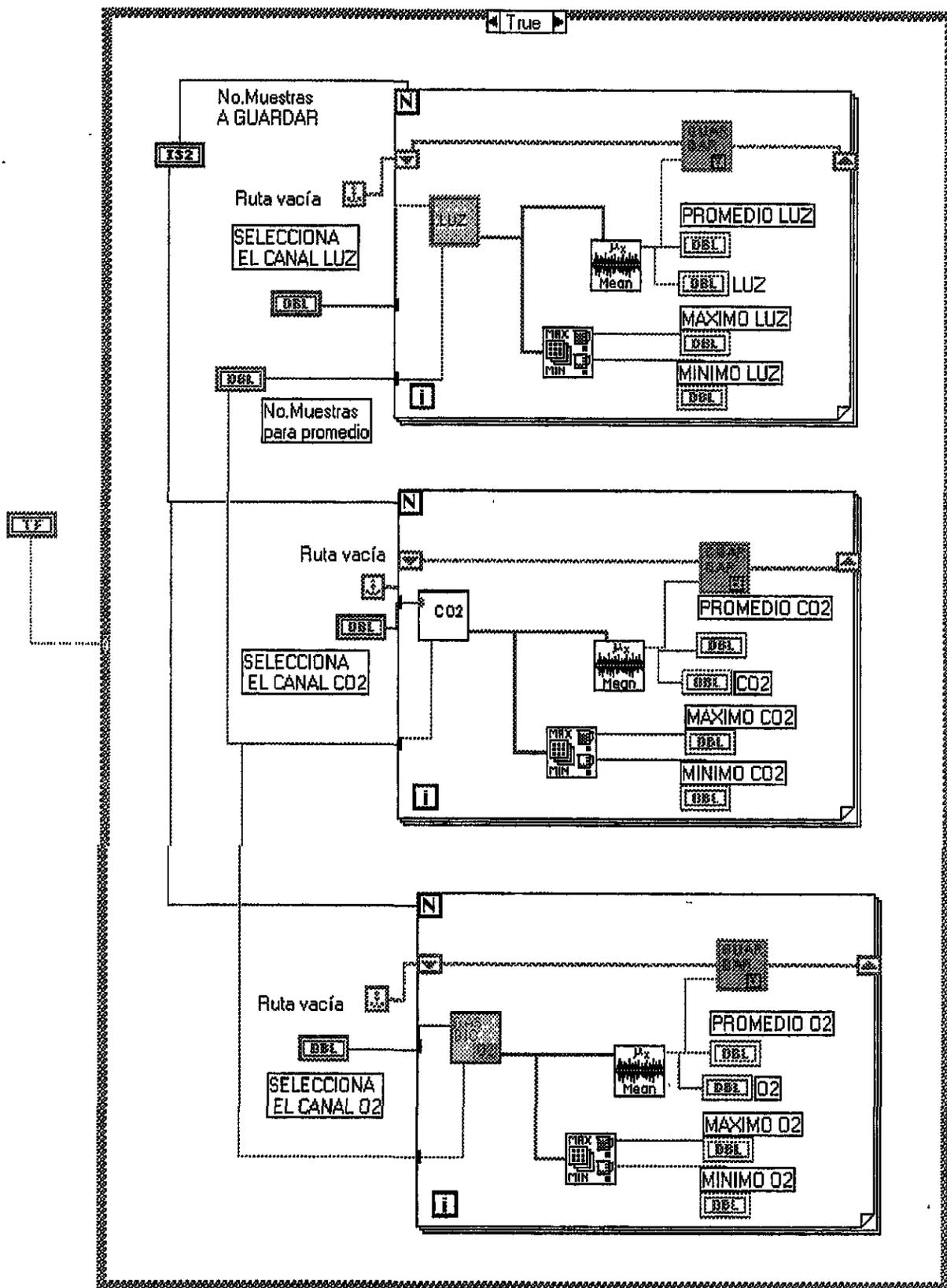


Fig. V.3 Diagrama de bloques del programa fotosíntesis.

*PROGRAMA LUZ.VI*

Este programa adquiere los datos correspondientes al sensor de luz teniendo como entrada el valor digital que toma la tarjeta a través del canal analógico, el cual se convierte a un valor de luxes mediante el ajuste a una recta,  $y=mx+b$ , en donde la pendiente  $m=120.409292$  y la ordenada al origen  $b=245327.82$  son determinados por la calibración, las operaciones anteriores se aprecian en el diagrama de bloques que muestra la figura V.5, así como su respectivo panel frontal en la figura V.4. La calibración se realizó en forma similar para todos los sensores, variando en los valores obtenidos para la pendiente y la ordenada al origen.

Para realizar la comparación entre el valor digital que proporciona la tarjeta y el valor correspondiente a cada variable, se tomó como referencia el valor que se obtuvo del programa MultiPurpose Lab Interface (MPLI), software para adquisición de datos de Vernier, el cual cuenta con los archivos de calibración necesarios para proporcionar los valores de las lecturas en las unidades específicas de cada variable.

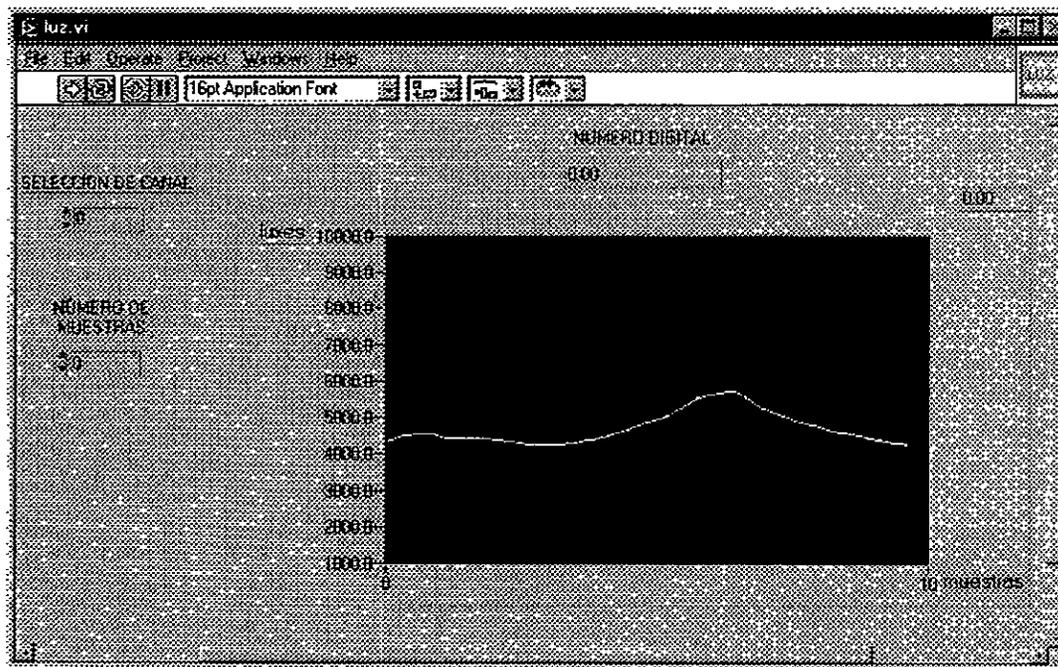


Fig. V.4 Panel frontal del programa luz.

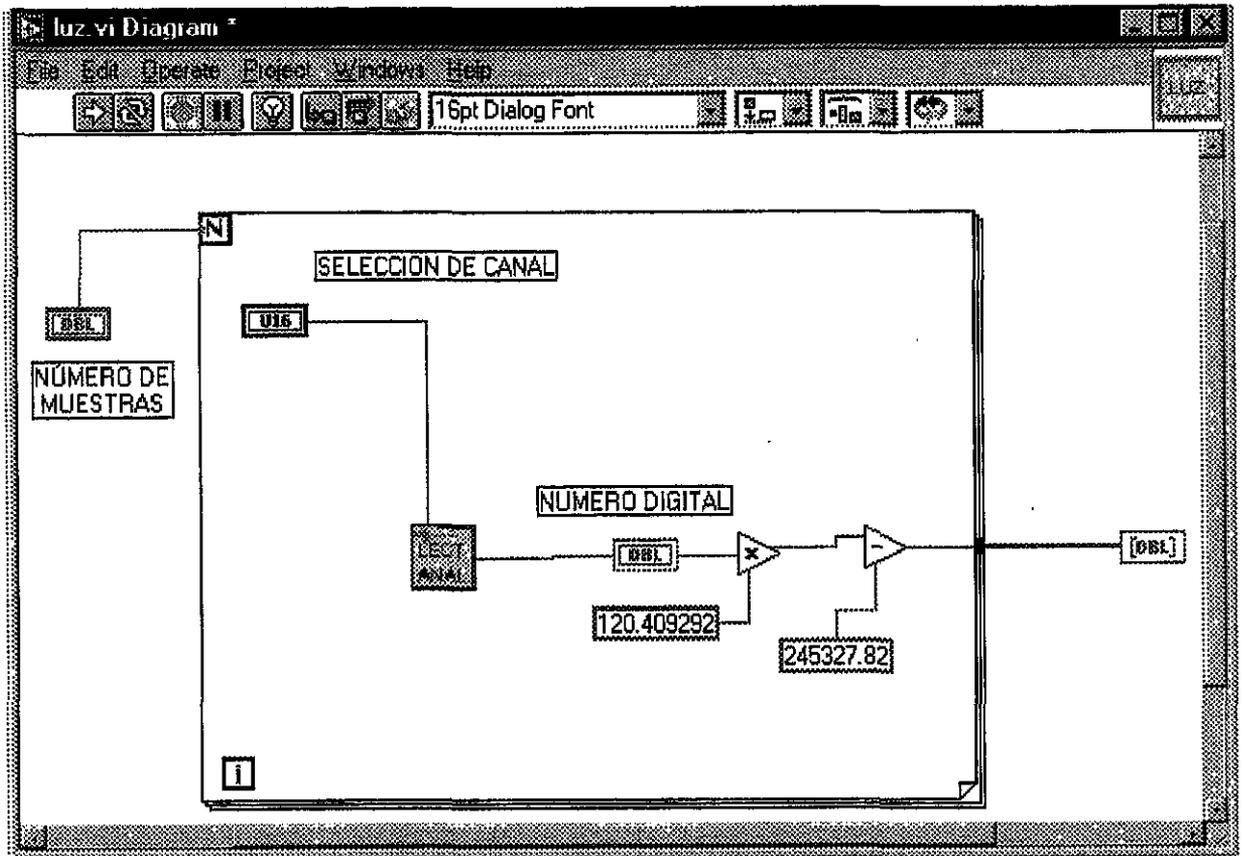


Fig. V.5 Diagrama de bloques del programa luz.

*PROGRAMA CO2.VI*

Éste adquiere los datos obtenidos del sensor de dióxido de carbono mostrados en la figura V.6, al igual que el anterior programa toma los valores desde un canal analógico, este valor digital es convertido mediante la calibración a un valor de ppm (partes por millón), de la misma forma que el programa anterior, con los valores;  $m=4.718538$  y  $b=9566.12722$ , ambos se observan en el diagrama de bloques de la figura V.7.

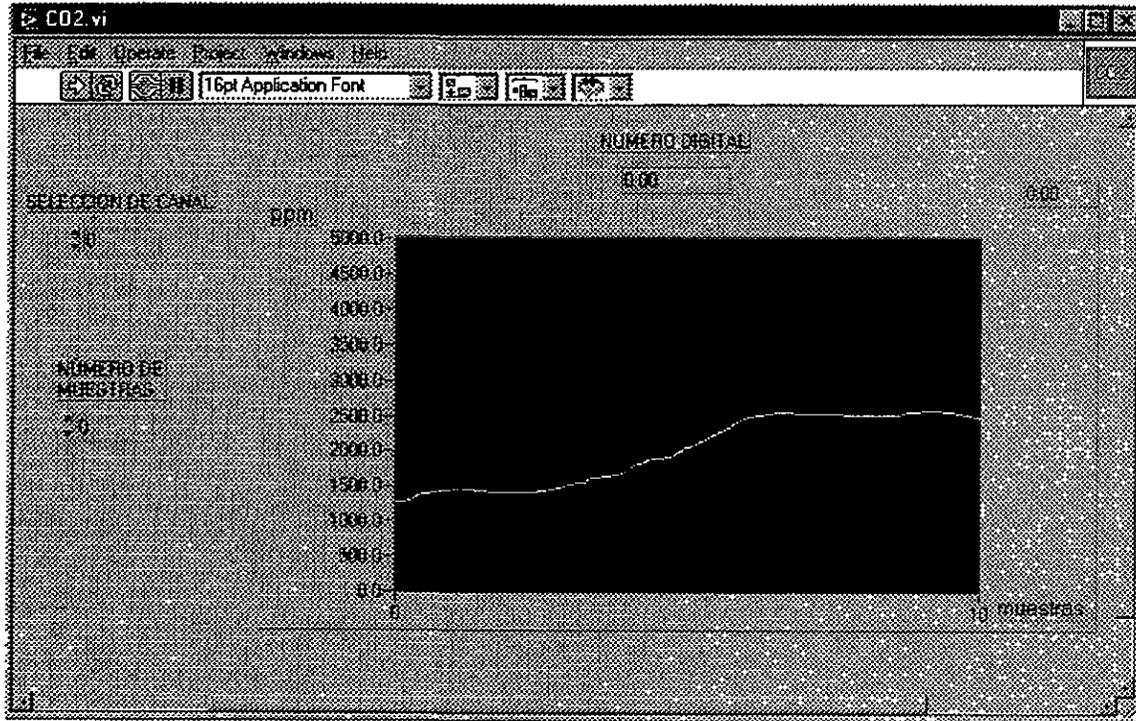


Fig. V.6 Panel frontal del programa CO2.

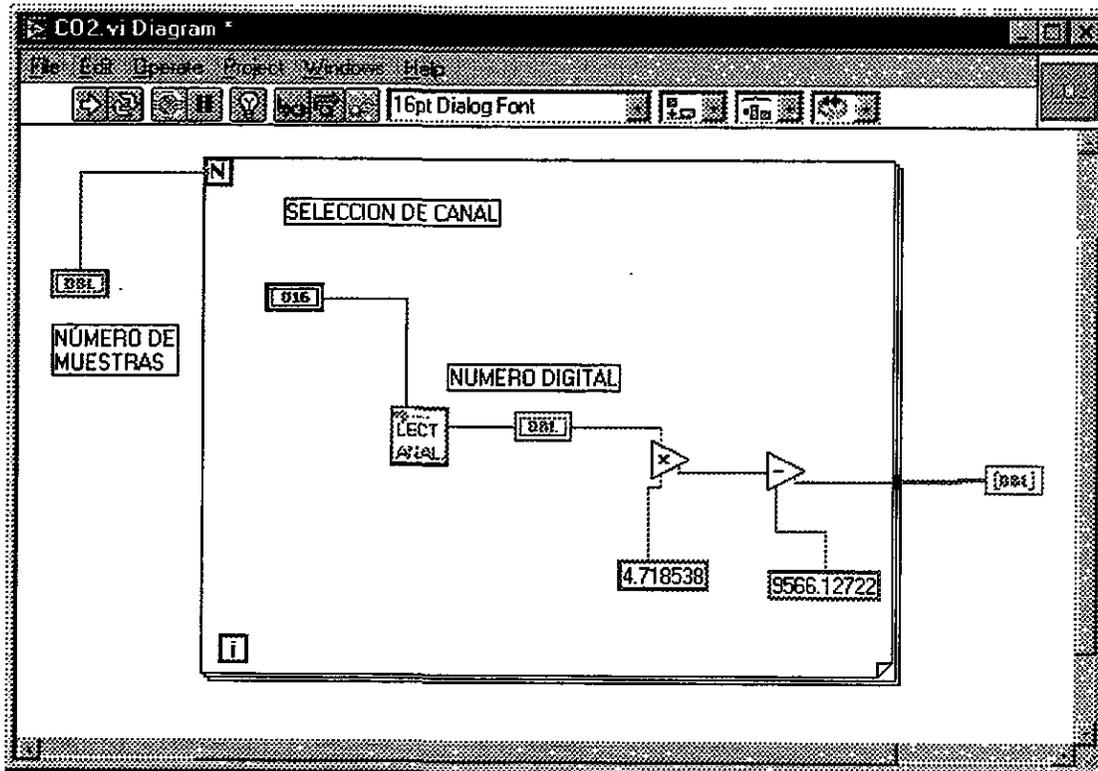


Fig. V.7 Diagrama de bloques del programa CO2.

*PROGRAMA GASBIO.VI*

Este programa adquiere los datos correspondientes al sensor de presión de gas biológico, con el cual se toma el valor del oxígeno en unidades de atm (atmósferas), mostrado en la figura V.8. La calibración de este sensor tiene los valores,  $m=0.000475$  y  $b=-1.241625$  para la aproximación a una recta, éstos se pueden observar en la figura V.9.

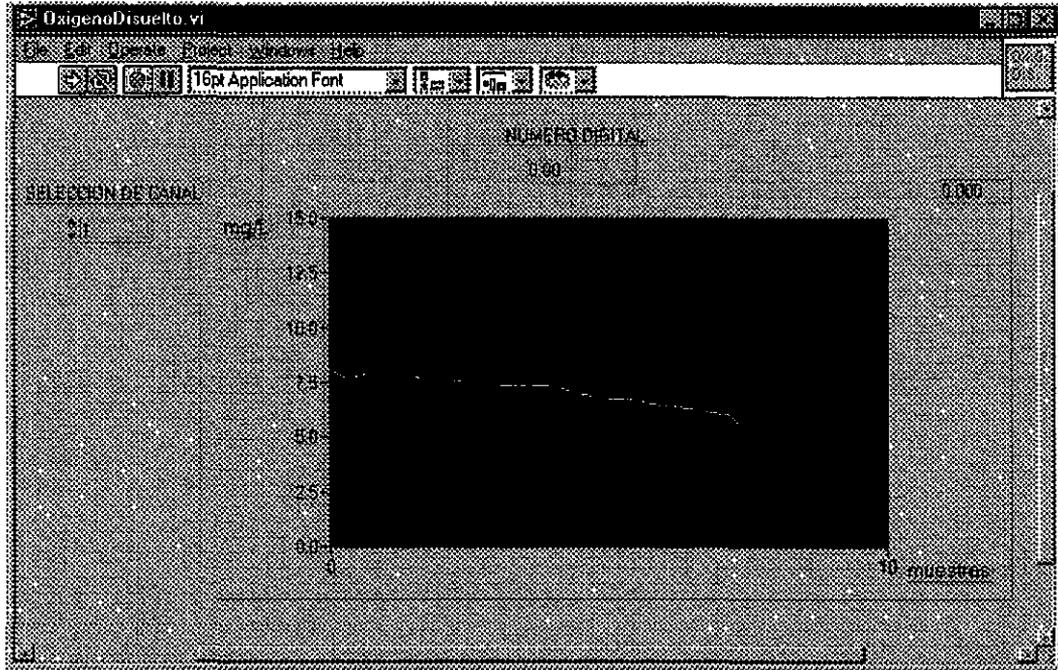


Fig. V.8 Panel frontal del programa gasbio.

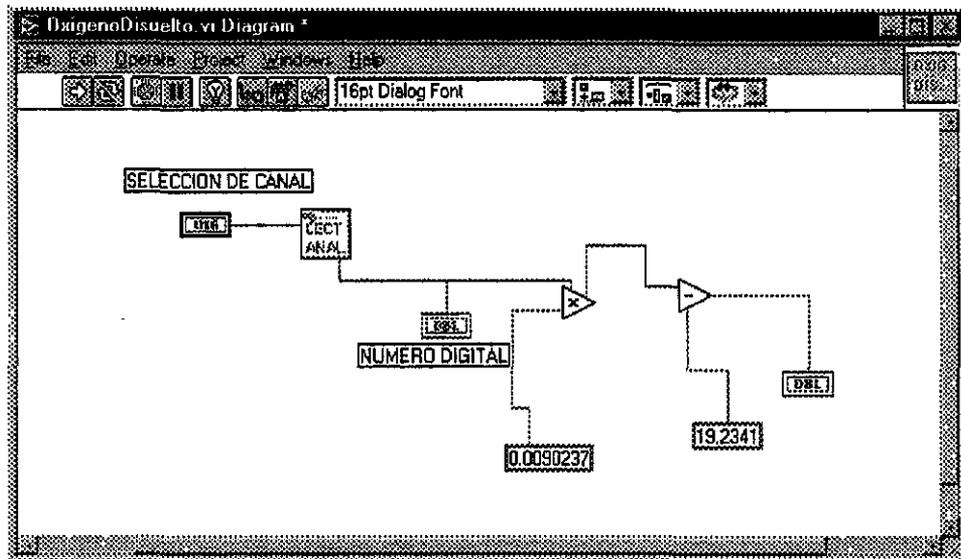


Fig. V. 9 Diagrama de bloques del programa gasbio.

PROGRAMA GUARDAR.VI

El programa guarda los datos obtenidos por cada sensor en un archivo, el cual especifica el usuario, ver figura V.10.

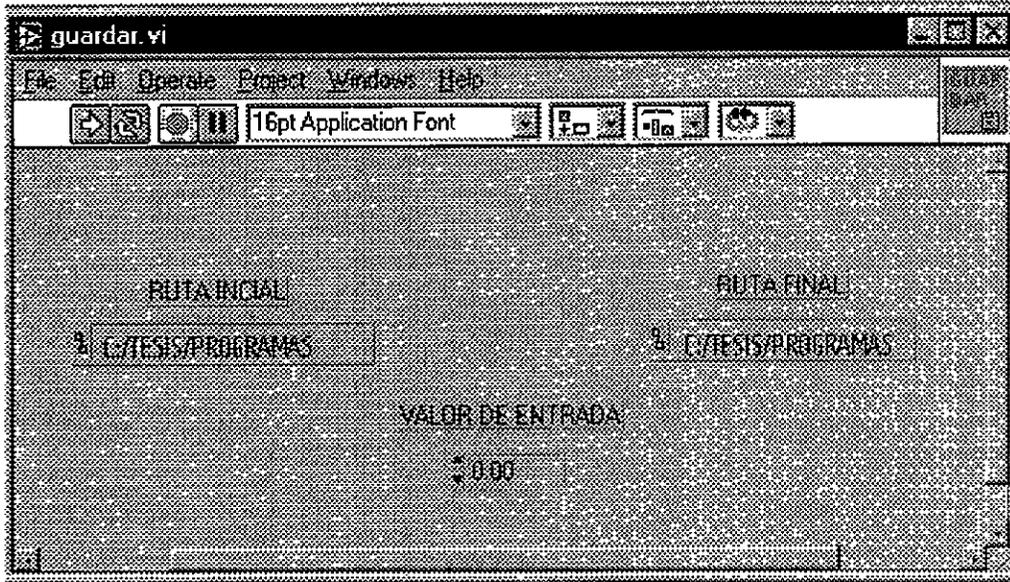


Fig. V.10 Panel frontal del programa guardar.

En el diagrama de bloques mostrado en la figura V.11, se observa el código de este programa, el cual contiene varios iconos para convertir el valor de entrada a un formato en cadenas de caracteres los valores obtenidos con una precisión determinada por el usuario para ser concatenados dentro del archivo donde se van a guardar.

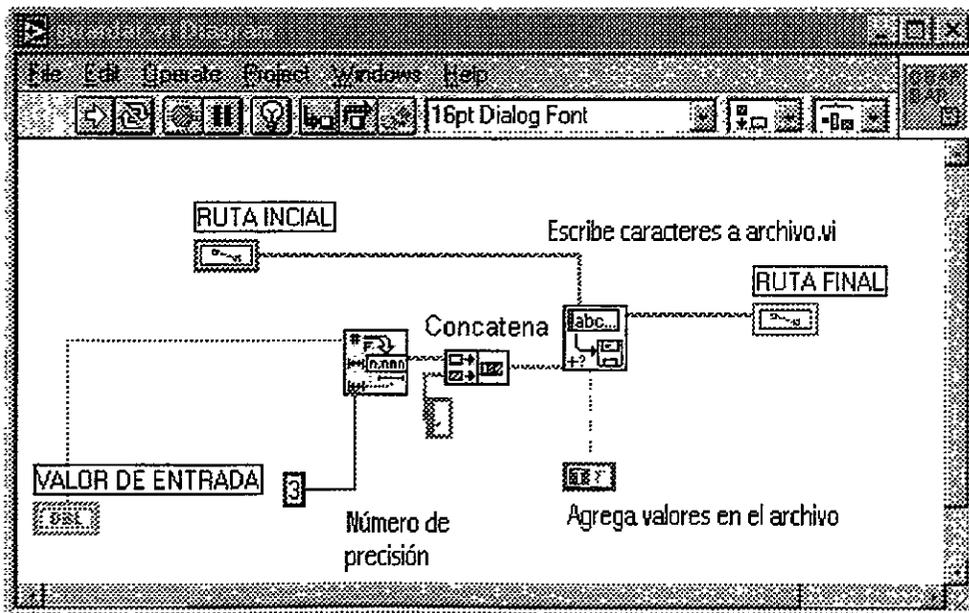


Fig. V.11 Diagrama de bloques del programa guardar.

*PROGRAMA LECTURA.VI*

Cada uno de los programas que adquieren datos de los sensores utilizan este programa como se muestra en la figura V.12, el cual contiene el programa representado en forma de icono contenido en el panel frontal de la figura V.13, siendo este quien realiza la conexión entre el software LabVIEW y la tarjeta ProControl PA-CP12 por medio de una interface denominada CIN

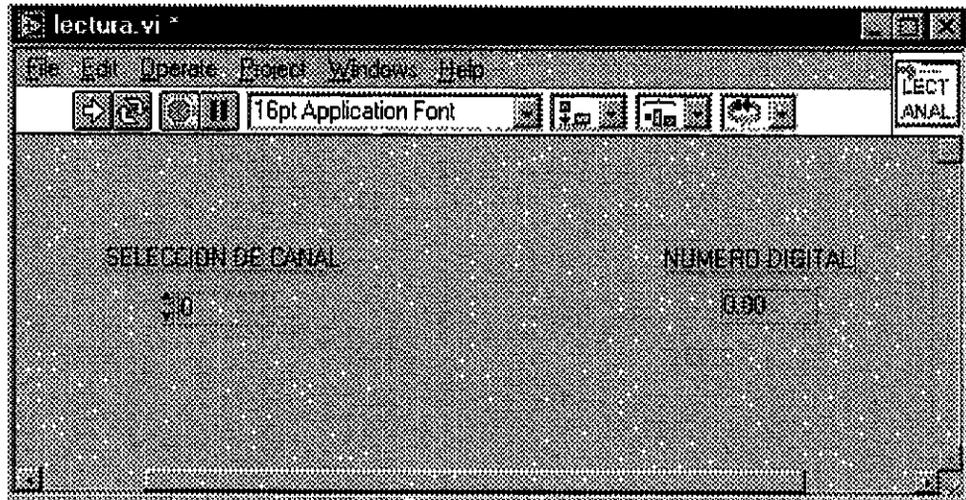


Fig. V.12 Panel frontal del programa lectura.

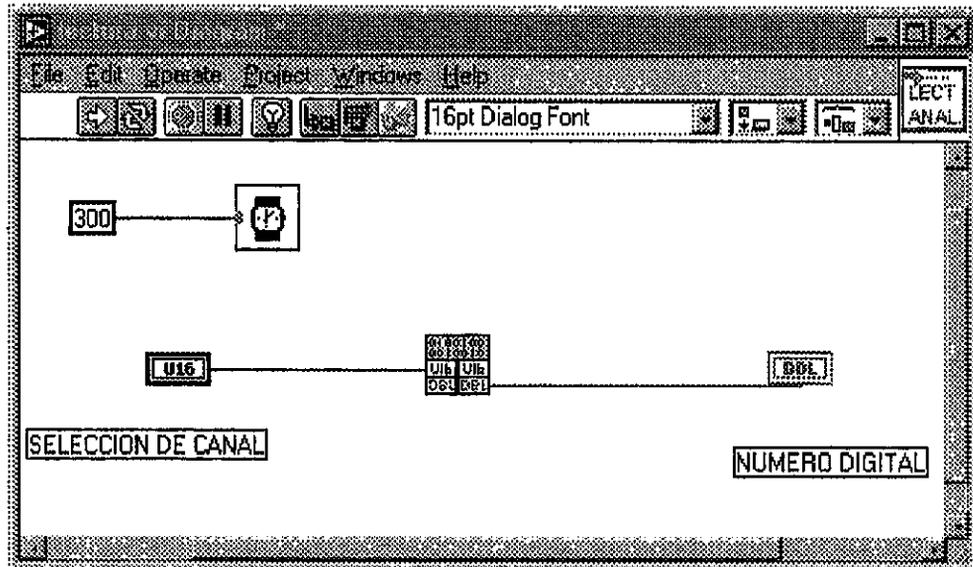


Fig. V.13 Diagrama de bloques del programa lectura.

### SIMULACIÓN DE FOTOSÍNTESIS

El IV de simulación realiza una ejecución de la fotosíntesis sin necesidad de muestrear el experimento durante un periodo amplio de tiempo, ya que se simula la luz solar por medio de un foco, el cual se controla con el botón de control que se encuentra en el panel frontal.

También se realiza un despliegue de los datos adquiridos por medio de gráficas y un análisis estadístico como promedio, valor máximo y mínimo. Además de guardar los datos en un archivo.

Los programas de luz.vi, CO2.vi y O2.vi son los mismos que en el programa de fotosíntesis, así como el de guardar.vi y lectura.vi, que se utilizan en los demás programas de adquisición. Este IV utiliza además los sub instrumentos virtuales llamados: control.vi y escritura.vi, ambos explicados más adelante.

A continuación se muestra el panel de control de este programa en la figura V.14, el diagrama de bloques en la figura V.15, así como la jerarquía y todos los iconos que se utilizan.

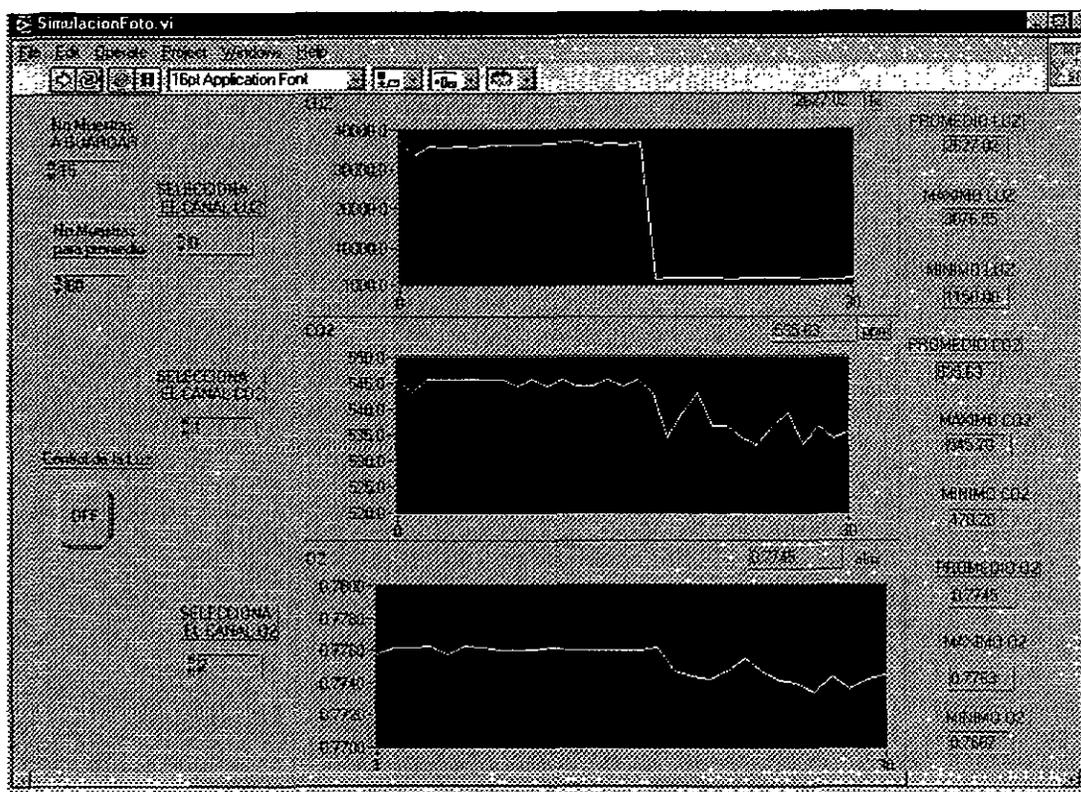


Fig. V.14 Panel frontal del programa simulación de fotosíntesis.

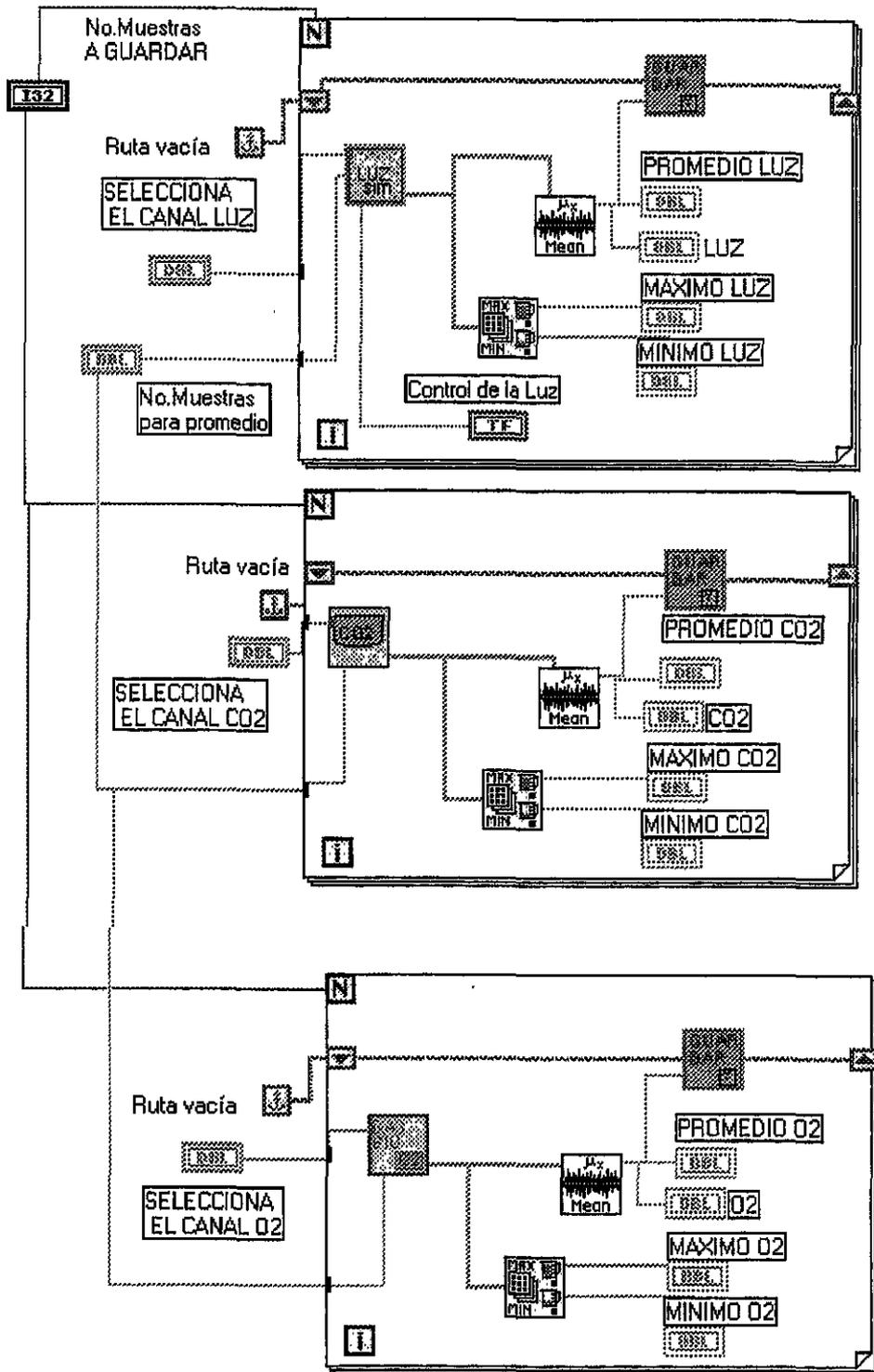


Fig. V.15 Diagrama de bloques del programa simulación de fotosíntesis.

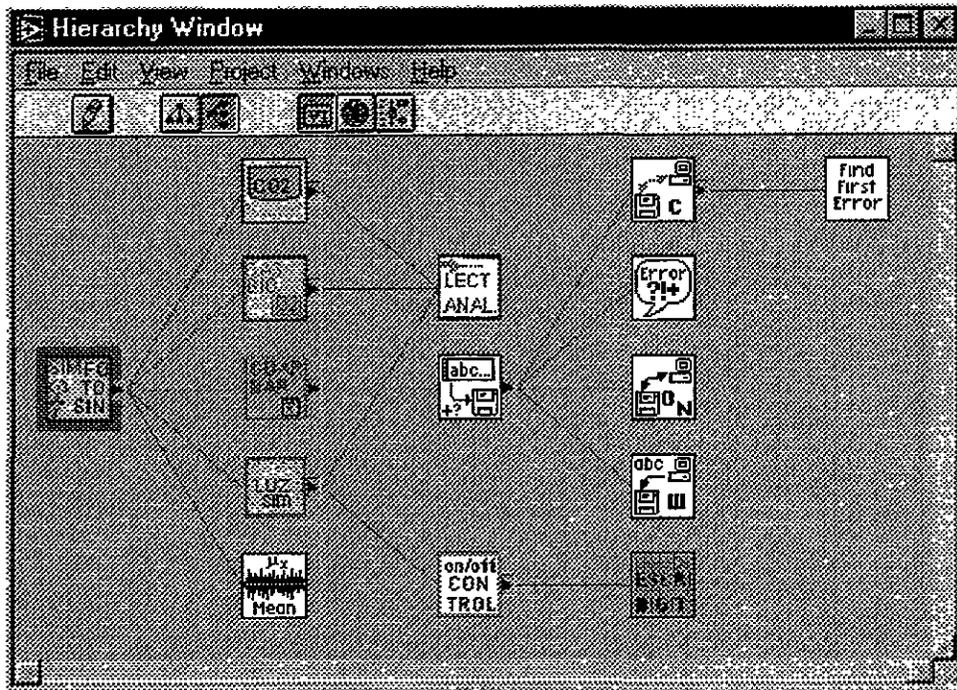


Fig. V.16 Jerarquía del programa simulación de fotosíntesis.

*PROGRAMA CONTROL.VI*

Por medio de este programa podemos controlar el encendido y apagado del foco, a través del canal de escritura analógica, para simular el día y la noche durante el fenómeno de la fotosíntesis. El panel frontal del control se presenta en la figura V.17 y su respectivo diagrama de bloques en la figura V.18.

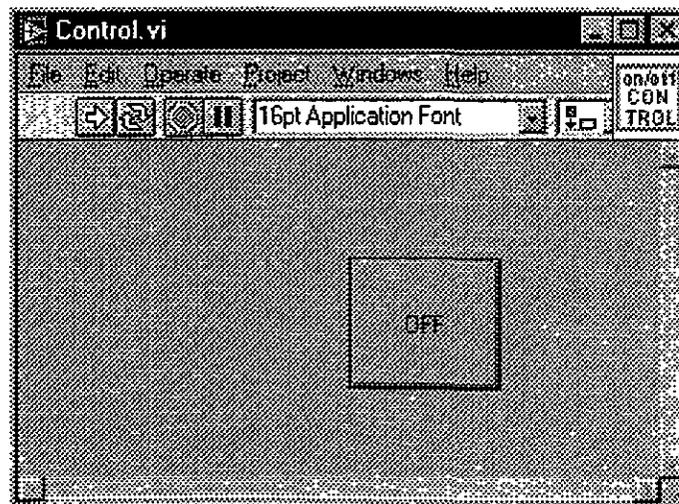


Fig. V.17 Panel frontal del programa control.

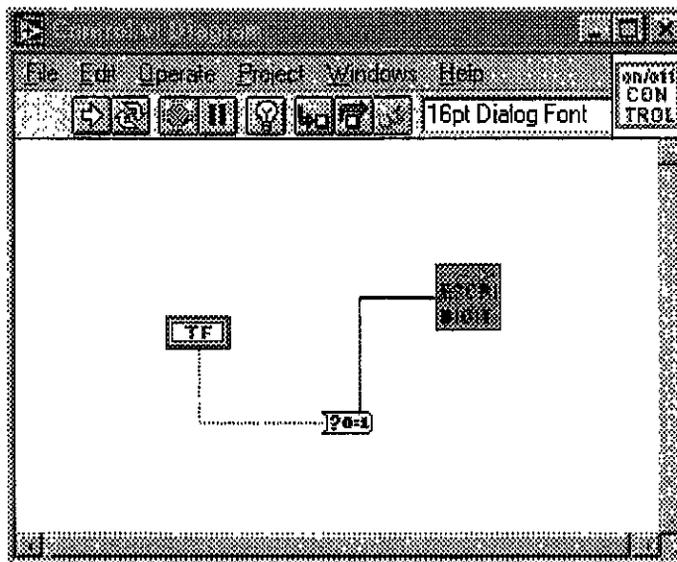


Fig. V.18 Diagrama de bloques del programa control.

*PROGRAMA ESCRITURA.VI*

Este realiza la escritura de un valor específico en el canal digital de la tarjeta, el cual se le proporciona desde el programa de control como un valor 1 o 0

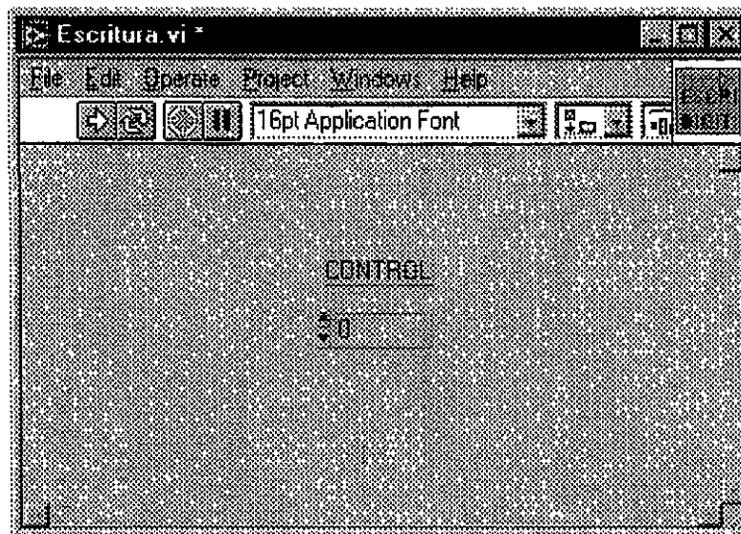


Fig. V.19 Panel frontal del programa escritura.

En el diagrama de bloques de la figura V.20, se muestra el programa que realiza la conexión llamada CIN entre LabVIEW y la tarjeta Procontrol PA-CP12

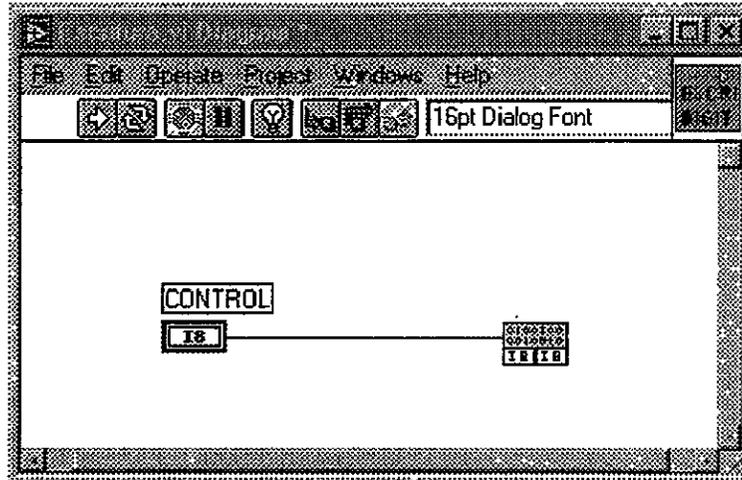


Fig. V.20 Diagrama de bloques del programa escritura.

Además de los programas anteriores, se realizó un programa que lee los datos desde archivos con la finalidad de poder acceder a los datos adquiridos en cualquier momento. En seguida se muestra el IV para fotosíntesis y simulación de fotosíntesis, y posteriormente el que lee los datos para el programa de transferencia de calor.

#### *LECTURA DE ARCHIVOS*

El IV que realiza la lectura de los archivos en donde se grabaron los datos de los experimentos de fotosíntesis se llama LeeFotosíntesis.vi, este programa realiza una lectura desde un archivo especificado. Proporcionando una gráfica para cada variable, así como el análisis estadístico de los datos almacenados en el archivo.

El panel frontal de este programa es muy similar a los de fotosíntesis y simulación de fotosíntesis, obsérvese la figura V.21

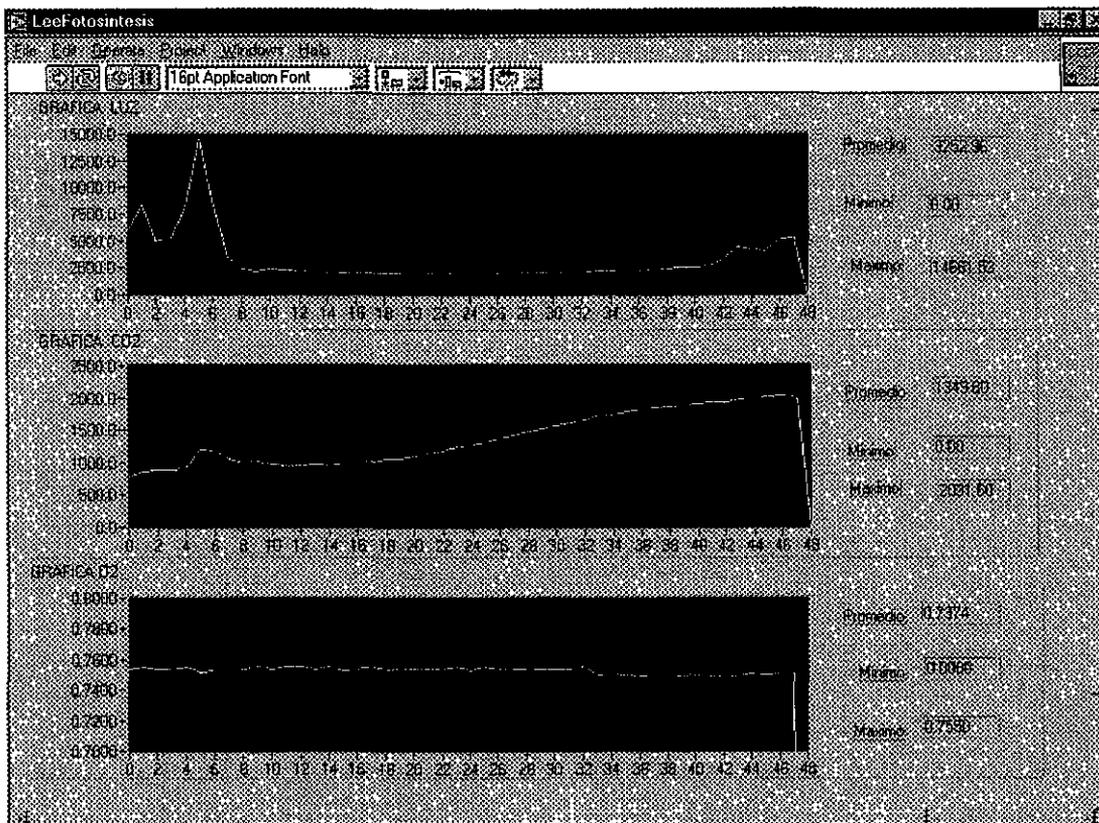


Fig. V.21 Panel frontal del programa que lee fotosíntesis.

En el diagrama de bloques de la figura V.22 se pueden observar los diferentes iconos que se utilizan para realiza la lectura de datos desde un archivo previamente grabado, para lo cual se extraen los datos en forma de cadena de caracteres transformándolos en un arreglo de números que serán introducidos a la gráfica.

Adicionalmente el programa obtiene el promedio y el valor máximo y mínimo de los datos leídos.

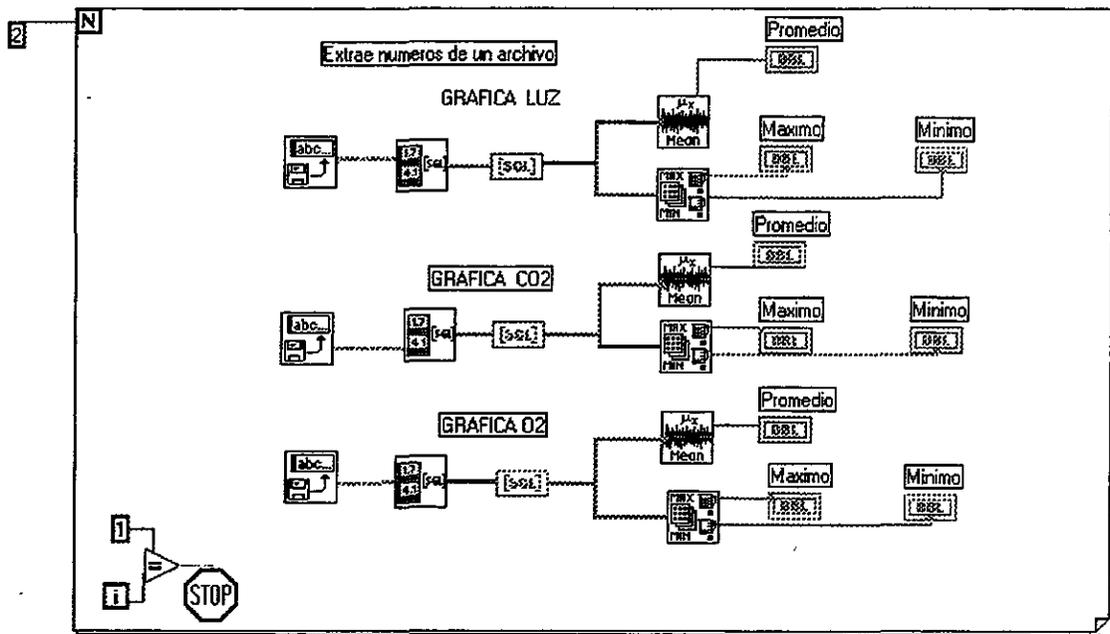


Fig. V.22 Diagrama de bloques del programa lee fotosíntesis.

**TRANSFERENCIA DE CALOR**

El IV llamado transferencia, realiza la adquisición de datos del experimento de transferencia de calor en un sistema acuático, dentro del cual contiene los subIV temperatura.vi y oxidis.vi, los cuales realizan la adquisición de datos de cada variable, mostrado en la figura V.23.

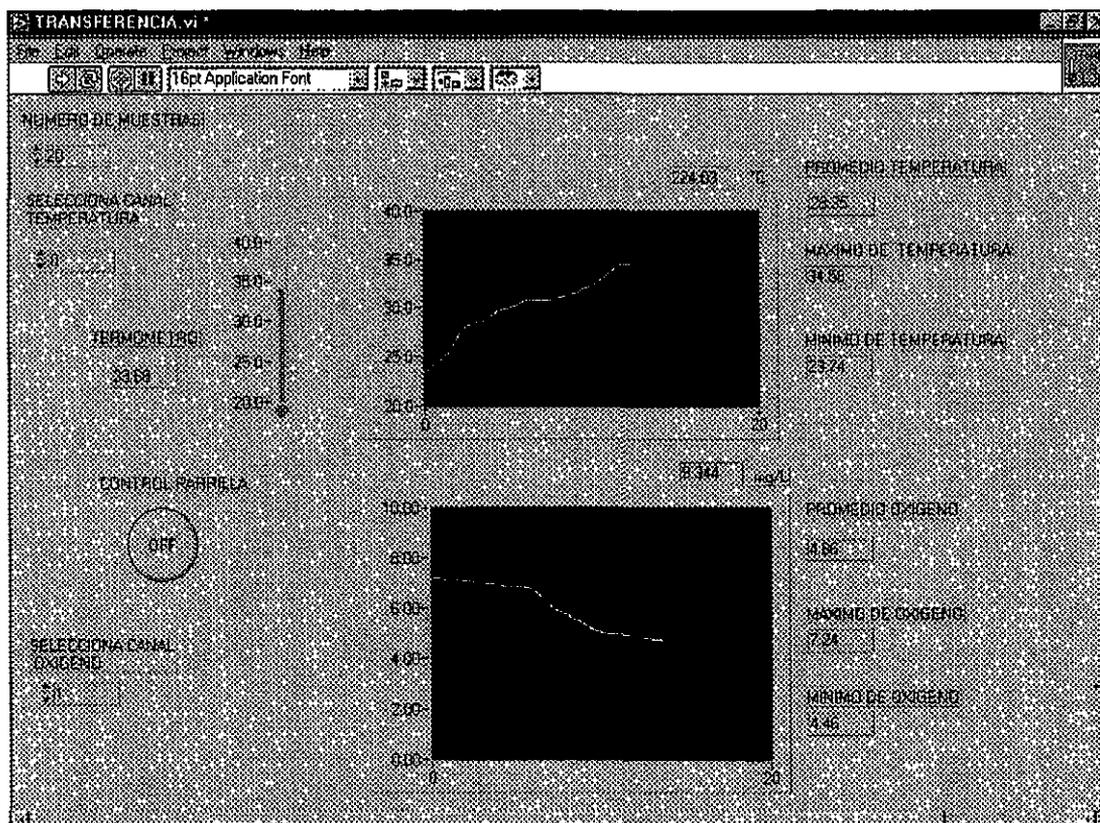


Fig. V.23 Panel frontal del programa transferencia.

Este IV presenta los valores adquiridos en forma gráfica, realiza un análisis numérico proporcionando promedio, valor máximo y mínimo. Además el programa guarda los datos a un archivo que se le indique

A este programa se le agregó una parte de control que maneja el encendido y apagado de una parrilla eléctrica para incrementar la temperatura en el experimento. El diagrama de bloques que ilustra el código de programación se puede observar en la figura V.24, así como la jerarquía de los iconos utilizados en la figura V.25.

Es importante mencionar que además se utilizan los programas<sup>77</sup> de control.vi, guardar.vi, lectura.vi, y escritura.vi, siendo éstos los mismos que se utilizan para el programa de fotosíntesis y simulación de fotosíntesis, con la diferencia que el control enciende y apaga un calentador eléctrico para el caso de transferencia.

<sup>77</sup> Véase las figuras V.10 y V.11 del programa guardar.vi, figuras V.12 y V.13 del programa lectura.vi, figuras V.17 y V.18 del programa control.vi, figuras V.19 y V.20 del programa escritura.vi.

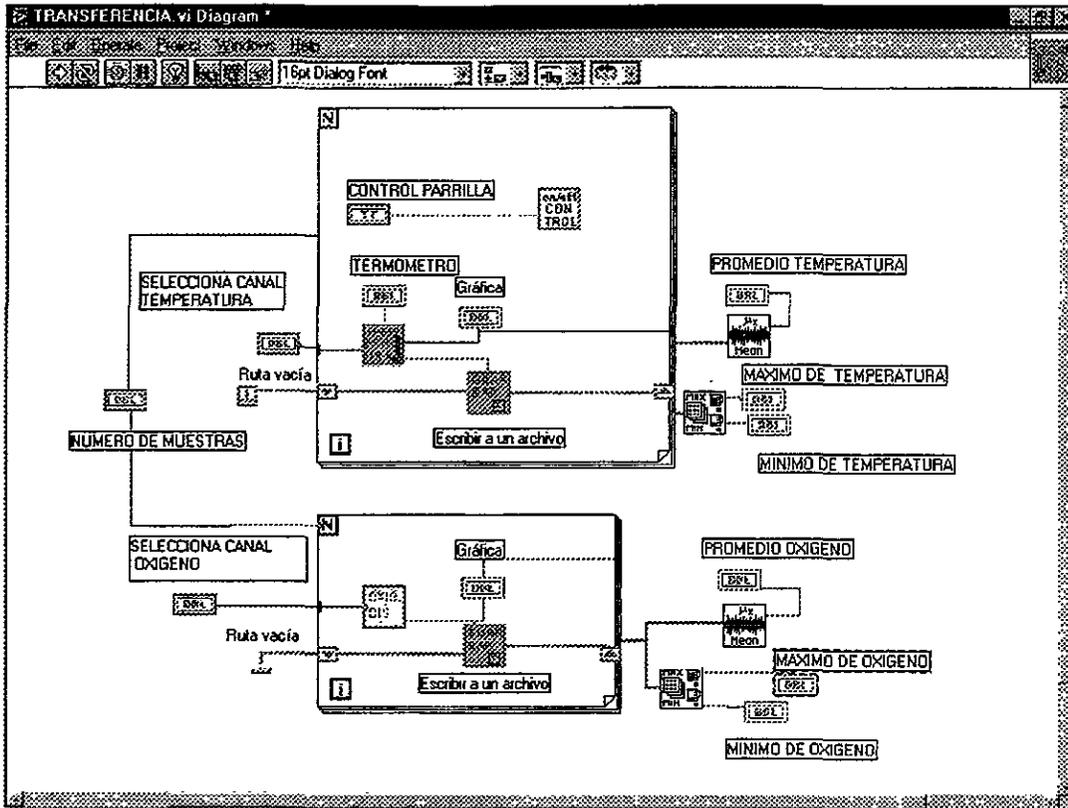


Fig. V.24 Diagrama de bloques del programa transferencia.

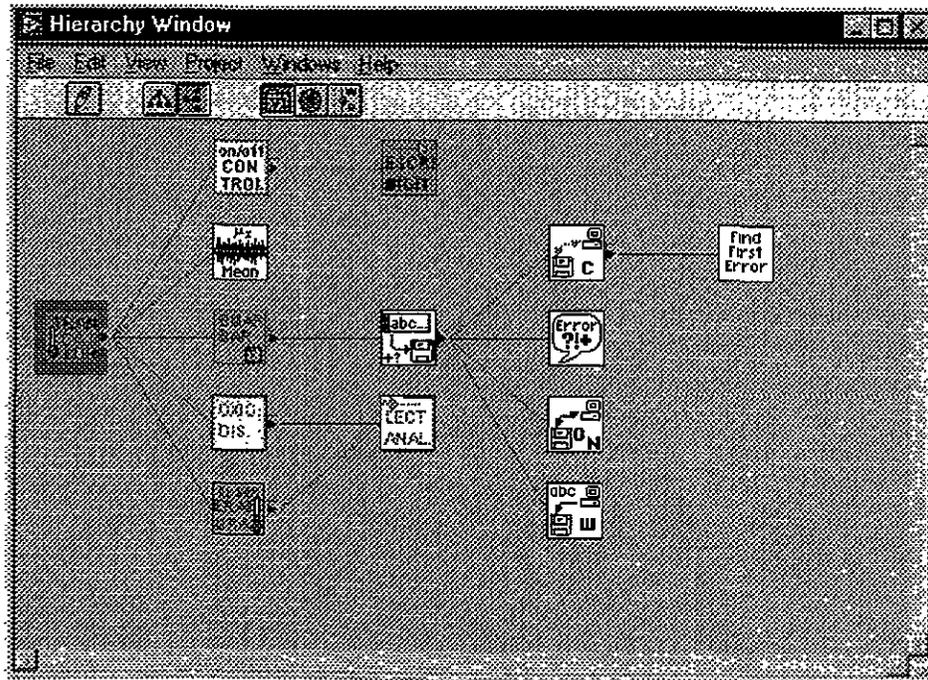


Fig. V.25 Jerarquía del programa transferencia

PROGRAMA TEMPERATURA.VI

Con TEMPERATURA.VI se adquieren los datos correspondientes al sensor de temperatura, vease panel frontal de la figura V.26, teniendo como entrada el valor digital que toma la tarjeta a través del canal analógico, debiendo éste convertirse a valores en grados centígrados mediante el proceso de calibración, el cual es similar para el programa de OXÍGENO DISUELTO.

La calibración se realiza ajustando los valores a una recta, con la ecuación  $y=mx+b$ ; en donde la pendiente  $m=0.1326$  y ordenada al origen  $b=-318.96628$ , estos valores y sus respectivos operandos se pueden observar en la figura V.27.

Cabe aclarar que estos valores se tomaron de comparaciones entre el valor digital registrado por la tarjeta y el valor en temperatura utilizando un termómetro convencional.

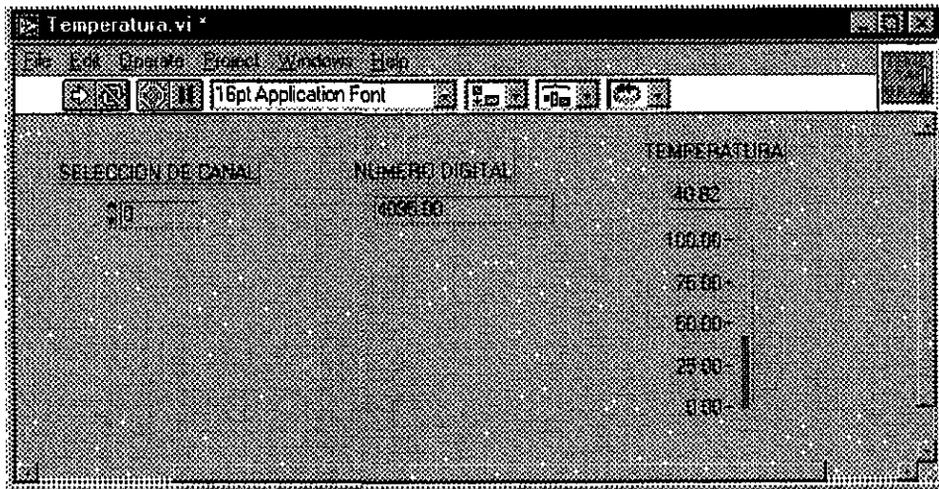


Fig. V.26 Panel frontal del programa temperatura.

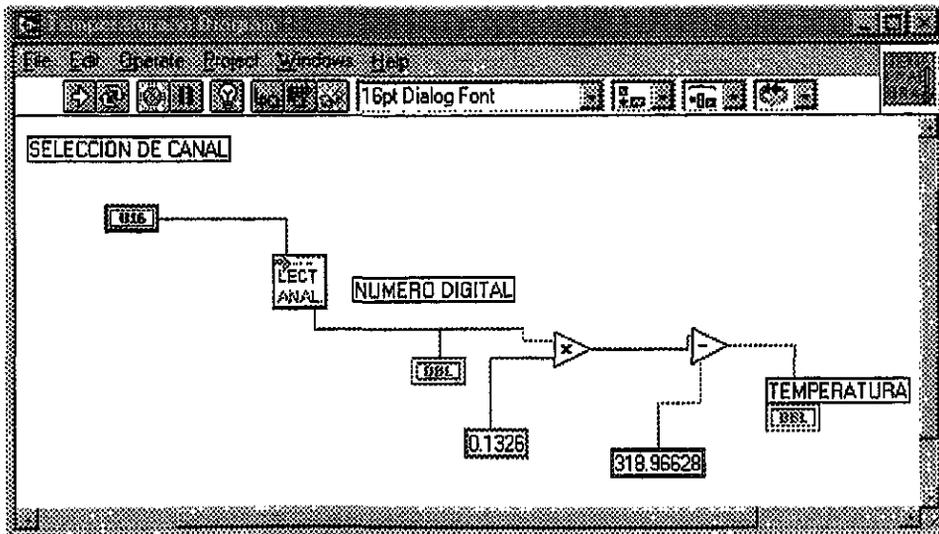


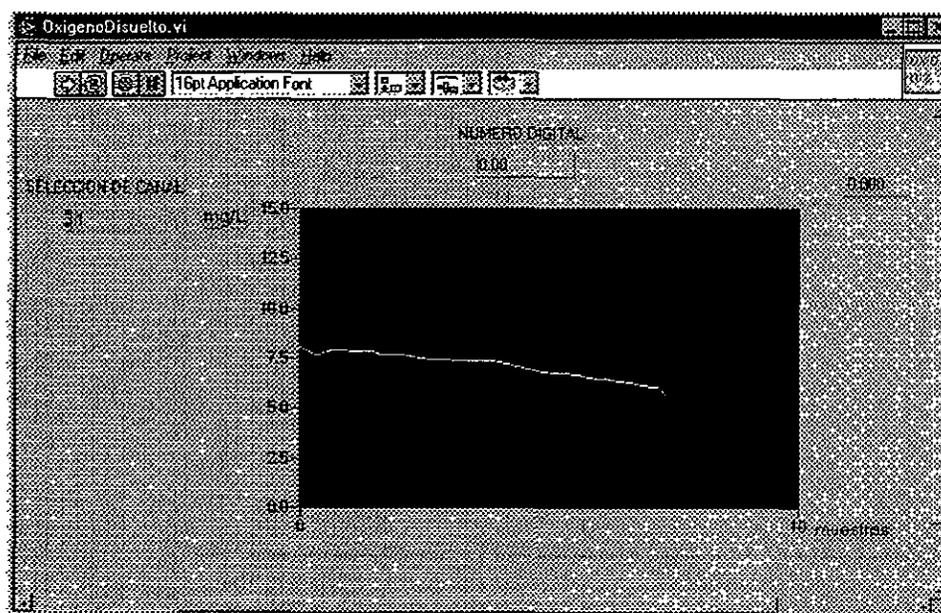
Fig. V.27 Diagrama de bloques del programa temperatura.

*PROGRAMA OXIGENO DISUELTO.VI*

Con este programa se adquieren los datos que genera el sensor de oxígeno disuelto, del cual se muestra el panel frontal en la figura V.28.

Al igual que los demás programas, éste tiene en la entrada un valor digital que proviene de la tarjeta a través del canal analógico seleccionado, por lo cual, es necesario convertirlo a unidades de la variable medida que son mg/L.

La calibración se realiza mediante el ajuste a una recta  $y=mx+b$ , donde la pendiente  $m=0.0090237$  y la ordenada al origen  $b=-19.2341$ , ambas operaciones se observan en la figura V.29 donde se localiza el código en un diagrama de bloques



ig. V. 28. Panel frontal del programa oxígeno disuelto.

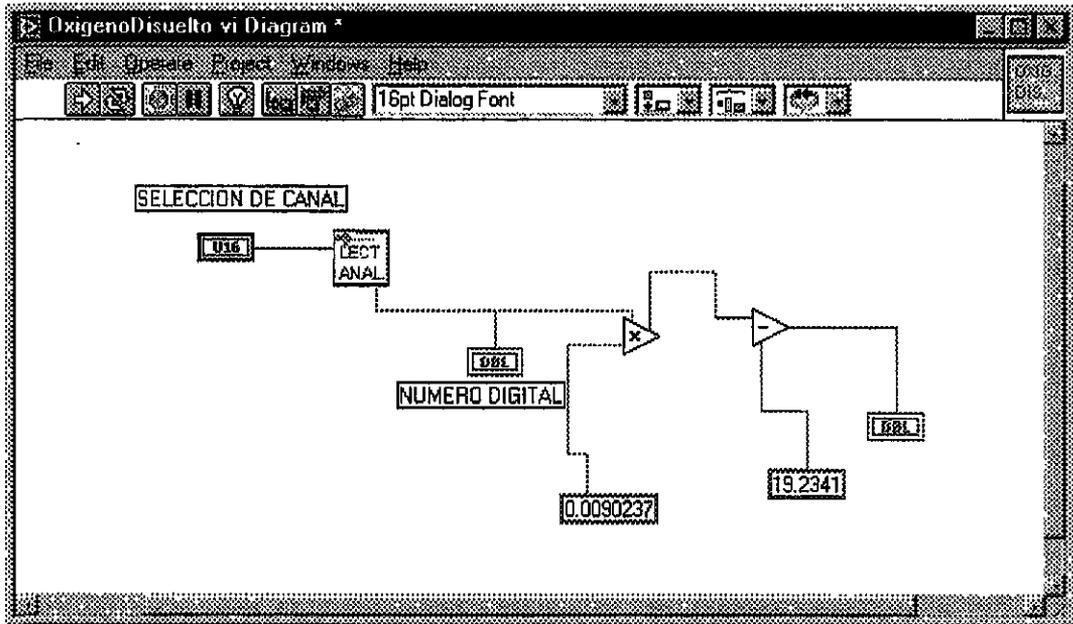


Fig. V.29 Diagrama de bloques del programa oxígeno disuelto.

Para el caso de transferencia de calor al igual que para fotosíntesis se realizó un programa que permita leer los datos almacenados en archivos anteriormente, facilitando la consulta de los datos.

#### *LECTURA DE ARCHIVOS PARA TRANSFERENCIA DE CALOR*

El IV lecturatransferencia.vi, lee los archivos de temperatura y oxígeno disuelto para el experimento de transferencia de calor, al ejecutarlo se solicita al usuario introducir el nombre de los archivos a leer. Los valores se muestran en una gráfica para cada variable, así como su respectivo análisis estadístico como promedio, valor máximo y mínimo del total de datos coleccionados.

El panel frontal donde se visualizan los resultados y el diagrama de bloques que muestra el código de programación se observan respectivamente en la figuras V.30 y V.31.

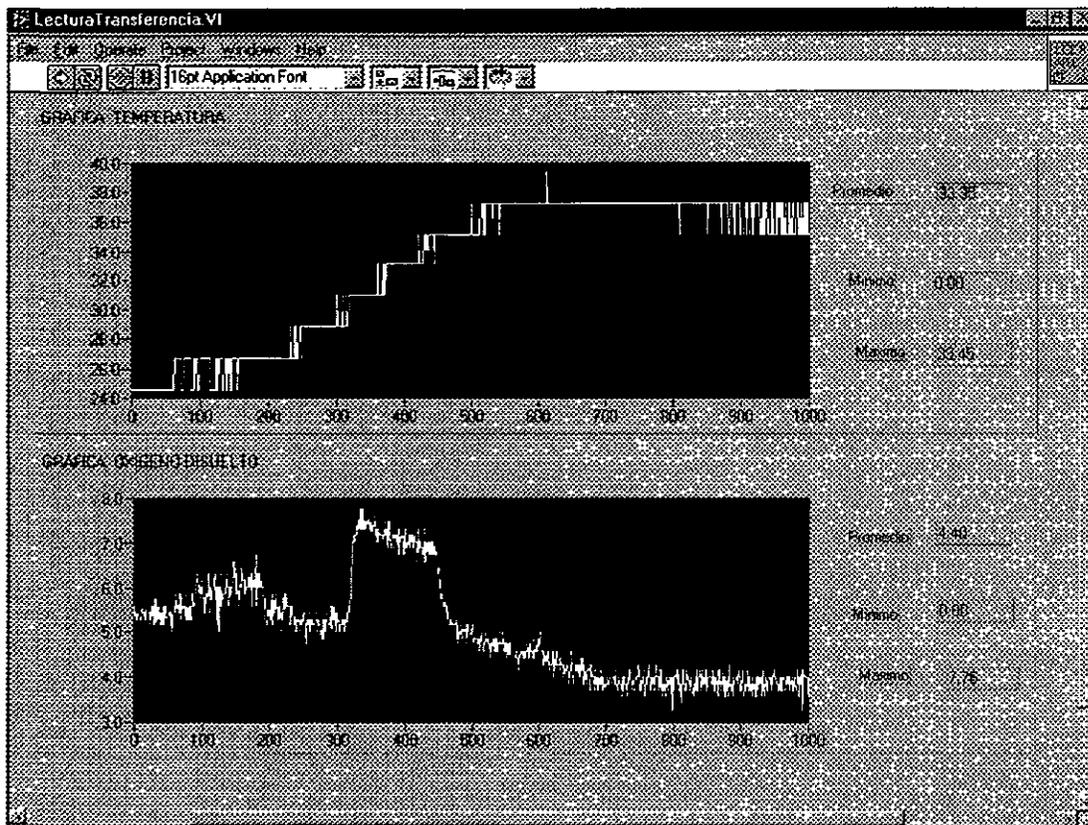


Fig. V.30 Panel frontal del programa lee transferencia.

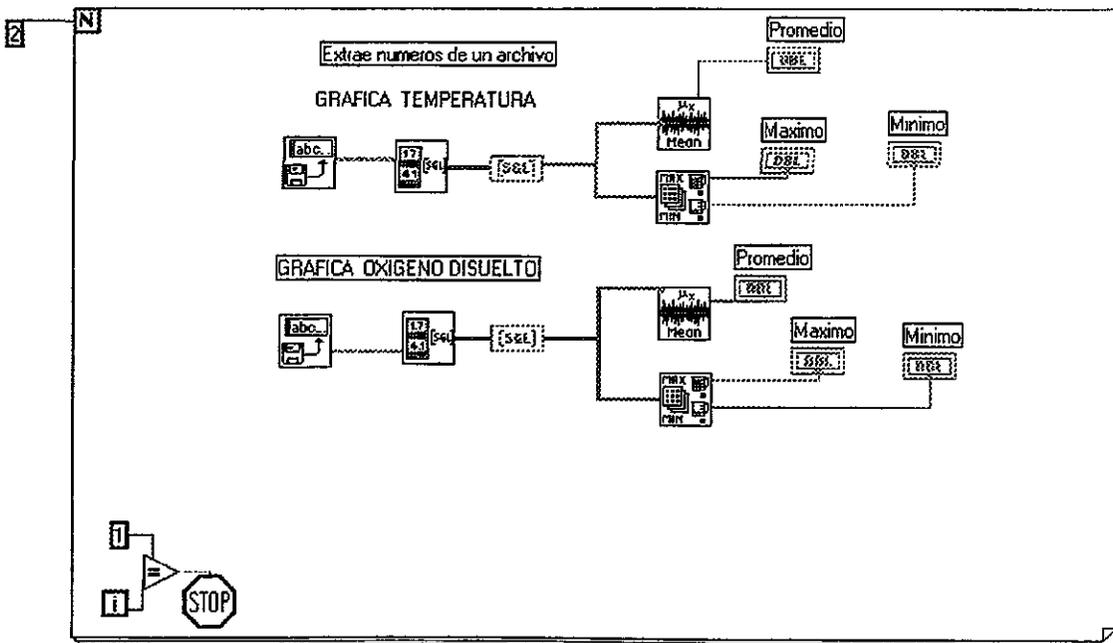


Fig. V. 31 Diagrama de bloques del programa lee transferencia.

## PROGRAMA MICRÓFONO.VI

Al ejecutar el programa microfono.vi se captura una señal de sonido a través de un micrófono, a ésta se le aplica transformada rápida de Fourier para obtener el espectro en frecuencias, mostrado en el panel frontal contenido en la figura V.32.

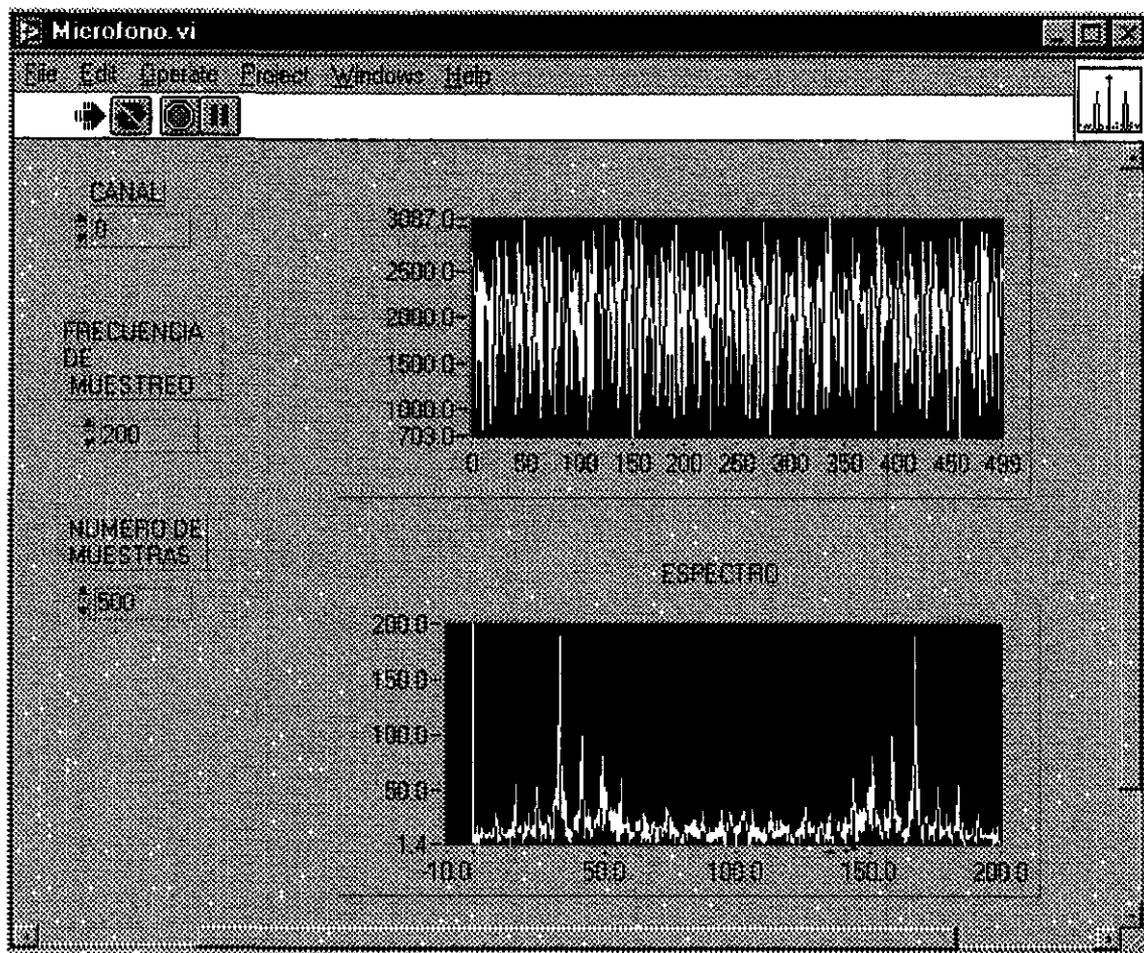


Fig. V.32 Panel frontal del programa micrófono.

En el diagrama de bloques de la figura V.33 se muestra el código de programación en el cual se aplica la transformada rápida de Fourier real, estos valores se muestran en una gráfica para la que se realizó una conversión de números complejos a polares. Además se proporciona una gráfica de los datos antes de ser transformados.

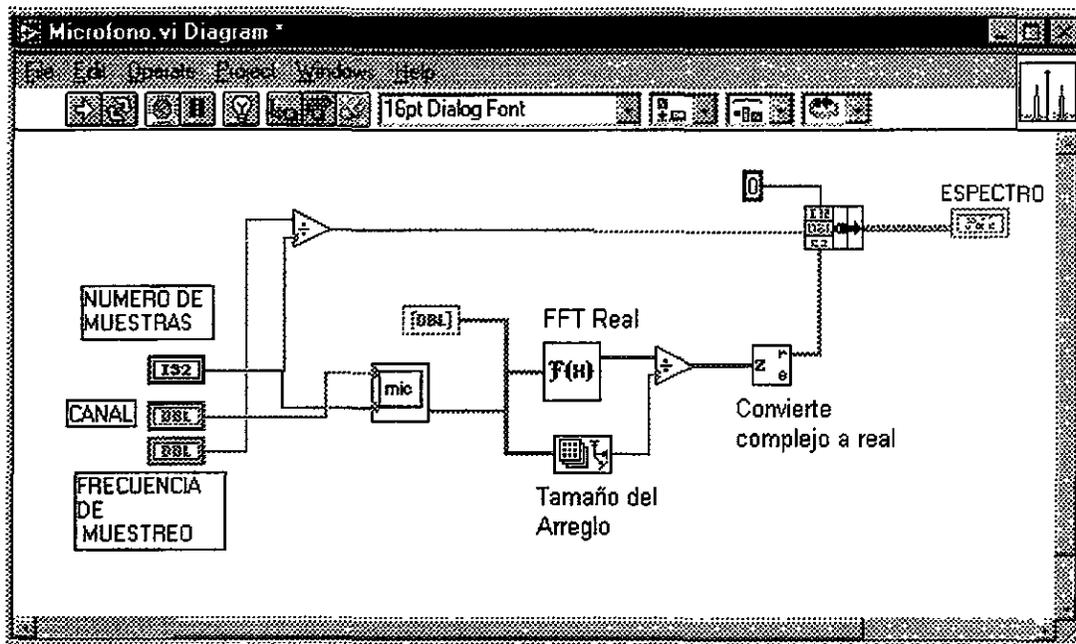


Fig. V.33. Diagrama de bloques del programa micrófono.

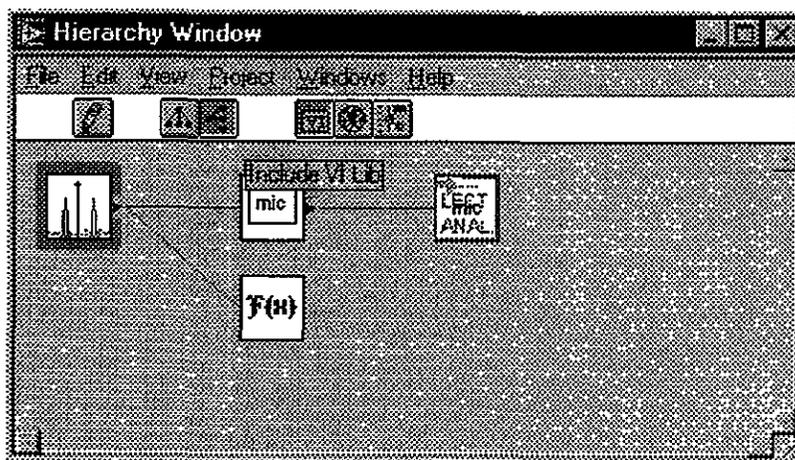


Fig. V.34 Jerarquía del programa micrófono.

PROGRAMA MICRO.VI

La adquisición de la señal la realiza este programa, véase panel frontal de la figura V.35. el diagrama de bloques mostrado en la figura V.36 presenta la estructura del programa, el cual hace una llamada a otro llamado lecturamic.vi.

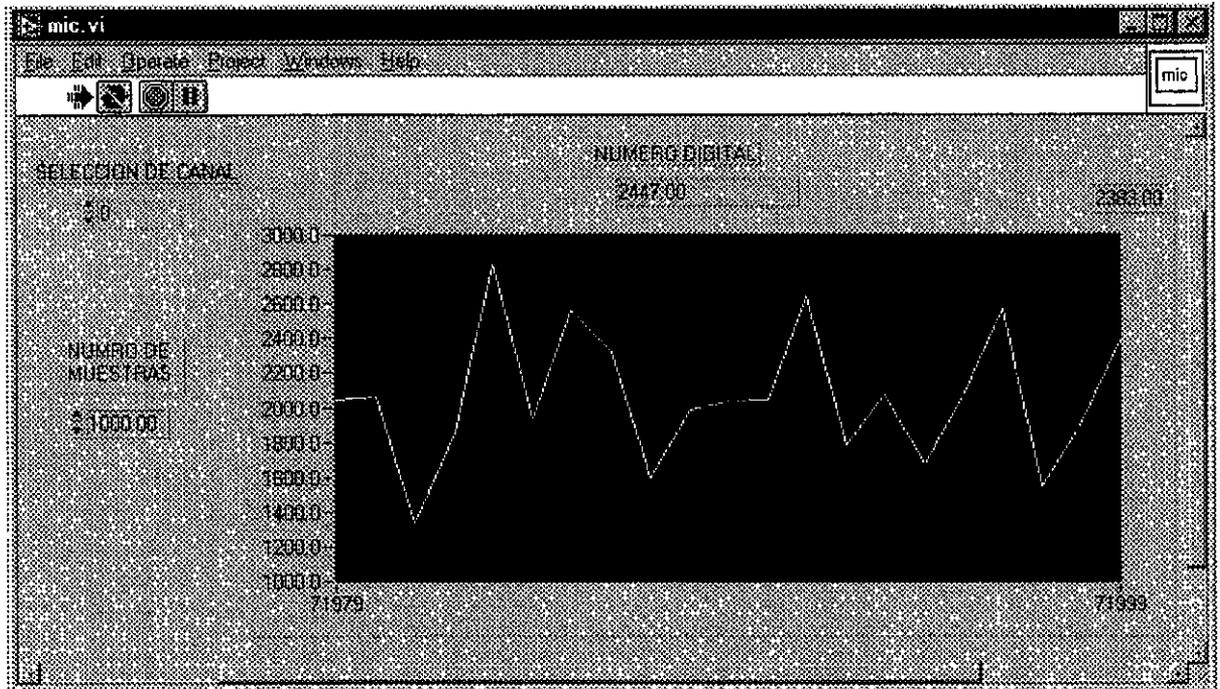


Fig. V.35 Panel frontal del programa micro.vi

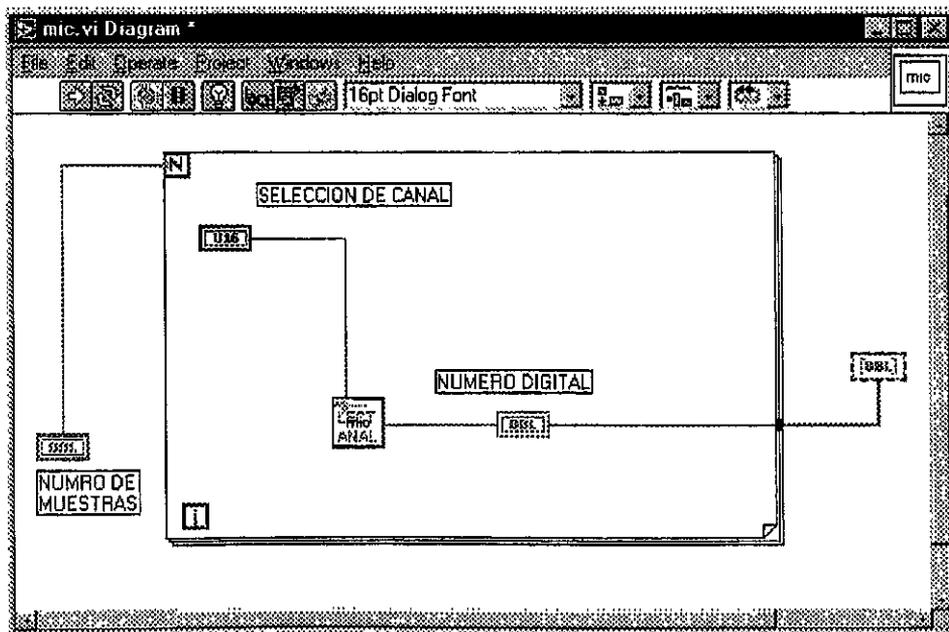


Fig. V.36 Diagrama de bloques del programa micro.vi

PROGRAMA LECTURAMIC.VI

Con este programa se obtiene la lectura del micrófono desde un canal analógico, para lo cual se realiza un código interfase denominado CIN que permite la conexión entre la tarjeta ProControl y LabVIEW.

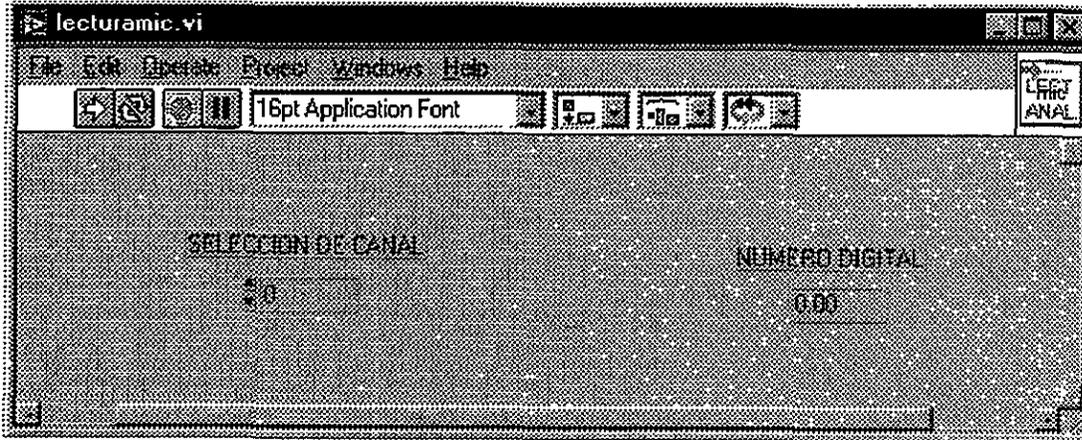


Fig. V.37 Panel frontal del programa lecturamic.vi

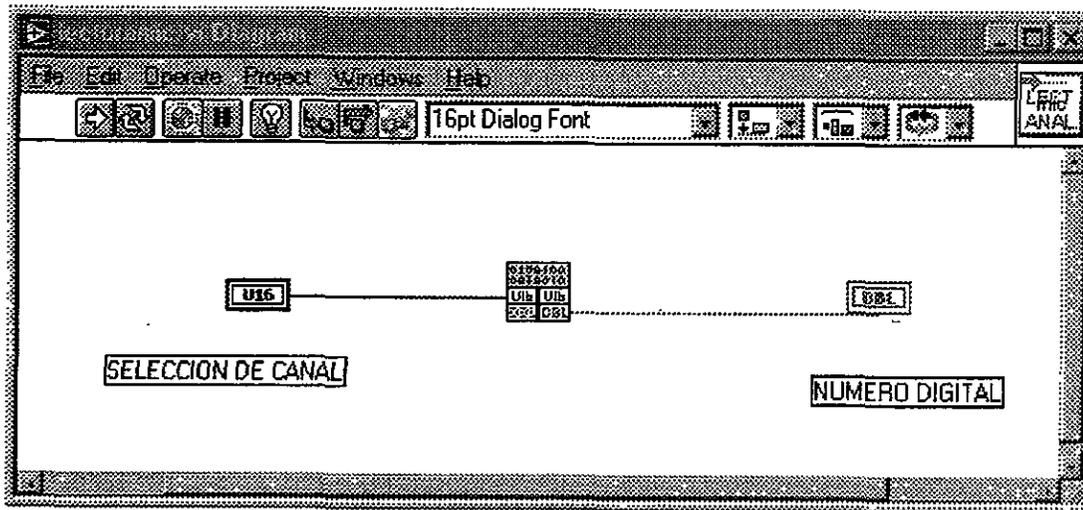
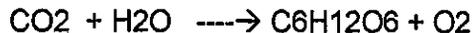


Fig. V.38 Diagrama de bloques del programa lecturamic.vi

### V.3 PROPUESTA DE PRÁCTICAS EXPERIMENTALES

#### FOTOSÍNTESIS

La fotosíntesis es el proceso mediante el cual las células vegetales transforman el dióxido de carbono y agua en azúcares y oxígeno.



Este proceso se realiza en los cloroplastos, organelos especializados que presentan 2 membranas, la externa e interna que delimitan al organelo y un tercer tipo de membrana interna que forma estructuras llamadas tilacoides. Los cloroplastos, se encuentran presentes en células foliares de plantas superiores y en algas fotosintéticas.

Los fenómenos químicos de la fotosíntesis ocurren en 2 fases:

- I) Fase luminosa (fotosíntesis): Mediante este proceso las plantas obtienen la energía para sus necesidades vitales a partir de la luz. Estas células también utilizan la energía solar para incorporar el carbono del anhídrido carbónico de la atmósfera a la molécula orgánica elemental de glucosa. (Absorción de CO<sub>2</sub> (bióxido de carbono))
- II) Fase oscura (fotorespiración): Este proceso consiste en la asimilación enzimática y la conversión del CO<sub>2</sub> en carbohidratos. En esta fase las células obtienen su energía mediante la combustión u oxidación de estos combustibles complejos. Fabricados por otras células, en el proceso llamado respiración usando oxígeno molecular (O<sub>2</sub>).

Al analizar las dos fases que ocurren en el proceso de fotosíntesis, se observa que la luz es el factor primordial para que ocurra este proceso.

En resumen podemos mencionar que el O<sub>2</sub> y CO<sub>2</sub> se comportan de manera inversa en las dos fases, es decir:

#### FOTOSÍNTESIS.

Absorción de CO<sub>2</sub>  
Liberación de O<sub>2</sub>

#### FOTORESPIRACIÓN.

Absorción de O<sub>2</sub>  
Liberación de CO<sub>2</sub>

La fotosíntesis es un proceso que transforma en carbono orgánico el gas carbónico tomado del aire o disuelto en el agua. La sustancia orgánica primaria formada es la glucosa, que típicamente se almacena bajo la forma de almidón. En el proceso interviene un pigmento, la clorofila, que es una sustancia capaz de absorber las radiaciones luminosas. Esta captación de energía luminosa se realiza en una primera etapa de la fotosíntesis, en la cual se produce energía química en forma de moléculas de

adenosintrifosfato (ATP) y se desprende oxígeno, que procede de la escisión de una molécula de agua. En una segunda etapa, que se denomina fase oscura porque puede tener lugar en ausencia de la luz, el dióxido de carbono ( $\text{CO}_2$ ) se combina con una pentosa para formar glucosa a través de una serie de reacciones químicas.

#### *OBJETIVO:*

- Estudiar el efecto de la luz en el proceso de la fotosíntesis.
- Comparar el comportamiento de los gases  $\text{O}_2$  y  $\text{CO}_2$  durante los procesos de fotosíntesis y fotorespiración.

#### *MATERIAL*

- Sensor de  $\text{CO}_2$
- Sensor de gas biológico
- Sensor de luz
- Energía solar
- Planta acuática (Elodea)
- Pecera con 2 orificios para conectar los sensores

#### *METODOLOGÍA*

1. Montar el experimento como se muestra en la figura V.39.
2. Conectar los 3 sensores a la interface, seleccionando un canal para cada uno de ellos.
3. Ejecutar el programa que adquiere los datos de la fotosíntesis (fotosíntesis.vi), para que corra durante un periodo de 24 horas, con intervalos de muestreo de cada 30 minutos; guardando estos valores en archivos para cada una de las variables (luz,  $\text{CO}_2$  y  $\text{O}_2$ ).
4. Posteriormente a la adquisición de datos, ejecutar el programa que lee los datos adquiridos desde el archivo donde se guardaron.
5. Analizar los resultados obtenidos mediante las variaciones del oxígeno y dióxido de carbono respecto a los cambios de luz, es decir, en la fotosíntesis y en la fotorespiración.

#### *DESARROLLO:*

Se ejecuta el programa que adquiere los datos de la fotosíntesis, el cual adquiere datos para las variables luz, dióxido de carbono y presión de gas biológico.

Introducir una planta acuática (Elodea, recomendada), en una pecera pequeña, se introduce la manguera de plástico al agua y el sensor de dióxido de carbono colocado aproximadamente 5cm por encima de la superficie del agua. Se sella la pecera y se inicia la toma de lecturas.

Cabe mencionar que el sensor utilizado para medir el oxígeno funciona midiendo la presión del gas biológico que desprende la planta, en este caso el oxígeno molecular el cual se utiliza para observar las variaciones de este gas durante la fotosíntesis.

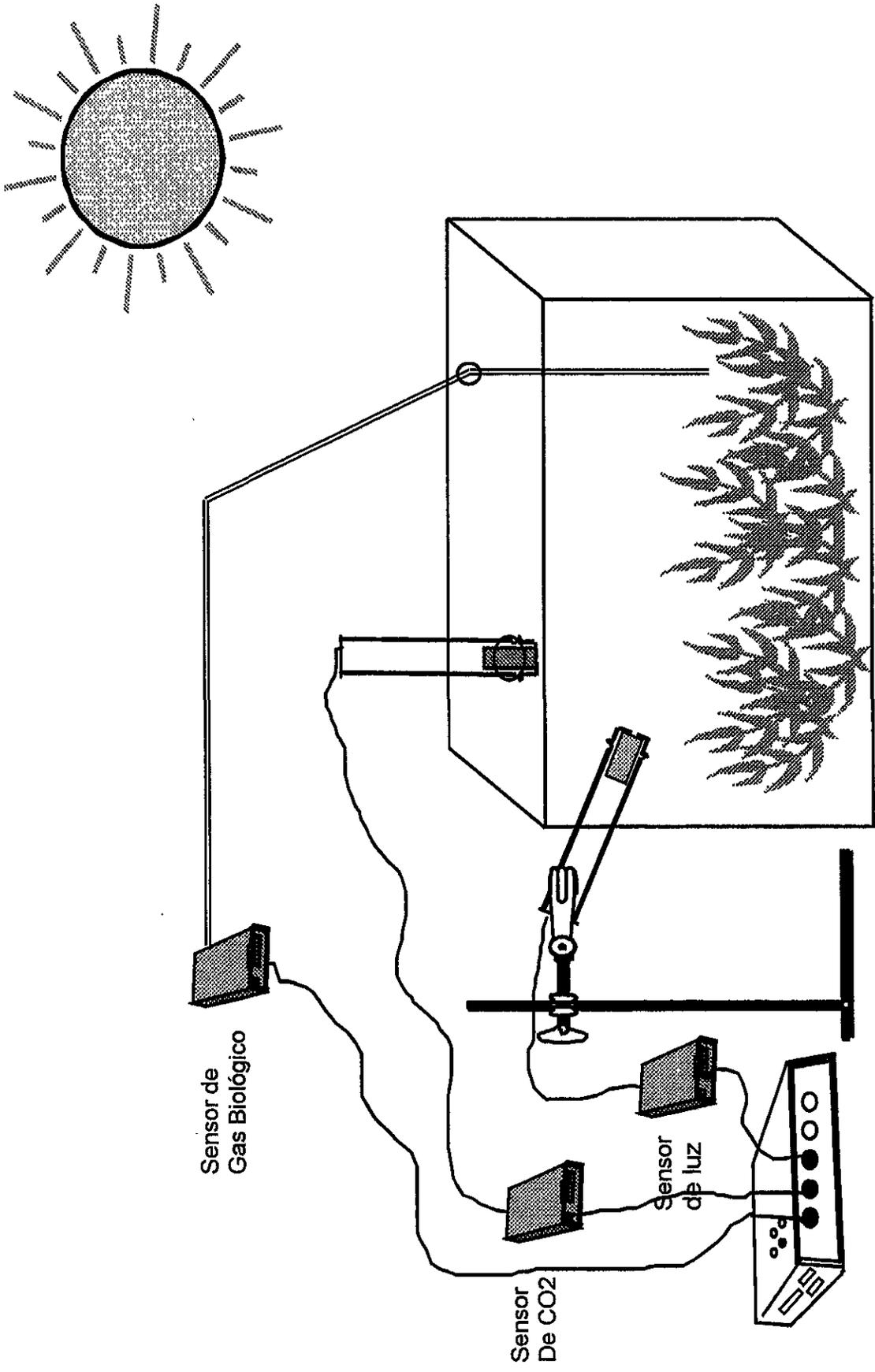


Fig V.39. Esquema para la práctica de Fotosíntesis.

## ***SIMULACIÓN DE FOTOSÍNTESIS***

La simulación de un ciclo de fotosíntesis se basa en simular la luz del día y la noche con la luz que emite un foco. El tiempo que se encienda y apague el foco, se recomienda de 15 minutos cada uno, pero este valor se puede modificar hacia un tiempo mayor para lograr una mejor apreciación del fenómeno.

### ***OBJETIVO:***

Observar las variaciones del fenómeno de fotosíntesis mediante la simulación de energía luminosa del sol por medio de la luz que emite un foco

### ***MATERIAL:***

- Sensor de CO<sub>2</sub>
- Sensor de O<sub>2</sub>
- Sensor de luz
- Foco
- Planta acuática (Elodea)
- Pecera pequeña sellada con dos orificios

### ***METODOLOGÍA:***

1. Construir el experimento como se muestra en la figura V.40, en donde se simulará la luz del sol con un foco.
2. Ejecutar el programa que adquiere datos para la simulación de la fotosíntesis (simulaciónfoto.vi).
3. Activando el control del foco, se simula el día, para observar el fenómeno de fotosíntesis por un lapso de quince minutos.
4. Apagar el foco con el botón de control, simulando la noche, para observar el fenómeno de fotorespiración por un lapso de quince minutos.
5. Ejecutar el programa de lectura de datos desde un archivo.
6. Analizar los resultados obtenidos observando la variación de las variables en ambos ciclos simulados.

### ***DESARROLLO:***

Se ejecuta el programa que adquiere los datos para simulación de la fotosíntesis, el cual adquiere datos para las variables luz, dióxido de carbono y presión de gas biológico.

Se realizan los pasos de igual manera que la practica de fotosíntesis. Se enciende el foco por 15 minutos y después se apaga otros 15 minutos, para simular el día y la noche.

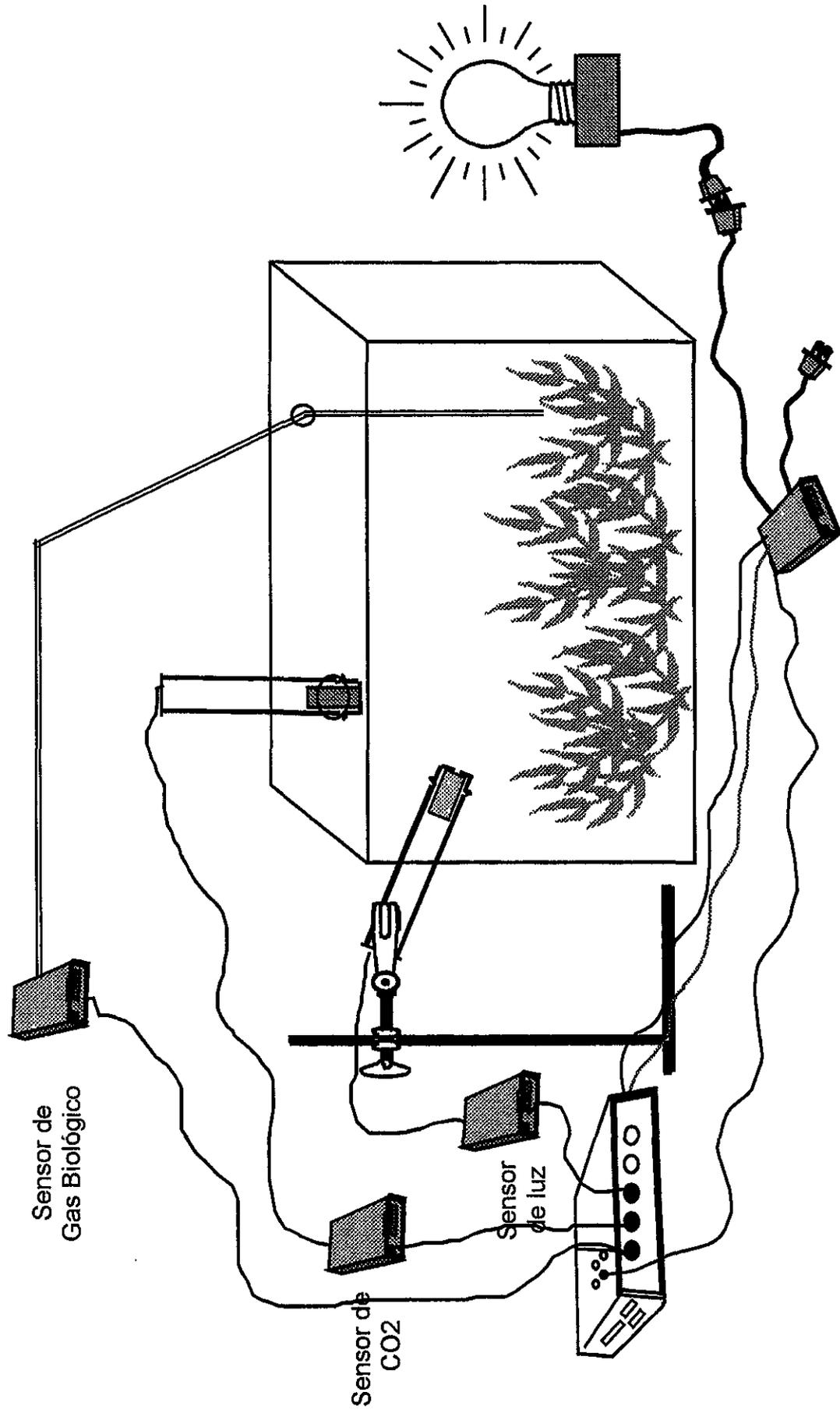


Fig. V.40. Esquema de conexión para la práctica de Simulación de Fotosíntesis.

## TRANSFERENCIA DE CALOR EN UN SISTEMA ACUÁTICO

La transferencia de calor de un sistema acuático proporciona cambios en cada una de las variables del sistema, como lo es el caso del oxígeno disuelto en el agua.

La transferencia de calor en el agua, produce la liberación de moléculas hacia la superficie en forma de vapor. Es por esto, que cuando se hierve agua se produce vapor, en el cual el oxígeno se desprende y el volumen inicial del líquido disminuye debido al decremento de moléculas de oxígeno.

Cuando un líquido se encuentra a temperatura ambiente tiene mayor concentración de oxígeno disuelto, al incrementarse la temperatura el nivel de concentración de oxígeno disminuye.

### OBJETIVO

- Estudiar la variación de los niveles de oxígeno disuelto en el agua dependiendo del incremento o disminución de la temperatura debido al calor transmitido por corrientes de convección.

### MATERIAL

- Sensor de oxígeno disuelto
- Sensor de temperatura
- Recipiente de un litro
- Parrilla térmica

### METODOLOGIA

1. Construir el experimento como se muestra en la figura V. 41.
2. Conectar los 2 sensores a la interface, seleccionando un canal para cada uno de ellos.
3. Ejecutar el programa que adquiere los datos de transferencia de calor (transferencia.vi), en cualquier período de tiempo, guardando estos valores en archivos para cada una de las variables temperatura y oxígeno disuelto.
4. Posteriormente, a la adquisición de datos, ejecutar el programa que lee los datos adquiridos desde el archivo donde se guardaron.

### DESARROLLO

Se corre el programa que adquiere los datos de la transferencia de calor por medio de los sensores de temperatura y oxígeno disuelto.

Colocando un recipiente de cristal con 500mL de agua e introduciendo ambos sensores al mismo nivel. A continuación se incrementa la temperatura, encendiendo el calentador eléctrico, hasta obtener 35°C; al llegar a éste valor se debe apagar el calentador.

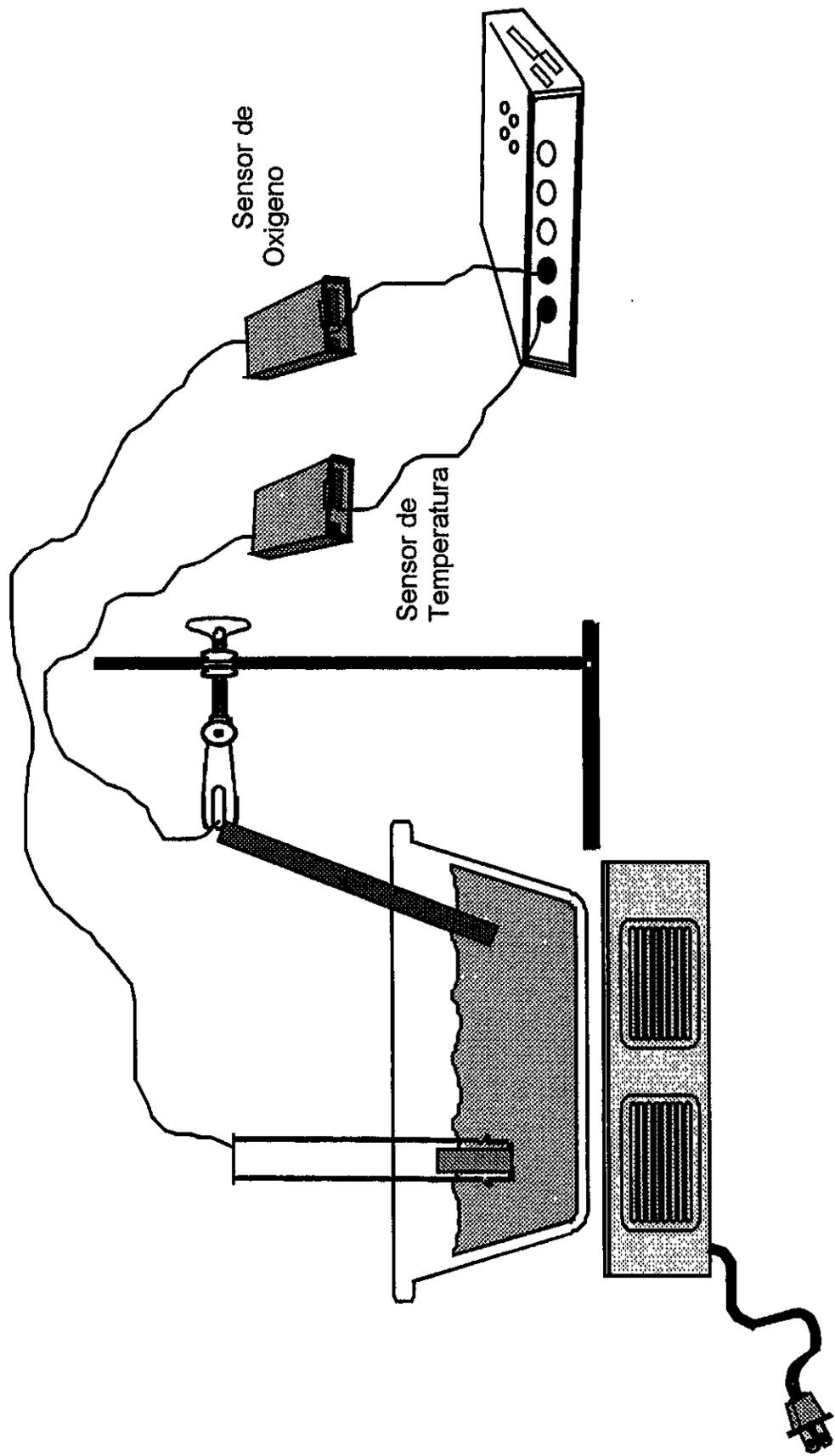


Fig.V.41. Esquema para la práctica de Transferencia de Calor

## ANÁLISIS DE SONIDO

La aplicación de la transformada rápida de Fourier a una señal de sonido permite conocer el espectro en frecuencias de ésta, para describir los rangos de valores en los que digitalmente la señal tiene variaciones significativas de amplitud y el número de armónicas que presenta.

### OBJETIVO:

Estudiar la variación del timbre cuando un mismo sonido es emitido por varias fuentes sonoras.

### MATERIAL:

- Micrófono
- Fuente sonora

### METODOLOGÍA:

1. Conectar el micrófono a cualquiera de las entradas de la interface.
2. Ejecutar el programa micrófono.vi, aplicando varias señales sonoras al micrófono.
3. Variar la fuente sonora.
4. Analizar los resultados obtenidos, respecto a las variaciones de los espectros en frecuencias de las señales adquiridas.

### DESARROLLO:

Ejecutar el programa que adquiere los datos del micrófono con tres diferentes voces de personas. Se propone la voz de un niño, de una mujer y de un hombre, o en su defecto voces totalmente distintas.

Analizar los espectros en frecuencias obtenidos para cada persona, observando las armónicas que presenta y su amplitud relativa.

**V.4 EJEMPLOS DE REPORTES DE LAS ACTIVIDADES EXPERIMENTALES.**

**FOTOSÍNTESIS**

Al terminar la práctica experimental en donde se analizaron las variables que intervienen en la fotosíntesis se obtuvieron los siguientes resultados.

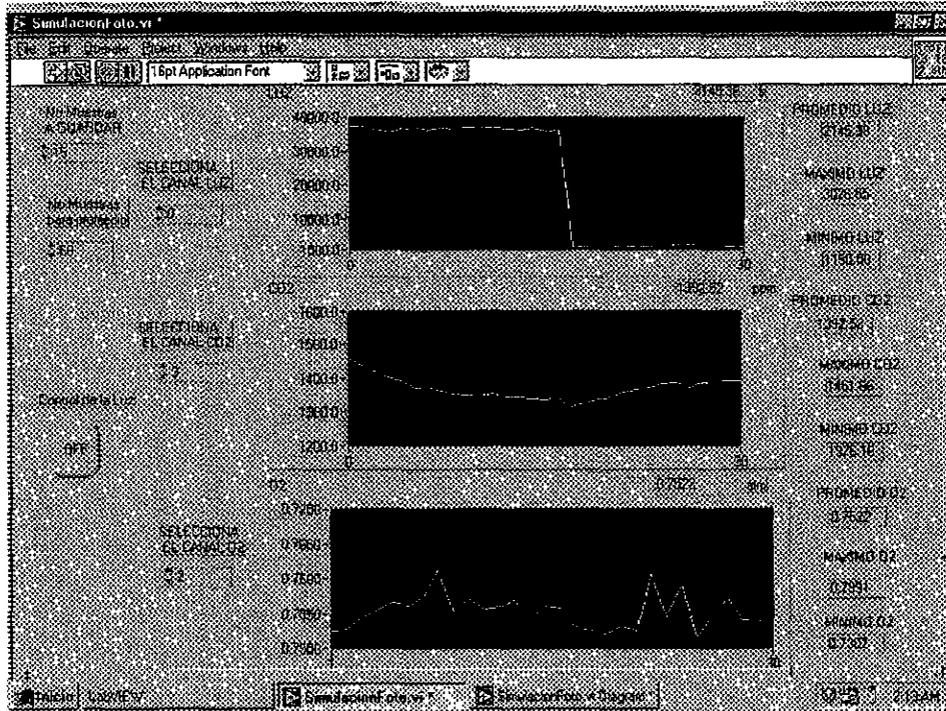


Fig. V.42 Panel frontal de los resultados de la fotosíntesis.

Estos valores se presentan en la tabla V.1.

VARIABLE	PROMEDIO	VALOR MÁXIMO	VALOR MÍNIMO
LUZ	2145.38	3076.55	1150.00
CO2	1392.52	1451.66	1376.16
OXIGENO	07522	0.7991	0.7307

Tabla V.1 Resultados de la fotosíntesis completa.

### SIMULACIÓN DE FOTOSÍNTESIS

En la simulación de la fotosíntesis, las variaciones se presentan de forma similar a las anteriormente mostradas, con la diferencia que el tiempo de muestreo se reduce a treinta minutos, simulando el día y la noche con un foco para observar las variaciones respectivos a los ciclos de fotosíntesis y fotorespiración.

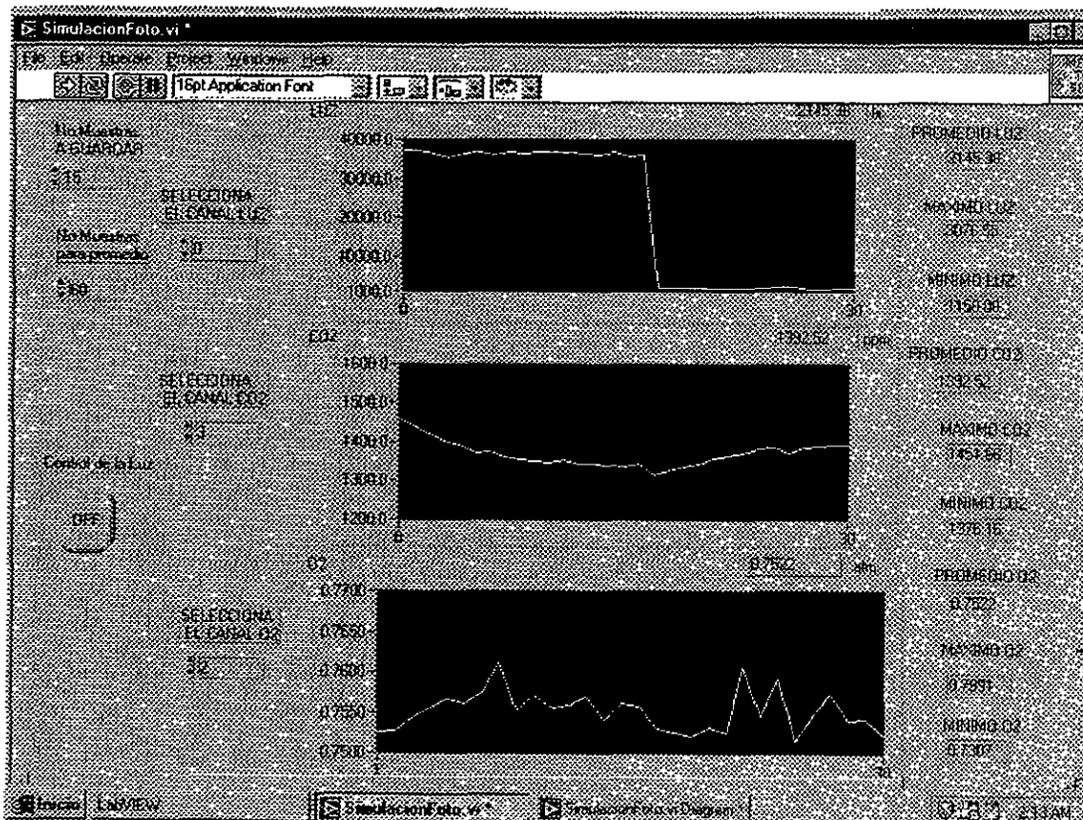


Fig. V.43 Panel frontal de los resultados de la simulación de fotosíntesis.

Los valores obtenidos se presentan en la tabla V.2.

VARIABLE	PROMEDIO	VALOR MAXIMO	VALOR MINIMO
LUZ	2145.38	3076.55	1150.00
CO2	1392.52	1451.66	1376.16
OXIGENO	0.7522	0.7991	0.7307

Tabla V.2 Resultados de la simulación de la fotosíntesis.

Los valores obtenidos en esta simulación varían cuantitativamente con respecto a los de la fotosíntesis completa pero muestra la misma tendencia cualitativamente, considerando que la intensidad del foco y del sol son diferentes.

*TRANSFERENCIA DE CALOR EN UN SISTEMA ACUÁTICO.*

Los resultados de la práctica de transferencia de calor en un sistema acuático son los siguientes.

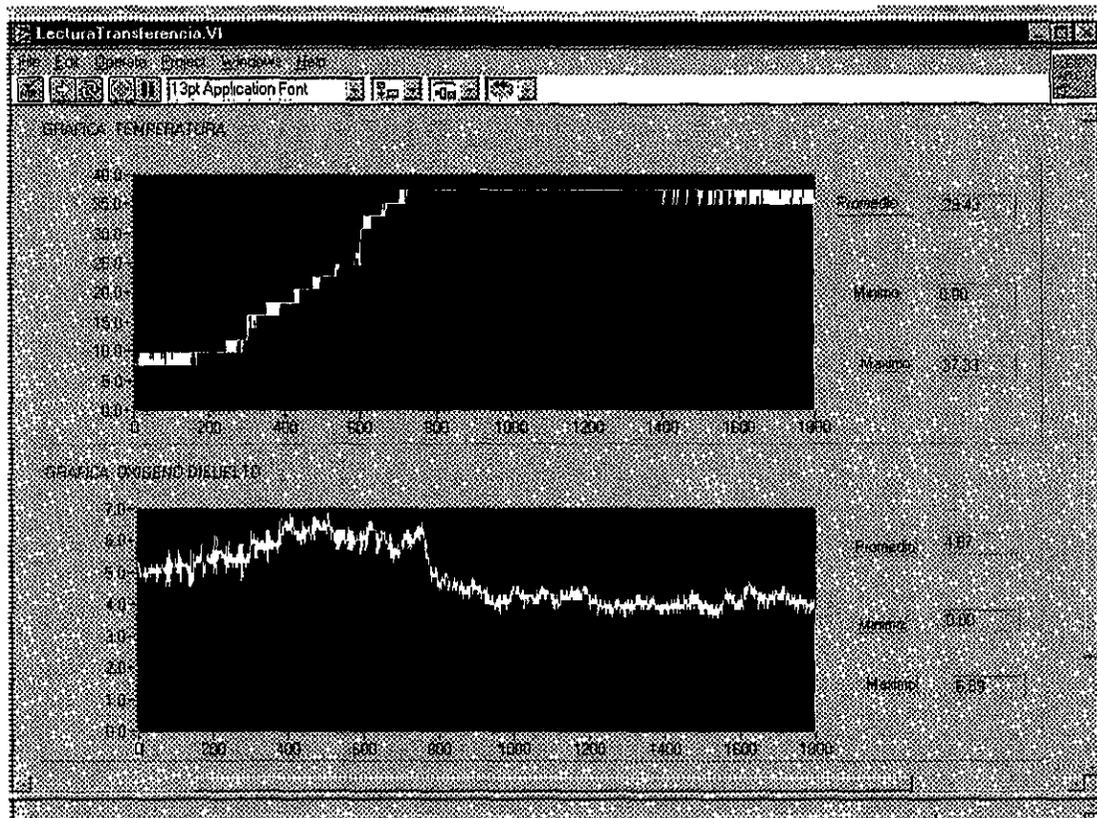


Fig. V.44 Resultados de la práctica transferencia de calor en un sistema acuático.

La tabla V.3 muestra los resultados con valores de promedio, valor máximo y mínimo.

VARIABLE	PROMEDIO	VALOR MAXIMO	VALOR MINIMO
TEMPERATURA	29.43	37.33	24.80
OXÍGENO DISUELTTO	4.87	6.89	3.57

Tabla V.3 resultados de la transferencia de calor en un sistema acuático.

### ANÁLISIS DE SONIDO.

A continuación se muestran los resultados obtenidos de ejecutar el programa que utiliza el sensor micrófono para analizar la señal de sonido mediante transformada rápida de Fourier.

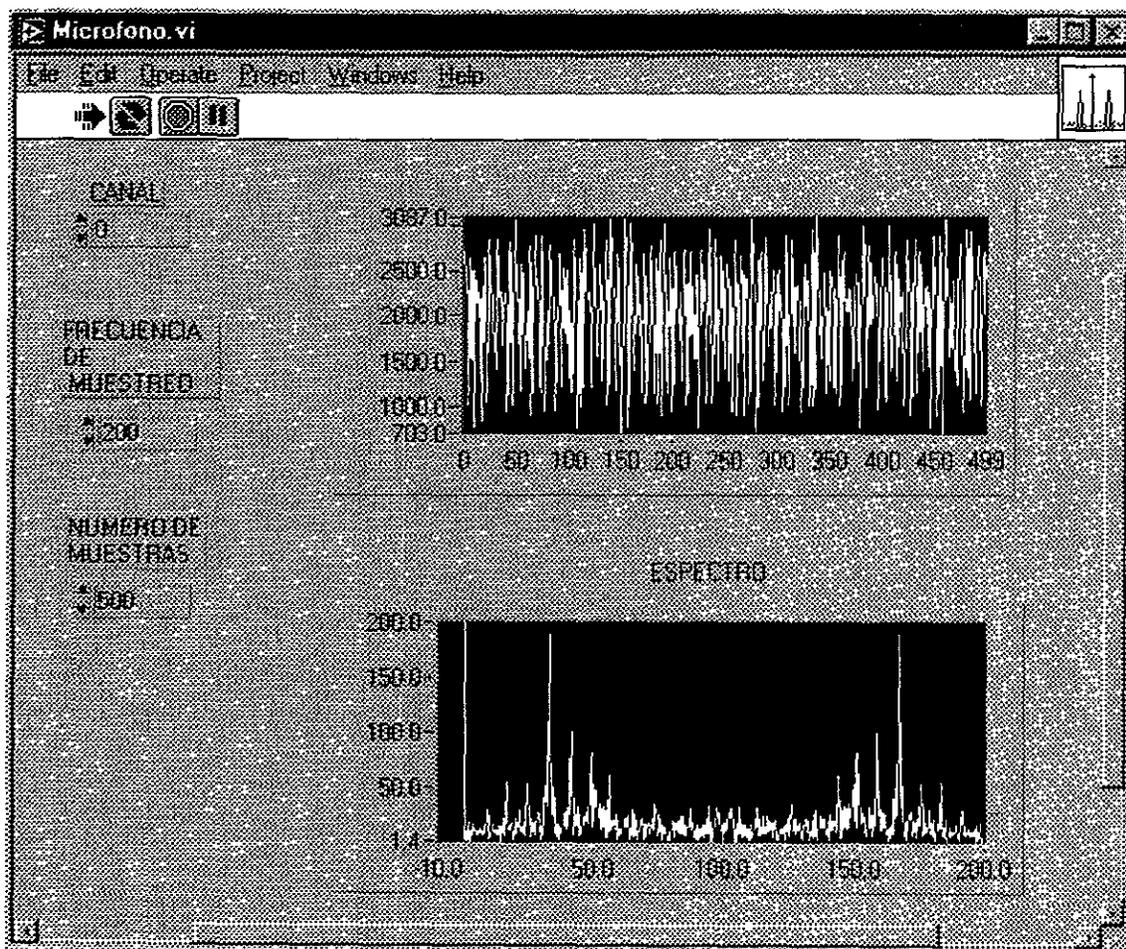


Fig. V.45 Resultado de la práctica micrófono.

Es importante mencionar que el micrófono se puede conectar en cualquier entrada de la interface, para este caso se decidió conectarlo en la entrada cero.

## CONCLUSIONES

Al concluir este trabajo, llegamos a la demostración de establecer la comunicación entre la tarjeta ProControl PA-CP12 y el software LabVIEW, proceso que parecía incompatible por pertenecer cada uno a diferente compañía, sin embargo, a través de esta investigación se logró adaptar estos componentes para cumplir objetivos importantes en nuestro medio educativo.

La conexión de la tarjeta ProControl PA-CP12 y el software LabVIEW se realizó mediante programas realizados en lenguaje C y compilados en el programa Watcom C++, esta conexión se puede hacer por medio de otros lenguajes pero en nuestro caso se optó por ésta opción, ya que al programar en el lenguaje C no es muy complicado, además de que las instrucciones usadas no observan mayor complejidad.

Consideramos que la utilidad de la conexión de una tarjeta de adquisición de datos y un software de diferente compañía, provee una solución un poco más amplia, es decir, el usuario tiene la posibilidad de escoger entre una serie de tarjetas diferentes en vez de las que se usen propiamente con el software, esto depende de los recursos y necesidades que requiera cada aplicación como lo pueden ser frecuencia de muestreo, número de canales, entre muchas otras. En este trabajo se presentó una solución específica entre dos componentes, pero con esto permitimos que el lector se de cuenta de que existe la posibilidad de crear una comunicación entre elementos diferentes.

Propusimos actividades experimentales en donde se aplica esta posibilidad de interconexión enfocado a profesores y estudiantes y así encuentren a la computadora como una herramienta útil tanto en el aula y como en el laboratorio experimental, para así facilitar contenidos teóricos-prácticos de la enseñanza-aprendizaje.

## GLOSARIO

*ASCII* : Código americano normalizado para el intercambio de información.

*Automatizar* : Realizar un trabajo para simplificarlo y conseguir una ejecución mejor de forma automática.

*Baud* : Bits por segundo. Unidad binaria de transmisión de información por segundo.

*Bit*: Contracción de *binary digit* (dígito binario). Unidad de información que puede adoptar dos valores o estados distintos.

*Buffer* : Almacenamiento temporal para datos adquiridos o generados.

*Bus* : Término relativo a un conductor común por el que se transmiten los datos entre las distintas partes de un equipo de cómputo.

*CPU* : (Central Procces Unit). Elemento que se encarga de buscar, decodificar y ejecutar instrucciones en una computadora

*Cursor* : Punto móvil luminoso, que adopta diferentes formas, mostrado en la pantalla que indica el lugar de desplazamiento del caracter.

*Chasis* : Estructura física de la computadora.

*dB* : (Decibel) Unidad de medida de la intensidad sonora.

*DDE* : (Dynamic Data Exchange). Protocolo estándar de software en Windows para comunicación de procesos. DDE es utilizado cuando LabVIEW envía mensajes y comparte datos con otras aplicaciones.

*Disparo Externo*: Pulso de voltaje desde una fuente externa que inicia un evento, tal como una conversión analógica/digital.

*DLL* : (Dinamic Link Library). Archivos de intercambio de información entre aplicaciones que corren sobre Windows.

*DMA* : (Direct Memory Acces). Método por el cual los datos pueden transferirse a la memoria de la computadora desde un dispositivo en el bus (o desde la memoria a un dispositivo), mientras el procesador realiza otra acción.

*Entradas de una sola terminal*: Entradas analógicas que se miden con respecto a una tierra común.

*Ethernet* : Estándar de red de área local que utiliza señales de frecuencia de radio llevadas por medio de cables coaxiales.

**FIFO** : (First in First Out) Estructura en la cual los datos se depositan en un extremo y se extraen del otro, según la orden de llegada.

**GPIB** : (General Purpose Interface Bus). Bus estándar utilizado para control de instrumentos electrónicos con una computadora.

**Hardware** : Es la parte física, electrónica de una computadora.

**Hz** : (Hertz) Unidad de frecuencia igual a un ciclo por segundo.

**Icono** : Símbolo gráfico que es representativo de un programa o una función en sí.

**Interface** : Conexión entre dos puntos.

**Interruptores (jumper)**: Conexión plástica que se localiza en los tableros de circuitos. Por lo regular dos o tres puntas en forma de alfiler sobresalen del tablero para que el interruptor se deslice sobre ellos.

**Manejadores (driver)** : Software que controla un dispositivo de hardware específico tal como una tarjeta de adquisición de datos.

**PCMCIA** : (Personal Computer Memory Card International Association). Arquitecturas de buses de expansión localizada en computadoras "notebook".

**Potenciómetros** : Resistencia eléctrica variable. Estos al ser variados modifican valores de características de la tarjeta ( en este caso) como por ejemplo rangos de voltaje.

**RAM** : (Random Access Memory). Memoria de lectura y escritura.

**Ranura (slot)** : Orificio donde se conectan las tarjetas de expansión o tarjetas de adquisición de datos.

**Ratón (mouse)** : Dispositivo de control que marca la posición del cursor en la pantalla.

**Reloj** : Componente de hardware que controla la regulación de tiempo para lecturas o escrituras.

**RISC** : (Reduced Instruction Set Computer). Un sistema RISC es una computadora veloz, debido a que su microprocesador solo puede realizar un número limitado de operaciones.

**ROM** : (Read Only Memory). Memoria solo de lectura.

**RS-232**: Método estándar de transmisión de datos or medio de cables seriales.

**SCXI** : (Signal Conditioning eXtensions for Instrumentation). Es la línea de productos de National Instruments para señales acondicionadas de bajo nivel con un chasis externo cerca de los sensores, para que solo señales de alto nivel sean enviadas a la tarjeta de adquisición de datos.

**Sistema Binario**: Sistema de numeración que solo utiliza unos y ceros para representar el almacenamiento de información en la computadora

**Software** : Término que hace referencia a los programas.

**Tarjetas de adquisición de Datos**: Componente de hardware capaz de capturar datos (señales analógicas) y convertirlos en señales que pueda leer la computadora (señales digitales)

**TaskID** : Un número generado por LabVIEW, el cual codifica el número disponible y el número de grupo después de configurar un grupo.

**Teclado** : Dispositivo de entrada de datos (alfanuméricos) a la computadora.

**Valor Booleano** : Valores Verdadero o Falso, que se utilizan con frecuencia en la programación.

**VISA** : Manejador desarrollado por National Instruments, para el uso diferentes tipos de transmisión de datos.

## BIBLIOGRAFIA.

BECKER W. M.

The world of the cell.

USA. Edit. The Benjamin Cummings Pub. 1995

DEL CORSO D., KIRRMAN H.

Microcomputer buses and links.

Edit. Academic Press, 1986.

DOWDEN, Tony.

Inside the EISA Computers.

Edit. Addison Wesley, 1990.

FLOY, T.L.

Fundamentos de Sistemas Digitales.

Madrid, Edit. Prentice Hall, 1997.

GOOKIN, Dan, WALLY Wang y VAN BUREN Chris.

Diccionario Ilustrado de Computación para Inexpertos.

México. Edit. Noriega Editores, 1995.

HARLEY, Hahn.

Unix sin Fronteras.

México, Edit. Mc Graw-Hill, 1995.

HAYES, John P.

Diseño de Sistemas Digitales y Microprocesadores.

España, Edit. Mc. Graw Hill, 1986.

HOLMAN P. Jack.

Métodos experimentales para ingenieros.

México, Edit. Mc. Graw Hill, 1988.

KERN , Donald.

Procesos de Transferencia de Calor

México, Edit. Continental. 1978.

LEE, Lisa.

MacWEEK upgrading and repairing your Mac.

USA. Edit. Hayden Books, 1995.

STANLEY, Wolf. SMITH Richard.

Guía para mediciones electrónicas y prácticas de laboratorios.

México, Edit. Prentice Hall, 1991.

## CATÁLOGOS

Instrumentation Reference and Catalogue 1998  
National Instruments, 1998.

LabVIEW  
Manual for Windows  
National Instruments Corporation, 1994.

LabVIEW  
Code Interfase Reference Manual  
National Instruments Corporation, 1994.

LabVIEW  
Data Acquisition VI  
Reference Manual for Windows  
National Instruments Corporation, 1994.

LabVIEW  
Function and VI Reference Manual  
National Instruments Corporation, 1998.

LabVIEW  
G Programming  
Reference Manual  
National Instruments, 1998.

LabVIEW  
User Manual  
National Instruments Corporation, 1998.

Instruction Manual  
PA-CP12 A/D, D/A and Digital I/O Card  
Edición 1, 1994.

Watcom C/C++ 10.0  
Getting Started  
Canada, 1994.

## PAGINAS WEB CONSULTADAS.

### INTRODUCCIÓN

[http://jever.phys.ualberta.ca/~gingrich/phys395/notes\\_1996/node151.html](http://jever.phys.ualberta.ca/~gingrich/phys395/notes_1996/node151.html)

<http://www.scimedia.com/chem-de/electron/daqintro.htm>

<http://helena.phy.vanderbilt.edu/e864/drawings/daq.html>

<http://www.chiltonco.com/ics/Natspcgd.htm>

<http://www.iotech.com/prsigcon.html>

### ADAC

<http://www.adac.com/pressrelease/pr5508lf.html>

### ANALOGDEVICES

[http://www.analog.com/product/Product\\_center.html](http://www.analog.com/product/Product_center.html)

### ATEasy

<http://www.geotestinc.com/asteasy.htm>

### BSOFT

<http://www.bsoft.com/a100prod.htm>

### HP VEE 4.0

<http://www.tmo.hp.com/tmo.pia/HPVee/>

### LEYBOLD DIDACTIC - CASSY

[http://www.leybold-didactic.de/date\\_e/](http://www.leybold-didactic.de/date_e/)

### LR INFORMATICA INDUSTRIAL

<http://www.lrinfo.com/160016.htm>

### MICROLINK

<http://www.microlink.co.uk/>

NATIONAL INSTRUMENTS.  
<http://www.natinst.com/>

OMEGA ENGINEERING  
<http://www.omega.ca/ce/prodinfo.htm>

PASCO  
<http://www.pasco.com/>

PICOTECH  
<http://www.picotech.com/scope.html>

RD13 DAQ.  
[http://www.rd13doc.cern.ch/public/doc/summerSchool94\\_2.html](http://www.rd13doc.cern.ch/public/doc/summerSchool94_2.html)

SABLE SYSTEMS.  
<http://www.sablesys.com>

SDS  
<http://www.scidata.com/boards.htm>

VERNIER  
<http://www.vernier.com/>

WINDAQ  
[http://www.turnkey\\_solutions.com.au/](http://www.turnkey_solutions.com.au/)

WINDMILL  
<http://www.windmill.com.uk>