

01170



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA
DIVISION DE ESTUDIOS DE POST-GRADO

"SISTEMA DE ADQUISICION DE DATOS PARA LABORATORIOS MOVILES DE ANALISIS DE EMISIONES EN FUENTES FIJAS"

T E S I S
PARA OBTENER EL GRADO DE:
MAESTRO EN INGENIERIA
(E L E C T R I C A)
P R E S E N T A:
ING. JOSE AMANDO ORTIZ ISLAS

ASESOR: DR. VLADIMIR TCHIOV T.



MEXICO D. F.

ABRIL 2000

277658



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.



Dios nos ha dado dos alas para volar hacia él: el amor y la razón.
Platón

La solución de una duda es el descubrimiento de la verdad.
Aristóteles

Según las leyes de la física comprobada por los experimentos en el túnel aerodinámico, la abeja no debería poder volar, ya que el tamaño, el peso y la configuración de su cuerpo no guardan la debida proporción con la envergadura de sus alas.

Pero la abeja, que ignora estas verdades científicas, se lanza a volar, y no tan solo vuela, sino que fabrica su poco de miel todos los días.

Bernard Shaw

AGRADECIMIENTOS

Al Instituto Mexicano del Petróleo que a través de su programa de formación y desarrollo del factor humano me brindó la posibilidad de realizar estos estudios de Maestría.

A la Universidad Nacional Autónoma de México, que a pesar de todo, sigue siendo nuestra máxima casa de estudios y por lo cual me siento orgulloso de ser puma.

Al Dr. Vladimir Tchijov por su dedicación, ayuda y sabios consejos dirigiendo este trabajo de tesis.

Al Ing. Carlos F. Valdes Olmedo, por su gran apoyo en la consecución de este logro.

A Julio, Jorge y el Sr. Peña, compañeros de trabajo y amigos.

A Hugo, Omar y Emmanuel por el empeño puesto en su trabajo.

A los miembros del honorable jurado por el tiempo dedicado a la revisión de este trabajo.

DEDICATORIAS

A dios por su grandeza y bondad

A mis padres los señores Martín Ortiz Ortiz y Martha Islas de Ortiz, por la mejor herencia que pudieron darme, su ejemplo de tenacidad, honradez y amor al trabajo.

A mis hermanas Emma, Vero, Elizabeth y Tere.

A mis tíos Víctor López y Carmen Ortiz, mis grandes apoyos en los momentos difíciles.

A Laura, mi compañera, deseando que podamos encontrar el camino.

A Citlalli, mi gran orgullo.

*A Angel y Ana Laura, mis
pequeños grandes motivos.*

INTRODUCCIÓN	1
1. LOS SISTEMAS DE ADQUISICIÓN DE DATOS	8
INTRODUCCIÓN.....	8
1.1 RESOLUCIÓN.....	11
1.2 EXACTITUD.....	12
1.3 VELOCIDAD DE MUESTREO.....	14
1.4 RANGO DE LA SEÑAL DE ENTRADA.....	16
1.5 SEÑALES ANALÓGICAS DE CORRIENTE.....	17
1.6 ENCUBRIMIENTO (<i>ALIASING</i>).....	18
1.7 CONVERTIDORES ANALÓGICO/DIGITAL.....	19
1.8 TENDENCIAS ACTUALES DE LOS CONVERTIDORES A/D.....	23
1.9 MODOS DE TRANSFERENCIA DE DATOS.....	23
1.10 ENTRADA SENCILLA O DIFERENCIAL.....	24
1.11 AISLAMIENTO.....	25
1.12 EL SISTEMA DE ADQUISICIÓN DE DATOS PROPUESTO.....	26
2. LA INTERFACE DE COMUNICACIÓN	27
2.1 BUS vs. PUERTO.....	27
2.2 ALGUNAS CARACTERISTICAS IMPORTANTES DE LAS COMUNICACIONES	30
2.3 ESTÁNDARES DE COMUNICACIÓN EN UNA PC	32
2.3.1 ESTÁNDARES DE TIPO PARALELO	32
2.3.2 ESTÁNDARES DE TIPO SERIAL	36
2.4 OTRAS CONSIDERACIONES IMPORTANTES	41
2.5 CONCLUSIONES	42
3. LA TARJETA DE ADQUISICIÓN DE DATOS	45
INTRODUCCIÓN	45
3.1 CARACTERÍSTICAS DE LA TARJETA DE ADQUISICIÓN DE DATOS	46
3.2 EL MICROCONTROLADOR <i>68HC11</i>	47
3.3 LA IDEA PRINCIPAL	49
3.4 UNA PRIMERA PROPUESTA	53
3.5 SEGUNDA APROXIMACIÓN	54
3.6 ETAPA DE MULTIPLEXAJE	55

3.7 ETAPA DE FILTRAJE Y AMPLIFICACIÓN	56
3.8 PRUEBAS REALIZADAS	58
3.9 VERSIÓN FINAL	61
3.10 CONVERSIÓN ANALÓGICA/DIGITAL	69
3.11 CODIFICACIÓN DE LOS DATOS	73
3.12 CONEXIÓN TARJETA DE ADQUISICIÓN DE DATOS-COMPUTADORA	78
3.13 CPU Y MEMORIA	80
3.14 COMENTARIOS FINALES	81
4. LA APLICACIÓN	83
INTRODUCCIÓN	83
4.1 EL SOFTWARE DE DESARROLLO	84
4.2 VISUAL BASIC	86
4.3 LA FILOSOFIA DE PROGRAMACIÓN DE VISUAL BASIC	88
4.4 ESTRUCTURA DE LA APLICACIÓN	94
4.5 ARRANCANDO LA APLICACIÓN	98
4.6 LA VENTANA PRINCIPAL	101
4.7 MONITOREAR	104
4.8 ADQUIRIR DATOS	112
4.9 CONFIGURAR EL No. DE LABORATORIO MÓVIL	121
4.10 CONFIGURAR EL PUERTO DE COMUNICACIONES	124
4.11 CONFIGURAR CADA CANAL ANALÓGICO	129
4.12 CONFIGURACIÓN DE TIEMPOS	141
4.13 CONFIGURACIÓN DE DATOS GENERALES	144
4.14 REVISIÓN DE RESULTADOS	147
4.15 LOS MÓDULOS ESTÁNDAR QUE CONTIENEN CÓDIGO	152
CONCLUSIONES	154
APENDICE: EL REGISTRO DE WINDOWS 95 Y 98	160
REFERENCIAS	164

En la Subdirección de Protección Ambiental del Instituto Mexicano del Petróleo existen cinco laboratorios móviles, como el mostrado en la figura 1, encargados de realizar el análisis de las emisiones contaminantes en fuentes fijas, principalmente las que son producto de la combustión, tales como chimeneas en refinerías y complejos petroquímicos.



Figura 1. Laboratorio Móvil

El objetivo de realizar dichos análisis es finalmente hacer recomendaciones relacionadas con el proceso de combustión para disminuir al mínimo posible las emisiones contaminantes o en su caso saber si se encuentran dentro de los límites permisibles.

Cada laboratorio móvil puede contar con hasta 8 posibles analizadores de diferentes gases contaminantes, cada uno de los cuales proporciona la información respecto al nivel de concentración del gas que le corresponde analizar.

Los monitoreos se realizan durante algunas horas tomando lecturas por lo menos cada minuto lo que, como es natural, genera una buena cantidad de información al final de cada periodo de monitoreo.

Los dos laboratorios móviles más recientemente adquiridos cuentan desde un inicio con un sistema de adquisición de datos que captura y procesa la información proveniente de los distintos analizadores de manera automatizada, entregando al final un reporte de los resultados obtenidos.

Los sistemas de adquisición de datos con que cuentan los más recientes laboratorios móviles fueron desarrollados en España debido a que en México no se cuenta con alguna compañía que los implemente, según argumentó la compañía vendedora [1].

En los otros tres laboratorios móviles, la información es capturada de manera semi-manual, es decir, el analista lee los datos emitidos por cada analizador directamente en un instrumento llamado *datalogger*, para luego anotarlos en una tabla junto con la hora en que fueron tomados.

El *datalogger* es un sistema de adquisición de datos que basa su funcionamiento en un microprocesador, es típicamente portable, y su principal función consiste en coleccionar datos de manera autónoma y durante periodos largos de tiempo, la información es almacenada en su memoria RAM para que posteriormente esos datos puedan ser leídos por el operador o descargados a una computadora para su procesamiento y análisis.

En el caso particular que nos ocupa, sirve como un concentrador de los datos emitidos por cada analizador ya que de otra forma el laboratorista tendría que leer la información en el indicador de cada uno de los equipos de análisis, mientras que así todos los datos son presentados en la pantalla del *datalogger*.

Posteriormente, los datos de las tablas son alimentados manualmente a hojas de cálculo electrónicas (Excel por ejemplo) para su tabulación, integración y graficación. Como es de suponerse este procedimiento no está exento de errores, tanto de lectura como de escritura, provocando retrasos en la entrega de resultados debido a múltiples revisiones y correcciones.

En un buen número de ocasiones los resultados del análisis hecho a los datos obtenidos en cada monitoreo necesitan estar disponibles con la mayor brevedad posible, pues de ellos dependen acciones preventivas y correctivas que impactan directamente en el área ambiental.

De lo anterior se desprende la necesidad de contar con un sistema de adquisición de datos capaz de capturar y procesar la información de manera automática, rápida y confiable y que inclusive tenga la posibilidad de enlazar los resultados de cada prueba con los diversos programas de manejo de datos para su posterior tratamiento e incorporación a los reportes finales.

Para que los laboratorios móviles que no tiene sistema de adquisición de datos pudieran contar con uno, existen dos posibilidades, una es adquirirlos con la misma compañía antes mencionada, la otra es desarrollarlos.

Inicialmente los responsables de los laboratorios móviles escogieron la primera opción, por lo que dichos sistemas fueron requisitados por el área correspondiente, pero debido a diversos tramites, a su elevado costo (20 000 dólares) y a recortes presupuestales, su adquisición se canceló.

En consecuencia se optó por la segunda opción, por lo que se pensó en desarrollar un sistema que en un momento dado sustituyera a los requisitados y que finalmente no se pudieron adquirir.

Dado que las actividades sustantivas del grupo de trabajo al cual pertenece el autor de la tesis son básicamente de mantenimiento preventivo y correctivo al diverso equipo analítico con que cuentan cada uno de los laboratorios de la Subdirección de Protección Ambiental, este trabajo de tesis se desarrolló de manera paralela a dichas actividades.

La idea original de este trabajo de tesis era realizar la menor cantidad de gastos posibles echando mano de todos los recursos con que se contaba en ese momento y solo adquirir lo estrictamente necesario.

En este trabajo de tesis se desarrolló un sistema de adquisición de datos que permite la captura y procesamiento, de manera automática, rápida y libre de errores de los datos emitidos por los diversos analizadores utilizados en los monitoreos realizados por los laboratorios móviles de la Subdirección de Protección Ambiental del Instituto Mexicano del Petróleo que no cuentan con él.

Las características principales del sistema que se propone son el que esta basado en una computadora personal, tiene la posibilidad de enlazar hasta 8 analizadores diferentes a la computadora por medio de una interface conectada a algún puerto serie de la misma. Posee una interface encargada

de realizar la conversión analógica/digital de cada una de las 8 posibles señales por medio de *multiplexaje*, además de poder manejar la ganancia adecuada para cada uno de las señales conectadas.

La computadora recibe los datos provenientes de la interface a través del puerto serie, decide a cual canal pertenecen esos datos y los va guardando en un archivo con extensión *adq*, para que el final del muestreo se puedan hacer los cálculos necesarios para la presentación de los resultados.

También por medio de la computadora el usuario podrá configurar el sistema de acuerdo a cada analizador y a las características del puerto donde se encuentre conectada la interface.

En la actualidad no existe un estándar que indique el rango de voltaje que la salida analógica de un analizador debe tener [2], pero en general todos los analizadores utilizados en los laboratorios móviles tienen una salida analógica de voltaje que caen en los siguientes rangos: 0-1Vdc, 0-5Vdc y 0-10Vdc.

Estas señales son proporcionales al rango de respuesta de cada analizador, por lo que realizando el escalamiento adecuado se puede interpretar un nivel de voltaje entregado por el analizador, como el grado de concentración de la variable que esté monitoreando. Lo anterior nos permite, mediante una interface adecuada, la posibilidad de hacer la conversión analógica/digital de ese voltaje para su posterior procesamiento en la computadora. También es importante considerar que en la actualidad todos los laboratorios móviles cuentan ya con una computadora personal la cual es suficiente para enlazar todos los analizadores.

Como se puede ver este trabajo consta de dos partes principales. Una es desarrollar e implementar una tarjeta de adquisición de datos que pueda ser conectada en cualquiera de los puertos serie disponibles de una computadora personal por el usuario y que cuente con 8 diferentes canales analógicos de entrada que trabajen en los tres rangos anteriormente mencionados.

La otra parte es desarrollar e implementar el software que posibilite la interconexión de la tarjeta de adquisición de datos con una computadora personal para poder capturar, procesar y presentar la información, todo bajo el ambiente de Windows 95 o posteriores, debido a que las computadoras instaladas en los laboratorios móviles cuentan con ese sistema operativo.

En la actualidad los sistemas de adquisición de datos por computadora son un campo de la informática ampliamente abordado [3-5]. Lo anterior no implica que el tema este agotado, debido principalmente a la rápida evolución de las computadoras, lo que involucra también la pronta actualización de los sistemas ya desarrollados.

Por otro lado, la evolución antes mencionada ha permitido que los sistemas de adquisición de datos puedan encontrar otros campos de acción que antes les estaban vedados, principalmente por limitaciones tecnológicas. Las distintas áreas de aplicación de estos sistemas van desde cuestiones tan complicadas como el análisis de datos atmosféricos con instrumentos analíticos montados en un avión [6], hasta cuestiones más simples como el control de un motor [7] por ejemplo.

Adicionalmente, la tendencia actual es que casi todos los instrumentos analíticos tengan la posibilidad de contar con un sistema de adquisición de datos y control del mismo, como parte del equipo o como un accesorio, permitiendo con ello un manejo más rápido y eficaz de la información aportada por el instrumento.

En relación con el trabajo que aquí se propone, un sistema de adquisición de datos utilizando una computadora personal, se están llevando a cabo un buen número de proyectos en diversas instituciones educativas alrededor del mundo e inclusive en nuestro país, cada uno de ellos con una aplicación específica [8-13].

La mayoría de esos proyectos intenta desarrollar una tarjeta que pueda ser instalada en el bus de la tarjeta madre de la computadora, en cualquiera de los diferentes tipos de ranuras de expansión que puedan existir. Tan solo la UNAM tiene registrados más de 60 trabajos de tesis relacionados con el tema de adquisición de datos de 1976 a la fecha, aunque solo los más recientes [14-21] tienen alguna relación con una computadora.

Un punto importante a tratar lo constituye el ambiente bajo el cual va trabajar el sistema; los ambientes gráficos se están imponiendo debido a que representan una interfaz más amigable para el usuario, en ese sentido es innegable la necesidad de desarrollar, para este caso en particular, un sistema que trabaje bajo el ambiente Windows, en especial en Windows 95 debido a que en un 100% computadoras de los laboratorios móviles antes mencionados tienen como sistema operativo el Windows 95.

Es también importante señalar que los compiladores visuales tales como Visual Basic [22], Visual C++ [23], etc. son ahora una herramienta bastante útil para el desarrollo de tales sistemas, ya que quitan al programador la necesidad de realizar grandes códigos de programa tan solo para poder desplegar una ventana del tipo Windows, permitiéndole enfocar sus esfuerzos a otro tipo de problemas relacionados con el proyecto.

El tipo de conexión que se va a utilizar para conectar a la PC con los diferentes instrumentos analíticos debe ser tomada en cuenta; la mayoría de sistemas utiliza una tarjeta que se inserta directamente en el bus del CPU, lo cual tiene como inconveniente que tal tarjeta necesita ser instalada

por gente con cierto grado de conocimientos al respecto, debido a que la computadora tiene que ser abierta para poder instalarla.

La comunicación entre la computadora y los diversos equipos analíticos conectados a ella a través de la tarjeta antes mencionada, puede ser soportada por diversos estándares de comunicación pero básicamente caen en dos categorías: comunicación serie y paralela [24].

La comunicación en paralelo por su naturaleza es más rápida debido a que pueden transmitirse hasta 8 bits a la vez, aunque cuenta con el inconveniente de que tradicionalmente las computadoras personales solo tienen un puerto de este tipo, el cual es prácticamente esclavo de la impresora. Este tipo de comunicación es utilizado solo cuando los instrumentos y la computadora tienen distancias cortas entre sí (hasta 3 m), debido a la inducción de ruidos en el cable cuando su longitud sobrepasa ese límite.

La comunicación en serie esta soportada por diversos estándares, ya que es más ampliamente utilizada que la anteriormente mencionada, debido a que cubre distancias más grandes y a que una computadora personal puede tener disponibles más de un puerto de este tipo. Este tipo de comunicación es más lento que la de tipo paralelo, pero aún así es suficiente para trabajar de manera adecuada.

En el trabajo aquí presentado se desarrolló una tarjeta que soporta la comunicación de tipo serial, utilizando el estándar RS-232. Dicha tarjeta puede ser conectada a uno de los puertos serie de la computadora sin necesidad de abrir esta, lo que significa que podrá ser soportada por cualquier tipo de computadora personal (escritorio, portátil), siempre y cuando tenga instalado el software que posibilite el trabajo del sistema de adquisición de datos.

La utilización de un microcontrolador como el MC68HC11 [25] para desarrollar la tarjeta de adquisición de datos alrededor de él permite una gran versatilidad tanto en el manejo del número de posibles canales analógicos de entrada como en la cantidad de posibles escalas que se puedan manejar.

En la actualidad los sistemas de adquisición de datos que no cuenten con por lo menos dos canales analógicos de entrada, están prácticamente obsoletos, esto se debe a que un mismo instrumento puede emitir varias señales a la vez o a que se tiene la necesidad de monitorear varios instrumentos al mismo tiempo.

En cuanto a la estructura de este trabajo de tesis, en el capítulo 1 se intenta dar una visión general de las características que deben ser tomadas en cuenta al momento de desarrollar un sistema de adquisición de datos cualquiera que sea su aplicación final. Se pone especial énfasis en el hardware ya que es probablemente la parte vital del sistema.

El capítulo 2 se ocupa de revisar las diversas opciones de comunicación entre la tarjeta de adquisición de datos y la computadora. Primeramente se comparan las posibilidades de utilizar una tarjeta conectada directamente al bus de la tarjeta madre o una conectada a alguno de sus puertos de comunicación. Después se revisan los estándares de comunicación más comúnmente utilizados para que finalmente se argumente la decisión de utilizar uno de los puertos de tipo serial.

En el capítulo 3 se enumeran las características de la tarjeta de adquisición de datos, como se implemento, las pruebas y modificaciones que se le hicieron.

El software de captura y procesamiento de datos desarrollado con Visual Basic es abordado en el capítulo 4. Primeramente se comparan los programas de desarrollo especializados con los de aplicación general. Después se enumeran las principales características de Visual Basic haciendo énfasis en la programación controlada por eventos. Por último se explica la estructura y funcionalidad de la aplicación así como los puntos clave de ella.

En el apéndice se aborda el tema del registro de Windows, parte fundamental en el desarrollo e implementación del programa.

Finalmente se enumeran las conclusiones obtenidas al finalizar el desarrollo de este trabajo.

INTRODUCCIÓN

En los sistemas de adquisición de datos basados en PC y en todos los sistemas de adquisición de datos en general, existen algunos conceptos que es importante manejar al momento de bosquejar el sistema que se requiere desarrollar con el fin de que este pueda cubrir de manera eficiente los requerimientos del usuario.

Los sistemas de adquisición de datos basados en computadora se componen de varios subsistemas o etapas, cada uno con una función específica bien determinada [26]. La primera etapa es la de sensado o detección, en esta, el sensor o transductor se encarga de detectar y medir variables físicas tales como presión, flujo, temperatura, movimiento, entregando una señal eléctrica (voltaje o corriente) proporcional a la magnitud de la variable que se está monitoreando.

La siguiente etapa se encarga de filtrar y amplificar la señal proveniente del sensor, con el fin de que esta pueda tener las características adecuadas para que pueda ser alimentada a algún tipo de tarjeta analógica, a esta etapa se le conoce regularmente como de acondicionamiento.

Otra etapa es la de conversión analógica/digital, en la cual la señal de voltaje o corriente es convertida a un formato digital, de tal forma que pueda ser leída por la computadora. Una señal analógica está definida por una función continua en el tiempo, esta señal debe ser convertida a una señal discreta en el tiempo para que pueda ser representada por la computadora.

La última etapa está conformada básicamente por la computadora, en esta, el software se encargará de cuantificar, analizar y presentar de diferentes maneras los datos emitidos por la interface de adquisición de datos. En la figura 1.1 se puede apreciar de manera gráfica cada una de las etapas mencionadas y la secuencia de ellas.

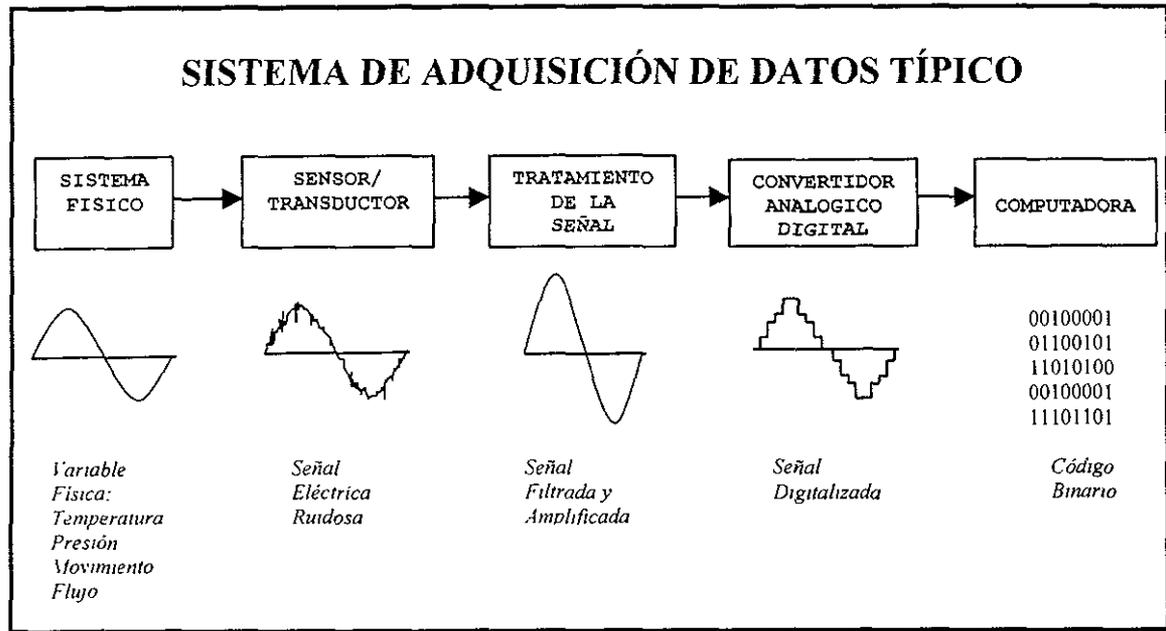


Figura 1.1. Etapas de un sistema de adquisición de datos típico.

Las etapas presentadas pueden variar según el sistema de adquisición de datos de que se trate, pero al final de cuentas, todas las antes mencionadas deben estar presentes aun cuando aparentemente, no puedan apreciarse a simple vista.

En la figura 1.2 se puede apreciar una de estas variaciones, en ella, la etapa de detección y la de acondicionamiento de la señal forman una sola como comúnmente sucede, ya que los instrumentos de análisis o detección necesitan tratar a la señal proveniente del sensor para que esta pueda ser alimentada a su sistema de medición, normalmente un voltímetro o amperímetro ya sea analógico o digital, después existe otra etapa de acondicionamiento para que la señal pueda ser transformada por el convertidor analógico/digital de manera más exacta, por último está la computadora encargada de manejar todos los datos provenientes de las anteriores etapas.

En la actualidad existe una tendencia a implementar los sistemas de adquisición de datos como se muestra en la figura 1.3 [27]. En esta figura se puede apreciar en primer termino, un equipo de análisis, el cual se encarga de detectar y medir la magnitud de la variable a analizar. Este equipo indicará en una pantalla digital o analógica dicha magnitud para que el usuario la lea y pueda llevar un registro escrito de ella. Como un agregado, el equipo de análisis tiene una salida analógica de voltaje o corriente, la cual proporcionará una señal cuyo nivel es directamente proporcional a la magnitud indicada en la pantalla.

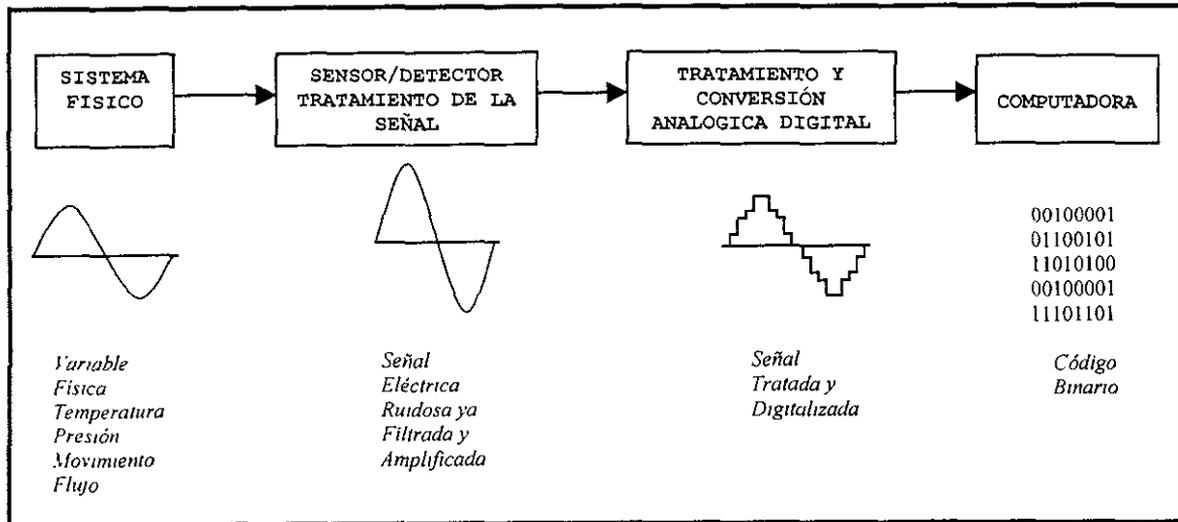


Figura 1.2 .Variación de sistema de adquisición de datos

Por ejemplo, si el equipo analizador puede mostrar en su pantalla lecturas entre 0 y 1000 PPM, y su salida analógica de voltaje tiene un rango entre 0 y 10 Vdc, un nivel de 2 volts representará una lectura de 200 PPM. Como es evidente, el equipo analizador necesitará de circuitos electrónicos que se encarguen de linealizar, filtrar y amplificar la señal que emite el detector para que pueda ser leída en pantalla por el usuario o alimentada a un sistema de adquisición de datos a través de su salida analógica.

La siguiente etapa es la interface de adquisición de datos la cual se encargará de utilizar la señal proveniente de la salida analógica del analizador para, mediante una conversión analógica/digital, poder transformarla e introducirla a la computadora con el fin de que esta pueda procesar la información.

La mayoría de las interfaces de adquisición de datos tienen como característica actual, la capacidad para manejar más de una señal analógica, es decir, se pueden conectar los voltajes provenientes de más de un equipo analizador, lo cual permite manejar la información que se esta recopilando de manera más integral [28]. Dicha interface de comunicación se conecta a la computadora básicamente de dos formas, ya sea a través de alguno de los puertos o puede ser insertada directamente en alguno de los buses de la misma.

La comunicación a través de los puertos es la más utilizada, ya que existe un buen numero de estándares para ese fin, los de tipo serie como el RS-232, RS-422, RS-485 y los de tipo paralelo como el IEEE-488, el Hewlett-Packard Interface Bus y el General Purpose Interface Bus [29].

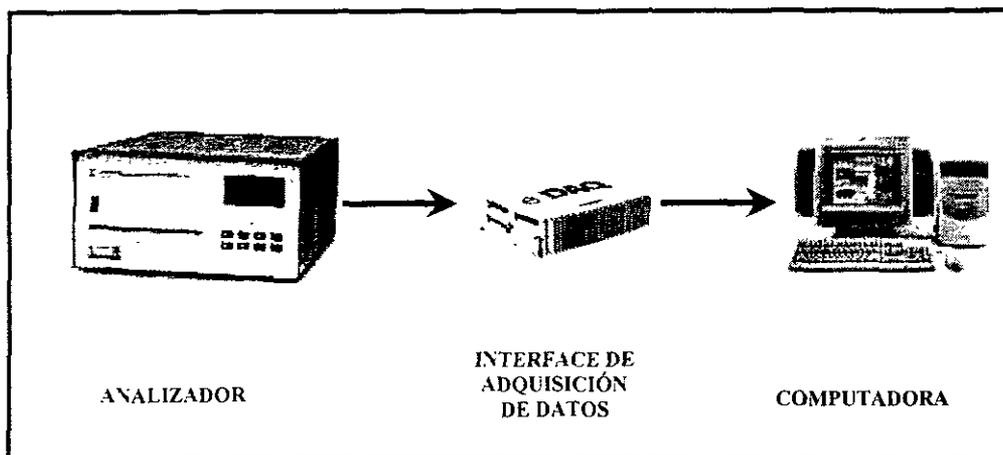


Figura 1.3 Interconexión analizador-computadora.

La inserción de la interface de adquisición de datos en alguno de los buses de la computadora permite velocidades de intercambio de datos entre mayores entre ellas por lo que esta opción es muy útil cuando se necesita hacer un muestreo de señales cuya frecuencia es muy alta. Estos sistemas de adquisición de datos tienen la ventaja de que pueden ser utilizados con cualquier analizador que tenga una salida analógica disponible

Otra tendencia actual de los fabricantes de equipos analíticos, es la de integrar una interface de adquisición de datos dentro del equipo de análisis con lo que el usuario solo tiene que conectar un cable de intercomunicación entre el analizador y una computadora, generalmente una PC [30]. Este tipo de sistemas de adquisición de datos tiene el inconveniente de que solo pueden ser utilizados en el equipo de análisis para el cual fueron construidos. Como es evidente, deben estar presentes todas las etapas mencionadas anteriormente aunque no se puedan apreciar a simple vista.

Para el diseño de un sistema de adquisición de datos se deben tomar en cuenta varios puntos con el fin de que este sea lo más exacto y confiable posible. A continuación se abordan estos puntos.

1.1 RESOLUCIÓN

La resolución de un sistema de adquisición de datos es el cambio más pequeño detectable durante una medición [5][31], y es directamente determinada por la resolución del convertidor analógico/digital.

Existen varias formas de expresarla [32], por ejemplo, como un % de la escala total, aunque la más frecuentemente utilizada es con un número de bits. Se puede calcular la resolución de la siguiente manera:

$$\text{Resolución} = \text{Valor de escala total} / 2^n,$$

donde n es igual al número de bits de resolución del convertidor analógico/digital.

Los sistemas de adquisición de datos tienen, por lo general, resoluciones de 8, 12, 16 y en la actualidad hasta 24 bits. La resolución también se puede obtener en volts, dividiendo el valor absoluto del rango del convertidor entre la resolución en bits. Como ejemplo, si tenemos un sistema con una resolución de 12 bits y un rango de escala total de 5 volts, la resolución en volts será:

$$\text{Resolución} = 5 \text{ Volts} / 2^{12}$$

$$\text{Resolución} = 5 \text{ Volts} / 4096$$

$$\text{Resolución} = 0.00122\text{V} \approx 1.22 \text{ mV}$$

Debido a que, por lo general, el precio de un convertidor analógico/digital es directamente proporcional al número de bits de resolución del mismo, es importante utilizar un convertidor con la resolución adecuada para la aplicación que se está desarrollando, es decir, si no es necesario utilizar un convertidor con una resolución grande, es por demás pagar el costo de un convertidor con esas características.

1.2 EXACTITUD

Aunque la exactitud de un sistema de adquisición de datos está íntimamente relacionada con su resolución, no significan lo mismo. La exactitud depende de muchos factores, no únicamente de la resolución, entre los factores que influyen en la exactitud del sistema se encuentran, la exactitud del convertidor A/D y el error introducido por los circuitos electrónicos necesarios para manejar y acondicionar la señal tales como amplificadores de ganancia variable y los multiplexores de entrada.

La exactitud de un sistema se puede definir de varias formas [33-34], la exactitud absoluta para un código dado, es la diferencia entre el voltaje que lo está produciendo actualmente y el voltaje necesario para producirlo teóricamente.

Por otro lado, la exactitud relativa es la desviación del valor teórico después de que el rango de escala total ha sido calibrado.

Es importante tener en cuenta que la exactitud total de un sistema solo puede ser tan buena como la de aquel dispositivo que tenga la menor exactitud y que forma parte del mismo.

La exactitud se puede calcular de varias formas dependiendo de las especificaciones proporcionadas por el fabricante del convertidor. Por ejemplo, para un convertidor A/D de 8 bits y con una escala total de 5 Volts:

Especificaciones: 0.024 % de la lectura \pm 1 bit

$$\text{exactitud de la medición} = 5 \text{ V} * (0.024/100 + 1/2^8) = 20.73 \text{ mV}$$

Especificaciones: \pm 2 bits

$$\text{exactitud de la medición} = 2 * (5 \text{ V}/2^8) = 39.06 \text{ mV}$$

Especificaciones: 0.048 % del rango de escala total

$$\text{Exactitud de la medición} = 5 \text{ V} * (0.048/100) = 2.4 \text{ mV}$$

Como es obvio, entre menor sea el valor obtenido al calcular la exactitud el convertidor A/D tendrá mejores características.

A la exactitud del convertidor A/D también se le conoce como error de cuantificación [31]. A los tres métodos para especificar la exactitud se les conoce como (a) error en bits menos significativos, (b) error de voltaje para un rango especificado y (c) porcentaje de error de lectura respectivamente [35].

En los ejemplos anteriores se calculó la exactitud solo para el convertidor A/D, pero las especificaciones de exactitud de un sistema deben incluir los efectos combinados de repetibilidad, histéresis, linealidad y resolución de todos los dispositivos que conforman el sistema. Las especificaciones de exactitud deben incluir el peor de los casos o error pico a las condiciones de referencia, por ejemplo temperatura 25°C, Voltaje de alimentación 15 Volts e impedancia de carga 250 ohm.

El error total se puede calcular así

$$\sigma_{TOTAL} = \sqrt{\sum_i \sigma_i^2}$$

donde σ_i es el error atribuible a cada uno de los dispositivos que forman el sistema.

1.3 VELOCIDAD DE MUESTREO

La velocidad de muestreo es una de las especificaciones más importantes de cualquier sistema de adquisición de datos. Esta característica no es otra cosa que el número de veces que el sistema puede tomar una "muestra" de la señal que se está analizando, en un periodo determinado de tiempo, típicamente un segundo [31][36]. Este dato nos permite saber si el sistema tiene la capacidad para hacer un muestreo adecuado de la señal o señales que se están monitoreando y así poder obtener un grupo de datos que la representen de la manera más exacta posible, además de contar con la información a tiempo en los sistemas de tiempo real.

La velocidad de muestreo se especifica normalmente en muestras por segundo y sus multiplicadores, muestras/segundo, kmuestras/segundo y Mmuestras/segundo

Por lo general, un sistema de adquisición de datos cuenta con un simple convertidor A/D y un circuito multiplexor de entrada que le permite conmutar entre varias señales de entrada para poder tomar muestras de cada una de ellas de forma independiente. Esta es la característica multicanal de los sistemas modernos y se puede observar en la figura 1.4.

La especificación más importante, relacionada con la velocidad de muestreo, es la velocidad de muestreo máxima, que no es otra cosa que el máximo número de muestras que el sistema puede tomar.

Como la velocidad de muestreo depende directamente del tiempo que necesita el convertidor A/D para realizar una conversión, este dato se obtiene dividiendo la máxima velocidad de muestreo del convertidor A/D entre el número de canales analógicos con que el sistema cuenta.

Por ejemplo, si un sistema de adquisición de datos cuenta con 8 canales analógicos de entrada y su velocidad de muestreo máxima es de 50 000 muestras/segundo, un solo canal puede ser muestreado a esa velocidad, pero si dos canales del sistema trabajan simultáneamente, solo pueden ser muestreados a 25 000 muestras/segundo como máximo para ambos canales.

El dato de máxima velocidad de muestreo en muchos casos es proporcionado considerando que todos los canales tienen la misma ganancia. El hecho de que los canales tengan diferentes ganancias, hace que la velocidad de muestreo sea más lenta, debido principalmente a que el sistema tiene que conmutar a la ganancia adecuada cada que se va a realizar el muestreo de una entrada analógica en particular. Dicha conmutación implica el cambio de estado de varios circuitos electrónicos los cuales tienen un tiempo de respuesta más lento que el convertidor analógico digital por lo que la velocidad de conmutación depende directamente del circuito electrónico más lento.

A velocidades de muestreo altas la memoria de la computadora es ocupada rápidamente, por lo que es necesario asegurarse de que el sistema que uno diseña tiene la memoria suficiente para guardar los datos adquiridos a la velocidad de muestreo que se esta implementando. Tradicionalmente los sistemas de adquisición de datos que tienen velocidades de muestreo muy altas, solucionan este detalle de dos formas, una es adicionando una tarjeta adicional de memoria RAM y la otra es desarrollando rutinas de software que permitan el acceso a altas velocidades al disco duro [36-37].

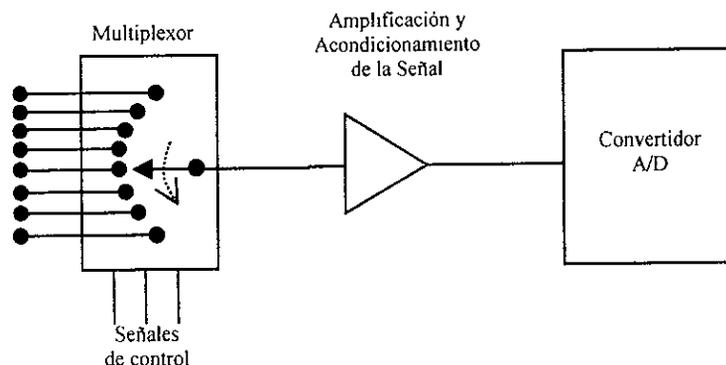


Figura 1.4. Sistema Multicanal

Como se verá un poco más adelante, la teoría general de muestreo dice que se deben tomar muestras de la señal que se esta analizando como mínimo a una frecuencia de por lo menos dos veces la del componente de más alta frecuencia de dicha señal, con el fin de que los datos recabados representen de manera más exacta a dicha señal.

1.4 RANGO DE LA SEÑAL DE ENTRADA

Como ya se apuntó antes, en la realidad, la mayoría de señales analógicas que se conectan a un sistema de adquisición de datos no pueden ser conectadas directamente a las entradas del convertidor A/D. Dichas señales tienen que ser atenuadas, amplificadas, filtradas o de alguna manera adecuadas a las especificaciones del convertidor. Si las señales analógicas tuvieran características fuera del rango de trabajo del convertidor, este invariablemente se dañaría al trabajar con ellas.

Para un uso correcto del convertidor A/D, es conveniente que la magnitud de las señales de entrada coincida con el voltaje de escala total del convertidor. Por ejemplo, si el convertidor tiene un rango de voltaje de entrada de 0-5V y un sensor nos entrega una señal de -1 a $+3$ V es necesario adecuar la señal proveniente del sensor para que coincida con el rango de entrada de 0-5 V del convertidor. Esto se logra por medio de circuitos electrónicos que realizan todas las modificaciones necesarias. A este proceso se le conoce como *acondicionamiento de la señal*.

La tendencia actual es que los sistemas de adquisición de datos puedan manejar señales de entrada de distintas magnitudes, aunque existen los sistemas que tienen rangos de entrada fijos [38-39]. Para aquellos sistemas que manejan varios rangos de entrada es claro que debe existir alguna manera de seleccionar el más adecuado para la señal proveniente de un analizador determinado. La selección del rango más adecuado para una señal determinada se realiza ya sea por hardware utilizando interruptores (*switches*) y puentes (*jumpers*) o a través del software.

Los sistemas que tienen un rango fijo de entrada trabajan bien con señales de magnitud grande o donde la señal es acondicionada por el instrumento de cual el dato es adquirido, pero principalmente con señales provenientes de analizadores que tienen el mismo nivel de salida analógica y que nunca cambia, inclusive si se sustituyera el analizador por otro.

Cuando se tiene un sistema de análisis que incluye varios analizadores que tienen niveles de salida analógica diferentes entre sí, y además los analizadores pueden ser conectados a un canal de entrada diferente cada vez, es más conveniente utilizar sistemas con rangos de entrada ajustables.

Los sistemas con rangos de entrada seleccionables por hardware, son normalmente fáciles de entender y utilizar pero pueden resultar engorrosos cuando se requiere cambiar su configuración frecuentemente. La selección por software se puede utilizar en casi cualquier aplicación pero tienen un mayor costo de adquisición y tienden a ser más lentos, tal como se explicó en el capítulo de la velocidad de muestreo.

La necesidad de tener distintos rangos de entrada se desprende también del hecho de que no existe un estándar relacionado con el nivel de la señal analógica que proporcionan los instrumentos analizadores, por lo que los fabricantes de dichos equipos utilizan el que les es más conveniente, sobre todo por cuestiones económicas, a ellos.

1.5 SEÑALES ANALÓGICAS DE CORRIENTE

Existen instrumentos, tales como el transmisor de temperatura ambiente y humedad relativa [40] o el transmisor de velocidad de viento [41], que tienen salidas analógicas pero no precisamente de voltaje sino de corriente, para las cuales existe ya un estándar para el nivel de esta señal, el rango que se maneja es de 4-20mA. En este rango, 4mA representa el límite inferior de la escala total del instrumento de análisis, mientras que 20mA representa el límite superior.

Este tipo de salidas analógicas son muy frecuentes en equipos de medición y análisis utilizados en plantas o en redes de monitoreo remotas, donde se tiene la necesidad de tener sensores localizados en diferentes partes de las mismas y a gran distancia del centro de control, ya que tiene la característica de manejar grandes distancias entre el analizador y el sistema de adquisición de datos, además de tener una gran inmunidad al ruido. Estos equipos son normalmente conocidos como transmisores de variable que analizan.

La forma en que esta señal analógica de corriente es transformada a una señal que un convertidor A/D pueda manejar, es muy simple, la corriente se hace circular a través de una resistencia de una magnitud tal que el voltaje que se genera entre sus terminales es de un nivel que puede ser adecuado por una serie de etapas de amplificación. La exactitud del voltaje que se genera, depende únicamente de las características de la resistencia utilizada. Es común utilizar una resistencia con un valor de 250 Ω para poder obtener un rango de voltaje de 1 a 5 V. También existen en el mercado circuitos integrados que se encargan de manejar este tipo de señales facilitando su conversión a voltaje [42-43].

Otras etapas de adecuación de la señal se encargan de ajustar el rango de la misma a un nivel adecuado para el convertidor A/D.

1.6 ENCUBRIMIENTO (ALIASING)

Un factor que es importante considerar en relación con la velocidad de muestreo es el fenómeno llamado *aliasing*. Este fenómeno se presenta cuando se obtiene una señal falsa de baja frecuencia al momento de reconstruir una señal analógica la cual fue muestreada a una velocidad insuficiente.

Un buen ejemplo de este fenómeno se presenta cuando las aspas del rotor de un helicóptero aparecen en una película moviéndose lentamente en dirección inversa a la que en realidad tienen.

En la figura 1.5 se puede apreciar el resultado de que el sistema de adquisición de datos tenga una velocidad de muestreo no adecuada para la frecuencia de la señal que se esta adquiriendo. Como se puede observar, la señal resultante es totalmente diferente a la original, por lo que los datos obtenidos no son representativos y pueden provocar fallas, errores y malos resultados.

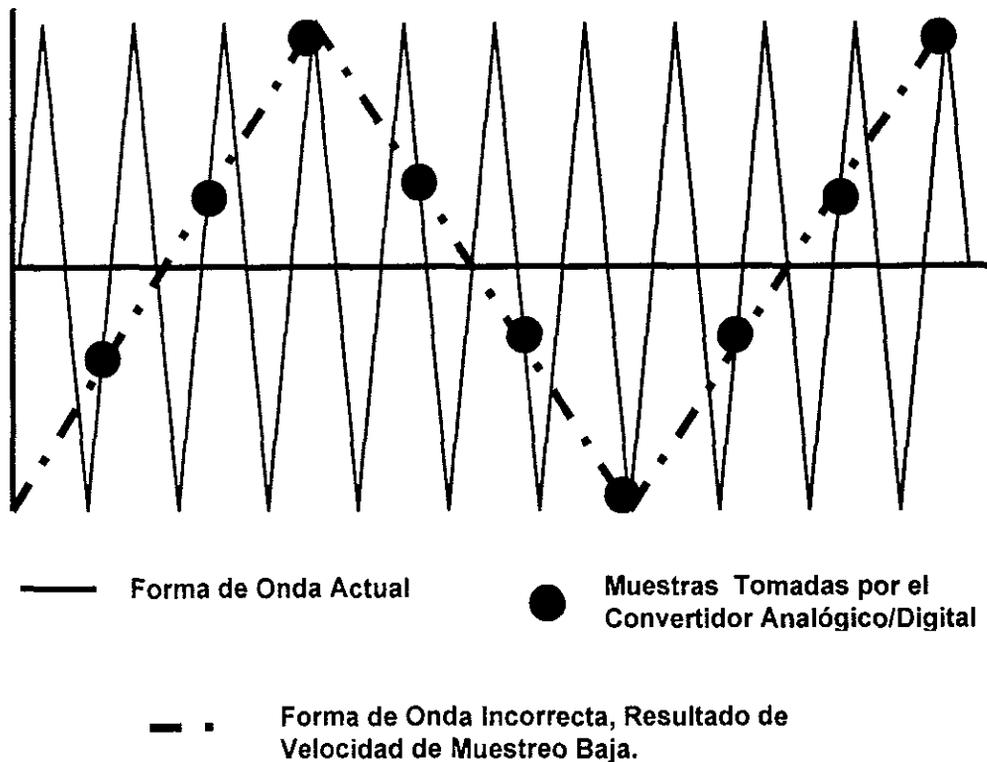


Figura 1.5. Ejemplo de Aliasing.

Si la señal tiene componentes cuyas frecuencias son mayores a la mitad de la frecuencia de muestreo entonces, los datos que se obtengan nos arrojarán una señal discreta muy alejada de la real.

Lo anterior está enunciado por el Teorema de muestreo del matemático americano Claude Elwood Shannon [4][44], el cual desarrolló la teoría básica del muestreo discreto. Dicho teorema enuncia que si una señal está compuesta por frecuencias menores a una frecuencia de corte f_c , toda la información de la señal puede ser capturada si es muestreada a $2f_c$, la cual es conocida como la frecuencia de Nyquist.

En el mundo real siempre existirán componentes cuyas frecuencias se encuentren por encima de este límite. Sin embargo, se ha encontrado [45] que si se está seguro de que las desviaciones introducidas por ellos son lo suficientemente pequeñas el error puede ser insignificante y por lo tanto ignorado.

En caso de que dichos errores no puedan ser ignorados se debe buscar la manera de atacar este problema. Para implementar un sistema *anti-Aliasing* las dos cosas que se deben tener en cuenta son (a) la velocidad de muestreo del sistema de conversión analógica/digital y (b) las frecuencias presentes en las señales muestreadas.

En la práctica [46] es muy común encontrar velocidades de muestreo entre 3 y 5 veces la del componente de más alta frecuencia de la señal, con el fin de reducir al mínimo los efectos del enmascaramiento.

Otra práctica *anti-Aliasing* muy típica es el utilizar filtros pasa-bajos, con una frecuencia de corte cercana a la del componente de interés de más alta frecuencia de la señal, es decir, el filtro debe permitir pasar a las señales válidas al mismo tiempo que remueve las componentes de alta frecuencia que pueden producir error, por lo general este filtro tiene una frecuencia de corte de 0.33 la velocidad de muestreo [47].

1.7 CONVERTIDORES ANALÓGICO/DIGITAL.

El tipo de convertidor A/D que se va a utilizar en el sistema de adquisición de datos debe ser una consideración importante al momento de diseñarlo.

Como ya se ha mencionado antes, el convertidor A/D convierte un voltaje analógico a un número digital el cual representa al voltaje de entrada con intervalos discretos de resolución finita.

En la actualidad hay muchos tipos diferentes de convertidores A/D disponibles, cada uno con sus características de velocidad, resolución y exactitud que los hacen adecuados para algunas aplicaciones y

no muy útiles para otras. Los tipos más comunes son el paralelo (*Flash*), el de aproximaciones sucesivas, el convertidor de voltaje a frecuencia y el de integración [48].

El convertidor analógico/digital paralelo (*Flash*) es el convertidor de más simple implementación [31][37]. Este tipo de convertidor utiliza un voltaje de referencia que cubre la escala completa del rango de entrada y un divisor de voltaje compuesto de $2^n + 1$ resistencias en serie. En este caso n es la resolución del convertidor A/D en bits. El valor del voltaje de entrada es determinado utilizando un comparador en cada uno de los 2^n voltajes de referencia creados a lo largo del divisor de voltaje.

El convertidor A/D tipo *Flash* realiza una conversión en un solo paso pero tiene que utilizar un banco de comparadores lo que los vuelve bastante caros ya que, como ejemplo, un convertidor con una resolución de 16 bits utiliza un banco compuesto por 65536 comparadores.

Los convertidores tipo *flash* son bastante rápidos (hasta 500MHz) por lo que es común encontrarlos en aplicaciones relacionadas con vídeo, tales como osciloscopios digitales.

Otro tipo de convertidores A/D frecuentemente utilizado es el de aproximaciones sucesivas. Este convertidor utiliza internamente un convertidor digital/analógico y un solo comparador. Para realizar una conversión el circuito compara la entrada analógica a convertir, con la salida analógica de su convertidor digital/analógico interno para que progresivamente el bit más significativo (MSB) sea determinado primero, enseguida el MSB - 1 y así hasta determinar el bit menos significativo (LSB) al último [31]. La mayoría de estos convertidores usan la tecnología CMOS. Un ejemplo de este tipo de convertidores se muestra en la figura 1.6.

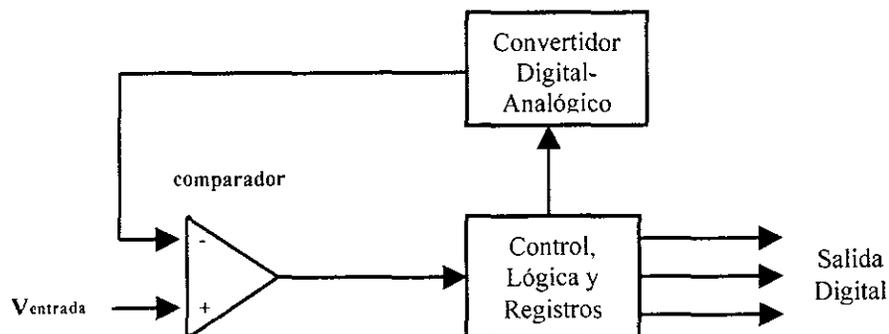


Figura 1.6. Convertidor A/D de Aproximaciones Sucesivas

En algunos casos los convertidores A/D de aproximaciones sucesivas utilizan una red de resistencias, en lugar de un convertidor digital/analógico, para formar un divisor de voltaje y así obtener la señal con que se va a comparar el voltaje analógico de entrada. Los convertidores más actuales de este tipo utilizan una red de distribución de cargas capacitivas en lugar de la red resistiva, por lo que tienen disipaciones de potencia y tamaños más pequeños. Otra característica importante que ha mejorado en este tipo de convertidores es el voltaje de referencia, que ha pasado de ser fijo a ser ajustable lo que permite adaptarlo a cualquier necesidad en particular.

Los convertidores A/D de aproximaciones sucesivas tienen la característica de ser más lentos que los de tipo *flash* debido a que las comparaciones deben realizarse por pasos, además de que el convertidor A/D debe hacer una pausa en cada paso para poner el convertidor digital/analógico y esperar que este ajuste.

Para este tipo de convertidores existen velocidades de conversión por encima de 200 KHz. Dispositivos de este tipo con resoluciones de 12 y 16 bits son relativamente baratos, por lo que son los convertidores analógico/digital más comúnmente utilizados y se pueden encontrar en muchos sistemas de adquisición de datos basados en PC.

Una clase más de convertidores A/D son los convertidores voltaje/frecuencia. Estos convierten un voltaje analógico de entrada en un tren de pulsos de salida cuya frecuencia es proporcional a la magnitud del voltaje de entrada [31]. La frecuencia de salida es determinada contando el número de pulsos existentes en un intervalo de tiempo fijo por lo que la magnitud del voltaje es calculada de una relación ya conocida.

Los convertidores voltaje frecuencia tienen un alto grado de rechazo al ruido debido a que la señal de entrada es *integrada de manera efectiva*. Este tipo de conversión se utiliza comúnmente para convertir señales lentas, es decir de baja frecuencia, y muy ruidosas.

Es también muy útil en aplicaciones de monitoreo remoto de señales en ambientes ruidosos. La conversión se realiza localmente y el tren de pulsos es transmitido por medio de cables a un contador, esto elimina el posible ruido que se pueda introducir si la señal analógica fuese transmitida. En la figura 1.7 se muestra un diagrama a bloques de este tipo de convertidores.

Existe otro tipo de convertidores analógico/digital que utilizan la integración como técnica de conversión. La integración consiste en medir el tiempo de carga o descarga de un capacitor para determinar el voltaje de entrada [31]. Por medio de una corriente proporcional al voltaje de entrada, un capacitor es cargado durante un periodo fijo de tiempo, el voltaje de entrada es determinado midiendo el

tiempo necesario para descargarlo usando una corriente constante, esta técnica es conocida como de doble pendiente

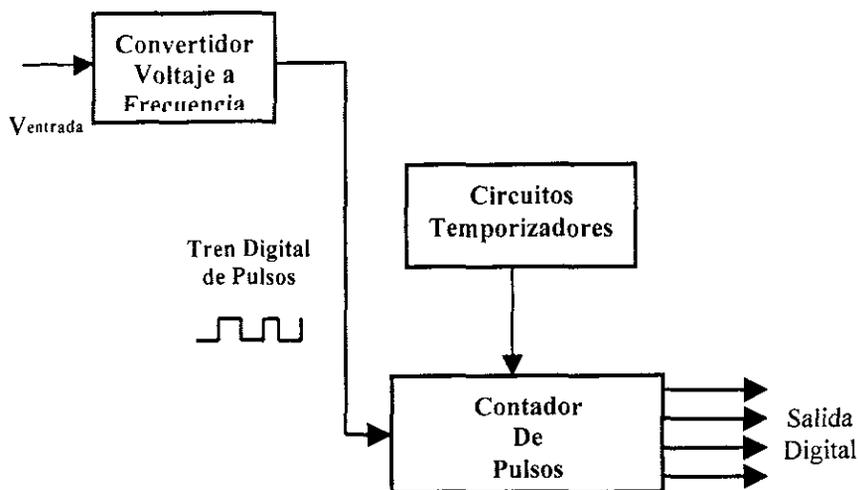


Figura 1.7 Convertidor Voltaje/Frecuencia

La integración permite reducir el ruido debido a la frecuencia de las líneas de AC si se utiliza un tiempo de integración que sea múltiplo de dicha frecuencia. Debido a esta característica este tipo de convertidores se utilizan frecuentemente en multímetros y paneles de medición digitales muy precisos. Resoluciones de hasta 20 bits para estos convertidores son comunes. Su principal desventaja es su relativamente lento tiempo de conversión de 60 Hz como máximo.

Los cuatro tipos de convertidores mencionados anteriormente son los más conocidos, pero existen otros que en algunos casos son variaciones de los anteriores con el fin de mejorar sobre todo la velocidad de conversión.

El convertidor Delta-Sigma ($\Delta-\Sigma$) [31][49] tiene la característica de digitalizar la señal a muy bajas resoluciones, 1 bit por ejemplo, y muy altas velocidades. Sobremuestreando y utilizando filtros digitales se pueden alcanzar resoluciones de hasta 20 bits. Estos convertidores son útiles con señales de baja frecuencia y que necesitan alta resolución. Son frecuentemente utilizados en el procesamiento de señales de audio. En resumen, este convertidor hace muestreos a una velocidad más alta y una resolución más baja de lo necesario, y por medio de varias técnicas de filtraje desarrollan una mejora en la resolución.

1.8 TENDENCIAS ACTUALES EN LOS CONVERTIDORES A/D

Como se mencionó en párrafos anteriores, una de las características más importantes en cualquier sistema de adquisición de datos es la velocidad de muestreo la cual ha sido mejorada con el uso de nuevas técnicas y tecnologías para la implementación de los convertidores, lo que les ha permitido incursionar en aplicaciones que antes les estaban vedadas.

Una de las técnicas implementadas para realizar conversiones a altas velocidades es la conocida como canalización (*pipeline*) que no es otra cosa que un convertidor tipo *flash* multipasos compuestos por varios convertidores con resoluciones ascendentes, es decir, en el primer paso la entrada analógica es aproximada para después pasar a otra etapa para ser refinada mientras que la etapa anterior puede ser utilizada para hacer un muestreo otra vez aproximar nuevamente el dato [37].

Con lo anterior, para producir la primer conversión de manera completa se necesitan un numero finito de ciclos de reloj, pero todas las conversiones siguientes serán producidas secuencialmente con cada ciclo de reloj. En resumen, en este tipo de convertidores, mientras se completa una conversión, en cada una de las etapas anteriores ya se están llevando a cabo conversiones posteriores.

Otra técnica implementada comúnmente es la de utilizar en una misma tarjeta de adquisición de datos multicanal un convertidor A/D para cada canal, es decir, al contrario de los sistemas multicanal que utilizan un solo convertidor para procesar varias señales por medio de multiplexaje, en estas cada canal tiene asignado un convertidor que no tiene que dividir su trabajo entre otros canales. Esto permite aproximarse mucho a la conversión simultanea de varias señales. Como es obvio, este tipo de tarjetas tienen un costo bastante elevado.

Una técnica más, resultado de la descrita anteriormente, es que cuando existen más convertidores que canales se puede utilizar más de uno para mejorar la velocidad de conversión, esto es, si un solo convertidor tiene un tiempo de conversión de 10 mS y se utilizan dos de ellos se pueden combinar de tal forma que se puedan realizar muestreos cada 5 mS, a esta técnica se le conoce como "*ping-ponging*".

1.9 MODOS DE TRANSFERENCIA DE DATOS

Para la adquisición de datos a alta velocidad, las tarjetas que tienen esta característica utilizan ya sea Acceso Directo a Memoria (DMA por sus siglas en ingles) [31][50] o transferencia de datos manejada por interrupciones [51].

Debido a que en el primer método, la transferencia es controlada completamente por hardware, se pueden obtener altas velocidades para aplicaciones donde la velocidad es crítica.

En el método de transferencia por acceso directo a memoria, el dato es llevado directamente de la tarjeta de adquisición de datos a la memoria de la computadora. El software que inicializa este modo de transferencia puede desarrollar de esta manera otras tareas mientras el DMA esta desarrollándose.

En el caso de la transferencia de datos manejado por interrupciones, una interrupción provoca que la computadora detenga la operación de su programa actual y salte a un programa diferente, este otro programa tomará el dato de la tarjeta de adquisición de datos, lo pondrá en memoria, desarrollará cualquier otra tarea que se desee y después regresará el control de la computadora al programa original.

1.10 ENTRADA SENCILLA O DIFERENCIAL

Existen dos opciones básicas para conectar las señales analógicas provenientes de los diferentes sensores o analizadores a la tarjeta de adquisición de datos, (a) en modo sencillo o (b) en modo diferencial [52]. La razón de que deba ponerse atención a la configuración de la entrada es la existencia del Voltaje de Modo Común (CMV por sus siglas en Inglés) [50][53]. Este voltaje aparece en las terminales de entrada de cada uno de los canales con que cuenta el sistema y se encuentra en fase con la señal de entrada.

Hay dos casos en los que debe tenerse especial cuidado con el CMV, el primero es cuando su nivel es de tal magnitud que sumado al de la señal analógica provoca lecturas incorrectas provocando errores en la cuantificación. El segundo es más crítico, debido a que si el CMV alcanza niveles que sobrepasen al nivel máximo de sobrevoltaje del sistema puede provocar además del daño físico a los circuitos, también puede llegar a ocasionarlo a las personas que se encuentre en contacto con el sistema.

La presencia del CMV y su magnitud determinan el modo de conexión de las entradas e inclusive si algún tipo de aislamiento es necesario.

La conexión en modo sencillo de las entradas, es la más sencilla de entender y conectar; esta consiste solo de dos cables, uno de señal y otro de tierra común. Esta configuración es la recomendada para hacer mediciones en sistemas donde no existe CMV, situación que es muy común cuando la señal analógica proviene de la salida de un amplificador. Todos los amplificadores tienen salidas conectadas en modo sencillo y la única forma de que exista CMV es una diferencia entre las tierras del amplificador y el sistema, situación que no se presenta cuando ambos son alimentados localmente como sucede normalmente.

La conexión en modo diferencial es muy útil y requerida cuando se hacen mediciones de fuentes que emiten señales de balanceo o cuando el CMV existe. Un ejemplo de señales balanceadas son las provenientes de puentes de Wheatstone cuya salida es diferencial. En este modo, la magnitud de una señal de entrada es el resultado de la diferencia entre dos entradas, debido a que una de las entradas conduce a la señal de interés más el CMV y la otra conduce solo el CMV, la diferencia entre las dos arrojará como resultado solo a la señal de interés. Es importante estar seguro de la magnitud del CMV para prevenir resultados como los mencionados anteriormente.

Las ventajas de la entrada diferencial sobre la sencilla son su habilidad para realizar mediciones y rechazar al CMV. Una entrada diferencial ocupa dos canales sencillos uno para la entrada positiva y otro para la entrada negativa. No es común encontrar sistemas que tengan una mezcla de entradas diferenciales y sencillas aunque un mismo sistema puede ser configurado de manera total ya sea en forma diferencial o en forma sencilla [54].

1.11 AISLAMIENTO

El aislamiento en un sistema de adquisición de datos significa no tener una conexión eléctrica directa o enlace de baja impedancia entre dos o más puntos o circuitos [31]. Hay dos razones principales para realizar aislamiento [55]: 1) para romper *loops* de tierra potenciales, y 2) para proteger al equipo de picos y transitorios de alto voltaje.

El aislamiento asegura que no exista conexión entre la tierra de la computadora y la tierra de la señal analógica con lo cual el CMV puede flotar a un nivel igual al que el aislamiento permita. Por este motivo un sistema de adquisición de datos que incluya aislamiento puede ser aplicado prácticamente cualquier situación de medición sin riesgo de daño para la tarjeta de adquisición de datos, la computadora o el usuario mismo del sistema.

Hay cuatro posibles combinaciones de aislamiento:

Entrada: La señal de entrada se encuentra aislada o flotando con respecto a los voltajes de polarización y la salida que comparten un mismo común.

Salida: La salida se encuentra flotando con respecto a la entrada y los voltajes de polarización los cuales son comunes entre ellos. Se utiliza para enviar señales de un sistema aterrizado al campo.

Tres vías (*Three Way*): La entrada, la salida y las polarizaciones se encuentran completamente aisladas una de las otras. Se utilizan para interfazar tres sistemas aterrizados por separado.

Polarización aislada: La entrada y salida son comunes entre ellas y aisladas del común de la polarización. Este sistema y el anterior brindan protección de sistemas cuya polarización es aterrizada de manera separada de las fuentes de señal.

1.12 EL SISTEMA DE ADQUISICIÓN DE DATOS PROPUESTO

La variante que se relaciona con este trabajo de tesis es muy parecida a la de la figura 1.2, en este caso las variables físicas a sensor son las concentraciones de emisiones contaminantes en el aire tales como CO₂, CO, O₃, HCl, NH₃, etc.

Estas concentraciones son determinadas por un equipo llamado *analizador de gases*, el cual como un todo, esta formado por una etapa de detección en la cual el elemento sensor es el elemento principal, después existen una etapa de filtraje y acondicionamiento de la señal y una de amplificación.

El analizador tiene un indicador ya sea digital o de aguja, donde el usuario puede leer directamente el nivel de concentración del gas que esta detectando; también posee una salida analógica de voltaje la cual es directamente proporcional al nivel de concentración de la variable a analizar. Esta salida analógica es utilizada para conectarse directamente a una tarjeta perteneciente al sistema de adquisición de datos.

La tarjeta se encarga de adquirir los datos a petición de la computadora para después enviárselos. El enlace entre la tarjeta y la computadora es a través de uno de los puertos seriales de esta última.

Finalmente la computadora se ocupa de manejar los datos adquiridos para su identificación y almacenamiento con el fin de obtener el reporte correspondiente.

En los siguientes tres capítulos de este trabajo de tesis, se aborda de manera más detallada todo el sistema brevemente descrito en estos últimos párrafos.

Lo primero que se tiene que decidir al momento de diseñar un sistema de adquisición de datos, como el presentado en este trabajo de tesis, es lo relacionado con la interface que servirá como medio de comunicación entre la tarjeta de adquisición de datos y la computadora. Para tomar esa decisión es necesario analizar los requerimientos de la aplicación que se va a desarrollar, para elegir la interface más adecuada. Comparar es inevitable cuando se necesita decidir que tipo de interface utilizar para poner los datos adquiridos directamente a disposición del CPU.

Entre los parámetros que se deben tomar en cuenta al hacer el análisis están la velocidad de transmisión, la distancia entre dispositivos, si la interface se conectará directamente en el bus o a través de alguno de los puertos de la computadora.

El primer punto a analizar es el relacionado al tipo de tarjeta que se utilizará para interfazar a los diferentes analizadores con la computadora y obviamente con la aplicación que este corriendo en ella. A continuación se analiza esta situación.

2.1 BUS vs PUERTO

Para los sistemas de adquisición de datos basados en PC, solo existen dos opciones en lo que se refiere a la forma como se conectará la tarjeta de adquisición de datos a la computadora [56]. La primera consiste en conectarla directamente al bus, lo cual tiene la ventaja de poder manejar altas velocidades de transferencia de datos, además de menor costo de fabricación ya que el chasis y las fuentes de alimentación no tienen que incluirse en ella, porque son tomadas de la computadora. A este tipo de interfaces se les conoce comúnmente como *PLUG-INS*.

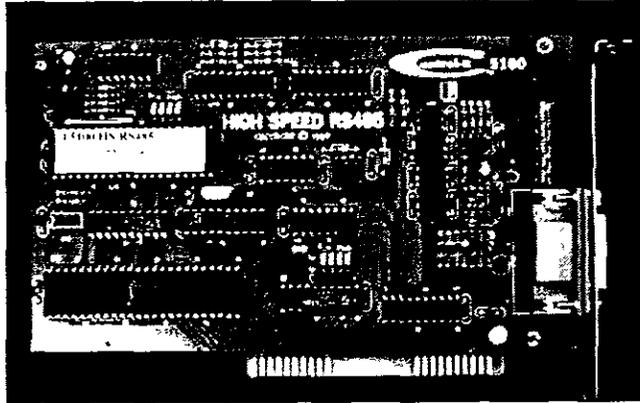


Figura 2.1. Interface tipo PLUG-IN

Como es obvio, la interface debe cumplir con las especificaciones de alguno de los buses de la tarjeta madre además de que debe haber una ranura de expansión disponible en la computadora para poder insertarla. En la figura 2.1 se muestra un ejemplo de este tipo de interfaces.

Esta opción tiene el inconveniente de que el usuario o alguien con cierto tipo de conocimientos debe instalar la interface dentro de la computadora con el riesgo que esto acarrea, sin contar también con que pueden presentarse conflictos de hardware que requieren un conocimiento más profundo para resolverlos y que en la mayoría de los casos el usuario no tiene.

En el caso de computadoras portátiles (para poder ser utilizadas en campo, por ejemplo), la situación se vuelve mucho más complicada debido a que la dificultad de instalar la tarjeta de adquisición de datos directamente en un bus se incrementa notablemente.

La otra opción consiste en conectar la tarjeta de adquisición de datos a alguno de los puertos de comunicación de la computadora para que a través de éste fluyan los datos que ambas necesitan, tal como se muestra en la figura 2.2.

Como es muy bien sabido las PC's suelen contar con solo dos tipos de puerto según sea la forma en que los datos se manejen, serial o paralelo. Entre los primeros se encuentran los que cumplen con el estándar RS232 y últimamente los llamados Buses Seriales Universales (USB por sus siglas en Inglés). En la segunda categoría se encuentran los ya muy conocidos *Centronics*.

El inconveniente principal cuando se utiliza algún puerto de la computadora, es su menor velocidad de transferencia de datos comparada con la opción de conectar la interface al bus, es decir, si la aplicación que se está desarrollando, requiere de manejar velocidades de transferencia de datos muy altas, es posible que ninguno de los puertos disponibles pueda soportarlas.

Si la velocidad de datos no representa ningún problema para la aplicación que se va a desarrollar, se puede tener la ventaja de que casi siempre existe algún puerto disponible, además el usuario no tendrá la necesidad de abrir su computadora para instalar la interface de adquisición ya que solo necesitará el cable adecuado para realizar la conexión.

Cuando el sistema de adquisición de datos no esté en uso, se puede desconectar la interface del puerto para poder utilizarlo con algún otro dispositivo sin tener que hacer ajustes complicados en el software.

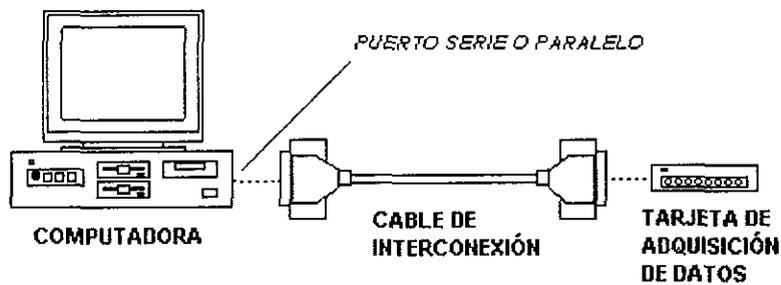


Figura 2.2. Interface conectada a un puerto

En ambos casos, ya sea conectada al bus o conectada a alguno de los puertos de la computadora, la interface debe cumplir con algún estándar que le permita transferir datos o recibirlos de manera adecuada.

Los estándares se desarrollan para asegurar la compatibilidad entre equipos fabricados por diferentes compañías, además de permitir transferencias de datos exitosas sobre distancias entre equipos y/o velocidades de transferencia de datos bien específicas.

Una de las características que se pensó debía tener este trabajo era el echar mano de los recursos que una PC tiene de manera regular, es decir, no pensar en adicionar algún dispositivo a menos que fuera estrictamente necesario.

La opción que se ajusta más a esta condición es el utilizar alguno de los puertos de la PC, ya que esto requerirá que el usuario tan solo conecte la tarjeta de adquisición de datos, por medio del cable adecuado, al puerto escogido, además de que el software desarrollado pueda manejar ese tipo de comunicación.

A continuación se analizan algunas de las características más importantes de las comunicaciones de datos, para después revisar los principales estándares de comunicación con los cuales se puede trabajar en una computadora personal.

2.2 ALGUNAS CARACTERISTICAS IMPORTANTES DE LAS COMUNICACIONES

Una importante característica que debe tomarse en cuenta también, es lo relacionado con la forma cómo los datos fluirán entre ambos dispositivos. Con respecto a esto existen básicamente dos tipos de comunicaciones [57]: *simplex* y *dúplex*.

Las comunicaciones tipo *simplex* son aquellas en que la información viaja siempre en un solo sentido, desde un transmisor hacia un receptor y ninguno de los dos puede cambiar su papel. Un buen ejemplo de ella sería una estación de radio y un receptor de esta señal. Este tipo de comunicación no es muy utilizada entre computadoras ya que no se puede verificar cuando un dato ya ha sido recibido, pero es muy útil cuando se tiene que distribuir una gran cantidad de información a un buen número de receptores.

En la comunicación tipo *dúplex* ambos dispositivos pueden ser tanto emisores como receptores, es decir, los datos fluyen en ambas direcciones con lo que puede existir un control y verificación de las operaciones de transmisión/recepción.

Dentro de las comunicaciones tipo *dúplex* se encuentra el *full dúplex* que se caracteriza por que los dispositivos pueden transmitir y recibir datos al mismo tiempo. Un buen ejemplo de esto lo representa el estándar RS232. Esta comunicación se puede lograr debido a físicamente existen líneas separadas para la recepción y emisión de señales lo que permite que puedan fluir datos en ambas direcciones simultáneamente.

El *half dúplex* tiene la característica de permitir comunicación bidireccional pero no de manera simultánea lo que significa que uno de los dispositivos puede transmitir mientras que el otro solo puede recibir al mismo tiempo. El estándar RS485 opera de esta manera.

Con relación a la operación de generadores y receptores de datos existen básicamente dos modos: no balanceado y balanceado.

La transmisión de datos no balanceada, como la figura 2.3, utiliza un solo conductor con un voltaje referenciado a la tierra de señal para diferenciar los niveles lógicos. Esto significa que solo una línea tiene que ser manejada en este tipo de comunicación.

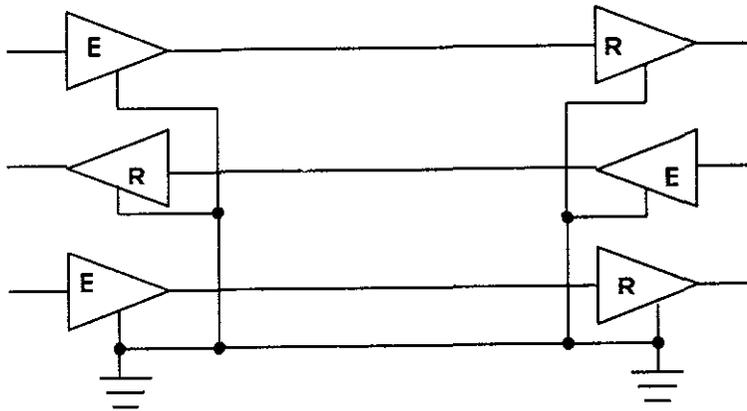


Figura 2.3. Comunicación no balanceada

Entre las ventajas de la comunicación no balanceada está el que cuando se requiere manejar múltiples canales solo tiene que utilizarse una tierra común lo que minimiza la cantidad de cable y conectores a utilizar y en consecuencia reduce el costo del sistema. La desventaja principal de este tipo de comunicación está en que no es muy confiable cuando se maneja en ambientes ruidosos, debido a sus pequeños márgenes de inmunidad contra el ruido.

La transmisión de datos balanceada, mostrada en la figura 2.4, requiere de dos conductores por señal. Los niveles lógicos se obtienen de la diferencia de voltaje entre esas dos líneas, no con respecto a tierra. Esta característica permite utilizar más eficazmente este tipo de transmisión en ambientes ruidosos. Debido a que el ruido se induce en ambas líneas, al realizar la diferencia de voltaje entre ellas el ruido inducido se nulifica.

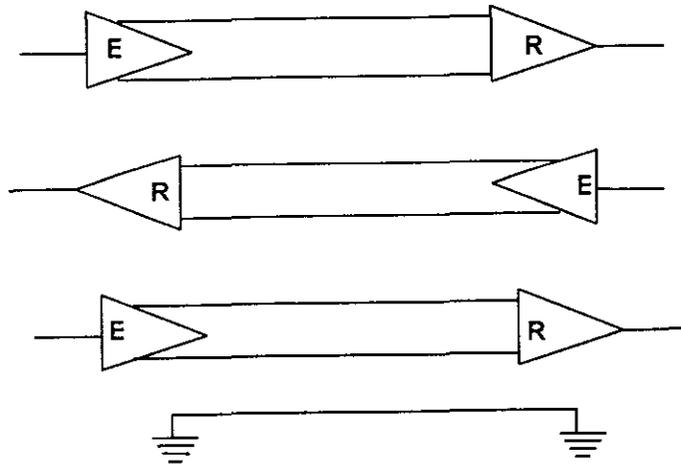


Figura 2.4. Comunicación balanceada

Después de haber abordado las anteriores características de las comunicaciones a continuación se enumeran los principales estándares de comunicación que pueden ser soportados en una PC, con el fin de poder estar en condición de escoger el que se adapte mejor a las características que este trabajo debe tener y que se enumeran en párrafos más adelante.

2.3 ESTÁNDARES DE COMUNICACIÓN EN UNA PC

En los sistemas de adquisición de datos basados en PC, como ya se mencionó anteriormente, una de las opciones para introducir los datos de la tarjeta de adquisición hacia la computadora es utilizar uno de los puertos de esta, opción por la cual nos decidimos. Los puertos de la computadora solo son de dos tipos, seriales y en paralelo, a continuación analizaremos las principales características de cada uno de ellos.

2.3.1 ESTÁNDARES DE TIPO PARALELO

Todos los puertos de la computadora se basan en algún estándar de comunicación establecido por algún organismo encargado de esto.

Los estándares de comunicación más comunes del tipo paralelo son los conocidos como *Centronics* e IEEE488 [24]. Estos estándares pueden manejar datos de 8 o más bits a la vez.

El estándar *Centronics* es el que tradicionalmente se utiliza para conectar las impresoras a las computadoras y por medio de este estándar la información que se necesita imprimir es enviada a la impresora.

Se le conoce como *Centronics* porque la primera gran compañía que lo utilizó se llamaba así y también fue una de las primeras en fabricar impresoras para computadora a gran escala. En la figura 2.5 se pueden ver los conectores para esta interface y las señales correspondientes a cada pin.

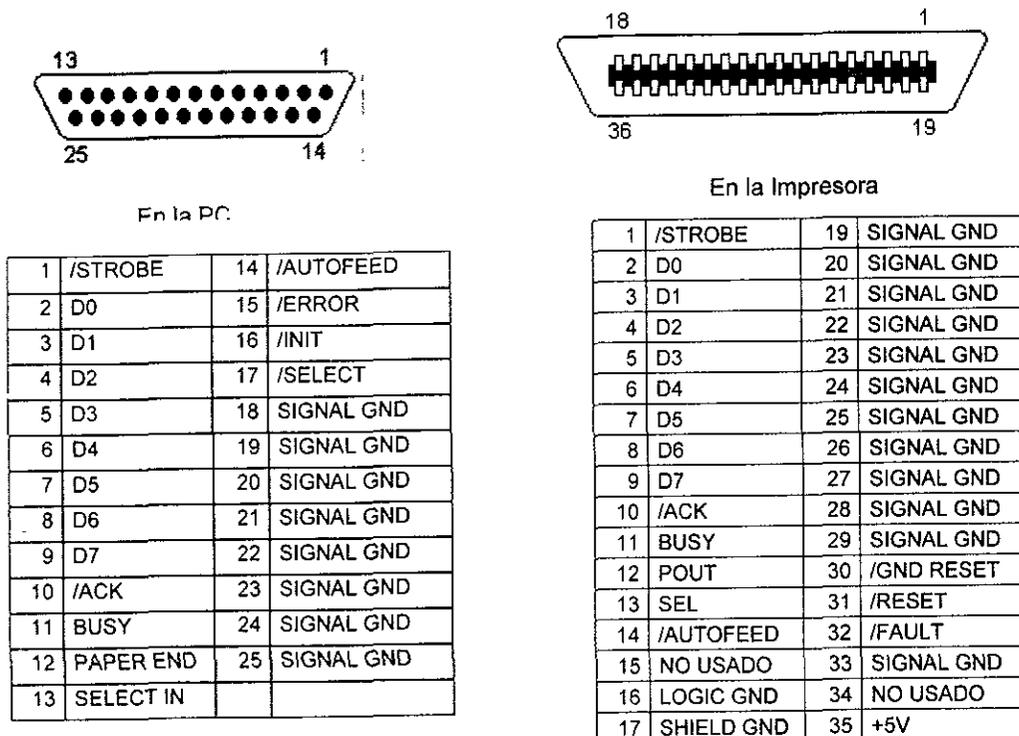


Figura 2.5. La interface *Centronics*

Para cuando la interface *Centronics* fue desarrollada, el periférico más importante que existía era la impresora, debido a esto la interface se hizo unidireccional ya que solo se necesitaba que los datos fluyeran de la computadora hacia cualquier periférico. Con el paso del tiempo se inventaron y

desarrollaron nuevos periféricos que adoptaron la interface paralela pero con la característica de bidireccionalidad.

Lo anterior obligó a los fabricantes de periféricos a pensar en la necesidad de una interface paralela, tal como la *Centronics*, pero mejorada. En 1991 Lexmark, IBM, Texas Instruments y otros se reunieron para discutir un estándar que ofreciera más velocidad de comunicación y que fuera bidireccional [58]. Como resultado de este encuentro y con el auspicio del IEEE (Institute of Electrical and Electronics Engineers) se formó el comité IEEE1284.

El estándar IEEE1284 fue aprobado en marzo de 1994. Este estándar proporciona las características eléctricas y físicas para conectar una computadora a otros periféricos. Las características físicas son iguales a las del ya mencionado *Centronics*.

Cinco modos de operación son especificados por el estándar IEEE1284:

1. **Modo de Compatibilidad.** Este no es más que el original estándar *Centronics* y esta pensado para utilizarse con impresoras de matriz de puntos e impresoras láser anticuadas.
2. **Modo Nibble.** Este modo permite transferir datos hacia la computadora a través de las líneas de estatus. Se le llama así porque las líneas de estatus son enviadas en dos ciclos de transferencia en cada uno de los cuales se transmiten 4 bits. Este modo se utiliza con impresoras.
3. **Modo Byte.** Este modo utiliza programas manejadores (software *drivers*) que deshabilitan a los otros manejadores que controlan a las líneas de datos, es decir, tanto los datos como las señales de control son enviadas a la misma velocidad. Un dato es enviado seguido por los dos ciclos del *modo nibble*.
4. **Modo ECP (*Enhanced Capability Port*).** Es un modo avanzado bidireccional que se utiliza con impresoras y escaners. Permite compresión de datos y comunicación bidireccional a altas velocidades desde 2 hasta 4 Mbytes/segundo. Una característica avanzada de este modo es su direccionamiento de canales aplicable y muy útil al interfazar periféricos multifunción a la computadora. Esto significa que un periférico multifunción puede ejecutar varias funciones a la vez a través de la interface con solo direccionar cada una de ellas a un canal.
5. **Modo EPP (*Enhanced Parallel Port*).** Este modo fue diseñado por Intel, Xircom y Zenith Data Systems para crear una interface de tipo paralelo de alto rendimiento. En este modo se utilizan los llamados ciclos de datos los cuales se utilizan para transferir datos desde la computadora hacia algún periférico o viceversa y los ciclos de direcciones que se utilizan para asignar direcciones de canales o información de comandos. En este modo se pueden alcanzar velocidades de transferencia desde 0.5 hasta 2 Mbytes/segundo y es adecuado para adaptadores de red, adquisición de datos, discos duros portátiles y otros dispositivos.

La computadora debe determinar las características del periférico conectado a ella para escoger el modo más adecuado. A esto se le conoce como *negociación* y no es más que una secuencia de eventos para determinar cual de los modos del estándar IEEE1284 puede manejar el periférico. De este modo un dispositivo viejo no responderá a la secuencia de negociación por lo que se utilizara el modo de compatibilidad para operarlo. Un dispositivo más reciente responderá a la negociación por lo que se puede utilizar un modo más avanzado.

Otro estándar de tipo paralelo es el IEEE488 [24], más comúnmente conocido como GPIB (*General Purpose Interface Bus*) o HPIB (*Hewlett Packard Interface Bus*) puede manejar hasta 15 dispositivos conectados al bus con altas velocidades de transferencia de datos. Fue desarrollado originalmente por la compañía fabricante de instrumentos de análisis Hewlett Packard como una interface para interconectar sus equipos.

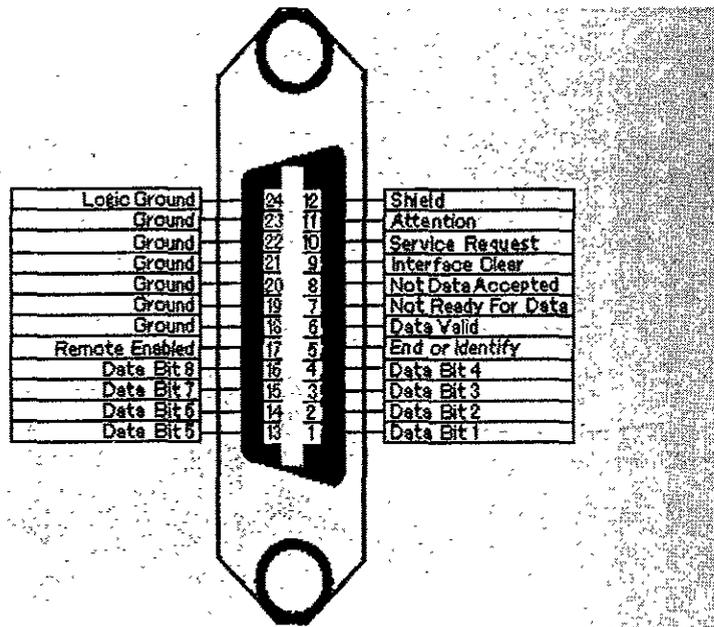


Figura 2.6. Conector de Estándar IEEE488

Debido a que su popularidad creció, otros fabricantes lo incluyeron también en sus instrumentos por lo que después llegó a conocerse como GPIB. En 1975 fue adoptado por el IEEE (Institute of Electrical and Electronic Engineers) como su estándar IEEE488.

El IEEE488 consta de 24 líneas: 8 para datos, 5 de control, 3 de protocolo y 8 para retorno de tierra, como se muestra en la figura 2.6. Como los datos se transmiten en paralelo se pueden alcanzar velocidades de hasta 1 Mbyte por segundo.

La principal limitación del IEEE488 es la longitud de su bus ya que es de 20 metros como máximo con una separación entre cualquiera dos dispositivos de hasta 4 metros.

Los detalles anteriores limitan al bus para un uso industrial, pero lo hacen ideal para un laboratorio debido a que Hewlett Packard cuando lo diseñó lo hizo pensando solo en sus instrumentos de laboratorio.

En la actualidad muchas compañías se dedican a la fabricación de tarjetas para PC que manejan este tipo de estándar las cuales son conectadas directamente al bus de la computadora por medio de alguna ranura de expansión libre de la tarjeta madre y tan solo necesitan que se instale el correspondiente *driver*.

Este tipo de tarjetas ha mejorado algunos de los aspectos antes mencionados a cambio de un menor rendimiento en otros. Por ejemplo, se pueden tener velocidades de transferencia de datos de hasta 2 MByte/segundo pero con una distancia máxima entre cualquiera dos dispositivos de hasta 2 metros.

2.3.2 ESTÁNDARES DE TIPO SERIAL

Con respecto a la conexión tipo serial los tres, estándares más conocidos y utilizados en la actualidad son el RS 232, el RS 422 y el RS 485 [59]. De los tres el RS 232 es el que se desarrolló primero.

En las computadoras personales que utilizan microprocesadores tipo *Pentium* se ha implementado otro tipo de bus serial, el llamado *Universal Serial Bus* o USB el cual en la actualidad comienza a ser ya un puerto que todas las tarjetas madre para ese tipo de microprocesadores tienen de manera corriente. El sistema operativo Windows 98 ya cuenta con *drivers* para este tipo de puertos.

En general, las transmisiones de tipo serial son más lentas que las de tipo paralelo aunque pueden manejar distancias más grandes y requieren de menos líneas de transmisión.

Los tres estándares mencionados tienen en común las letras iniciales RS, las cuales significan Estándar Recomendado por sus siglas en inglés. Actualmente estos estándares son marcados con el prefijo EIA para indicar la organización que avala a dichas normas.

Esto significa que los fabricantes que ponen esta interface en sus productos tienen la libertad de adecuarlo sin seguir estrictamente todas sus especificaciones casi siempre con el fin de obtener productos más baratos pero no siempre completamente funcionales.

El RS232 fue originalmente diseñado para normar la interconexión entre terminales y grandes computadoras *Host* a través de redes de telefonía pública. Para convertir las señales digitales provenientes de la computadora a señales analógicas de audio adecuadas para transmitirse en las redes telefónicas y hacer la función inversa en el otro extremo se ha utilizado hasta nuestros días el *módem*.

Durante los años 60, los enlaces para acceso remoto entre una terminal y un *host*, se realizaban invariablemente a través de una línea telefónica, el problema era que cada fabricante de equipo utilizaba su propia configuración para interfazar a las terminales con las computadoras. Lo anterior significaba que cables, conectores y niveles de voltaje eran diferentes e incompatibles por que el uso de partes de distintos fabricantes requería de adquirir convertidores de voltaje, conectores y cables especiales.

En 1969 la EIA (Electronic Industry Association), los laboratorios Bell y otros más se reunieron con el fin de establecer un estándar recomendado para interfazar lo que llamaron *Data Terminal Equipment* (DTE) con *Data Communications Equipment* (DCE), computadora y *módem* respectivamente [60]. El objetivo de este estándar era simplificar la interconexión de equipos de diferentes fabricantes.

Este estándar definía las características eléctricas, mecánicas y funcionales. Las primeras incluyen parámetros tales como niveles de voltaje e impedancias de cable. La sección mecánica las asignaciones de señales para cada pin del conector. La parte funcional se ocupa de explicar las funciones de cada una de las diferentes señales eléctricas que se usan.

Pronto se vio la necesidad de hacerle revisiones al estándar por lo que llegó a convertirse en el RS232C, estándar RS232 revisión C. Los europeos implementaron un estándar similar llamado V.24 para la descripción funcional y V.28 para sus especificaciones eléctricas, reconocido por el CCITT (Comité Consultatif de Internationale de Telegraphie et Telephonie) [61].

Después de haberse implantado, el RS232 fue ampliamente adoptado por los fabricantes de equipo de computo.

Durante el rápido crecimiento de la industria de las microcomputadoras en los años 80 los fabricantes se dieron cuenta que el RS232 representaba un estándar barato y adecuado para interconectar equipo periférico a las microcomputadoras por lo que decidieron utilizarlo para conectar impresoras, *plotters*, lectoras de cinta magnética, terminales e inclusive otras microcomputadoras.

En la actualidad todas las PC's son fabricadas con por lo menos dos puertos seriales que se rigen bajo el estándar RS232 y que comúnmente son conocidos como COM más un número que los diferencia entre sí.

Entre las principales características del RS232 esta el que admite longitudes de hasta 50 pies (15 metros aproximadamente) entre dispositivos, es una interface de comunicación entre dispositivos punto a punto y su transmisión de datos es del tipo no balanceado y *full dúplex*.

Entre sus principales desventajas se encuentran:

- La velocidad de transmisión de datos esta limitada a 20 kbauds
- La distancia de transmisión esta limitada a 15 metros.
- El estándar no especificaba un conector, lo que ocasionó que algunos diseños que utilizaban el conector de 25 pines no fueran compatibles entre sí.
- La interface utiliza transmisores y receptores no balanceados los cuales son susceptibles a interferencias por ruido inducido en sus líneas de comunicación.
- El diseño global de la interface fue hecho pensando en componentes de tecnología discreta.

En la figura 2.7 se pueden apreciar todas las señales del estándar RS232 en los dos conectores más comunes así como la dirección de cada una de ellas.

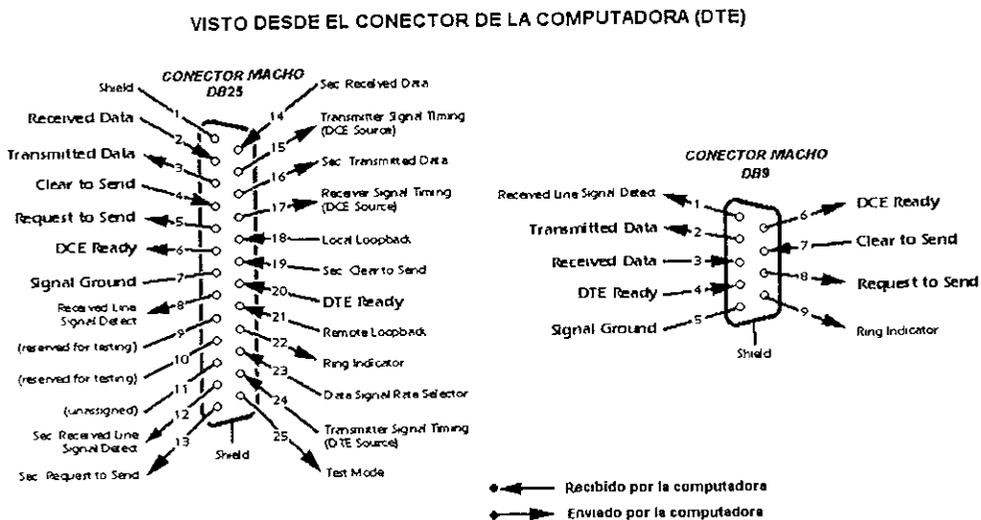


Figura 2.7. Conectores para el RS232

También es importante mencionar que algunas de las limitaciones del estándar RS232 se debieron a cuestiones de la tecnología que existía en el momento. Por ejemplo, la distancia límite de 15 metros se debe a que el cable que se utilizaba en esa época, tenía una gran capacitancia, en la actualidad ya hay cables con mejores características, por lo que hay reportes de distancias de transmisión más grandes que las indicadas por el RS232 sin ningún problema.

En 1973 [62], el RS232 fue revisado debido al rápido desarrollo de la tecnología. Como resultado de esta revisión surgieron tres nuevos estándares que utilizaban nuevas tecnologías de interfazamiento con el fin de superar las limitaciones del RS232.

Esos nuevos estándares se caracterizaron por:

- Mantener compatibilidad con el RS232.
- Soportar velocidades de transmisión de datos más altas y sobre distancias más grandes que las del RS232.
- Resolver el problema mecánico de la interface, resultado de la ausencia de especificaciones para un conector en el RS232.
- Mejorar las características eléctricas de la interface utilizando circuitos de comunicación balanceados.

Uno de los estándares resultantes de esa revisión fue el RS422, el cual define una interface eléctrica de doble terminación que puede manejar velocidades de transmisión de datos más allá de los 20 kbps del RS232 [63-64]. Las conexiones mecánicas para esta interface son proporcionadas por el estándar RS449.

Como se mencionó anteriormente, las comunicaciones de tipo balanceado o también conocidas como diferenciales ofrecen mejores características en muchas aplicaciones que sus contrapartes, las comunicaciones no balanceadas. Esto se debe a ayudan a nulificar los efectos de corrimientos de tierras e inducción de ruido en las señales.

El RS422 es un estándar de comunicación de tipo diferencial que permite longitudes de hasta 4000 pies (1300 metros aproximadamente), con una velocidad de transferencia de datos de hasta 100 kbps.

El RS422 es especificado para aplicaciones de multipunto donde un *driver* es conectado a un bus con hasta 10 receptores. Hay que tener en cuenta que este estándar no puede realizar funciones multipunto completas ya que para tener esta característica se necesitaría de una red de varios *drivers* y receptores conectados a un bus donde cualquier nodo puede transmitir y recibir datos.

Dispositivos que utilizan el RS422 se utilizan para implementar redes con las características mencionadas en el párrafo anterior. Este tipo de redes se utilizan en un modo *half dúplex* donde un solo dispositivo maestro envía un comando a uno de varios dispositivos esclavos de la red.

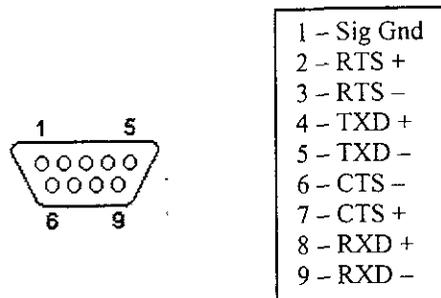


Figura 2.8. Conector para el estándar RS422

El RS422 utiliza un diseño de 4 líneas: transmitir +, transmitir -, recibir + y recibir -. En la figura 2.8 se muestra la configuración de este estándar para un conector DB9 y visto desde la computadora.

El otro estándar de tipo serial que abordamos aquí es el RS485; este aún no es considerado como un equipamiento estándar en las computadoras tipo PC al igual que el RS422, pero es muy utilizado en el mundo de los sistemas de adquisición de datos [63-64].

De entre las características principales del RS485 sobresale el hecho de que puede soportar hasta 32 *drivers* y 32 receptores conectados en una red multipunto. Esto significa que se pueden tener comunicaciones de tipo bidireccional, *half dúplex* y multipunto en un simple par de cables.

El RS485 permite longitudes de cable hasta de 4000 pies con una velocidad de transmisión de datos de hasta 100kbps ó una velocidad de 10 Mbps con una longitud de cable de hasta 40 pies.

La transmisión de datos en el RS485 es de tipo balanceado y puede operar con solo una fuente de 5 Volts.

Cabe señalar que este estándar especifica únicamente las características eléctricas tanto de los *drivers* como de los receptores pero no especifica ni recomienda algún protocolo de comunicación. Esto provoca que dispositivos con este estándar, pero fabricados por diferentes compañías, tengan problemas al intentar interconectarse en la misma red.

El uso típico del RS485 es en una computadora tipo PC conectada a varios dispositivos que pueden ser direccionados por ella y que comparten el mismo par de cables.

El cable que recomienda el estándar para implementar este tipo de comunicación es el conocido como par trenzado con una impedancia de 120 ohms y con terminadores también de 120 ohms en cada extremo de la línea de comunicación.

En resumen, el RS485 maneja longitudes de cable y velocidades de intercambio de datos similares al RS422, pero acepta hasta 32 *drivers* y 32 receptores conectados a la vez, cuenta con líneas positiva (+) y negativa (-) de señal y cuando un dispositivo necesita enviar datos conecta su transmisor a las líneas de señal, y cuando necesita recibirlos conecta su receptor.

2.4 OTRAS CONSIDERACIONES IMPORTANTES

La gran mayoría de las PC's son fabricadas con dos puertos que manejan la interface RS232 comúnmente conocidos como puertos serie (COM1 y COM2) y un puerto *Centronics* llamado puerto paralelo (LPT1).

Para poder tener en la computadora cualquiera de las otras interfaces (RS422, RS485 o IEEE488) se necesita ya sea de una tarjeta adaptadora que se inserte directamente en la tarjeta madre o de un convertidor de cualquiera de esas interfaces a la RS232 o *Centronics* así como del *driver* correspondiente.

Los convertidores tienden a limitar el rendimiento de los puertos especialmente si se trata del IEEE488. Las tarjetas adaptadoras se insertan en alguna ranura libre de la tarjeta madre, suelen ser más flexibles y económicas que los convertidores.

Todo lo anteriormente mencionado esta relacionado con el hardware, pero debe tenerse en cuenta también algunos aspectos acerca del software.

Hay un buen número de lenguajes [56] que soportan directamente las comunicaciones de los estándares RS232 y *Centronics* tales como BASIC, C FORTRAN, etc.

El desarrollar software para otro tipo de estándares diferentes al RS232 y al *Centronics* requiere de hacer uso de un programa especial llamado *driver* el cual debe ser proporcionado por el fabricante de la tarjeta adaptadora.

El *driver* permite al programador, direccionar al adaptador sin estar íntimamente familiarizado con los aspectos de hardware del controlador.

2.5 CONCLUSIONES

La idea principal con respecto a este trabajo de tesis era desarrollar un sistema de adquisición de datos echando mano de los recursos humanos y materiales con que se contaba en el laboratorio con el fin de realizar solo los gastos estrictamente necesarios.

En cuanto al hardware, el sistema debe tener como una de sus principales características, la facilidad de instalación para el usuario, es decir, que él pueda instalar y desinstalar la interface con un mínimo de conocimientos acerca del hardware.

Otra característica importante del hardware, es el hecho de que pudiera ser portable, es decir, que pudiese conectarse a cualquier tipo de PC que tenga el puerto de comunicación adecuado y el software de adquisición de datos instalado.

Entre los motivos para que el sistema y en especial el hardware tenga esta característica se encuentran el considerar que en un futuro la computadora pudiese descomponerse y ser sustituida por otra sin mayores problemas.

También se piensa utilizar el sistema en un futuro para realizar análisis de campo de otro tipo de parámetros, en un buen número de ocasiones, por lo que debe poder instalarse en una computadora de tipo laptop.

En relación con el software la condición principal es que la aplicación corra en la plataforma Windows ya sea Windows 95 o posteriores ya que este es el sistema operativo con el cual trabajan las PC's de los laboratorios móviles.

Una de las principales ventajas de utilizar Windows 95 como sistema operativo es que la computadora no estará esclava de la aplicación que se encargue de la adquisición de datos y permitirá que de manera simultánea se puedan desarrollar otros trabajos en la computadora, tales como análisis de datos, reportes y oficios.

Las velocidades de muestreo para este caso no se requiere que sean muy altas debido a que como máximo se requiere de obtener mediciones cada segundo.

Los analizadores de donde se va obtener la señal analógica se encuentran todos dentro de un laboratorio móvil y la máxima distancia posible entre ellos y la interface sería de 6 metros aproximadamente.

Después de revisar las características de los principales estándares de comunicación que puede uno encontrar en una computadora tipo PC y tomando en cuenta las consideraciones enumeradas en los párrafos anteriores se determinó que la opción más adecuada para cubrir las necesidades de la aplicación es el interconectar la tarjeta de adquisición de datos y la computadora a través de uno de los puertos con que cuenta la PC de manera estándar.

Las computadoras tipo PC con que los laboratorios móviles cuentan, y todas en general, son ensambladas usualmente con un puerto paralelo y dos puertos seriales comúnmente conocidos como LPT1, COMM1 y COMM2 respectivamente.

Ambos tipos de puertos pueden cubrir perfectamente las necesidades del sistema en cuanto a velocidad de transferencia de datos, longitud de cable entre la interface y los analizadores; además de que el utilizar alguno de estos puertos evitaría el realizar gasto alguno para adquirir dicha interface.

Otra ventaja importante y mencionada en la primera parte de este capítulo es que el usuario no tendrá la necesidad de abrir la computadora para instalar la tarjeta de adquisición de datos con los riesgos que esto implica.

Para conectar la tarjeta de adquisición de datos a la computadora el usuario solo tendrá que utilizar un cable adecuado para el puerto que se utilizará.

Ahora para decidir cual de los dos tipos de puertos se utilizaría, se tomó en cuenta que el puerto paralelo esta siempre ocupado por la impresora y que no es adecuado que el usuario la desconecte para poner en su lugar otro dispositivo a menos que sea estrictamente necesario.

La opción de utilizar alguno de los puertos seriales es la más conveniente para este trabajo, ya que las computadoras personales tienen por lo menos dos de ellos e inclusive pueden ser expandidos hasta cinco.

Las computadoras utilizadas en los laboratorios móviles tienen por lo menos un puerto serial disponible ya que el otro puede estar ocupado por el *Mouse* si es que no cuentan con un conector del tipo PS 2 para este dispositivo.

Finalmente la configuración del sistema queda como se muestra en la figura 2.9. La tarjeta de adquisición de datos concentra las señales provenientes de cada analizador, las digitaliza y las transmite a la computadora a través de uno de sus puertos seriales.

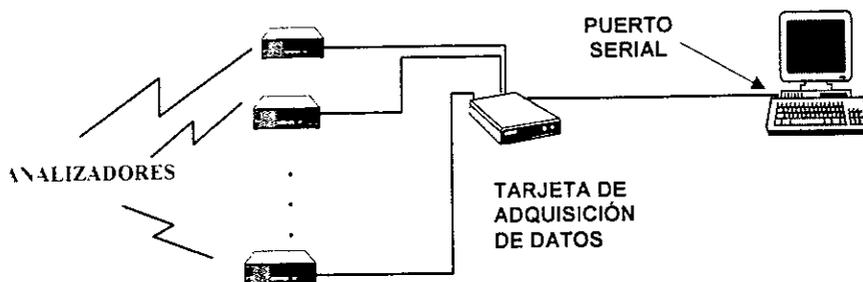


Figura 2.9 Configuración final

INTRODUCCIÓN

Después de que en los capítulos anteriores se han abordado temas relacionados con los sistemas de adquisición de datos basados en PC, por un lado, y la interface de comunicación por el otro, en este capítulo se explica como se desarrolló la tarjeta de adquisición de datos.

Es importante volver a hacer hincapié nuevamente en que la idea al desarrollar el sistema era el utilizar los recursos que se tenían a la mano para realizar el menor número de gastos económicos posibles.

Los recursos a que se refiere el párrafo anterior no son otra cosa que software, hardware y conocimientos.

En cuanto al punto que nos ocupa, como ya se mencionó en el capítulo de los sistemas de adquisición de datos, hay solo dos caminos posibles para tener una tarjeta de adquisición de datos que nos permita interfazar cada uno de los equipos analizadores con la computadora; el primero consiste en comprar una existente en el mercado con los pros y contras que esto implica, el segundo es desarrollar una que se ajuste a las necesidades que se tengan.

En el laboratorio del grupo de electrónica de la Subdirección de Protección Ambiental se contaba con un sistema de desarrollo del microcontrolador de Motorola *MC68HC11*, además de que algunos becarios que realizaban su servicio social contaban con la experiencia acerca de este circuito integrado.

En base a lo anterior se decidió desarrollar una tarjeta de adquisición de datos en el mismo laboratorio.

La tarjeta fue desarrollada con la valiosa ayuda de algunos becarios. En las primeras pruebas colaboraron Víctor Hugo López Rodríguez y Omar Tapia Laguna. La versión final fue desarrollada por el becario Emmanuel Dámaso, todos ellos bajo la dirección y supervisión del autor de esta tesis.

El trabajo de desarrollo de la tarjeta servirá a la última de las personas mencionadas como base para realizar su trabajo de tesis con el fin de obtener su título de Ingeniero en Comunicaciones y Electrónica en el Instituto Politécnico Nacional.

Me permito hacer un reconocimiento a todos por su dedicación y esfuerzo en esta tarea.

A continuación se hace un breve repaso de las características que debe tener la tarjeta de adquisición de datos.

3.1 CARACTERÍSTICAS DE LA TARJETA DE ADQUISICIÓN DE DATOS

Los siguientes, son los aspectos más importantes que esta tarjeta debe tener:

- Monitorear hasta 8 señales analógicas de manera simultánea.
- Manejar señales analógicas de voltaje en los rangos 0-1 Volt, 0-5 Volts y 0-10 Volts.
- Poder configurar la ganancia para cada canal analógico, de acuerdo al rango de trabajo de cada uno de ellos.
- Cada uno de sus canales analógicos debe poder manejar cualquiera de los tres rangos antes mencionados.
- Manejar el estándar de comunicaciones RS232 para comunicarse con la computadora personal donde se este corriendo la aplicación.
- Realizar, a solicitud de la computadora, una conversión de cada uno de los canales que se estén utilizando por lo menos cada segundo.

El número de analizadores que se encuentran en los laboratorios móviles puede ser de hasta 7, pero en un futuro se tiene contemplado agregar más con el fin de poder monitorear hasta 8 variables.

Esto significa que la tarjeta debe ser capaz de monitorear de manera simultánea hasta 8 señales analógicas distintas y enviar esa información a la computadora cuando esto se le solicite.

Debido a que los laboratorios móviles están equipados con analizadores de emisiones que cuentan con salidas analógicas de voltaje con rangos de 0-1 Volt, 0-5 Volts y 0-10 Volts se requería de una interface que pudiera trabajar con los tres rangos sin ningún problema.

Cada uno de esos rangos de voltaje representará de manera proporcional la escala de trabajo de cada uno de los distintos analizadores. Por ejemplo, si un analizador esta trabajando con una escala de 0-1000 PPM y tiene una salida analógica de voltaje de 0-5 Volts, 0 Volts corresponderán a 0 PPM, 2.5 Volts a 500 PPM y 5 Volts a 1000 PPM.

Los datos que la tarjeta obtenga para cada uno de los canales analógicos al realizar una conversión analógica/digital, deberán ser enviados a la computadora para que esta los opere además de guardarlos, desplegarlos y graficarlos.

Dicha información será enviada a la computadora por la tarjeta de adquisición de datos a través del puerto serie y utilizando el protocolo de comunicación RS232.

La información de tipo ambiental que manejan los analizadores no tiende a tener variaciones significativas en lapsos cortos de tiempo, es decir, no tienen componentes de alta frecuencia por lo que la velocidad máxima de muestreo que se necesita es del orden de 1 cada segundo para obtener el promedio de cada una de ellas en lapsos que pueden variar desde 1 hasta 5 minutos.

La obtención del promedio es recomendada por las normas oficiales mexicanas relacionadas [65-70] con este tipo de estudios y es ampliamente utilizada en los sistemas de adquisición de datos como una técnica para disminuir el ruido que pueda introducirse en el sistema.

Sobre la base de estas características se procedió a analizar la posibilidad de utilizar el sistema de desarrollo del microcontrolador *MC68HC11*. A continuación se hace una revisión del mismo.

3.2 EL MICROCONTROLADOR 68HC11

Es importante mencionar que los microcontroladores en general son dispositivos electrónicos que son la siguiente generación de microprocesadores. La diferencia básica entre un microprocesador tradicional y un microcontrolador es que el segundo además de poseer las mismas partes que el primero, ya puede contar con memoria RAM, ROM, puertos de E/S, posibilidad de manejar interfaces de comunicación como

las mencionadas en el capítulo anterior e inclusive convertidores analógico/digital dentro de un mismo circuito integrado; esto le da un potencial tremendo a este tipo de dispositivos.

El MC68HC11 es fabricado con un material producto de la tecnología de alta densidad de integración llamada Metal Oxido Semiconductor Complementaria (HCMOS por sus siglas en inglés).

A los microcontroladores se les conoce también como MCU (Unidad Micro Controladora) por sus siglas en inglés.

El MC68HC11 es un MCU de 8 bits, alta velocidad y bajo consumo de energía y posee un bus no multiplexado con una velocidad nominal de 2 MHz. En la figura 3.1 se muestra un diagrama a bloques de este dispositivo.

La tecnología que se utiliza para fabricarlo permite combinar un tamaño pequeño, velocidades altas y bajo consumo de energía.

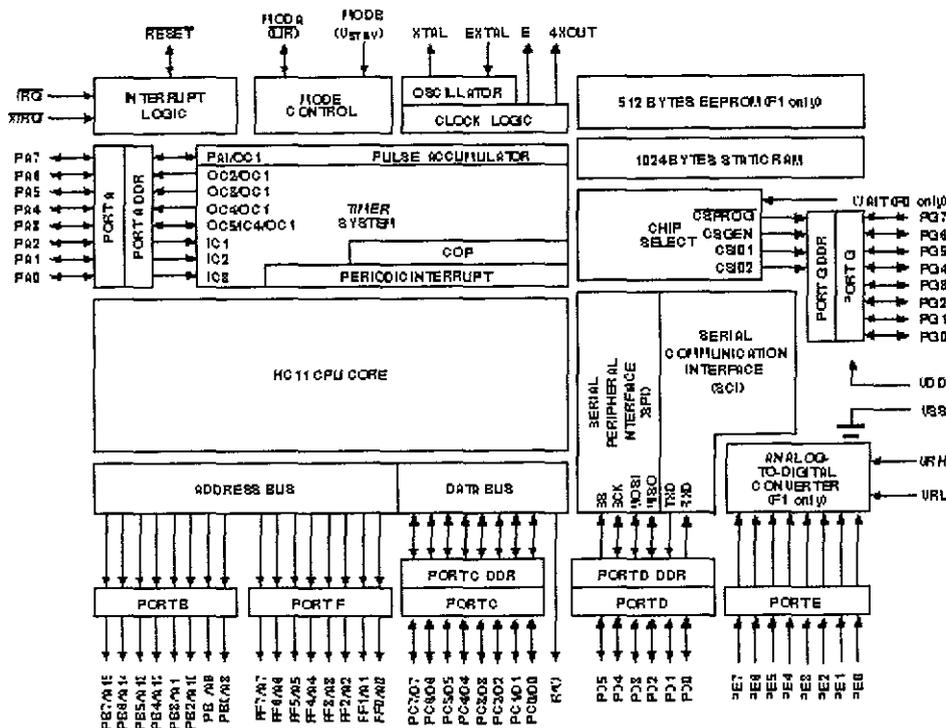


Figura 3.1. El microcontrolador MC68HC11

El MC68HC11 esta formado por las siguientes partes de hardware y software [71]:

- *Sistema temporizador expandido de 16 bits con predivisor programable de 4 opciones.*
- *Interfaz de Comunicación Serie (SCI) mejorada, con formato registro sin retorno acero (NRZ).*
- *Convertidor Analógico/Digital (A/D) de 8 bits y 8 canales.*
- *Mecanismo de protección de bloques de memoria para EEPROM y CONFIG.*
- *Bus expandido no multiplexado.*
- *Encapsulado de 68 pines.*
- *Modos STOP Y WAIT de ahorro de consumo de energía.*
- *Capacidad de direccionamiento de memoria de 64 K.*
- *Interfaz periférica serie (SPI).*
- *512 Bytes de EEPROM.*
- *Circuito acumulador de pulsos de 8 bits.*
- *1024 Bytes de RAM estática, salvada totalmente durante los modos de bajo consumo de energía (Stanby).*
- *Instrucciones de brinco y prueba de bits.*
- *Circuito de interrupción de tiempo real.*
- *Cuatro chips selects programables.*
- *Sistema vigilante de operación correcta de la computadora (COP).*

3.3 LA IDEA PRINCIPAL

Para que la tarjeta de adquisición de datos cumpla con su función, en primer lugar la aplicación que permite la adquisición de datos en la computadora deberá estar corriendo ya que es quien lleva el control fundamental del proceso.

En segundo lugar, la computadora y la tarjeta de adquisición de datos deberán estar interconectados físicamente por un cable que permite el flujo bidireccional de datos.

El cable de interconexión debe ser el adecuado para una comunicación de tipo serial RS232 y debe corresponder al que se describe más adelante.

En tercer lugar, el formato de comunicaciones deberá estar configurado adecuadamente en la aplicación que corre en la computadora para corresponder al de la tarjeta de adquisición de datos, es

decir, la velocidad, numero de bits por dato, paridad, etc. deben ser los mismos para que la información sea interpretada correctamente.

Debido a que se considera que no es conveniente que el usuario pueda estar modificando estos parámetros, porque potencialmente pueden existir errores en la configuración además de daños físicos, se decidió que la tarjeta de adquisición de datos trabaje con una configuración fija para el puerto serie y cuyas características se enumeran a continuación:

- Velocidad = 9600 bauds.
- Paridad = sin paridad
- Bits de parada = 1
- Bits de dato = 8

Es obvio que el puerto serial, al que este conectado el cable de comunicación, deberá corresponder al que se encuentra configurado en el programa de aplicación o de otra manera no se podrá establecer esta.

Después de haberse cumplido con las condiciones antes enumeradas, ya se podrá proceder a establecer la comunicación entre la tarjeta de adquisición de datos y la computadora para que esta última capture los datos provenientes de la primera.

En la figura 3.2 se muestra en un diagrama de flujo, el mecanismo de intercambio de información entre la computadora y la tarjeta.

Como se puede apreciar en la figura 3.2, la tarjeta al momento de ser energizada espera un carácter especial para poder establecer la comunicación con la computadora, ese carácter es "P" que fue escogido como una forma de indicar que la computadora le solicita a la tarjeta que programe el rango de trabajo de cada uno de los canales analógicos. La tarjeta en respuesta envía el carácter "R" para indicar que la petición fue recibida y que la comunicación puede establecerse correctamente.

Después de que la computadora recibe la respuesta y de que concluye que la comunicación se estableció correctamente envía una cadena de 8 caracteres que indican cada uno el rango de trabajo. A esa cadena le hemos llamado cadena de configuración. Los caracteres de la cadena de configuración vienen en orden ascendente, es decir, el primer carácter es el correspondiente al primer canal analógico, el segundo es el correspondiente al segundo canal analógico y así sucesivamente.

Los caracteres solo pueden tomar tres posibles valores como se indica a continuación:

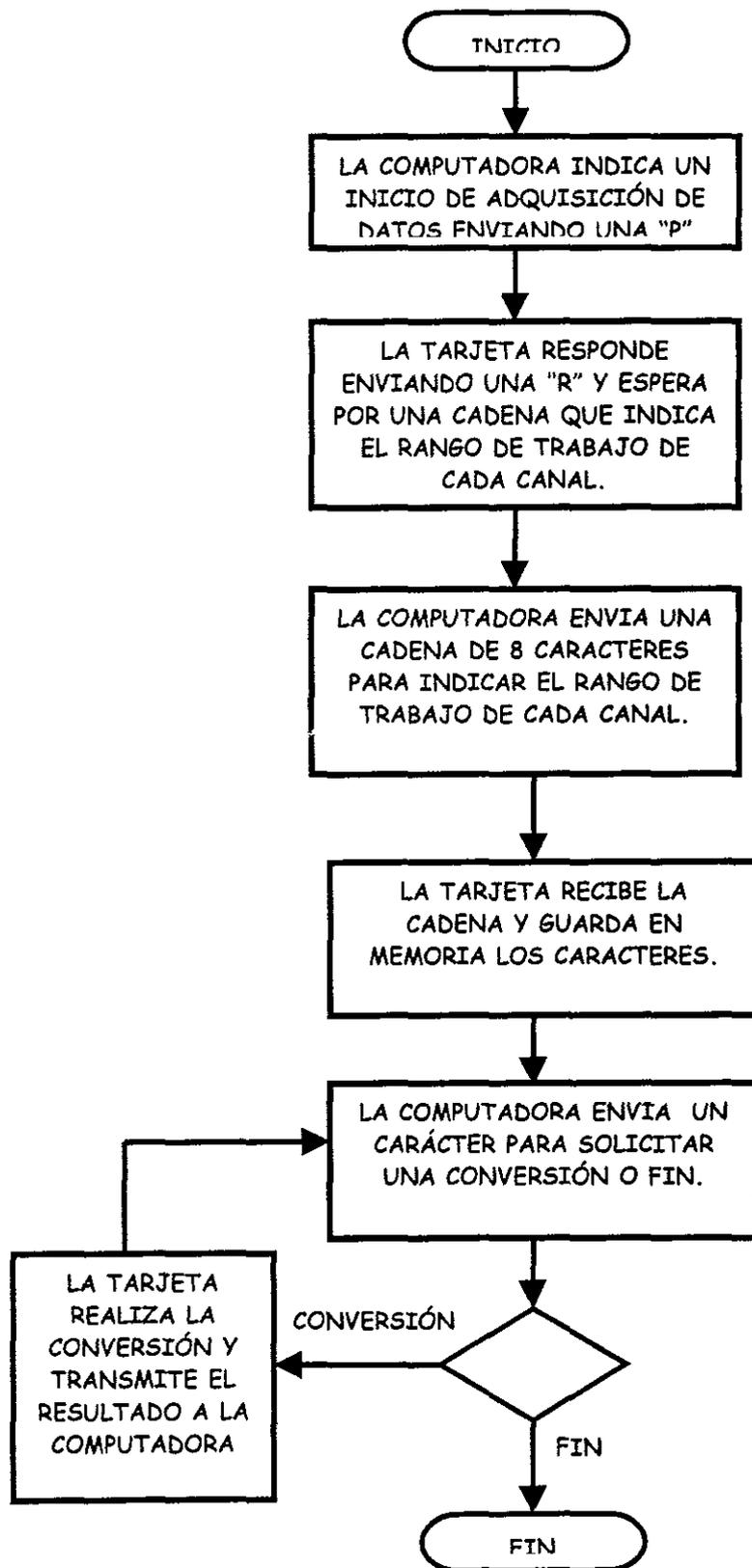


Figura 3.2 Comunicación Tarjeta-Computadora

- 0 ó código ASCII 48
- 1 ó código ASCII 49
- 2 ó código ASCII 50

En la tabla 3.1 se indica la interpretación de cada uno de los caracteres con relación al rango de trabajo.

Carácter	Rango del canal
0	0 - 1 volts
1	0 - 5 volts
2	0 - 10 volts

Tabla 3.1. Relación carácter – rango de trabajo

Una cadena de configuración como "22122012" significa que los canales 1, 2,4,5 y 8 tienen un rango de trabajo de 0 – 10 volts, los canales 3 y 7 tienen un rango de 0 – 5 volts y el canal 6 tiene un rango de 0 - 1 volts.

Inmediatamente después de recibir la cadena de configuración, el microcontrolador de la tarjeta los separa y guarda en su memoria RAM para que cada ocasión que tenga que realizar una conversión este dato sea leído y el microcontrolador realice los ajustes necesarios en la etapa de amplificación para que la señal sea leída correctamente.

De aquí en adelante la tarjeta solo espera que la computadora le envíe periódicamente la señal de conversión que es representada con el carácter "T" el cual se escogió para indicar que la tarjeta realice la conversión de los ocho canales e inmediatamente los envíe a la computadora.

La computadora es quien indica cada cuando debe hacerse una conversión en base a los tiempos programados por el usuario del sistema cuando este realiza la configuración del mismo. Esto se explicará detalladamente en el capítulo siguiente.

Siempre que la tarjeta recibe una solicitud de conversión por medio del carácter "T" responde a la computadora con el carácter "R" para indicar que ah recibido la solicitud. Esto se implemento como una forma de ir checando el estado de la comunicación entre la tarjeta y la computadora también.

En cuanto a la conversión de las señales analógicas, la idea es que las señales correspondientes a los ocho canales analógicos sean convertidas una por una y el valor digital obtenido en cada conversión sea guardado en memoria para que cuando se completen las ocho sean enviadas en forma de una cadena de caracteres. A esta cadena le nombramos la cadena de datos. Los caracteres de la cadena de datos también son enviados por la tarjeta en forma sucesiva y ascendente, es decir, el primer dato enviado corresponde al canal 1, el segundo al canal 2 y así sucesivamente para todos los demás.

Los caracteres de la cadena de datos son enviados en código ASCII, es decir, aunque arriban a la computadora como un carácter la información más importe es el código ASCII que representan. Por ejemplo, si la computadora recibe el carácter "A" lo que importa es el número que representa que no es otra cosa que el 65 decimal.

3.4 UNA PRIMERA PROPUESTA

De todas las características que se enumeraron en párrafos anteriores, el contar con un convertidor analógico/digital y hasta 8 entradas analógicas, más la posibilidad de manejar la comunicación de tipo serial fueron las que principalmente nos decidieron a utilizar este dispositivo.

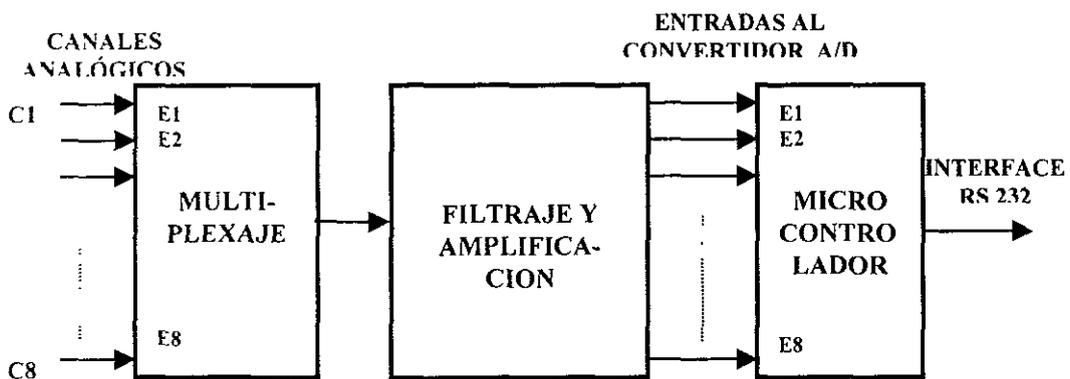


Figura 3.3. Primera aproximación

El hecho de que el microcontrolador ya cuente con un convertidor analógico/digital dentro del circuito integrado nos ahorra puertos de comunicación debido a que los datos que arroja este son accedidos directamente por el CPU.

El microcontrolador cuenta con hardware y software para manejar las comunicaciones de tipo serial, en especial la comunicación asíncrona del estándar RS232. Es decir, dentro de su conjunto de instrucciones tiene algunas dedicadas exclusivamente a manejar este tipo de comunicaciones, además de contar con los circuitos electrónicos internos adecuados [71].

Lo enumerado en los dos párrafos anteriores significa en pocas palabras que solo se tiene que desarrollar los circuitos electrónicos necesarios para multiplexar, filtrar, amplificar y aislar las señales analógicas provenientes de cada uno de los analizadores del laboratorio móvil.

En la figura 3.3 se puede observar la configuración de la tarjeta de adquisición de datos que fue originalmente propuesta.

3.5 SEGUNDA APROXIMACIÓN

Después de realizar algunas pruebas con esta configuración se llegó a la conclusión de que era innecesario utilizar las 8 entradas analógicas del microcontrolador como se explica a continuación.

En la etapa de amplificación y filtraje se utiliza un solo circuito para realizar dicha función, es decir, solo uno de los canales analógicos puede ser filtrado y amplificado a la vez debido a que la etapa de multiplexaje tiene esta característica también.

Si se quisiera que todas las entradas analógicas entraran simultáneamente al convertidor analógico digital del microcontrolador es necesario utilizar un circuito de filtraje y amplificación para cada una de ellas, lo que traería como consecuencia que físicamente la tarjeta de adquisición de datos fuese de tamaño muy grande.

Es importante aclarar que aunque el *MC68HC11* tiene 8 entradas analógicas no puede realizar una conversión analógica/digital simultánea de todas ellas, es decir, solo puede realizar la conversión de una a la vez [72].

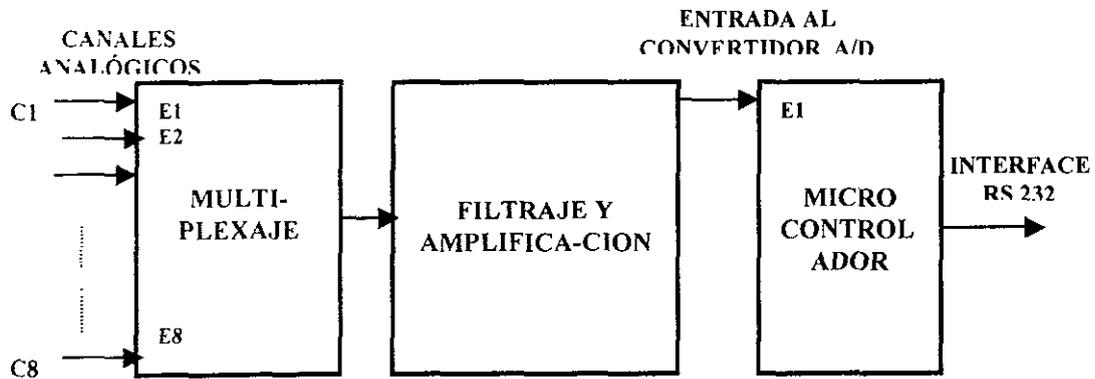


Figura 3.4. Segunda propuesta

En general, para el desarrollo de cualquier tarjeta de adquisición de datos es necesario decidir sobre que cual de las características debe prevalecer, si el tamaño de la interface o el muestreo simultaneo.

Cuando se tiene el caso en que es estrictamente necesario realizar una conversión simultanea se necesita implementar un circuito que contenga un convertidor analógico/digital para cada entrada analógica, además de sus correspondientes etapas de amplificación, filtraje, etc.

En nuestro caso, no se requiere que la conversión sea estrictamente simultanea y además existen técnicas, como las explicadas en la sección 3.9 de este capítulo, que permiten acercarse bastante a esa simultaneidad. Por otro lado, el circuito integrado que se utiliza para realizar la conversión tiene el inconveniente, ya mencionado, de que no puede convertir simultáneamente.

Finalmente se decidió utilizar una etapa de multiplexaje, un solo circuito de filtraje y amplificación y una sola entrada analógica del microcontrolador como se muestra en la figura 3.4.

3.6 ETAPA DE MULTIPLEXAJE

Esta etapa es la encargada de seleccionar cual de las entradas analógicas pasará a la etapa de amplificación y filtraje para ser convertida como se puede apreciar en la figura 3.5.

La etapa de multiplexaje es implementada utilizando switches analógicos y multiplexores además de circuitos de acoplamiento de impedancias como se verá en forma mas detallada en párrafos posteriores.

Como es entendible, el control de la secuencia de entrada de los canales es controlado por el programa que ejecuta el 68HC11 a través de algunos de sus puertos.

ETAPA DE MULTIPLEXAJE

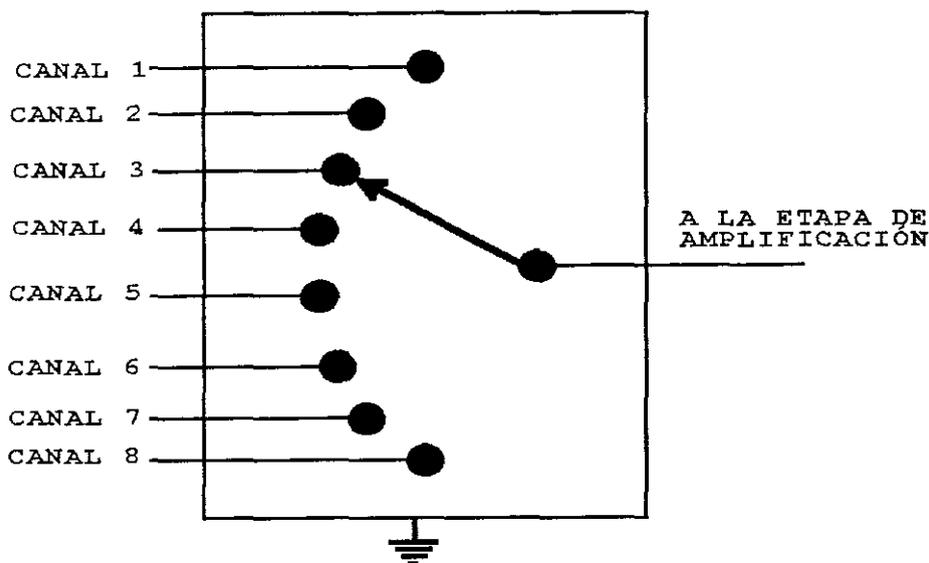


Figura 3.5. Etapa de Multiplexaje

En cada conversión son convertidas las señales presentes en cada uno de los ocho canales analógicos, aunque en la configuración de la aplicación que esta corriendo en la computadora algunos de ellos se encuentren deshabilitados ya que quien discrimina la información perteneciente a estos es el programa de la aplicación.

3.7 ETAPA DE FILTRAJE Y AMPLIFICACIÓN

En esta etapa, cada señal analógica proveniente de la etapa de multiplexaje es filtrada y amplificada de acuerdo su rango de voltaje configurado por el usuario en la computadora y enviado por esta a la tarjeta.

La amplificación para cada canal necesita ser adecuada debido a que no existen restricciones acerca del rango de trabajo para cada uno de ellos, es decir, cualquiera de los ocho canales puede ser configurado en cualquiera de los tres posibles rangos de trabajo.

El convertidor analógico/digital del 68HC11 solo acepta entradas de voltaje a un nivel máximo de 5 volts pues de lo contrario puede dañarse. Esto significa que cuando un canal de la tarjeta esta manejando señales en el rango de 0 – 10 volts debe reducirse su nivel a la mitad para que cuando a la entrada del canal se tengan 10 volts al convertidor solo llegue una señal de 5 volts.

En el otro extremo, cuando un canal esta configurado para trabajar con señales en el rango de 0 – 1 volts este debe ser amplificado 5 veces para que cuando a la entrada se tenga una señal de 1 Volt al convertidor leguen 5 volts.

ETAPA DE FILTRAJE Y AMPLIFICACIÓN

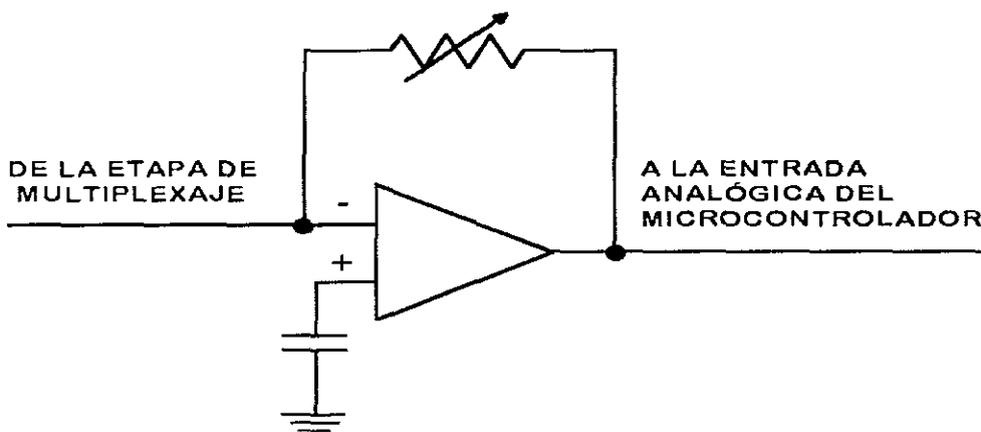


Figura 3.6. Etapa de filtraje y amplificación

El último caso es cuando un canal es configurado para trabajar con señales en el rango de 0 – 5 volts. En este caso la señal proveniente de la entrada del canal no necesita ser ni amplificada ni reducida, es

decir, debe llegar con el mismo nivel de voltaje con que entró o lo que es lo mismo, debe tener una ganancia unitaria en la etapa de amplificación.

El manejo de tres diferentes rangos de trabajo por la tarjeta de adquisición de datos fue implementado utilizando una etapa de amplificación de ganancia variable. La ganancia para cada canal depende de su correspondiente carácter en la cadena de configuración mencionada unos párrafos antes.

Siempre que la señal de un canal va a ser convertida el microprocesador revisa en memoria el dato correspondiente a su rango de trabajo para que, por medio de unos switches analógicos y multiplexaje, ponga la ganancia adecuada en la etapa de amplificación. En este caso también el control de los switches es realizado a través de algunos de los puertos del microcontrolador como se detallará más adelante.

Otra opción para poner la ganancia adecuada para cada canal era el utilizar ocho etapas de amplificación y filtraje antes de la etapa de multiplexaje, cada una para un canal, pero tenia que dejarse fija. Esto significa que los canales solo podrían trabajar en un rango predeterminado o que el usuario lo cambiara por medio de hardware con las consabidas molestias.

3.8 PRUEBAS REALIZADAS

Las pruebas realizadas tanto al hardware como al software del sistema de adquisición de datos siempre habían sido realizadas en el laboratorio del grupo de trabajo en las instalaciones del Instituto Mexicano del Petróleo.

Debido a que los laboratorios móviles se encuentran casi de tiempo completo realizando sus servicios en las distintas instalaciones de PEMEX y a que solo regresan por periodos muy cortos de tiempo para que se les brinde mantenimiento preventivo, es muy difícil contar con alguno de ellos con el suficiente tiempo libre para realizar pruebas.

Como se podrá comprender existieron pocas oportunidades de poder disponer de los laboratorios móviles para llevar a cabo pruebas del sistema de adquisición de datos, pero finalmente en enero de 1999 se pudo realizar pruebas a todo el sistema de adquisición de datos en el laboratorio móvil número 3.

Estas pruebas consistieron en instalar el sistema en el laboratorio para observar el comportamiento del mismo, tanto del hardware como del software y compararlo contra el sistema semi-manual con que se

trabaja cotidianamente. Otro de los objetivos era mostrarlo a sus futuros usuarios para obtener sus opiniones y sugerencias acerca del mismo.

Una de las cosas que principalmente se quería observar era lo relacionado al nivel de ruido con que las señales analógicas provenientes de los diferentes equipos analizarían a la tarjeta de adquisición de datos. Este parámetro es importante de observar debido a que el nivel de ruido puede ser de tal magnitud que puede distorsionar de manera significativa a la señal como se explicó en el primer capítulo de este trabajo.

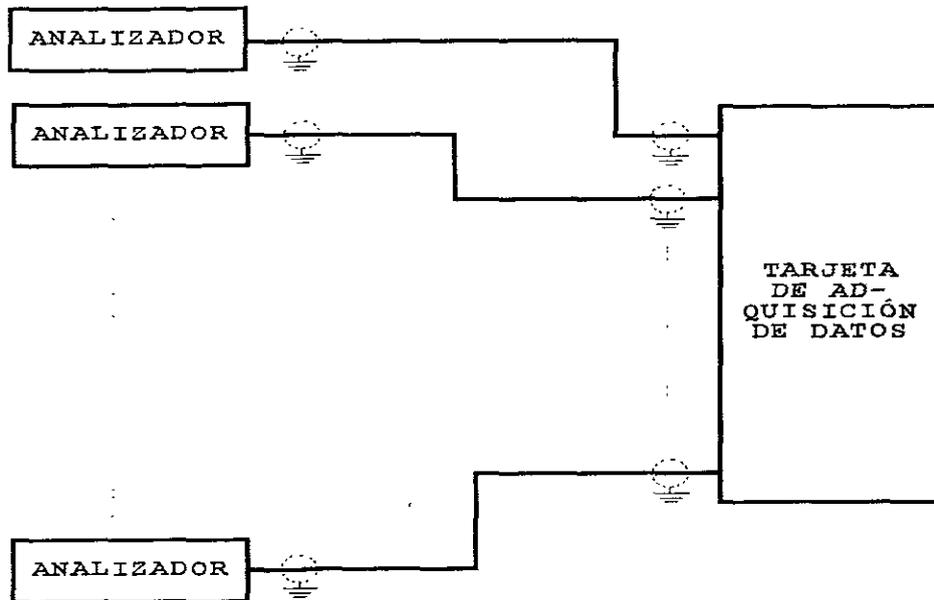


Figura 3.7. Blindaje de las señales

Observando el nivel de la señal leída por el sistema de adquisición de datos y comparándolo con el leído por el sistema con que se trabaja cotidianamente se determinó que el nivel de ruido era bastante grande debido a que en el laboratorio móvil hay bastantes fuentes potenciales de ruido siendo las principales las de los sistemas de clima artificial y el compresor de aire.

Para resolver este problema se decidió probar el llevar la señal de los analizadores hacia la tarjeta de adquisición de datos por medio de cable blindado con dos cables internos. Los dos cables internos son para llevar la señal analógica (positivo y negativo) y el blindaje se utiliza para conectar la tierra física del sistema de alimentación eléctrico del laboratorio móvil. Esta técnica es ampliamente utilizada para que el blindaje del cable evite se induzca ruido en los cables que llevan la señal. En la figura 3.7 se puede observar esta implementación. Con la configuración mostrada en dicha figura, la magnitud del ruido inducido en las señales analógicas se volvió insignificante, de tal forma que ya no representó un problema.

Otro parámetro importante era el relacionado al voltaje de modo común que se podía generar entre cada una de las entradas analógicas y la tarjeta de adquisición de datos y que puede ser capaz de crear fuegos artificiales en los circuitos electrónicos como ya se explico en el capítulo de los sistemas de adquisición de datos. En dicho capítulo también se explico que existe una técnica para aislar la etapa analógica de la etapa digital como una forma de prevenir los efectos de dicho voltaje.

El sistema que se desarrolló no contaba con dicha etapa al momento en que se probó por lo que existía la posibilidad de que se necesitara implementarla si el voltaje antes mencionado fuera excesivamente grande.

Al realizar las pruebas ese voltaje resultó ser muy bajo de tal forma que no fue necesario implementar la etapa de aislamiento antes mencionada. La razón es muy simple, como todos los analizadores estan conectados a la misma tierra física, el voltaje de modo comun es pequeño o prácticamente nulo.

Tal vez el parámetro más importante que fue observado en estas pruebas es el relacionado a la resolución del convertidor analógico/digital de la tarjeta. Al momento de realizar las pruebas y comparar las lecturas del sistema contra las del *datalogger* y contra las de los analizadores se observó que estaban un tanto cuant alejadas, es decir, las lecturas de estos dos últimos eran muy semejantes y las diferencias entre ellos se pueden atribuir a las limitaciones del *datalogger*.

Por el otro lado la diferencia entre las lecturas del sistema de adquisición de datos y los analizadores eran más grandes. Después de analizar las posibles razones de esta situación se determino que esas diferencias se debían también a limitaciones del convertidor analógico/digital en cuanto a su resolución como se explica a continuación.

Después de revisar las especificaciones técnicas del *datalogger* se encontró que cuenta con un convertidor analógico/digital cuya resolución es de 12 bits y una entrada analógica que permite un máximo de 5 Volts por lo que puede resolver hasta 1.22 mVolts lo que para una escala de 5000 PPM representa una resolución de 1.22 PMM lo cual es suficiente para los requerimientos de los usuarios.

En el caso del sistema que se estaba probando, el convertidor tiene una resolución de 8 bits y una entrada analógica de 5 Volts máximo por lo que puede resolver hasta 19.5 mVolts lo que para una escala de 5000 PPM representa una resolución de 19.5 PPM. Como es obvio la diferencia entre las resoluciones es casi 16 veces mejor en el caso del *datalogger*. Al consultar la opinión de los usuarios estos consideraron inadecuada la resolución del sistema por lo que hubo necesidad de implementar otro tipo de convertidor con mejor resolución.

3.9 VERSIÓN FINAL

El hecho de tener que utilizar un convertidor analógico/digital diferente al que el *68HC11* posee, significó modificar de manera importante el circuito originalmente propuesto sobre todo en lo relacionado a los puertos del microcontrolador que se tienen que utilizar para capturar los datos que el convertidor arroje, pues en el caso del *68HC11* toda la comunicación entre el microprocesador y el convertidor es transparente para el programador.

Es evidente que el nuevo convertidor debía ser por lo menos de 12 bits de resolución para poder estar al parejo del que contiene el *datalogger*. Después de buscar los convertidores de 12 bits o de mayor resolución disponibles en el mercado mexicano, para conservar la idea original de solo utilizar componentes que se consiguieran fácilmente en México, se optó por utilizar el convertidor *ADC1241* de *National Semiconductor* mostrado en la figura 3.8.

De entre los motivos que nos decidieron a utilizar este convertidor tal vez el principal es el hecho de que es uno de los de mayor resolución de que se pueden encontrar fácilmente en México, aunque no menos importante es su relativo bajo costo, 15 dólares por unidad aproximadamente.

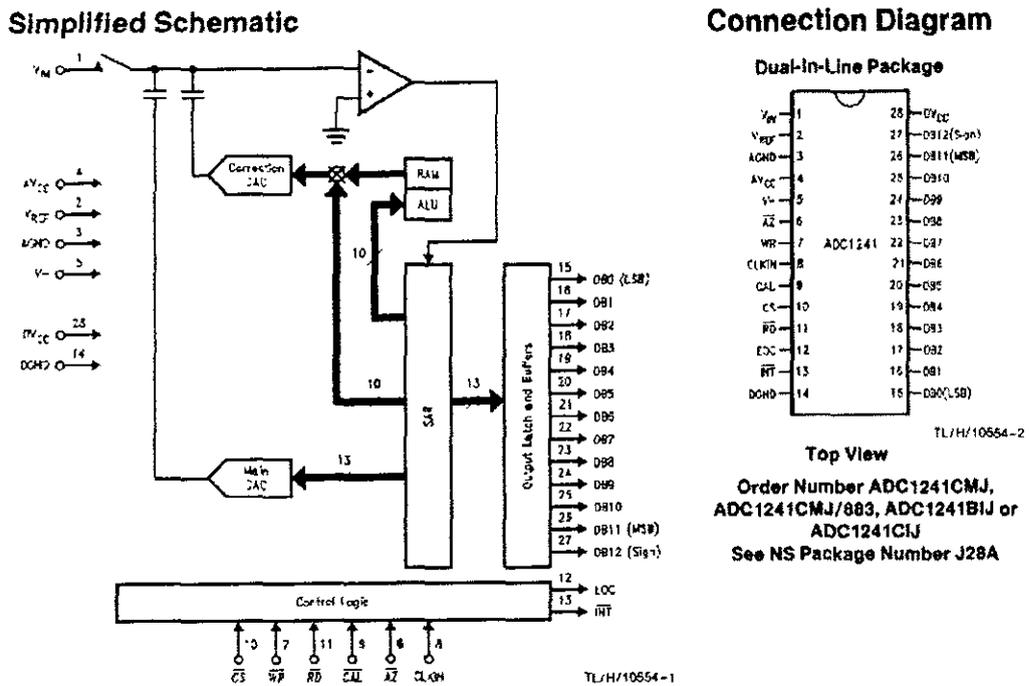


Figura 3.8. El convertidor ADC1241

Sus características técnicas más importantes para nosotros son [73]:

- Es compatible con μ procesadores.
- Cuenta con posibilidad de autocalibración.
- Puede realizar un ciclo de auto-cero para corregir el error de cero durante cada conversión.
- Muestreo y retención de voltaje internos.
- Puede manejar señales de entrada bipolares si se polariza con $\pm 5 V_{DC}$ y un voltaje de referencia sencillo de $5 V_{DC}$.
- Sus entradas/salidas digitales son compatibles con los niveles lógicos TTL o CMOS.
- Puede determinar el signo de la señal analógica que convierte.
- Tiene un tiempo de conversión máximo de $13.8 \mu s$.
- Utiliza la técnica de aproximaciones sucesivas para realizar las conversiones.

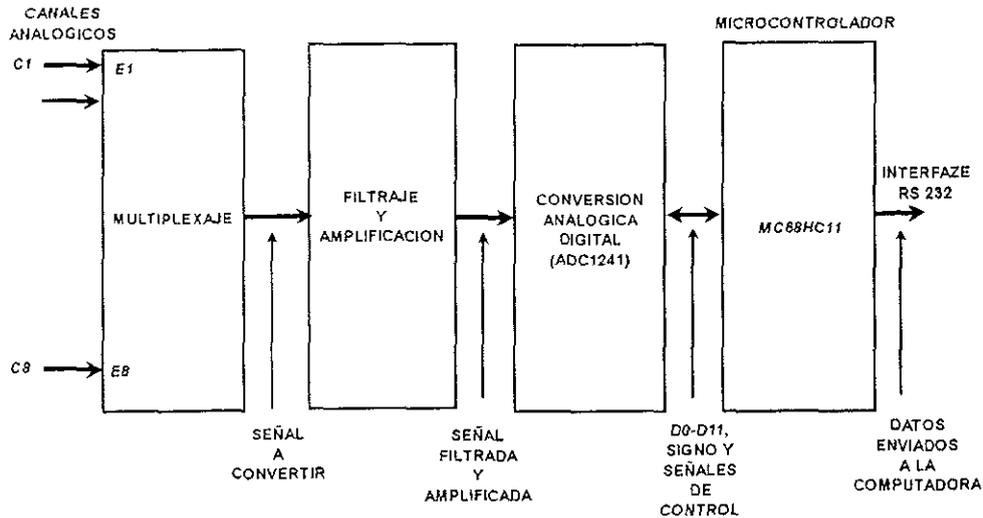


Figura 3.9. Versión Final

Este convertidor es recomendado por el fabricante para aplicaciones tales como procesamiento digital de señales, control de procesos con alta resolución e instrumentación. Esta última muy relacionada con el tema de este trabajo de tesis.

Para utilizar este convertidor se requiere que el microcontrolador tenga libres 12 puertos para el dato, 1 puerto para el signo y 5 puertos para el control de la conversión, mientras que en la versión que utiliza el convertidor del 68HC11 esto no era necesario ya que su convertidor solo ocupa 8 de los puertos para las 8 entradas analógicas que puede manejar.

Al tener que utilizar al ADC1241 es obvio que no solo lo relacionado con los circuitos electrónicos tenía que cambiar, sino que el programa que el 68HC11 tiene que ejecutar para controlar a los diferentes circuitos también tuvo que ser modificado significativamente lo cual requirió de más tiempo para realizar los cambios y pruebas necesarios.

En la figura 3.9 se puede apreciar el diagrama a bloques de la versión final de la tarjeta de adquisición de datos después de realizar los cambios. Como se puede ver, se añadió la etapa del convertidor que en las versiones anteriores no existía debido a que se encontraba dentro del microcontrolador.

ETAPA DE MULTIPLEXAJE

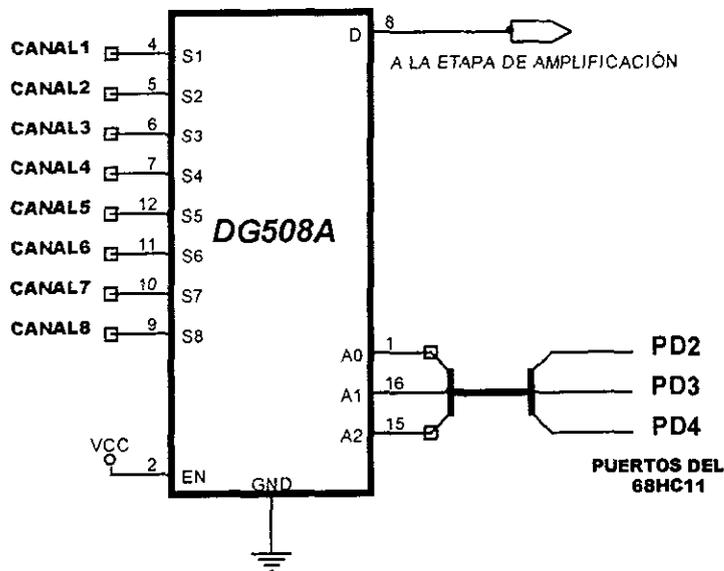


Figura 3.10. Etapa de Multiplexaje con el CI DG508

A continuación se aborda la etapa de multiplexaje con un poco más de detalle, esta finalmente fue construida alrededor del *circuito integrado* DG508 ya que después de varias pruebas con distintos dispositivos electrónicos de este tipo resultó ser el que mejor comportamiento tiene. El DG508A [74] se puede conseguir fácilmente en México y su costo es relativamente bajo.

Otros circuitos integrados con los cuales se hicieron pruebas para construir esta etapa son el 4051 y el 4066 de la serie CMOS pero no tienen un buen comportamiento sobre todo cuando las señales de voltaje en los canales alcanzan niveles cercanos a los 10 volts, pues su respuesta no es lineal.

Otro problema que se presentó con estos circuitos es el hecho de que las señales de control que provienen del 68HC11 tienen niveles lógicos TTL mientras que estos manejan niveles lógicos CMOS por lo que era necesario utilizar un circuito convertidor de niveles lógicos de TTL a CMOS, situación que no se presenta con el DG508. En la figura 3.10 se puede apreciar un *diagrama electrónico* de esta etapa ya construida con dicho circuito integrado.

En esta figura se aprecia que se utilizan tres puertos del 68HC11 para realizar el control del multiplexaje (PD2-PD4), por lo que dependiendo de la combinación de los valores lógicos de estas

señales de control uno solo de los canales analógicos saldrá por la pata 8 del DG508A para ir hacia la etapa de filtraje y amplificación.

En la tabla 3.2 se muestra la combinación necesaria para que un canal específico sea seleccionado. El microcontrolador 68HC11 es el encargado de poner la combinación adecuada para seleccionar al canal deseado. Esto lo realiza ejecutando una subrutina escrita en lenguaje ensamblador.

PD2	PD3	PD4	CANAL SELECCIONADO
0	0	0	CANAL 1
0	0	1	CANAL 2
0	1	0	CANAL 3
0	1	1	CANAL 4
1	0	0	CANAL 5
1	0	1	CANAL 6
1	1	0	CANAL 7
1	1	1	CANAL 8

Tabla 3.2. Multiplexaje de canales

Después de que un canal analógico específico es seleccionado, su señal pasa a la etapa de filtraje y amplificación para ser adecuada y que pueda llegar a la etapa de conversión analógica/digital con la forma y nivel más adecuados.

La etapa de filtraje y amplificación esta implementada principalmente con circuitos electrónicos basados en amplificadores operacionales. En la figura 3.11 se aprecia que el primer circuito es un amplificador operacional con una configuración conocida como seguidor la cual tiene la característica de tener una ganancia unitaria pero sirve para acoplar las impedancias de esta etapa y la anterior, esto con el fin de que la última no represente una carga eléctrica para la primera. Como se puede apreciar, se utiliza un circuito integrado LF356 [75] el cual es un amplificador operacional con entrada de tipo FET con una muy alta impedancia de entrada.

El siguiente circuito de la etapa de filtraje y amplificación es un inversor de ganancia unitaria, el cual es necesario para invertir la polaridad de su voltaje de entrada ya que en el siguiente circuito se le vuelve a invertir para dejarla como es originalmente. Este circuito es implementado también utilizando un LF356. La ganancia unitaria se logra poniendo resistencias del mismo valor tanto de en la entrada como en la

de 0 – 5 volts, la ganancia de la etapa de amplificación debe ser cambiada cuando se pase de la conversión del canal 1 a la conversión del canal 2.

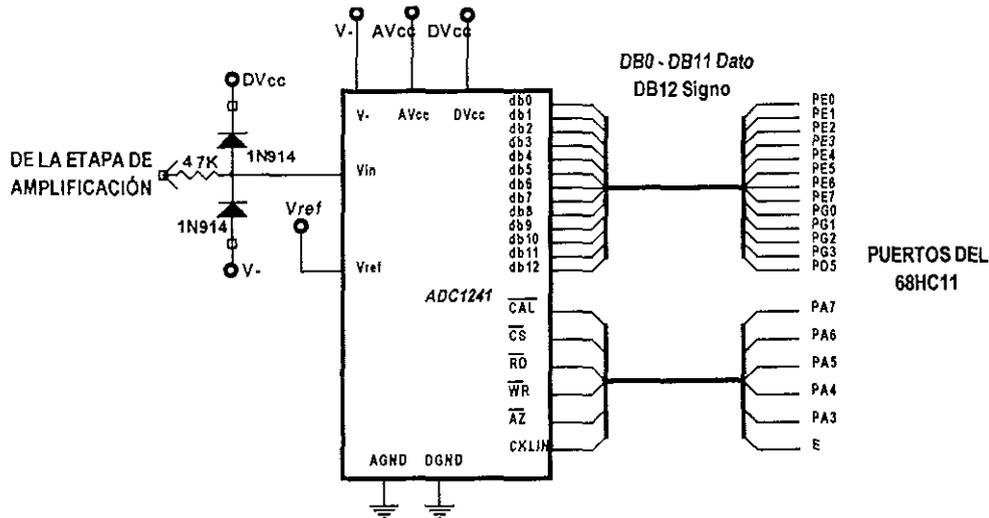


Figura 3.12. Conversión A/D con el ADC1241

Esto se logra utilizando *switches* analógicos para seleccionar una resistencia de entrada al amplificador operacional y así junto con la resistencia de retroalimentación obtener la ganancia adecuada. Para escoger esa resistencia de entrada se usa el circuito integrado DG509A [74] el cual está compuesto de 4 *switches* analógicos que pueden ser seleccionados uno solo a la vez por medio de unas señales de control

Para cuando el rango de trabajo de la señal de un canal analógico es de 0-1 volts este debe ser amplificado en una magnitud de 5 para que como máximo se obtengan 5 volts. Para un rango de 0-5 volts la ganancia es de 1 y para un rango de 0-10 volts la ganancia es de 0.5.

Anteriormente se explicó que la aplicación que está corriendo en la computadora se encarga de mandar una cadena de 8 caracteres llamada cadena de configuración. Cada uno de esos caracteres indica el rango de trabajo para un canal específico. Esos caracteres son guardados en la memoria de la

tarjeta de adquisición de datos para que cada ocasión que esta deba hacer una conversión analógica/digital los lea y ponga la ganancia adecuada para cada canal.

Este último circuito de la etapa de amplificación es implementado con un circuito integrado de baja voltaje de *offset* de entrada como es el *OP07* [76], el cual es muy utilizado en cuestiones de instrumentación.

En la figura 3.11 también se puede apreciar que existe un circuito de balanceo para que se pueda ajustar el "cero" de toda la etapa, es decir, que cuando se tengan cero volts a la entrada de la etapa, en su salida se obtengan también cero volts.

Cabe anotar que las resistencias utilizadas en este último circuito deben ser de muy alta precisión para eliminar cualquier error posible y que la ganancia sea de la magnitud exacta que se necesita. Por tal razón se utilizan resistencias de muy bajo coeficiente de temperatura ya que este es el parámetro que más comúnmente hace variar el valor de una resistencia.

Después de que la señal analógica ha sido multiplexada y amplificada es alimentada a la etapa de conversión analógica/digital para ser digitalizada y que el *68HC11* pueda leerla y enviarla a la computadora.

Como ya se explicó antes, para construir esta etapa se utiliza el circuito integrado *ADC1241* como convertidor analógico/digital. En la figura 3.12 se muestra esta etapa.

Esta etapa de conversión analógica/digital es mucho más precisa que la que se tenía con el convertidor del *68HC11* por las razones antes mencionadas, pero requiere de circuitos electrónicos adicionales para funcionar adecuadamente.

Esos circuitos son una fuente bipolar de ± 5 Volts_{DC} para polarizar al convertidor, un voltaje de referencia de 5 Volts_{DC} y la señal de reloj que es tomada del microprocesador.

La fuente bipolar se utiliza para que el convertidor pueda manejar señales positivas y negativas y así poder aprovechar su característica de proporcionar el signo o polaridad del voltaje de entrada.

El voltaje de referencia es necesario para que el convertidor pueda comparar el voltaje presente en su entrada analógica cuando se le solicita una conversión contra una referencia y así entregar un código binario que sea proporcional a la relación entre el voltaje de entrada y el voltaje de referencia. Este último

voltaje puede ser directamente la fuente positiva de polarización pero es recomendable que sea independiente y formada por componentes de muy alta estabilidad con la temperatura.

Por último, la señal de reloj es necesaria para que el convertidor pueda realizar las diferentes etapas de una conversión. La velocidad de conversión es directamente proporcional a esta señal de reloj. En nuestro caso se aprovecha que el *68HC11* tiene una salida de reloj de 2 MHz que puede ser utilizada por otro dispositivo.

En la entrada analógica del *ADC1241* es necesario poner un circuito de protección para él, debido a que si el voltaje presente en esta pata del circuito integrado es mayor a $DV_{cc} + 0.05$ volts o menor a $V - 0.05$ volts el convertidor puede sufrir un daño permanente. Este circuito de protección es muy sencillo y es implementado con una resistencia de 4.7 K Ω y dos diodos 1N914 con el cual nunca se tendrá un voltaje que sobrepase los límites mencionados.

La conversión es controlada y manejada directamente por el *68HC11* a través de sus puertos PA3 a PA7 con los cuales se manipulan las señales de control de la conversión AZ, WR, RD, CS y CAL las cuales deben cambiar de nivel lógico con cierta secuencia para poder realizar primero una conversión y después leer el dato obtenido de ella.

El dato de 12 bits obtenido de una conversión es leído por el microcontrolador a través de sus puertos PE0 – PE7 y PG0 – PG3 mientras que el signo es leído a través del puerto PD5.

3.10 CONVERSIÓN ANALÓGICA/DIGITAL

La tarjeta de adquisición de datos realiza una conversión de los 8 canales solo cuando se lo solicita el programa que esta corriendo en la computadora. La solicitud de conversión se hace cuando la computadora le envía el carácter "T" a la tarjeta de adquisición de datos.

Es necesario que antes de realizar cualquier conversión, la computadora transmita a la tarjeta de adquisición la cadena de configuración, de la cual ya se habló en párrafos anteriores. Esta cadena de 8 caracteres le indica a la tarjeta el rango de voltaje de cada canal analógico para que su correspondiente ganancia en la etapa de amplificación sea controlada por el *68HC11*.

Lo primero que hace la tarjeta de adquisición de datos al ser energizada es esperar por el carácter "P" que le transmite la computadora y que sirve para indicar que se va a comenzar una adquisición de datos y que la tarjeta debe esperar inmediatamente la cadena de configuración.

Como respuesta la tarjeta de adquisición transmite a la computadora el carácter "R", el cual después de ser leído por esta última indicara que la comunicación entre ambas pudo establecerse adecuadamente.

Inmediatamente después de que la comunicación se estableció, el programa en la computadora envía una solicitud de conversión de manera periódica como ya se explico antes. La frecuencia con que es solicitada una conversión es conocida comúnmente como periodo de muestreo.

Quien marca la frecuencia de conversión es el usuario por medio de la configuración de tiempos en la aplicación. La máxima frecuencia de conversión que la aplicación puede tener es una vez cada segundo. Esto debido a que esta frecuencia es la máxima que los operadores de los laboratorios móviles necesitan, pero la tarjeta de adquisición de datos tiene capacidad para manejar velocidades de muestreo más rápidas.

Cada vez que una conversión es solicitada, la tarjeta de adquisición realiza la conversión de los 8 canales sin importar que alguno o algunos de ellos no estén en uso. Quien se encarga de discriminar los datos es el programa en la computadora.

Es obvio que debido a la configuración de la tarjeta de adquisición de datos, esta no puede realizar la conversión de los 8 canales de manera simultanea sino que tiene que hacerlo uno a uno. La diferencia de tiempo entre conversiones de canales adyacentes es el tiempo que se tarda en realizar la conversión de un canal.

Como se explicó en el capítulo 1, existen varias técnicas para realizar un muestreo simultaneo de todos los canales que puede manejar una tarjeta de adquisición de datos. Aunque el hecho de tener que hacer un muestreo de manera simultanea depende directamente del fenómeno que este bajo monitoreo y del uso que se le vaya a dar a los datos colectados.

En nuestro caso, no es importante realizar muestreos estrictamente simultáneos debido a que las señales no tienen unas variaciones de frecuencias altas. Es por eso que se utiliza el multiplexaje para realizar un muestreo casi simultaneo.

Para realizar muestreos simultáneos en caso de que no se cuente con un convertidor para cada canal ni los circuitos de amplificación necesarios, como es el caso nuestro, existe otra posibilidad con la técnica de muestreo y retención simultáneos (*Simultaneous Sample and Hold* por sus siglas en Inglés) [77].

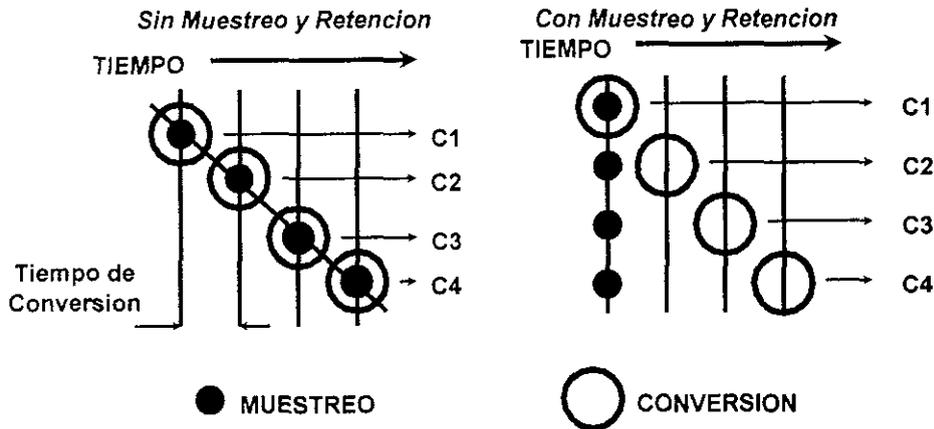


Figura 3.13. Muestreo y Retención Simultáneos

Esta técnica consiste en tener un circuito de muestreo y retención para cada canal analógico con que cuente la tarjeta de adquisición de datos, de tal forma que cuando se va a realizar una conversión analógica/digital de ellos, primero se realice el muestreo y retención de manera simultánea y después se realice la conversión de cada uno de ellos, de esta forma aunque la conversión no se realiza de manera simultánea los datos corresponden al mismo instante de tiempo. Los detalles se pueden apreciar en la figura 3.13 con un ejemplo para 4 canales.

En la figura 3.13 se puede apreciar que aunque finalmente en ambas opciones se tienen los datos disponibles en el mismo lapso de tiempo la diferencia estriba en que en el primer caso no pertenecen al mismo instante en el segundo caso sí. Debe ser evidente que el costo del segundo caso es mayor que el del primero ya que es incrementado por el costo del circuito de muestreo y retención simultáneos.

Existe una aproximación para obtener conversiones casi simultáneas utilizando la configuración del primer caso de la figura anterior. Esta consiste en realizar un muestreo y conversión de los canales analógicos a la máxima velocidad de la tarjeta de adquisición de datos y después esperar un periodo de tiempo definido por el usuario antes de realizar esta acción nuevamente.

Con esto, los canales son muestreados y convertidos con diferencias en el tiempo muy pequeñas, las cuales están en relación directa con las velocidades de retardo de los circuitos electrónicos involucrados. A esta técnica se le conoce como *burst* o ráfaga [77]. En la figura 3.14 se muestra la diferencia entre el modo estándar y el de *burst*.

La técnica de *burst* se adapta perfectamente a las necesidades de muestreo para las variables ambientales que se desea muestrear. Como se explico anteriormente, el programa que corre en la computadora es el indica en que momento debe realizarse una conversión de los 8 canales, en base a la configuración de tiempos de muestreo que el usuario elija.

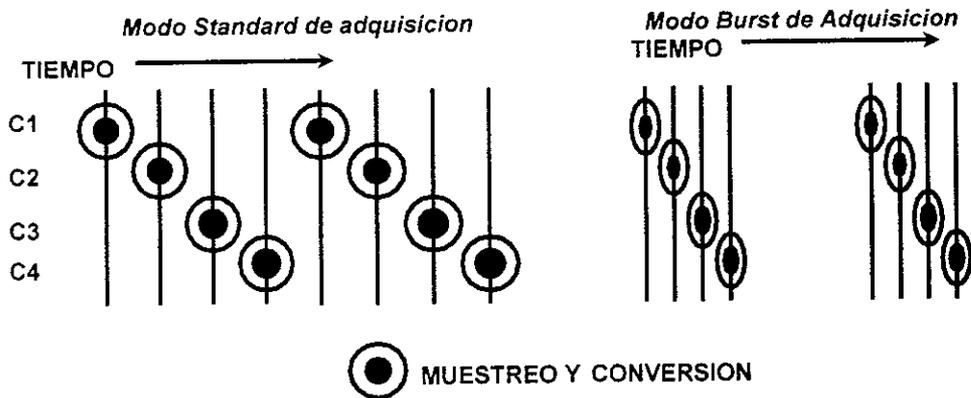


Figura 3.14. conversión estándar y *Burst*

Después de que la tarjeta recibe una solicitud, esta realiza la conversión de cada uno de los canales analógicos de manera ascendente, es decir comienza con el canal 1 continua con el canal 2 y así sucesivamente hasta finalizar con el canal 8.

Siempre que se realiza la conversión de algún canal analógico, su dato digital correspondiente es guardado en memoria para que cuando se finalice la conversión de todos los canales, todos los datos sean transmitidos de manera masiva a través de la interface que conecta a la tarjeta con la computadora, esto con el fin de facilitar su lectura e interpretación.

No sobra decir que debido a que los datos correspondientes son de 12 bits más 1 bit de signo y como las localidades de memoria son de 8 bits se necesita ocupar 16 bits, 2 localidades, para guardar un dato de conversión correspondiente a un solo canal. Como los datos son de 13 bits, 12 bits de dato más 1 bit de signo, sobran siempre 3 bits, dos de los cuales se utilizan como se explica más adelante.

Como ya se explico unos párrafos antes la conversión es controlada por algunos puertos del 68HC11.

3.11 CODIFICACIÓN DE LOS DATOS

Antes de que los datos sean transmitidos al puerto serial de la computadora, deben ser adecuados para que arriben en forma correcta.

El código que se utiliza para codificarlos es el muy conocido ASCII de 8 bits. Este código sirve para transmitir caracteres, por lo que una secuencia de 8 bits tiene un significado específico y único en este código. En consecuencia, cada dato correspondiente a un canal esta formado por dos caracteres, 16 bits.

Como el dato de cada canal son dos caracteres, los datos correspondientes a la conversión de los 8 canales son una cadena de 16 caracteres, 2 por canal. Esta cadena de 16 caracteres es la que es transmitida a la computadora y es leída por la aplicación de adquisición de datos que esta corriendo en la misma.

En resumen, cada vez que la computadora solicita una conversión a la tarjeta de adquisición, si todo esta operando adecuadamente, recibe como respuesta una cadena de 16 caracteres.

Los dos caracteres que forman el dato correspondiente a un canal contienen la siguiente información:

- El primer carácter corresponde a los 8 bits menos significativos del dato.
- Los Primeros cuatro bits del segundo carácter representan los bits más significativos del dato.
- El quinto bit del segundo carácter representa al signo del dato, es decir, contiene la información acerca de la polaridad de la señal leída en el canal.
- El sexto y séptimo bits del segundo carácter son usados para indicar si el primer y/o segundo carácter representa un cero, como se explicará más adelante.
- El octavo bit del segundo carácter nunca es utilizado por lo que siempre tiene como valor un cero lógico.

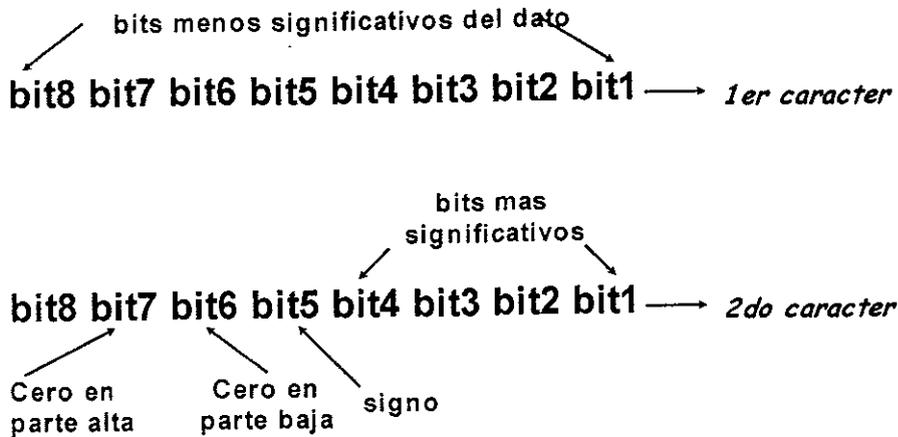


Figura 3.15. Significado de cada carácter

En la figura 3.15, se muestra de manera gráfica esta situación.

Por otro lado, en la figura 3.16 se muestra como ya unidos ambos caracteres tienen un solo significado, es decir, representan un dato de 12 bits, y como tal, son interpretados por el programa que corre en la computadora. El primer bit del primer carácter es el bit menos significativo del dato, mientras que el cuarto bit del segundo carácter es el bit más significativo del dato.

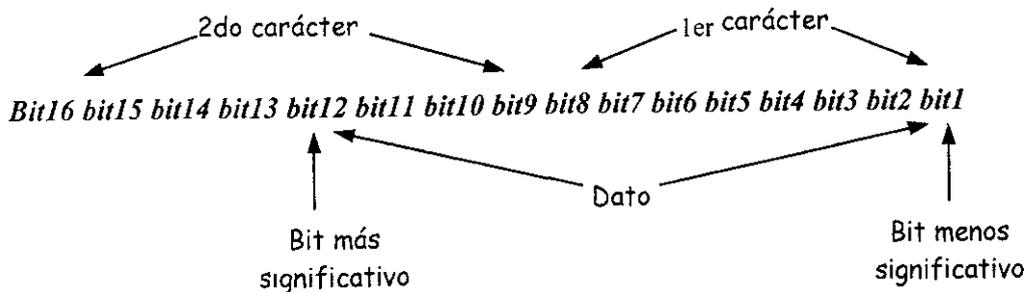


Figura 3.16. Suma de Caracteres

La cadena de 16 caracteres que se reciben en el puerto serial de la computadora es formada con un orden preestablecido. En primer lugar va el carácter que representa la parte menos significativa del primer canal y que es representado en la figura 3.17 como CB1 y representado como CA1, le sigue el carácter que representa la parte más significativa también del primer canal. A continuación están los caracteres bajo y alto del segundo canal y así sucesivamente hasta que por ultimo van los caracteres correspondientes al canal 8.

El carácter transmitido no es importante en si, sino el valor en código ASCII que representa. Por ejemplo, si la tarjeta de adquisición de datos transmite los caracteres A y 1 para la parte baja y alta respectivamente de algún canal, lo importante es el valor decimal que representan (65 y 49) y en consecuencia el binario (01000001 y 00110001).

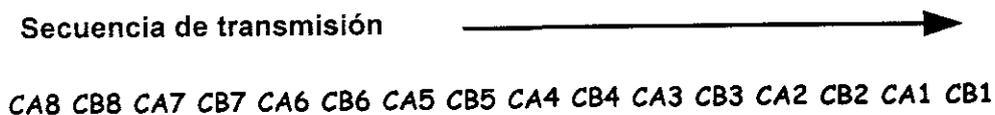


Figura 3.17. Orden de Transmisión

Seleccionando todo el valor binario de la parte baja y los primeros cuatro bits de la parte alta tenemos el valor binario 000101000001. Se debe recordar que la parte menos significativa de un dato siempre va del lado derecho del mismo. El dato 000101000001 en binario es el 321 en decimal. De esta forma tomando todos los 8 bits del primer carácter y los primeros 4 del segundo podemos obtener valores decimales de hasta 4095 como máximo los cuales no representan otra cosa que la magnitud de la variable que se esta monitoreando.

Después de ver como se codifican los datos de 12 bits que entrega el convertidor analógico/digital para ser transmitidos, ahora nos ocupamos del signo el cual es el bit 13 del dato del convertidor. Esto significa, como ya se explicó antes, que el convertidor puede manejar voltajes de entrada tanto positivos como negativos.

En relación al bit de signo si su valor es un cero lógico significa que el dato es positivo o cero pero si el valor de este bit es 1 lógico significa que el valor es negativo. Cuando los valores son negativos son

entregados por el convertidor en un formato tal que necesitan ser operados con un complemento a dos para obtener la magnitud real del mismo tal como se muestra a continuación:

0, 0000, 0000, 0010 - (2)

0, 0000, 0000, 0001 - (1)

0, 0000, 0000, 0000 - (0)

1, 1111, 1111, 1111 -(-1)

1, 1111, 1111, 1110 - (-2)

Como se puede apreciar en los datos representados arriba, los números positivos, incluido el cero, son representados de manera normal y el último bit que representa el signo tiene un valor de 0. Los números negativos necesitan ser complementados a dos y el bit de signo tiene un valor de 1.

Hasta aquí le hemos dado uso a los 8 bits del primer carácter y a los primeros cinco del segundo sobrando tres de este último. En párrafos anteriores se explicó que los dos siguientes bits (6 y 7) del segundo carácter son utilizados para indicar si es dato representado por los caracteres es un cero. A continuación se explica esta situación.

En las pruebas realizadas durante la etapa de experimentación del sistema se observó que en ciertos niveles de voltaje, los datos que eran desplegados en la computadora para los canales que tenían esos niveles no correspondían con los reales de manera marcada, e inclusive provocaban que todas las lecturas se comportaran de manera caótica.

Después de comprobar que la tarjeta de adquisición de datos estaba realizando las conversiones de manera adecuada y que estaba transmitiendo los caracteres correctamente, se fijo la atención en el programa de la computadora.

Los niveles de voltaje donde el programa tenía este comportamiento eran muy cercanos a cero volts por lo que se realizaron pruebas de transmisión de caracteres que representarían niveles alrededor de cero. Finalmente se pudo observar que el problema se presentaba cuando el carácter transmitido era el conocido como *NULL* y que se representa tan solo con ceros pues tiene el valor 0 en el código ASCII.

El problema es que el programa que corre en la computadora no reconoce este carácter y cuando lo lee hace de cuenta que no lo ha recibido, provocando con ello que tomara los caracteres que le "faltaban"

de la siguiente cadena de caracteres, correspondiente a otra conversión, enviada por la tarjeta de adquisición de datos.

El problema fue resuelto de una manera bastante sencilla. Cuando el convertidor analógico/digital arroja un dato que al ser representado con los caracteres alguno de ellos o los dos están compuestos tan solo de ceros, se le suma un 1 para evitar que se transmitan tan solo ceros.

Entonces hay que diferenciar cuando un carácter que representado un valor de 1 es realmente un 1 ó es un cero. La manera de hacer esa diferenciación es poniendo una "marca" dentro del mismo dato para que se distinga cuando un 1 lo es realmente o esta representando a un 0.

Como son dos los caracteres que pueden estar bajo esta situación se necesita de una "marca" para cada una de ellos. Los dos penúltimos bits del segundo carácter son utilizados para realizar esto. El bit 6 del segundo carácter es utilizado para indicar si el primer carácter es un cero o un 1, mientras que el bit 7 cumple la misma función para el segundo carácter.

Cuando la tarjeta de adquisición de datos transmite un carácter cuyo código ASCII es 1, en los bits 6 y 7 del segundo carácter se indica si este esta representando un 1 ó un cero de la siguiente manera, si el valor de esos bits es 1 quiere decir que se está representando un cero y cuando es cero se el dato es tomado como tal.

En la tabla 3.3 se resume el significado de los bits 6 y 7 del segundo carácter:

Código del carácter	Bit 6 ó bit 7	Valor que representa
1	0	1
1	1	0

Tabla 3.3. Significado de los bits 6 y 7 del 2º carácter

El programa que corre en la computadora debe poder distinguir esta situación para que los datos puedan ser interpretados adecuadamente, por lo que debe existir una parte del código que analice el segundo carácter de cada dato bit a bit. Esta misma situación se presenta con el bit 5 del segundo carácter que representa el signo ya que el dato correspondiente a cada canal siempre tiene signo para indicar la polaridad de la señal presente en la entrada del canal analógico. Aunque en la realidad no debía presentarse el caso en que un analizador entregue lecturas negativas como se explico anteriormente.

3.12 CONEXIÓN TARJETA DE ADQUISICIÓN DE DATOS-COMPUTADORA

El cable que une a la tarjeta de adquisición de datos con la computadora es el medio físico por donde la información viaja de una a otra y viceversa. Este cable debe estar construido de tal forma que la comunicación pueda establecerse de manera correcta.

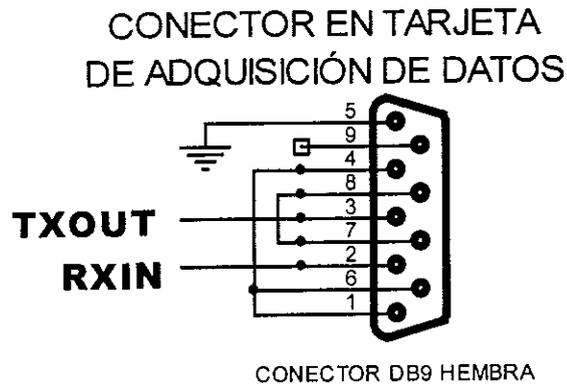


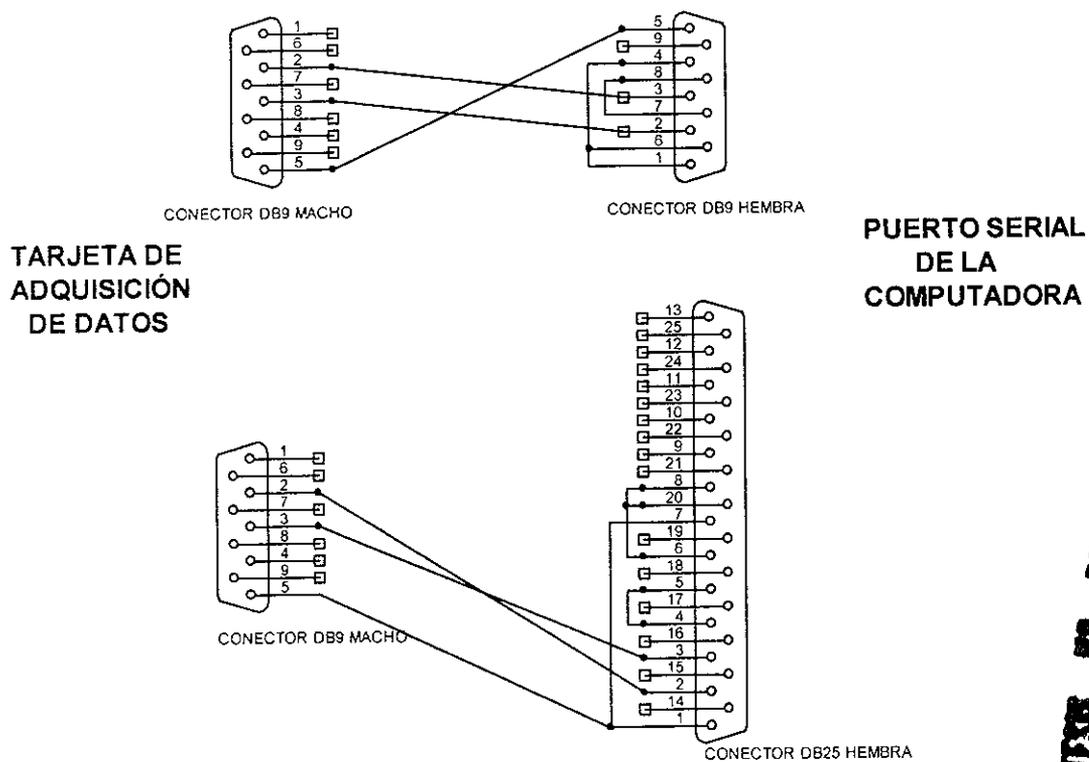
Figura 3.18. Conector de la tarjeta

En el capítulo 2 se discutió lo relacionado a la interface de comunicación entre la tarjeta y la computadora concluyéndose que lo más indicado es utilizar alguno de los puertos seriales de la computadora que soportan el estándar de comunicación RS232.

El 68HC11 que se utilizó tiene una tarjeta de desarrollo donde se encuentran además todos sus circuitos básicos como reloj y reset, memoria ROM y RAM y conectores para sus puertos.

Uno de los conectores es el de la interface de comunicación serial. Este conector tiene la configuración mostrada en la figura 3.18.

La configuración de este conector es la conocida como de módem nulo (*null modem*) en la cual las señales de control del estándar RS232 son puenteadas de tal forma que se presupone que no son necesarias aunque para efectos prácticos no pueden dejarse al aire sino tienen que interconectarse de esa forma para que la transmisión y recepción de datos se de correctamente.



ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

Figura 3.19. Configuración del cable de comunicación

Esta configuración (*null modem*), es muy utilizada para interconectar dispositivos que no necesitan manejar las señales de control. Debido a que el RS232 fue originalmente pensado para interconectar una computadora a un módem esas señales son necesarias, pero cuando se interconectan otros dispositivos puede utilizarse esta configuración.

El cable que interconecta a la computadora con la tarjeta de adquisición de datos debe tener la configuración mostrada en la figura 3.19. Del lado de la tarjeta el conector debe ser un DB9 macho que tan solo tenga los cables de transmisión y recepción de datos y el de tierra. Del lado de la computadora el conector es comúnmente un DB9 hembra pero puede necesitarse poner un conector DB25 hembra, ambas configuraciones se muestran en la figura 3.19.

Como el conector del puerto serial de la computadora no tiene puenteada ninguna señal de control, esto debe hacerse en el conector del cable tal como se muestra.

En el capítulo 2 se aborda de manera más profunda todo lo relacionado al estándar RS232 por lo que allí se puede consultar información relacionada con las señales que corresponden a cada una de las terminales de los conectores mostrados en las figuras 3.18 y 3.19.

3.13 CPU Y MEMORIA

Al inicio de este capítulo se presentaron las características del *68HC11*, dentro de las cuales una de las más importantes es el hecho de que puede manejar y soportar el estándar de comunicación RS232. Para manejar las comunicaciones de tipo serial, el microcontrolador se auxilia del circuito integrado *MAX232* que transforma los niveles lógicos del *68HC11* a los niveles lógicos del estándar RS232.

El *68HC11* que se utilizó tiene una tarjeta de desarrollo donde se encuentran además todos sus circuitos básicos como reloj y reset, memoria ROM y RAM y conectores para sus puertos.

El circuito integrado *27128* es una memoria *EPROM* DE 16k x 8 la cual contiene el programa en lenguaje ensamblador que el microcontrolador ejecuta. Dicha memoria es bastante grande para el tamaño del programa (350 Bytes aproximadamente) pero el direccionamiento y costo monetario para utilizar una memoria de capacidad más pequeña es prácticamente el mismo.

El circuito integrado *6264* es una memoria RAM de 8k x 8 y es utilizada para almacenar los datos que tienen un periodo de vida temporal.

En la figura 3.20 se muestra la configuración del microcontrolador junto con la memoria RAM y ROM que utiliza.

3.14 COMENTARIOS FINALES

Es evidente que los circuitos electrónicos mostrados en este capítulo, carecen de algunos detalles importantes como son la configuración de las fuentes de alimentación de todos los circuitos integrados, los circuitos de reloj y reset del *68HC11* etc.

Esta omisión es premeditada, debido a que como se explico antes, la tarjeta de adquisición de datos fue desarrollada principalmente por el C. Emmanuel Dámaso Mares, bajo la supervisión y asesoría del autor de la presente tesis, y servirá como su tema de tesis para obtener el título de Ingeniero en Comunicaciones y Electrónica en la Escuela Superior de Ingeniería Mecánica y Eléctrica del instituto Politécnico Nacional.

En dicha tesis se explicará detalladamente todo lo relacionado a la tarjeta de adquisición de datos. Lo anterior incluye al programa que debe ejecutar el microcontrolador *68HC11* y que solo es mencionado en este trabajo.

El programa del *68HC11* fue desarrollado totalmente por la persona antes mencionada y esta escrito en el lenguaje ensamblador de dicho dispositivo.

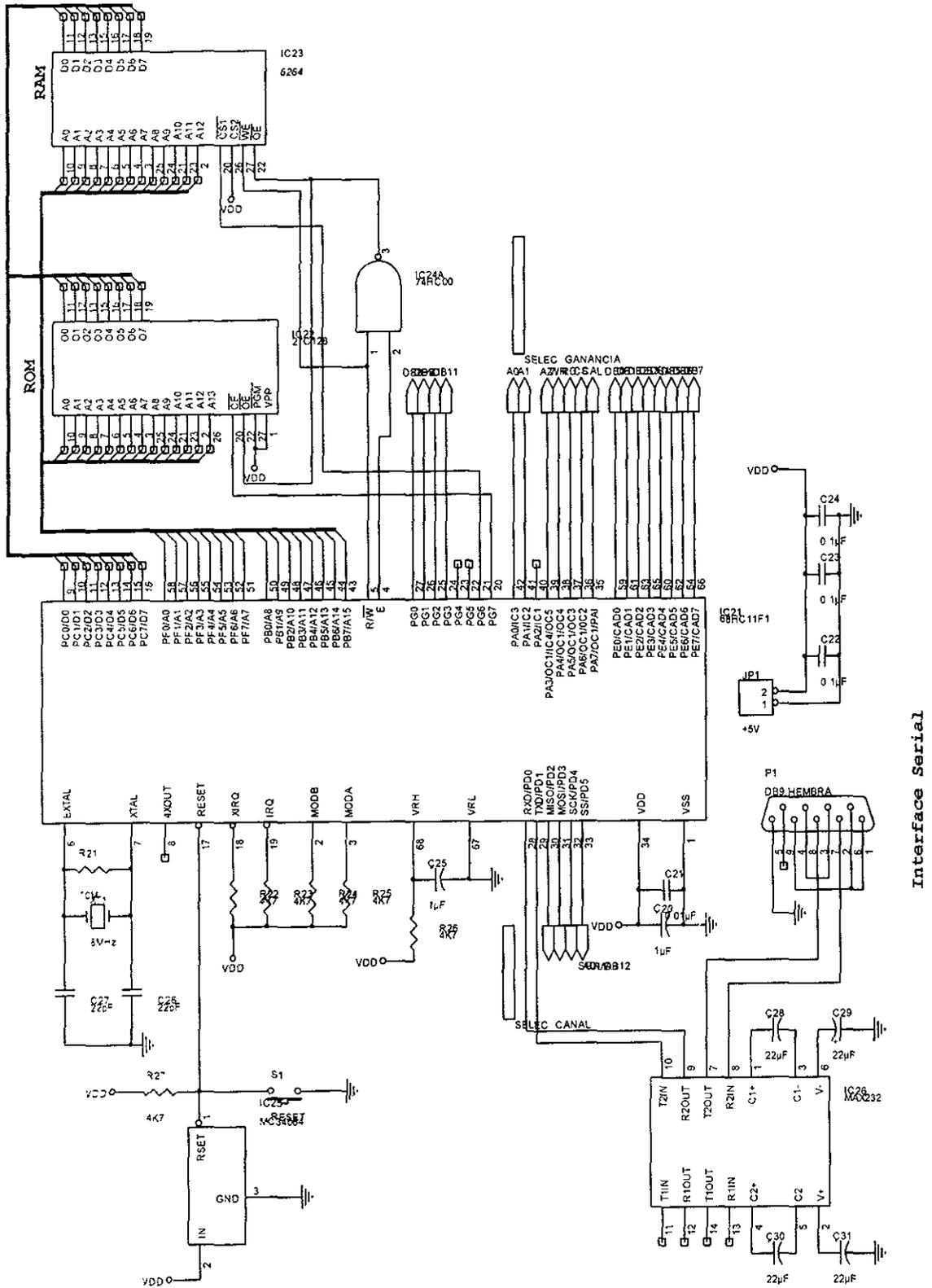


Figura 3.20. CPU y Memoria

INTRODUCCIÓN

Las computadoras personales se han convertido en la principal plataforma de los sistemas de adquisición de datos, contrario a lo que pudiera pensarse; existen varias causas de esta situación. A continuación se enumeran y describen las causas principales.

Tal vez la causa más importante, sea el hecho de que en la actualidad las computadoras personales son tan abundantes que es difícil encontrar un lugar de trabajo donde no se tenga por lo menos una de ellas.

El que las tarjetas madre cuenten con ranuras de expansión permite al usuario añadir tarjetas para adquisición de datos con solo insertarlas en alguna ranura libre. Aunque como ya se explico en un capitulo anterior, también se puede utilizar alguno de los puertos de la computadora personal para implementar el sistema, esto brinda la posibilidad de no estar restringidos a utilizar una computadora de escritorio sino hasta una de tipo portátil como las laptop o notebook.

El costo de las computadoras personales ha disminuido con el tiempo, además de ser más veloces y potentes. En la actualidad es relativamente barato adquirir una computadora de este tipo.

Es común que la gente de cualquier lugar de trabajo donde existan computadoras personales este familiarizada con el uso y manejo de ellas, esto significa que la inversión en aprendizaje es pequeña en los sistemas que la utilizan.

Las computadoras personales permiten implementar una buena cantidad de sistemas debido a su característica de arquitectura abierta y a causa de su gran popularidad se han desarrollado un buen número de estándares que permiten una compatibilidad entre componentes de diferentes fabricantes.

Años atrás las computadoras personales tenían bastantes limitaciones para ser consideradas como una opción en los sistemas de adquisición de datos. El rápido desarrollo de estas dio como resultado que en la actualidad sean bastante potentes y robustas y se conviertan en una buena opción. Hoy en día, pueden soportar tarjetas de adquisición de datos que manejan algunos cientos de canales de entrada/salida a altas velocidades con un gran volumen de información.

Otra buena razón para utilizar una computadora personal como base para desarrollar un sistema de adquisición de datos es lo relacionado con el software, ya que en la actualidad se han hecho una buena cantidad de aplicaciones que utilizan las distintas plataformas de este tipo de computadoras. El utilizar por ejemplo Windows le permite al usuario tener un panorama visual bastante bueno de su sistema, además de poder presentar y manipular su información con muy buena calidad.

Los campos de aplicación de las computadoras personales en sistemas de adquisición de datos y control son muchos y bastante variados. En la actualidad van desde la agricultura, la industria automotriz, otro tipo de áreas industriales, generación y distribución de energía, investigación y desarrollo de nuevas tecnologías y la que nos interesa, en el área ambiental, específicamente en sistemas de monitoreo y control de polución.

4.1 EL SOFTWARE DE DESARROLLO

Como es del conocimiento general, los ambientes de trabajo gráficos se han impuesto sobre otros, debido a que el usuario puede interactuar más fácilmente con ellos, como mejor ejemplo tenemos el ambiente Windows.

En el caso particular de este trabajo, el software que se desarrolló, debía tener esa característica, es decir, debe correr bajo Windows 95 o superiores.

Para el desarrollo del programa de aplicación de este trabajo, también existían dos posibilidades, una era el utilizar software de desarrollo especializado, muy de moda en la actualidad, la otra opción es el utilizar lenguajes de programación tradicionales como Basic o C en alguna de sus versiones que nos permitan desarrollar el sistema con las características requeridas [78].

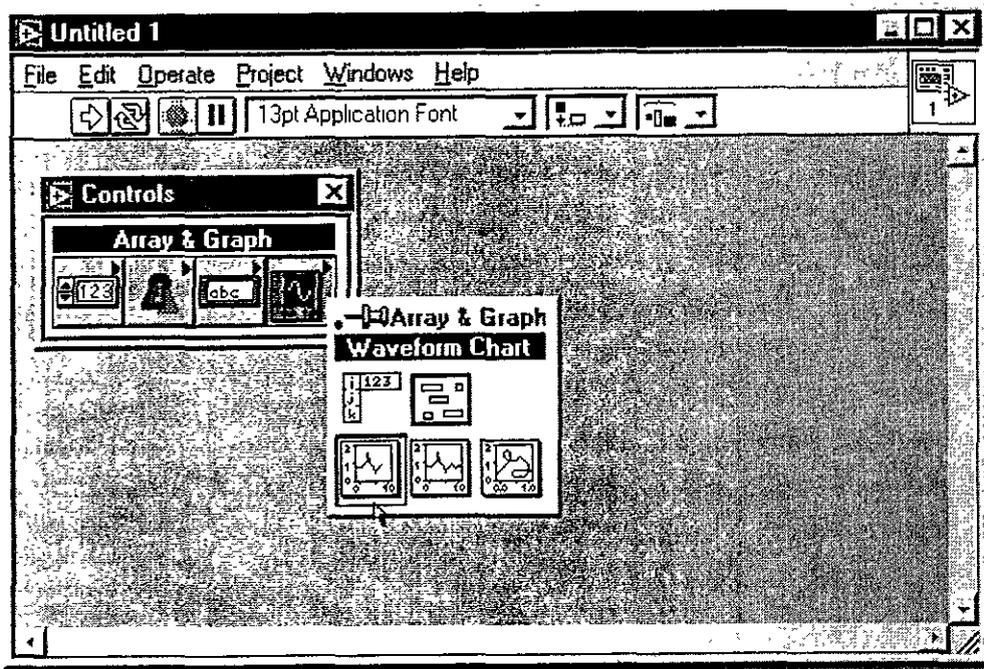


Figura 4.1. LabView

El software de desarrollo especializado en la actualidad está muy de moda y en general permite la implementación de casi cualquier sistema, ya que puede usarse para hacer desde un simple sistema de adquisición de datos hasta un sistema de control de un proceso muy complejo.

Dentro de esta categoría se encuentran los muy populares *LabView* de National Instruments [79] y *DaqView* de lotech [80]. El costo de este tipo de software de desarrollo es de aproximadamente 8000 USA dólares más aproximadamente 1200 USA dólares por un curso para aprender su manejo ya que requiere de un trabajo de programación muy especializada, además de que el usuario esta atado a la utilización del hardware de la misma compañía, el cual también tiene un costo bastante alto.

La característica básica de este tipo de software es que la programación es gráfica, ya que se utilizan iconos para representar distintas partes del sistema y líneas para relacionarlos. A este tipo de software se le conoce como *programación gráfica para instrumentos* aunque, como es obvio, esta característica no evita el tener que hacer un código de programa para que finalmente la aplicación desarrollada realice lo que se desea.

El más popular de este tipo de software es el *LabView* de National Instrument, mostrado en la figura 4 1, y que como ya se mencionó en párrafos anteriores, tiene un costo bastante elevado debido a que está diseñado para desarrollar sistemas de control de procesos bastante complejos, como los de plantas industriales, en los cuales siempre debe existir una etapa de adquisición de datos.

En lo personal, creo que adquirir un software como el *LabView* tan solo para desarrollar sistemas de adquisición de datos es un verdadero lujo ya que se haría una buena inversión monetaria tan solo para ocupar una mínima parte del software.

El utilizar un compilador de algún lenguaje de uso general para desarrollar la aplicación es la otra posibilidad. Como es obvio, comparando contra el software de propósito especial se tienen más dificultades para realizar la aplicación, es decir, se debe invertir más tiempo y esfuerzo.

Al realizar un recuento del software con que se contaba, se encontró que se poseía el compilador de Borland C++ en su versión 4.5 y el compilador de *Visual Basic* en su versión 5 profesional.

Finalmente se determino escoger alguno de los dos compiladores con que se contaba para desarrollar la aplicación. La decisión se inclino por el *Visual Basic*.

El decidirse por utilizar *Visual Basic* para hacer el desarrollo se debió principalmente a que la versión con que se contaba permite hacer aplicaciones que corren ya directamente en la plataforma Windows 95 mientras que el compilador de Borland es para aplicaciones que corren bajo DOS o Windows 3.11 y anteriores los cuales estan prácticamente descontinuados. Este software se adapta muy bien para resolver el problema ya que utiliza la filosofía de la programación orientada a objetos.

Otro aspecto importante es el hecho de que no se tendría que desembolsar dinero alguno para la adquisición del software de desarrollo.

4.2 VISUAL BASIC

El termino Visual esta relacionada con la forma como la interfaz gráfica de usuario se crea. El programador tan solo necesita arrastrar y colocar en ella los elementos que quiera ponerle. Esto significa que *Visual Basic* le evita al programador tener que escribir código para crear la apariencia y colocación de cada uno de los elementos que conforman a dicha interfaz, aunque existe la posibilidad de modificarlos por código como se explica en párrafos posteriores.

La palabra *Basic* es en referencia al lenguaje de programación de igual nombre (*Beginners All-Purpose Symbolic Instruction Code*) debido a que *Visual Basic* es el resultado de la evolución de ese lenguaje.

Visual Basic es en la actualidad uno de los programas de desarrollo más utilizado. Es un software que se caracteriza por su ambiente de trabajo visual.

Este software fue lanzado por la compañía Microsoft en 1991 [81] como *Visual Basic 1.0* y fue pensado para desarrollar aplicaciones especiales para la plataforma Windows de forma más rápida que cualquier otro lenguaje disponible en ese momento. A esta característica se le denominó Desarrollo Rápido de Aplicaciones (RAD por sus siglas en inglés) para la plataforma Windows.

Este software de desarrollo fue el primero que Microsoft puso en el mercado para competir con los tradicionales lenguajes de programación como C, C++ y PASCAL entre otros. Aunque las primeras versiones no tuvieron el éxito esperado, en la actualidad es una muy buena opción para desarrollar aplicaciones que corren bajo Windows.

El principal problema con las primeras versiones de *Visual Basic* era la velocidad con que corrían las aplicaciones desarrolladas con él; esto se debía a que los programas ejecutables no lo eran exactamente, sino que eran pseudo-código que un archivo llamado VBRUN###.dll se encarga de interpretar y pasar a comandos reales.

Esto significa que una misma aplicación desarrollada con *Visual Basic* y en algún otro lenguaje de programación como el C corría más rápido en el segundo caso.

La anterior situación se corrigió en las versiones más recientes logrando que *Visual Basic* tuviera un gran éxito de tal forma que incluso otros lenguajes de programación que tenían más tiempo en el mercado y que eran una buena opción para los programadores ya no lo son más.

A partir de la versión 5, *Visual Basic* ya cuenta con un modo de compilación nativo lo que significa que los programas ejecutables ya lo son realmente, esto se logra porque *Visual Basic* exporta el proyecto creado a un lenguaje intermedio que luego es compilado utilizando un compilador de C++.

El éxito de *Visual Basic* fue tal, que incluso las nuevas versiones de lenguajes más viejos, y que tradicionalmente dominaban el mercado, copiaron algunos de los conceptos introducidos por Microsoft en el *Visual Basic*. Tal es el caso de Delphi y VC++. Hasta ahora *Visual Basic* es la herramienta de desarrollo más vendida en todos los tiempos.

A pesar de todo el *Visual Basic* aun es considerado por algunos programadores como un lenguaje no profesional debido a que la palabra Basic normalmente hace referencia al lenguaje que muchos consideran como un lenguaje para principiantes.

4.3 LA FILOSOFIA DE PROGRAMACIÓN DE VISUAL BASIC

Hay tres conceptos importantes que se deben entender para trabajar con Windows, ventanas, eventos y mensajes.

En Windows hay un sin numero de ventanas, las cuales pueden definirse de la manera más simple, como una región rectangular con sus propios limites, esto significa que prácticamente todo son ventanas en Windows, desde las que conocemos tradicionalmente, hasta iconos, cuadros de texto, botones de opción, etc.

El sistema operativo Windows le asigna un numero identificador único a cada ventana con el fin de poderlas administrar. A este numero se le conoce como controlador de ventana o *hWnd*.

El sistema monitorea los eventos que puedan estar relacionados con las ventanas. Un evento es el resultado de alguna acción del usuario tal como hacer un clic con algún botón del *Mouse* o presionar alguna tecla. Los eventos por ejemplo, también pueden producirse por programación y como resultado de la acción de otras ventanas.

En toda ocasión que se produce un evento, el sistema operativo recibe un mensaje que procesa para transmitirlo a las ventanas relacionadas con ese mensaje; cada ventana realiza una acción apropiada de acuerdo a la programación relacionada con ese mensaje en particular.

En la programación de aplicaciones tradicional, la ejecución del programa comienza por la primera línea de código y sigue una ruta predefinida por el programador, tan solo llamando a los procedimientos cuando los necesita.

Una aplicación es un conjunto de instrucciones para que la computadora realice una o varias tareas determinadas. Todas las aplicaciones tienen una estructura que no es otra cosa más que la forma en que se organizan las instrucciones.

Las aplicaciones desarrolladas con *Visual Basic* tienen la característica peculiar de ser controladas por eventos, es decir, la ejecución del programa no tiene una ruta predeterminada ya que se ejecutan

distintas secciones del código en respuesta a los eventos que se generan. Finalmente la secuencia de esos eventos es la que determina el orden en que el código es ejecutado.

Lo anterior significa que el orden en que el código es ejecutado puede ser diferente cada vez que la aplicación es corrida, debido a que la secuencia de los eventos depende de la forma en que el usuario interactúe con la aplicación la cual puede variar en cada ocasión.

La diferencia básica entre los programas desarrollados de la manera tradicional (programación estructurada) y los desarrollados con la filosofía de programación controlada por eventos, es que mientras que en la primera el programa es quien tiene el control relegando al usuario a solo teclear datos cuando se requiere, en la segunda el usuario es quien tiene el control y decide que parte del código quiere ejecutar al hacer clic en botones de comando, seleccionar opciones en algún menú, etc.

Para crear una aplicación con *Visual Basic*, se deben seguir tres pasos básicos: crear la interfaz, establecer propiedades y escribir el código [82].

Para la creación de la interfaz se utilizan los llamados formularios que no son otra cosa que las ventanas principales con las que el usuario va interactuar, es decir, la parte visual. Los formularios deben ser los primeros en crearse al momento de desarrollar una aplicación para después colocar en ellos todos los objetos que cada formulario contendrá.

Los formularios son objetos que tienen propiedades para definir su apariencia, métodos para definir su comportamiento y eventos para definir la forma en la interactúan con el usuario [83].

Los formularios contienen objetos llamados controles. Los controles poseen también propiedades, métodos y eventos específicos que los hacen útiles para un fin determinado, mostrar texto, contener un gráfico etc.

Después de haber creado los formularios y haber colocado los objetos en cada uno de ellos el siguiente paso es establecer las propiedades de cada uno de los objetos que se han creado. *Visual Basic* cuenta con una ventana de propiedades desde la cual se pueden establecer estas de una manera muy fácil. Las propiedades también pueden ser modificadas en tiempo de ejecución por medio de código.

La sintaxis para cambiar las propiedades de un objeto por medio de código es:

Objeto.propiedad = valor

La sintaxis para obtener la propiedad de algún objeto por medio de código es:

variable=Objeto.propiedad

Las propiedades de los formularios y controles no son otra cosa que sus atributos, los métodos son sus acciones y los eventos son sus respuestas.

El último paso es escribir el código el cual sirve para definir el comportamiento de la aplicación. *Visual Basic* es un lenguaje de programación basado en objetos aunque con una naturaleza orientada a eventos. El código consta de instrucciones del lenguaje, constantes y declaraciones.

El código en las aplicaciones desarrolladas con *Visual Basic* está dividido en bloques menores llamados *procedimientos de eventos*, este código se ejecutará cuando se produzca el evento en cuyo procedimiento está contenido [82]. Un evento se produce cuando por ejemplo se hace clic sobre un botón, se pulsa una tecla, etc.

El código de una aplicación desarrollada con *Visual Basic* es organizado de manera jerárquica y está compuesto de módulos. Existen módulos de formulario, uno por cada formulario de que conste la aplicación, también hay módulos estándares los cuales son opcionales y se utilizan para poner el código que se va a compartir, por último hay módulos de clase también opcionales.

Los módulos de formulario son archivos con la extensión *.frm* y contienen procedimientos de evento que no son otra cosa que porciones de código que se ejecutará en respuesta a un evento específico. Como los formularios pueden contener controles, los módulos de formulario contienen un conjunto de procedimientos por cada control. También pueden contener procedimientos generales los cuales pueden ser llamados por cualquier evento del formulario.

Los módulos estándar son archivos con la extensión *.bas* y sirven para colocar en los procedimientos que son utilizados por diversos eventos en la aplicación con el fin de no multiplicarlos y evitar hacer más grandes los proyectos. Este tipo de módulos puede ser reutilizado en otras aplicaciones teniendo cuidado de no hacer referencia a un control en especial.

Los módulos de clase son archivos con la extensión *.cls* y se utilizan para crear objetos que pueden ser llamados desde cualquier procedimiento de la aplicación; a diferencia de los módulos estándar estos pueden contener además de código también datos.

Cualquiera de los tres tipos de módulos puede contener declaraciones y procedimientos. Las primeras sirven para declarar constantes, tipos de datos y variables. Los segundos como ya se mencionó, contienen el código que se va a compartir.

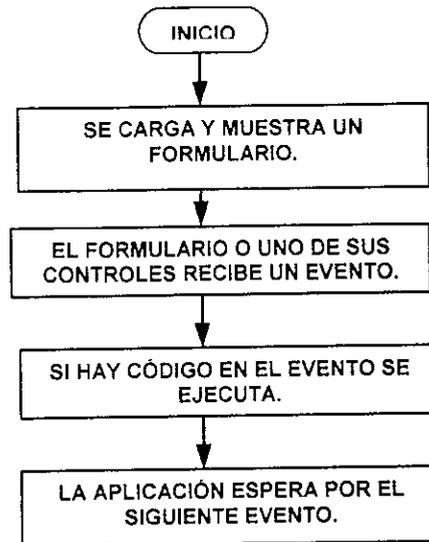


Figura 4.2. Aplicación controlada por eventos

En la figura 4.2 se muestra en un diagrama a bloques el funcionamiento de una aplicación controlada por eventos. Como ya se dijo anteriormente, en una aplicación de este tipo se ejecuta código *Basic* como respuesta a un evento. Cada formulario y control de *Visual Basic* tiene asociado un conjunto de eventos predefinidos, por lo que al producirse algún evento que tiene un código asociado, este es ejecutado.

Una aplicación desarrollada con *Visual Basic* es iniciada al correr el programa ejecutable de la misma, a partir de allí la ejecución del código dependerá de los eventos que sucedan y de su orden cronológico. Los eventos pueden ser causados por el usuario, por el sistema o de forma indirecta por el código contenido en algún evento.

El finalizar la aplicación será también el resultado de algún evento puesto que solo sucederá cuando ocurra un evento que contenga el código que finalice la aplicación.

Como ya se mencionó en párrafos anteriores, cuando se desarrolla una aplicación con *Visual Basic* se está trabajando con la filosofía de programación orientada a objetos. Se pueden utilizar los objetos proporcionados por *Visual Basic* como controles y formularios, también se puede tener acceso a objetos de otras aplicaciones además de que el programador puede crear sus propios objetos.

En *Visual Basic* los objetos se definen como una combinación de código y datos que se tratan como una unidad. Un objeto puede ser un control, un formulario o hasta una aplicación completa.

Otra de las características más importantes de *Visual Basic*, son los llamados controles *ActiveX*. Este tipo de controles son extensiones de su caja de herramientas y se utilizan de la misma manera que los controles que vienen con cada versión de *Visual Basic* de manera estándar. La tecnología de los controles *ActiveX* fue desarrollada primeramente por Microsoft.

Un control *ActiveX* es un elemento programable que puede agregarse a la aplicación que se esta desarrollando. También son conocidos como controles *OLE* o controles de usuario *OLE* [84].

Los controles *ActiveX* aumentan sensiblemente la funcionalidad y flexibilidad de la aplicación que se esta desarrollando, ya que son creados con el objetivo de mejorar las características de los ya existentes o realizar funciones que ningún otro hace [85].

Los archivos con la extensión *.OCX* corresponden a este tipo de controles para la versión 5 en adelante, para versiones anteriores la extensión de estos archivos era *.VBX*. Algunos de estos controles son incluidos en las versiones empresariales y profesionales de manera estándar, pero otros son desarrollados por compañías especializadas y pueden ser adquiridos por cierta cantidad de dinero e inclusive algunos programadores los comparten de manera gratuita.

Un ejemplo relacionado con este trabajo y los controles *ActiveX*, es el hecho de que la versión de *Visual Basic* con que se cuenta, trae un control para graficar datos y se decidió utilizarlo para mostrar el comportamiento en tiempo real de cada una de las variables que se estan monitoreando.

Cuando solo se estan monitoreando pocas variables, existe un mínimo parpadeo en la pantalla el cual es un poco molesto para el usuario, pero cuando se monitorean todos los 8 canales el parpadeo es de tal magnitud que puede representar problemas visuales para el usuario.

Para intentar resolver el problema se busco algún control *ActiveX* que realizara dicha función de manera más eficiente. En el mercado existen varios de estos controles pero los más eficientes tienen un costo superior a los 150 dólares el cual es un costo bastante alto.

La principal ventaja de estos componentes es que pueden ser rehusados en otras aplicaciones e incluso pueden ser creados utilizando lenguajes o herramientas de desarrollo tales como el mismo *Visual Basic*, *C++*, *Power Builder*, etc. Hasta ahora solo corren en Windows 95, Windows 98, Windows NT y Macintosh.

Al entorno de programación de *Visual Basic* se le denomina frecuentemente entorno integrado de desarrollo o IDE por sus siglas en Inglés. En la figura 4.3 se muestra el IDE y sus principales componentes.

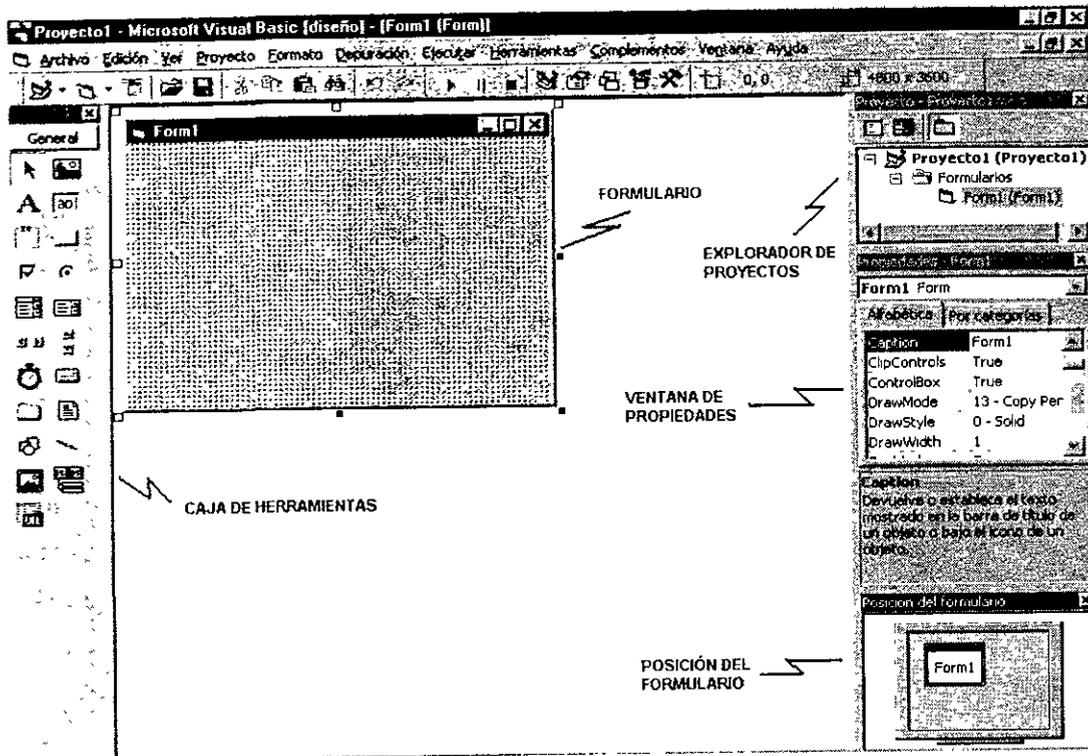


Figura 4.3. El entorno de desarrollo VB

Tal vez los componentes más importantes del IDE de *Visual Basic* son los formularios, la caja de herramientas y la ventana de propiedades. Los formularios son, como ya se mencionó antes, las ventanas donde el usuario va a interactuar con la aplicación. La caja de herramientas contiene los elementos, mejor conocidos como controles, que el programador puede agregar al formulario, tales como botones de comando, cuadros de texto, cuadros de dialogo, etiquetas, cuadros de imagen, etc. Con la ventana de propiedades se pueden adecuar muchas de las características tanto de los formularios como de los controles que contienen durante el tiempo de diseño.

4.4 ESTRUCTURA DE LA APLICACIÓN

A continuación se aborda la estructura básica del software de adquisición de datos desarrollado en este trabajo.

Como es obvio, antes de proceder a colocar los controles necesarios en cada uno de los formularios y escribir el código de programa relacionado con cada uno de ellos, se debe diseñar la estructura de la aplicación, es decir, el cómo va a interactuar cada uno de los bloques principales que la componen.

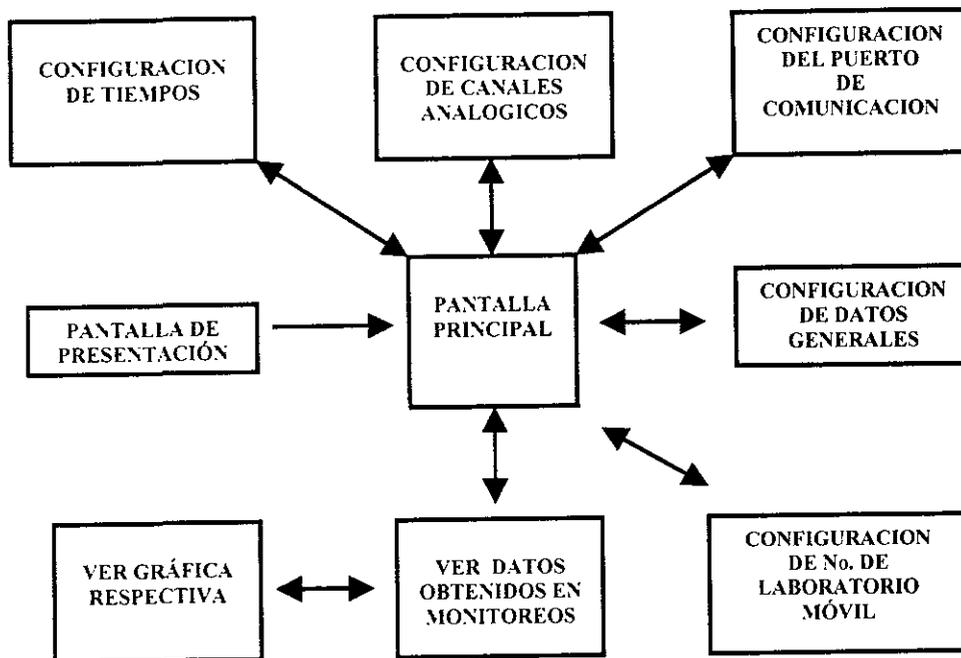


Figura 4.4. Estructura de la aplicación

Para el diseño de la aplicación se tomó en cuenta la opinión de los futuros usuarios acerca de las características que ella debe tener. Dentro de las principales se encuentran:

- Capacidad de configuración del sistema, esto incluye a cada uno de los canales de entrada analógicos, tiempos de muestreo y adquisición, características del puerto de comunicación, el número económico del laboratorio móvil y los datos generales del monitoreo.
- Capacidad de monitorear las señales de cada uno de los analizadores sin adquirir datos, esto es muy útil cuando se necesita calibrar y ajustar el sistema de monitoreo.
- Capacidad de adquisición de datos de 8 diferentes canales analógicos de entrada de manera simultanea.
- Capacidad de mostrar los datos obtenidos en la etapa de adquisición tanto en forma tabulada como en gráficas de tendencias en el tiempo.
- Capacidad de revisar archivos relacionados con monitoreos hechos anteriormente.

En la figura 4.4 se muestra el sistema propuesto en bloques que cubren todas las características enumeradas arriba y la forma de viajar a través de ellas.

En la realidad, cada bloque representa una ventana con la cual el usuario deberá tener contacto dependiendo de las acciones que quiera realizar. Cada ventana presenta una serie de opciones especiales para el usuario, relacionadas con la función principal de ellas.

La pantalla principal siempre será el centro de todo el sistema. Cuando la aplicación es arrancada muestra una pantalla de presentación del software por unos segundos para después entrar a la pantalla principal, a partir de la cual se podrá ir hacia las otras, pero siempre pasando por ella.

En las figuras 4.5 y 4.6 se muestra el diagrama de flujo general de la aplicación, en el se pueden apreciar todas las funciones enumeradas anteriormente y que se realizan a través de la ventana principal.

En el diagrama se puede ver que la ventana principal corresponde al bloque donde el usuario decide qué acción va a realizar, la ejecuta y finalmente vuelve a retornar a ella.

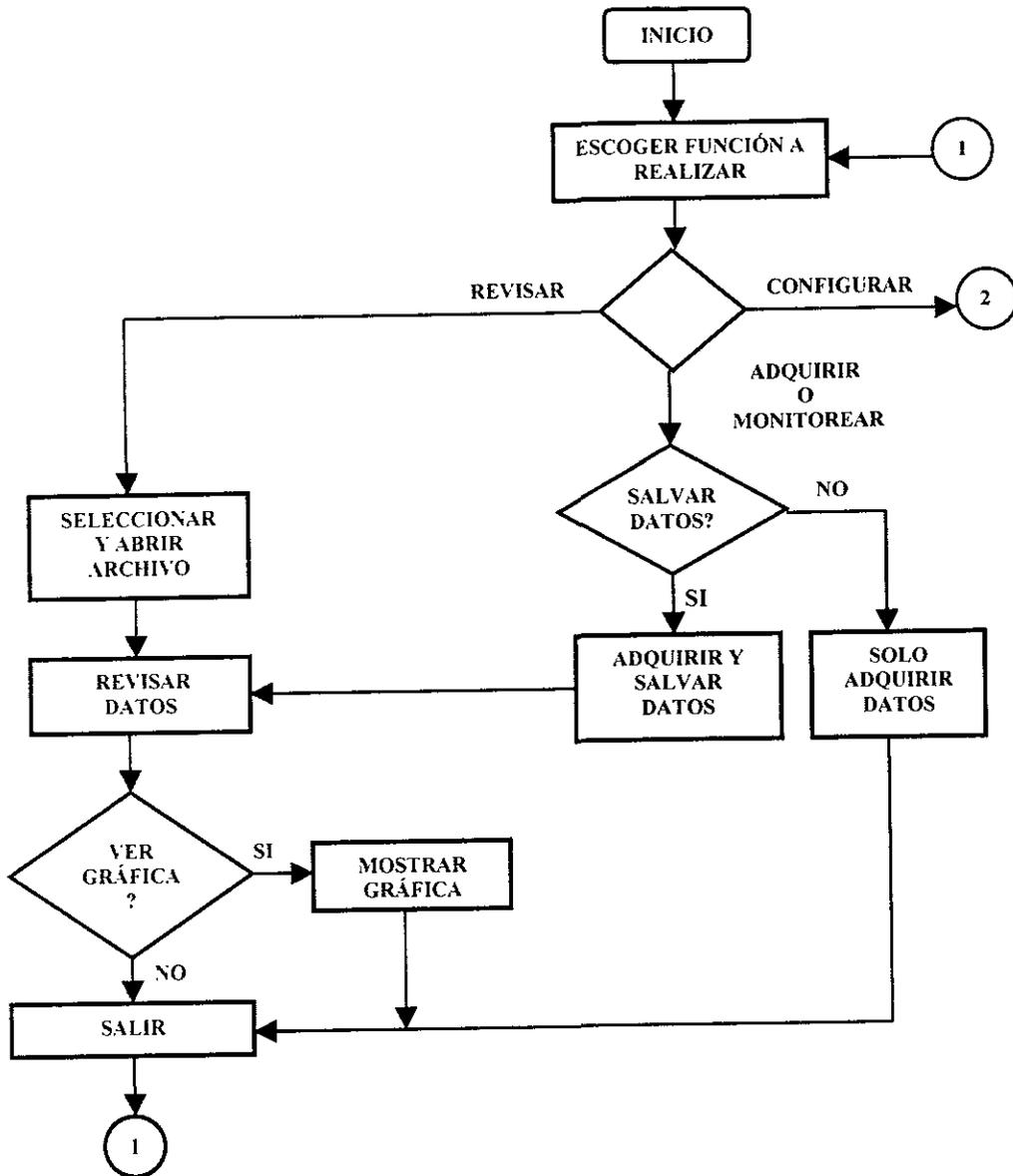


Figura 4.5. Diagrama de flujo general

Como se puede apreciar en el diagrama de flujo, el usuario decide la parte de la aplicación que quiere ejecutar y cuando finaliza dicha ejecución, el programa no termina sino que regresa al punto en que el usuario vuelve a tener la opción de escoger. También es obvio que una de esas opciones es el poder cerrar o finalizar la aplicación. Lo anterior no es otra cosa que la ejecución del programa controlada por eventos y que ya fue explicada anteriormente.

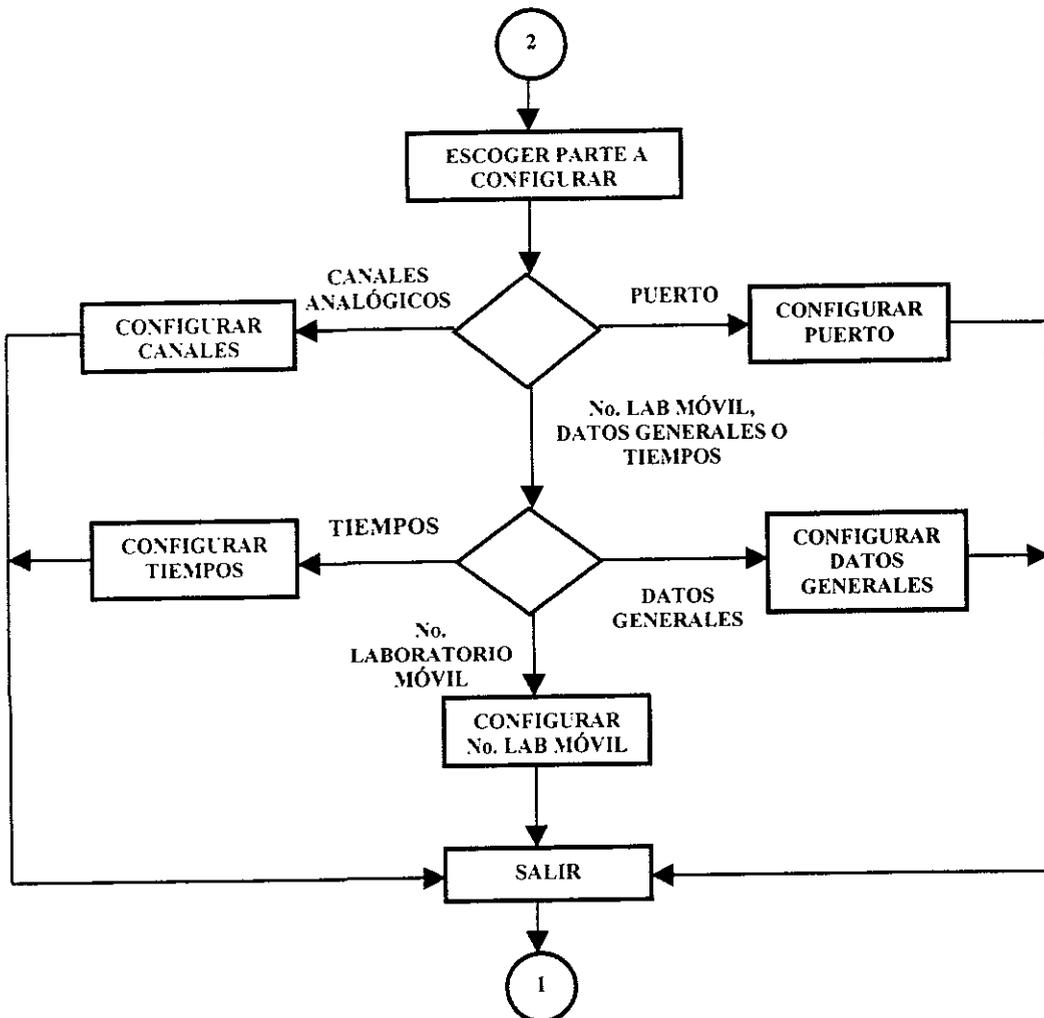


Figura 4.6. Diagrama de flujo general (continuación).

En el diagrama de flujo solo se muestran las funciones principales de la aplicación pero existen otras funciones secundarias que no son mostradas aunque su existencia pareciera obvia.

Una de esas funciones secundarias es la forma cómo la aplicación es cerrada o finalizada, lo cual se realiza por medio de código insertado en el menú correspondiente como se detallará posteriormente.

En la ventana de configuración de los canales analógicos, el usuario podrá escoger las opciones con las que cada uno de ellos trabajará, estas incluyen: variable a medir, escala de trabajo, rango analógico de entrada e inclusive desactivarlo si es que no es utilizado.

La ventana de configuración de tiempos se utiliza para programar dos tiempos necesarios para este tipo de monitoreo, uno es el tiempo de muestreo y otro es el intervalo de tiempo en que se obtendrá la media aritmética de los datos obtenidos para guardarla.

La comunicación a través del puerto serie también necesita ser adecuada a las características de la tarjeta de adquisición de datos para que la información fluya adecuadamente, esta operación se realiza por medio de la ventana de configuración de comunicaciones, en ella se podrán escoger desde la velocidad de transferencia, pasando por la paridad, hasta el número de bits por dato.

Como existe la posibilidad de que el sistema pueda ser instalado en cualquier laboratorio móvil, se creó una ventana para que el usuario pueda introducir el número económico de la unidad móvil de que se trate.

Cuando se realiza una adquisición de datos es fundamental poder guardar información relacionada con el lugar, la fecha y quien realiza la adquisición. Para poder hacerlo esta la ventana de configuración de datos generales.

Hay otra ventana para revisar los datos obtenidos en algún monitoreo, es decir, para que el operador revise la información que se obtuvo en el monitoreo que se está finalizando o en otros realizados con anterioridad. A partir de esta ventana se puede abrir una que presenta una gráfica de líneas de tendencias contra el tiempo de los datos del monitoreo que se está revisando.

Algo muy importante de señalar es el hecho de que la aplicación fue estructurada de acuerdo a las necesidades de los usuarios y con el visto bueno de los mismos en cuanto a las características que debía tener.

4.5 ARRANCANDO LA APLICACIÓN

El arrancar la aplicación es de lo más sencillo debido a que los programas desarrollados en Visual Basic generan un programa ejecutable que se puede correr de la manera tradicional en Windows, es decir, colocándose primero en el directorio donde se encuentra dicho programa para después hacer un doble clic sobre el icono que lo representa, de esta manera la aplicación inicia su ejecución. Es obvio comentar que es necesario que se encuentren en el mismo directorio todos los archivos que la aplicación necesita para correr correctamente, de otra forma no podrá hacerlo.

Visual Basic contiene un asistente para poder hacer el programa de instalación de la aplicación, con el cual se obtienen los discos que permitirán instalarla en cualquier computadora personal que corra bajo

el ambiente Windows y cuyo sistema operativo sea Windows 95, Windows 98 e incluso Windows NT. De esta forma, una vez que se termina de desarrollar y probar la aplicación se puede recurrir a este asistente para obtener dichos discos.

Después de que la aplicación es instalada con los discos de instalación, esta puede ser arrancada por medio del menú Programas del botón Inicio de Windows como cualquier otra aplicación.

Cuando la aplicación es arrancada de la forma que sea, y después de que la pantalla de presentación es mostrada por unos segundos, aparece la pantalla principal que, como ya se mencionó antes, es el eje sobre el cual gira toda la aplicación. Es necesario que cuando la aplicación sea inicializada se recupere alguna información relacionada con la configuración del sistema en general, esto porque sería bastante tedioso que el usuario tuviese que configurarlo cada vez que la aplicación es corrida.

Uno de los primeros eventos que se desencadenan al crear un formulario y mostrarlo en pantalla es el evento llamado *Form_Load*. Este evento tiene lugar al momento que el formulario es cargado en memoria por que normalmente solo ocurre una vez, aunque por medio de código puede descargarse y cargarse el formulario todas las veces que sea necesario.

El evento *Form_Load* es comúnmente utilizado para incluir toda la información que se requiere aparezca inicialmente en el formulario, en el caso nuestro es importante también recuperar la información de configuración de todo el sistema.

El formulario correspondiente a la ventana principal en nuestro caso es llamado *Form1* y en primer caso son declaradas las variables a nivel formulario en la sección de declaraciones.

```
Dim capturas() As Double 'Dato de los 8 canales
Dim comenzar As Boolean
Dim contador As Double
Dim conta As Integer
Dim adquirir As Boolean 'Verdadero=Adquirir, Falso=Monitorear
```

La variable *capturas()* es un arreglo que servirá para almacenar la información enviada por la tarjeta de adquisición de datos a través del puerto serie para su posterior tratamiento.

La variable *comenzar* es utilizada como un indicador para decidir si una adquisición o un monitoreo deben ser iniciados o terminados.

La variable *adquirir* sirve para que el programa sepa que los datos capturados deben ser mostrados y salvados o tan solo mostrados.

Todas estas variables son utilizadas por más de un evento, procedimiento o función del formulario, por lo que deben ser declaradas como globales.

Al arrancar el programa se utiliza el evento *Form_Load* para configurar y recuperar información con el siguiente código:

```
Private Sub Form_Load()
    'Se establece el nombre de la aplicación
    App.Title = "adqdatos"
    incom.FillColor = &HFF&
    comuntex.Caption = "No Comunicación"
    'Se recuperan tiempos
    recup_tiempos
    'Se fijan tiempos de promedio y muestreo
    muestra = configtiempos.tmuestreo.Text 'segundos
    media = configtiempos.tpromedio.Text 'minutos
    media = media * 60 'segundos
    'Se obtiene configuración de canales
    recup_config_canales
    'se inicializa puerto llamando a la siguiente fn'.
    recup_prop_puerto
    'Se obtiene el No. de laboratorio
    recup_num_lab
    'Se programa bandera de inicio
    comenzar = True
    'Se ponen a Cero las lecturas
    initcanales 8
End Sub
```

La primera instrucción de este código, `App.Title = "adqdatos"`, es una instrucción de *Visual Basic* utilizada para recuperar información de los llamados registros de Windows relacionada con esta aplicación a la cual se le llamó *ADQDATOS*. Cuando se guarda información en esos registros se necesita indicar el nombre de la aplicación para identificarla. La información es almacenada en diferentes carpetas

según el usuario desee estructurarla. Más adelante se explicará más a detalle todo lo relacionado a estos registros.

Es importante mencionar que todas las funciones incluidas en el código anterior y en toda la aplicación están incluidas en módulos estándar etiquetados de acuerdo a su función como se explicará más adelante.

4.6 LA VENTANA PRINCIPAL

Después de que la aplicación es arrancada aparece la ventana principal, toda la aplicación gira alrededor de esta ventana, lo que significa que para poder tener acceso a otras partes de la aplicación se debe pasar necesariamente por ella.

Como se ve en la figura anterior, todas las ventanas a excepción de la encargada de mostrar la gráfica, tienen una relación bidireccional con la principal, es decir, solo se puede entrar a ellas desde la principal y al salir de ellas se llega nuevamente a esta.

Siempre que se arranca la aplicación, aparece la ventana principal para que a partir de ella se puedan realizar básicamente cuatro operaciones:

1. Configurar alguna o todas las instancias que lo necesiten.
2. Monitorear los canales analógicos con el fin de revisar y posiblemente calibrar los analizadores sin la necesidad de adquirir datos.
3. Realizar un monitoreo ya formal con el fin de adquirir y guardar la información para su posterior análisis.
4. Revisar monitoreos realizados con anterioridad sin tener que capturar datos.

La ventana principal le permite al usuario realizar todas estas acciones por medio de botones de comando y menús, los cuales requieren de poco tiempo para familiarizarse con cada uno de ellos y su correspondiente funcionalidad.

La ventana principal está compuesta de botones de órdenes y menús que le permiten al usuario realizar acciones específicas ejecutando el código insertado en cada evento utilizado.

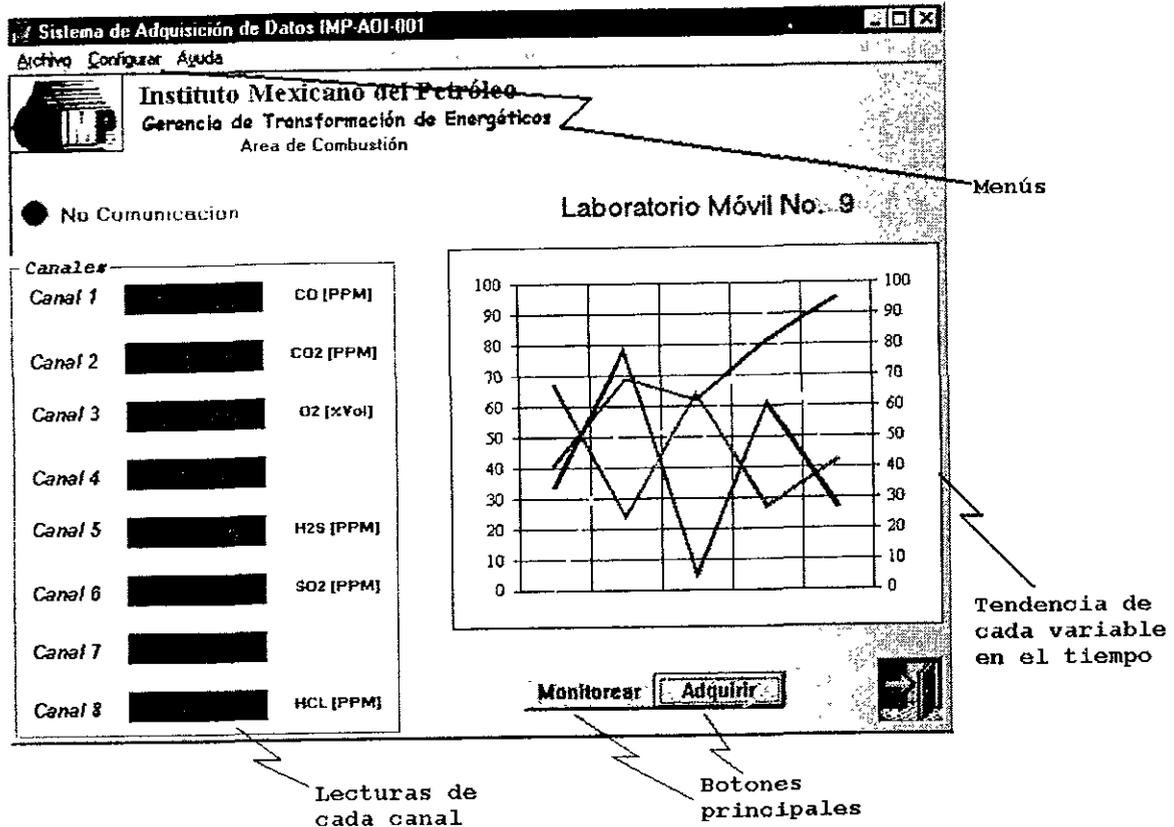


Figura 4.7. Ventana Principal

Las partes más importantes de la ventana principal, como se puede apreciar en la figura 4.7, son los cuadros de texto donde los valores de la variable asignada a cada canal son presentados, los botones para monitorear y adquirir datos, la gráfica donde se muestra la tendencia en el tiempo de cada variable y por último los menús utilizados por el usuario.

En la figura 4.7 se puede observar que los cuadros de texto donde se presentan los valores correspondientes a cada variable asignada a un canal específico, pueden desplegar dos posibles valores; cuando algún canal se encuentra habilitado se muestra el valor más actual de la variable asignada a ese canal junto con el nombre de la variable y las unidades en que esta siendo medida. Cuando algún canal se encuentra deshabilitado, en el cuadro de texto se muestra el texto "Libre" para indicar que dicho canal no está siendo utilizado y puede ser ocupado por el usuario para monitorear alguna variable que él desee.

Es importante señalar que los valores desplegados son actualizados en un periodo de tiempo dependiente directamente del periodo de muestreo determinado por el usuario en la ventana de configuración de tiempos, como se verá más adelante.

La ventana principal solo cuenta con dos botones de ordenes, uno es utilizado para iniciar o terminar un monitoreo, es decir, para que el usuario pueda ver los valores de las variables sin que el sistema guarde la información ya que solo es necesario que la muestre en pantalla. Como ya se dijo antes, este modo es muy útil para cuando se necesita ajustar y/o calibrar algún parámetro en los diferentes equipos involucrados en los monitoreos realizados en el laboratorio móvil.

El otro botón se usa para iniciar/terminar una adquisición de datos, la cual es la principal función de la aplicación. La adquisición consiste en solicitarle periódicamente a la tarjeta de adquisición de datos los valores correspondientes a la señal analógica presente en cada uno de los ocho canales; después de recibir dichos valores, estos son operados por factores tales como la escala total de cada canal y el periodo que debe transcurrir entre cada ocasión que debe obtenerse el promedio de las señales recibidas; al obtener los resultados estos son desplegados solo para los canales que se encuentran activos.

Siempre que una adquisición es iniciada se abre un archivo para guardar la información útil para el usuario tal como los valores correspondientes a cada variable y la hora del día a que corresponden. La información es salvada como texto pero es transparente para el usuario. Al terminar dicha adquisición se cierra ese archivo.

Otro archivo que se crea durante una adquisición es el correspondiente a los datos de identificación del monitoreo realizado, esos datos corresponden al nombre del usuario que realiza la adquisición, el lugar donde es realizado el monitoreo y la planta a que pertenece la chimenea que esta siendo monitoreada.

La gráfica donde se muestra el comportamiento en el tiempo de las variables que estan monitoreandose es otra de las partes importantes de la ventana principal, en ella se muestra una línea para cada variable y lo único que distingue a unas de otras es su color. Esta gráfica le da una idea global del comportamiento de la fuente de emisiones a l usuario, es decir, se puede apreciar como varían los valores de algunas variables al variar los correspondientes a otras.

El ultimo componente importante de la ventana principal son los menús, estos sirven para que el usuario pueda acceder a otras partes de la aplicación tales como la de configuración y revisión de archivos correspondientes a monitoreos realizados con anterioridad. También los menús sirven para iniciar/terminar un monitoreo o adquisición de datos sin utilizar los botones de ordenes. Dichos menús

están estructurados de tal forma que representen una idea lógica para el usuario al momento de realizarlos.

4.7 MONITOREAR

Una de las primeras cosas que tiene que hacer el usuario antes de realizar una de adquisición de datos en forma, es ajustar todos los parámetros y condiciones del laboratorio, necesarios para que la información recopilada pueda ser considerada válida.

Tal vez unas de las condiciones principales son que los analizadores que van a ser utilizados deben estar operando correctamente, que deben ser calibrados, que debe esperarse a que su tiempo de calentamiento se cumpla adecuadamente para posteriormente revisar que la información está llegando de manera adecuada a la tarjeta de adquisición de datos.

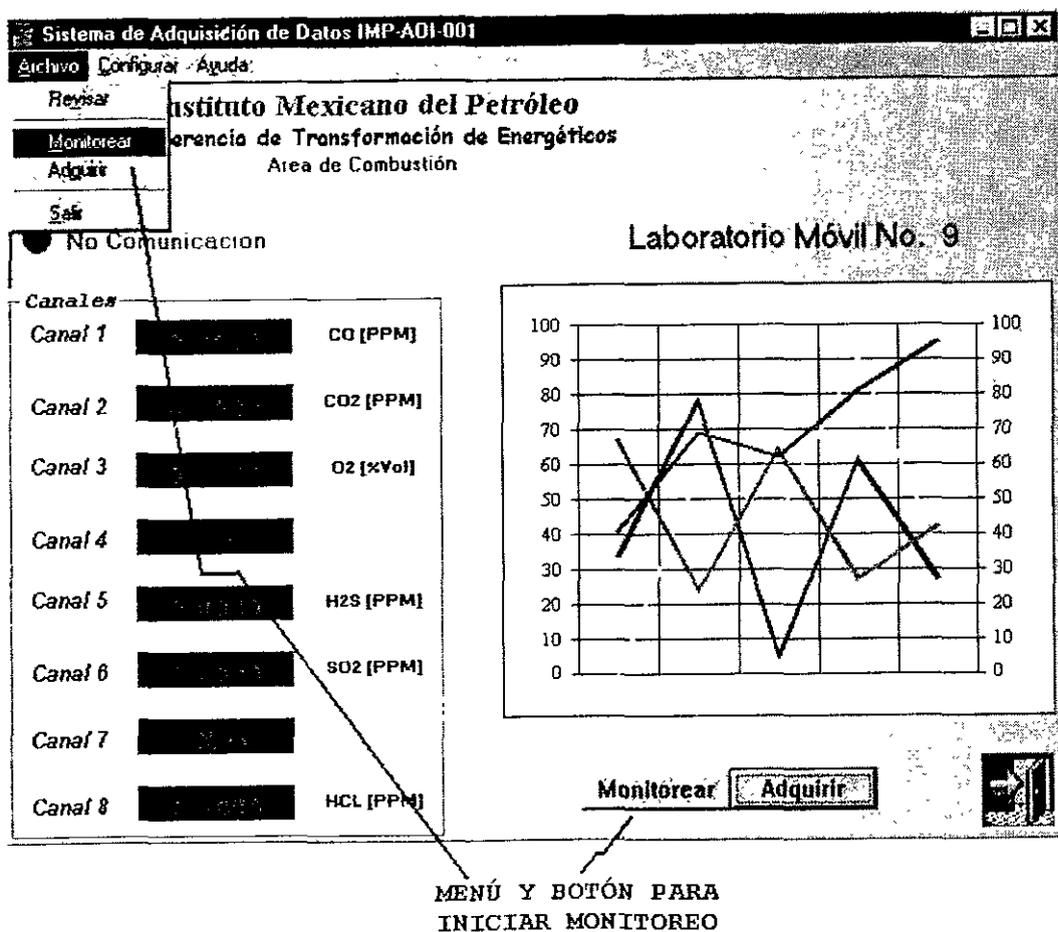


Figura 4.8. Inicio de Monitoreo

Durante esta etapa el sistema de adquisición de datos aquí presentado, debe capturar la información proveniente de los analizadores sin salvarla en alguna forma, ya que el usuario solo necesita poder comprobar visualmente que los datos estén siendo capturados de forma correcta y que correspondan a los desplegados por los analizadores.

A esta etapa se decidió llamarla *Monitorear* y a continuación se describe como se inicia y finaliza.

Para iniciar y finalizar una adquisición de datos sin que estos deban ser salvados se utiliza el botón etiquetado "Monitorear" y el menú del mismo nombre, los cuales son mostrados en la figura 4.8.

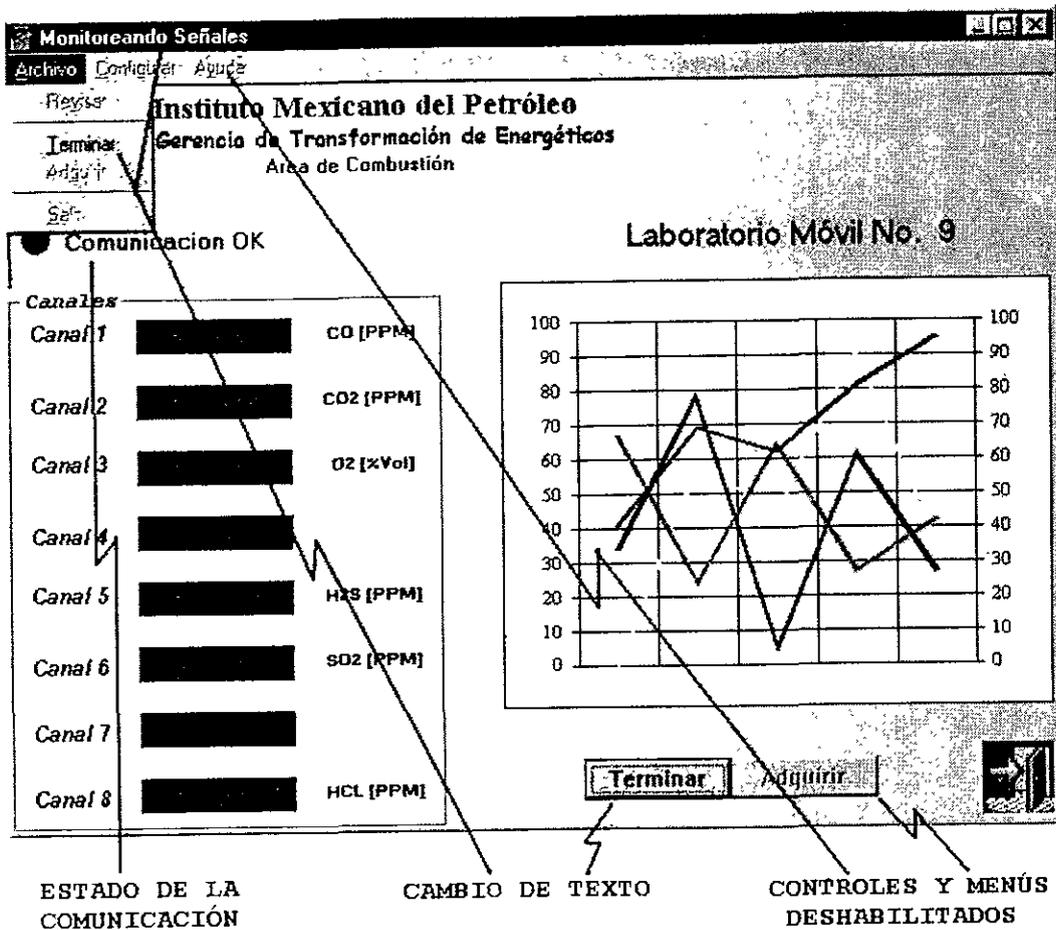


Figura 4.9. Estado de la ventana durante un monitoreo

Esta función es muy útil, como ya se mencionó en párrafos anteriores, cuando el operador necesita revisar que las señales provenientes de los analizadores están arribando correctamente al sistema y que la respuesta de los mismos es adecuada. También se utiliza esta función cuando el usuario calibra cada analizador. En todas estas situaciones el usuario necesita solo visualizar los valores de cada una de las variables sin tener que salvar dicha información.

El código relacionado con este botón, y en general con toda la aplicación, está estructurado de tal forma que el usuario solo tenga que seguir pasos lógicos al realizar cada acción que él necesite. Por ejemplo, el texto que aparece en el botón que sirve para iniciar o terminar un monitoreo cambia según el estado actual de la aplicación por lo que siempre será "*monitorear*" cuando no se está realizando ninguna operación de este tipo lo que le indica al usuario que si oprime este botón iniciará un monitoreo. Al momento de iniciar el monitoreo el texto cambia a "*Terminar*" indicando que al hacer clic en él se detendrá el monitoreo.

Lo anterior se logra cambiando en tiempo de ejecución una de las propiedades de los botones de ordenes llamada *caption* como se puede ver a continuación:

```
boton.caption="terminar"
```

para cuando no está en curso ningún monitoreo y se hace clic en el botón, y

```
boton.caption="monitorear"
```

para cuando está corriendo un monitoreo y se hace clic sobre el botón para detenerlo.

El estado de todos los menús y controles de la ventana principal, y de toda la aplicación, se activa o desactiva también según la operación que se esté ejecutando, como se puede observar en la figura 4.9. Esto se logra manipulando la propiedad *enabled* de todos los formularios, controles y menús de la aplicación.

Otra observación importante es el hecho de que en la ventana existe un indicador del estado de la comunicación entre la computadora y la tarjeta de adquisición de datos, para mostrarle al usuario el estado de esta.

Al momento de iniciar el monitoreo, la computadora envía el carácter "**P**" con el propósito de que la tarjeta de adquisición de datos lo reconozca y responda enviando el carácter "**R**". La aplicación lee el carácter de respuesta y si este es diferente al esperado o no se recibe ninguno se envía un mensaje de

error como el mostrado en la figura 4.10 y el monitoreo es abortado con el fin de que el usuario corrija la posible causa del problema.

Si el carácter esperado es recibido a tiempo, el programa procede a iniciar el monitoreo y el indicador cambia de color y muestra la leyenda "Comunicación OK".

Inmediatamente después de recibir el carácter de respuesta adecuado, la computadora envía una cadena de 8 caracteres que indican el rango de trabajo para cada canal analógico con el fin de que la tarjeta de adquisición de datos los guarde en memoria y los utilice durante cada conversión que realice.

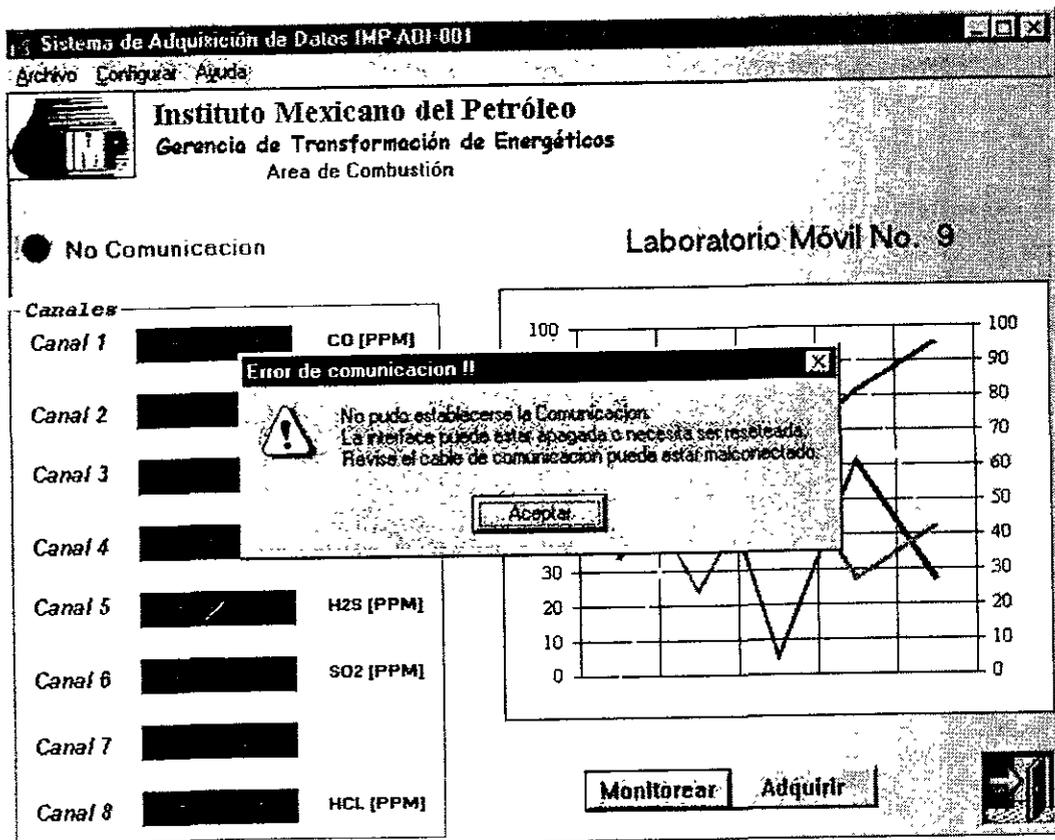


Figura 4.10. Falla de comunicación

Los caracteres que puede contener dicha cadena, de acuerdo al rango de trabajo de cada canal analógico, son los siguientes:

- 0 ó el carácter ASCII 48 para un rango de 0 – 1 Volt.

- 1 ó el carácter ASCII 49 para un rango de 0 – 5 Volts.
- 2 ó el carácter ASCII 50 para un rango de 0 – 10 Volts.

La secuencia en que la tarjeta de adquisición de datos debe leer los caracteres es:

rango canal 1, rango canal 2, rango canal 3,, rango canal 8

Por ejemplo, la cadena "22201102" tiene el significado mostrado en la tabla 4.1 que a continuación se muestra.

Después de que la aplicación ya envió la cadena de caracteres para indicar el rango de trabajo de cada uno de los canales analógicos, solo resta que la tarjeta de adquisición de datos envíe estos cada que la computadora se lo solicite, para mostrarlos en pantalla tanto en forma numérica como gráfica.

CANAL	CARACTER	RANGO DE TRABAJO
1	2	0 – 10 VOLTS
2	2	0 – 10 VOLTS
3	2	0 – 10 VOLTS
4	0	0 – 1 VOLTS
5	1	0 – 5 VOLTS
6	1	0 – 5 VOLTS
7	0	0 – 1 VOLTS
8	2	0 – 10 VOLTS

Tabla 4.1. Interpretación de la cadena de configuración

El rango de trabajo de cada canal analógico es seleccionado por el usuario en una ventana de configuración del canal respectivo como se podrá ver posteriormente. Como una característica importante de este sistema de adquisición de datos esta el hecho de que todos los 8 canales pueden trabajar en cualquiera de los tres rangos, solo se necesita que el usuario lo configure adecuadamente.

En la actualidad estan de moda los sistemas llamados *Inteligentes* [28], los cuales tienen como idea principal el vigilar a las condiciones de operación más importantes para que cuando salgan fuera del rango de trabajo normal se envíe un mensaje al usuario notificando el problema y sugiriendo las posibles soluciones. Esto es una forma bastante rápida de solucionar los problemas más comunes y vitales del sistema.

Al desarrollar el sistema presentado en este trabajo de tesis se pretendió que tuviera la característica de ser *inteligente*. Los mensajes que se muestran en la pantalla de la computadora cuando se detecta una situación que de no corregirse puede provocar que la información se pierda o se capture erróneamente, provocando con ello un gasto inútil de tiempo y recursos, tienen esa finalidad. Algunos de esos mensajes ya han sido explicados en secciones anteriores, otros se explican posteriormente.

No sobra decir que el usuario es quien determina el periodo de muestreo por medio de la ventana de configuración de tiempos, lo que significa que la aplicación va solicitar a la tarjeta de adquisición de datos una conversión de los 8 canales analógicos con una frecuencia indicada por el periodo de muestreo que el usuario escoja. Más adelante se explicará como opera dicha configuración.

El código que debe ser ejecutado cuando un monitoreo es iniciado es el siguiente:

```
Private Sub monitor_Click()
    mensaje = "No pudo establecerse la Comunicacion." _
    + vbCrLf + "La interface puede estar apagada o _ necesita ser _
    reseteada." _
    + vbCrLf + "Revise el cable de comunicacion puede _
    estar malconectado."
    estilo = 48
    titulo = "Error de comunicacion !!"
    ReDim capturas(7)
    If comenzar Then
        contador = 0
        adquirir = False
        'Se abre el puerto
        puerto.PortOpen = True
        'puerto.RThreshold = 8
        puerto.RThreshold = 2
        puerto.InputMode = comInputModeBinary
        'Se solicita establecer comunicación enviando primero 'una "P"
        indicando a la tarjeta de adquisición de datos 'que se inicia una
        adquisición.
        puerto.Output = Chr$(80)
        'Se espera un segundo para recibir la respuesta de la 'tarjeta
        ' de adquisisci3n de datos.
        retardo (1)
        puerto.InputMode = comInputModeText
    End If
End Sub
```

```

    puerto.InputLen = 1
'Espera una "R" como respuesta de la interface
    If puerto.Input <> "R" Then
        MsgBox mensaje, estilo, titulo
            GoTo alla
    Else
        incom.FillColor = &HC000&
        comuntex.ForeColor = &H0&
        comuntex.Caption = "Comunicacion OK"
        Form1.Caption = "Monitoreando Señales"
'Se envia configuracion de cada canal
'Intervalo de muestreo
        reloj.Interval = 1000
        reloj.Enabled = True
        'puerto.InputLen = 8
'Se leen 16 bits. 2 por cada canal
        puerto.InputLen = 16
        puerto.InputMode = comInputModeBinary
        monitor.Caption = "Terminar"
        comenzar = False
        encabezado = False
        solomonit

    End If

Else
    alla:
'Se interrumpe la comunicacion
        incom.FillColor = &HFF&
        comuntex.ForeColor = &H808080
        comuntex.Caption = "No Comunicacion"
        reloj.Enabled = False
        reloj.Interval = 0
        puerto.Output = "B"
        puerto.RThreshold = 0
        puerto.InputLen = 0
        puerto.PortOpen = False
        monitor.Caption = "Monitorear"
        Form1.Caption = "Sistema de Adquisición de Datos _
        IMP-AOI-001"

```

```

        comenzar = True
        initcanales (8)
        salirmonit
    End If
End Sub

```

En el código se puede apreciar que una de las primeras acciones que se tienen que realizar es abrir el puerto de comunicaciones para poder establecer el intercambio de información entre la computadora y la tarjeta de adquisición de datos a través de él.

Si no se puede abrir el puerto de comunicación no se puede realizar acciones de captura de datos debido a que no se puede establecer una conexión lógica entre la computadora y la tarjeta de adquisición de datos. Esta es la razón para que deba ser una de las primeras acciones.

Por otro lado, tanto el iniciar como el terminar un monitoreo es realizado utilizando un solo botón por lo que cuando se hace clic en ese botón es necesario que el programa decida si es para iniciar o finalizar un monitoreo. Para ello se utiliza una variable de tipo booleano llamada "comenzar", cuando es verdadera se inicia un monitoreo y cuando es falsa se debe finalizar.

Otra manera de iniciar o terminar el monitoreo es por medio de los menús de la aplicación para lo cual se utiliza el siguiente código.

```

Private Sub moni_Click()
    monitor_Click
End Sub

```

Se puede observar que cuando se hace clic en el menú correspondiente, por medio de código se genera el evento *monitor_click* y se ejecuta el código asociado a él. Ese código no es otro que el que se encuentra unos párrafos antes.

Esta es una de las ventajas de *Visual Basic* y de la programación controlada por eventos, ya que no es necesario volver a escribir el mismo código para cada control que lo necesite, sino que solo se necesita invocar al evento donde se encuentre ya escrito.

4.8 ADQUIRIR DATOS

La función principal del sistema aquí desarrollado es el adquirir datos de manera automatizada y en esta sección se aborda lo relacionado a ella. El mecanismo para adquirir datos es muy parecido al de monitoreo explicado en la sección anterior, la única diferencia sustantiva es que en este caso se necesita guardar los datos capturados con la periodicidad indicada por el usuario a través de la ventana de configuración de tiempos.

En este caso la información es guardada en un archivo para su posterior revisión e impresión. Los datos son salvados en un archivo con la extensión *adq* y con formato de texto como se explica en secciones posteriores.

Para iniciar ya sea un monitoreo o una adquisición es importante que el usuario previamente interconecte los analizadores a la tarjeta de adquisición de datos y esta a la computadora para que la comunicación se pueda establecer adecuadamente.

La adquisición de datos puede ser iniciada o terminada por medio de del botón o el menú mostrados en la figura 4.11. Como es obvio, el usuario debe haber ajustado y calibrado su sistema de análisis antes de iniciar una adquisición debido a que de otra formar corre el riesgo de obtener datos no válidos.

También, antes de iniciar alguna adquisición, el usuario debe introducir alguna información relacionada con el monitoreo, esto con el fin poder identificarlo posteriormente. En secciones más adelante se detallará cual información debe introducirse y el porque de esta situación.

Lo primero que el programa intenta hacer cuando una adquisición es iniciada, es establecerla comunicación entre la computadora y la tarjeta de adquisición de datos, si esta no puede establecerse se muestra en pantalla el mensaje mostrado en la figura 4.10 explicado en la anterior sección de este capítulo y se aborta la adquisición de datos para que el usuario pueda corregir la situación que esta impidiendo que la comunicación se pueda establecer.

Si la comunicación se establece, el programa continúa revisando que exista la información que identifica al monitoreo. Esta información es el lugar donde se esta trabajando, el nombre de la planta, el nombre del equipo y por ultimo el nombre del signatario. Si alguna de esta información no existe la adquisición es también abortada para que pueda ser introducida y en pantalla se despliega el mensaje mostrado en la figura 4.12.

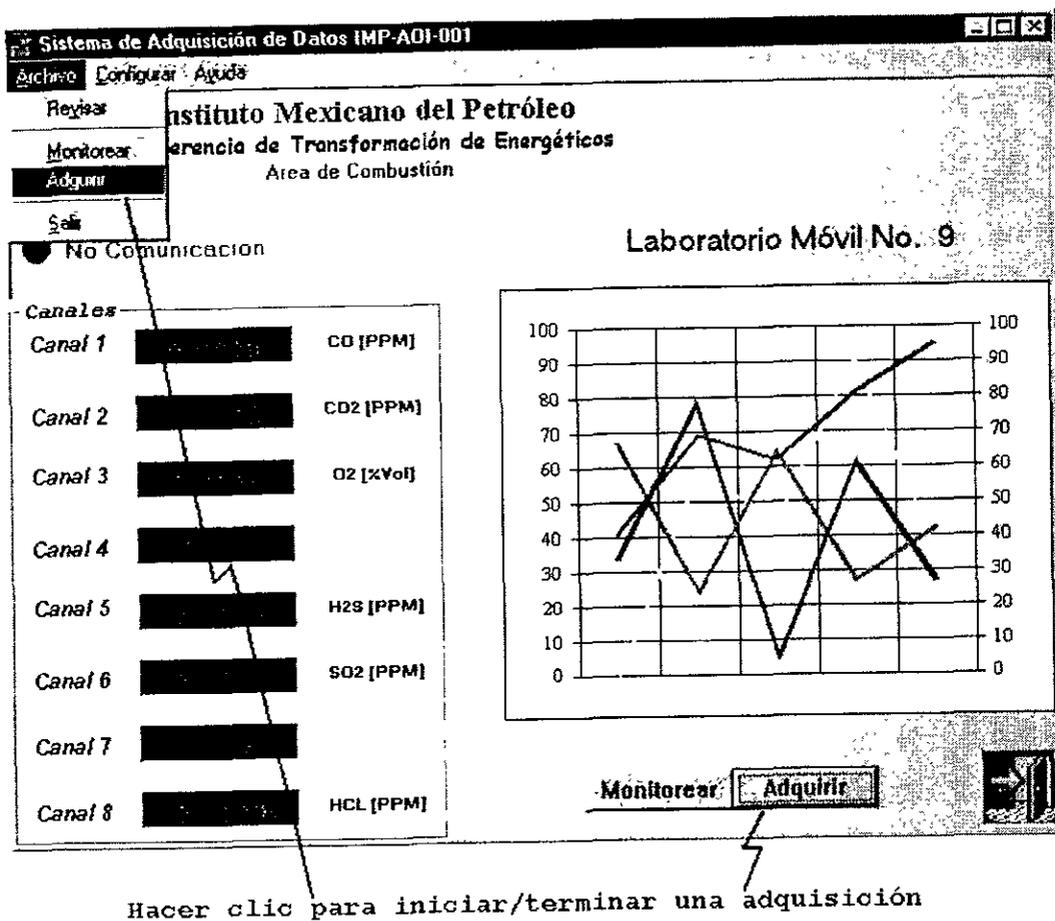


Figura 4.11. Opciones para iniciar/terminar una adquisición.

Después de que no hay ningún impedimento para que la adquisición pueda iniciarse, el programa envía a la tarjeta de adquisición de datos la cadena de configuración, tal como se explicó en la sección de monitoreo, para continuar solicitando una conversión de datos de manera periódica con el tiempo establecido por el usuario en la configuración de tiempos.

Los datos capturados por el sistema, son desplegados en forma numérica de acuerdo al canal correspondiente y además se muestra una gráfica con la tendencia de las señales de cada canal, esto se puede apreciar en la figura 4.13. Cabe aclarar que los datos mostrados en dicha figura no son de un monitoreo real por lo que pueden estar fuera de proporción.

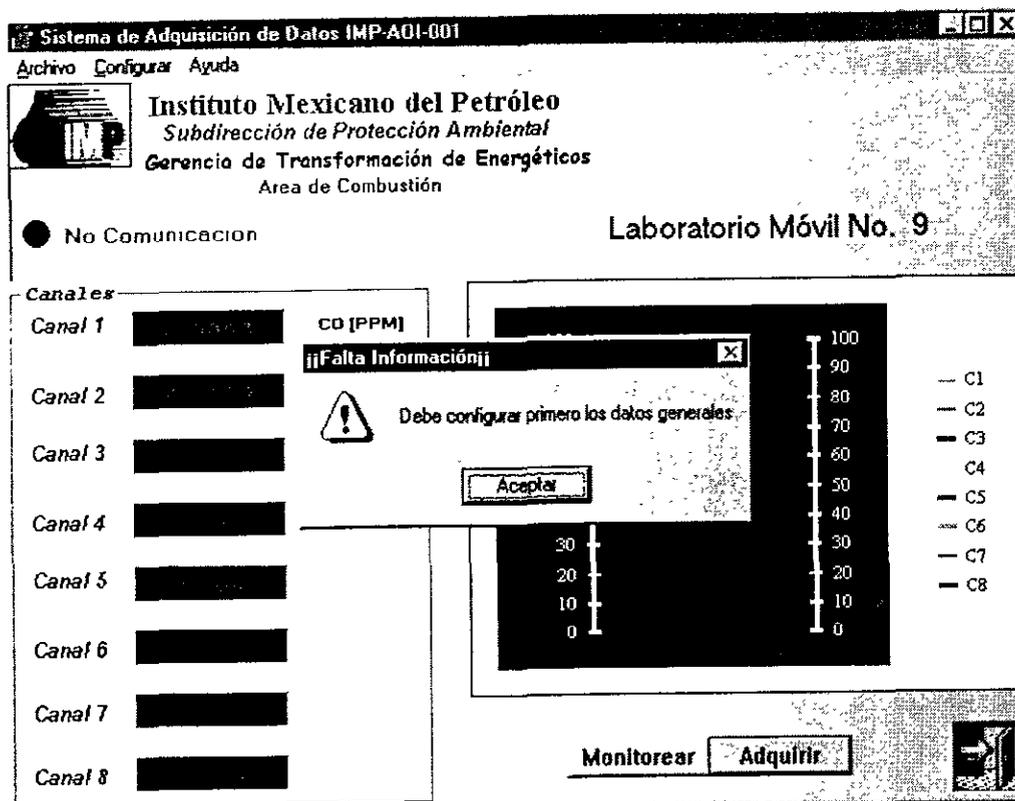


Figura 4.12. Falta de información.

En este caso solo se están adquiriendo datos correspondientes a las señales de los canales 1 y 2. En el primer caso para CO y el segundo para CO₂. La gráfica de tendencia puede tener dos ejes Y, esto con la intención de que los canales cuyo máximo valor posible sea menor o igual a 100 sean graficados con respecto al segundo canal para tener una mejor panorámica de ellos, ya que si todas las señales se graficaran en un solo canal, aquellas cuya magnitud sea muy pequeña comparada con las demás no podrán tener una representación gráfica clara.

En el caso de la figura 4.13 los datos del canal 1 son graficados en el segundo eje Y mientras que los del canal 2 lo son en el primer eje Y. Si ambos fueran graficados en el mismo eje, la tendencia de los datos del canal 1 no se podría apreciar tan claramente.

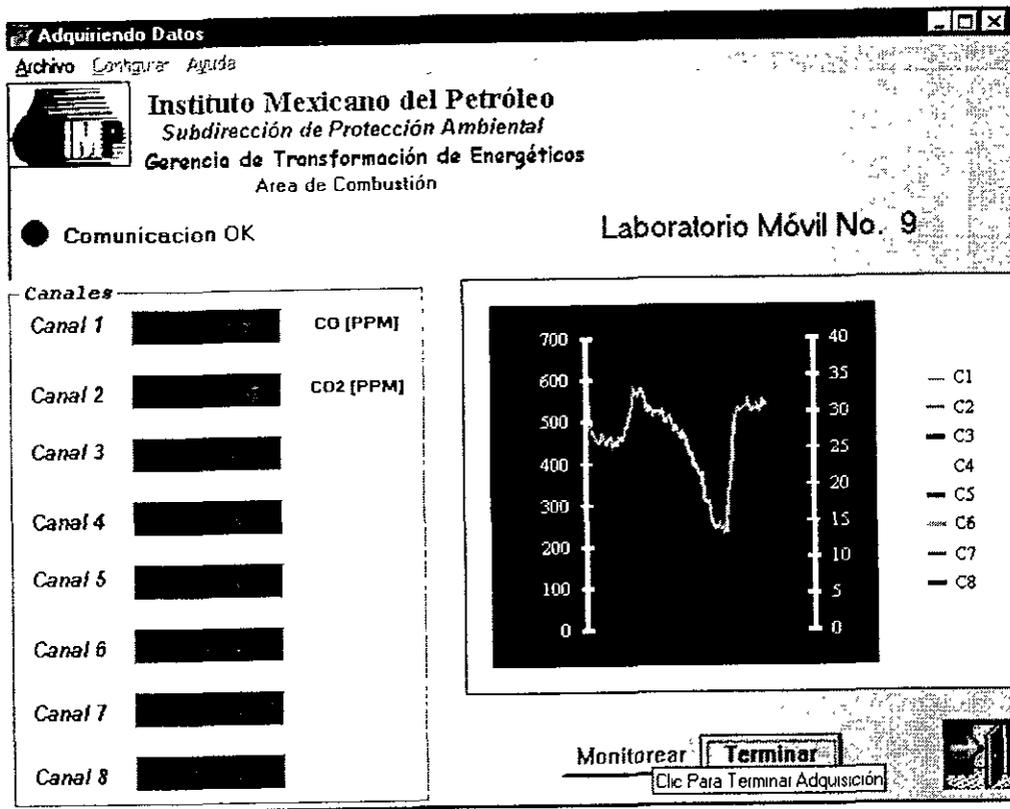


Figura 4.13. Adquiriendo datos.

Al momento de iniciar la adquisición el programa crea un archivo llamado monitoreo1.txt con el fin de que en él se almacenen los datos que el usuario necesita. La creación de este archivo es transparente para el usuario por lo que no tiene oportunidad de manipular los datos.

Los datos guardados en el archivo son los promedios de las muestras de cada canal activo en un periodo de tiempo. Cada canal es muestreado con la frecuencia indicada por el usuario.

Como se puede ver, a los usuarios del sistema solo les interesan los promedios no todo el cumulo de datos muestreados. Podría pensarse que se necesita una buena cantidad de memoria para ir almacenando los datos capturados para que en su momento se obtenga un promedio de ellos, es decir, si se necesitara obtener el promedio de los datos capturados durante un minuto con una frecuencia de muestreo de 1 segundo, sería necesario tener memoria disponible para almacenar 60 datos por cada canal activo. En el caso de que el periodo de tiempo en que se obtiene el promedio fuese más grande la cantidad de memoria necesaria también crecería de manera proporcional.

En nuestro caso los datos son manejados como del tipo doble precisión, los cuales en Visual Basic ocupan 8 Bytes de memoria cada uno. En el caso de que se tuvieran los 8 canales activos, se hiciera un muestreo cada segundo y se obtuvieran los promedios cada minuto sería necesario tener disponible 3840 Bytes de memoria tan solo para almacenar esos datos.

Para optimizar el uso de la memoria se hizo uso de la siguiente formula para obtener el promedio de los datos:

$$\bar{X}^{(n)} = \frac{n-1}{n} \bar{X}^{(n-1)} + \frac{X_n}{n}$$

para: $n \geq 2$

donde n es el número de muestreo más actual.

Para realizar los cálculos solo se necesitan tres variables, una que lleve el conteo de muestreos y que en este caso representa a n , otra donde se guarde el calculo del promedio (\bar{X}) y por ultimo una que represente al dato más recientemente muestreado (X). La primera variable puede ser del tipo *Integer* que ocupa 2 Bytes mientras que las dos ultimas son de tipo doble precisión, por lo que se requiere como máximo 130 Bytes para llevar un calculo del promedio de los 8 canales sin importar la frecuencia de muestreo y el periodo de tiempo en que se obtiene el promedio.

No sobra decir que con esta ecuación el valor del promedio es actualizado cada que se realiza un muestreo por lo que en el momento de terminar la adquisición de datos no se necesita realizar ningún calculo más

La adquisición de datos es finalizada por el usuario cuando este hace clic en el mismo botón con que la inicio, el cual ahora tiene la leyenda "Terminar" como se puede ver en la figura 4.13. También puede finalizar la adquisición por medio del submenú correspondiente en el menú "Archivo".

En la misma figura también se puede apreciar que los menús y botones que no tienen ninguna función mientras se esta realizando una adquisición se encuentran deshabilitados, esto con el fin de prevenir que el usuario intente realizar alguna acción que pueda hechar a perder el monitoreo.

Al momento en que el usuario oprime el botón para terminar la adquisición se producen varias acciones. La primera es que el archivo monitor1.txt, donde los datos han sido salvados, se cierra. El puerto de comunicación también es cerrado y el reloj que lleva el conteo de la frecuencia de muestreo es deshabilitado, todo lo anterior es transparente para el usuario.

Inmediatamente después, el programa despliega una pantalla como la mostrada en la figura 4.14, con el fin de que el usuario pueda guardar los datos obtenidos durante la adquisición, en un archivo cuyo nombre y ubicación él mismo debe seleccionar. Los datos son guardados en archivos cuya extensión es "adq". Al mismo tiempo se crea otro archivo con el mismo nombre pero con la extensión "dat" en el cual se guarda la información que identifica a los datos; esa información es el lugar donde se realiza la adquisición, la planta, el equipo, el signatario y la fecha. La creación de este último archivo también es transparente para el usuario.

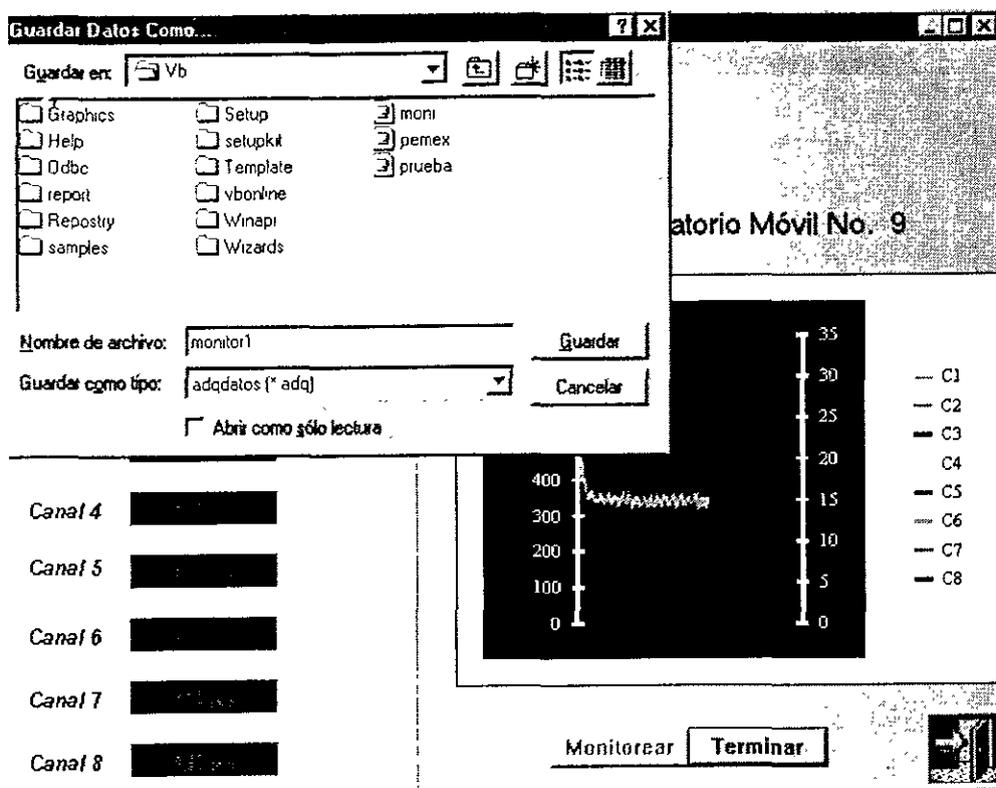


Figura 4.14. Salvar datos en archivo

Cualquier sistema de datos profesional debe salvaguardar la integridad de los datos capturados, es decir, debe impedir que estos puedan ser alterados por el usuario o cualquier otra persona con el fin de

que su tendencia en el tiempo sea diferente y así presentar mejores resultados. Esto se debe a que ciertos clientes intentan que los resultados sean maquillados para que sus plantas "cumplan" con las normas ecológicas vigentes.

Para prevenir esta situación los dos archivos creados al finalizar una adquisición, tienen el atributo de solo lectura por lo que no pueden ser modificados de ninguna forma. Como se puede entender todavía existe la posibilidad de modificar la información si los archivos son importados a una hoja de calculo, Excel por ejemplo, pero esto es responsabilidad única del usuario ya que los datos originales se conservarán íntegros.

Los archivos cuya extensión es "adq" pueden ser abiertos desde una hoja de cálculo, ya que son guardados en un formato fácilmente identificable. El separador de los datos es una simple coma. La idea es que el usuario pueda darles una presentación diferente a la que entrega el sistema para integrarlos a sus reportes.

El código ejecutado al iniciar o finalizar una adquisición de datos y que realiza todas las acciones enumeradas en esta sección es el siguiente:

```
Private Sub comenzaradq_Click()
    Dim archivo As String
    Dim directorio As String
    mensaje = "No pudo establecerse la Comunicacion." _
    - vbCrLf + "La interface puede estar apagada o necesita ser reseteada."
    -
    + vbCrLf + "Revise el cable de comunicacion puede estar malconectado."
    estilo = 48
    titulo = "Error de comunicacion !!"
    If comenzar Then
        If lugar = "" Or planta = "" Or equipo = "" Or signatario = ""
    Then
        MsgBox "Debe configurar primero" + vbCrLf + " los datos
    generales", 48, "¡¡Falta Información¡¡"
        GoTo alli
    End If
    adquirir = True 'Adquisicion
    'Establecer condiciones iniciales
    cond_ini
    'Se revisa si hay por lo menos un canal activo
```

```

canal_activo
If activo = 0 Then
    MsgBox "Debe estar activado por lo menos un canal" + vbCrLf +
"antes de comenzar;", 48, "CUIDADO;"
    GoTo alla
End If
'Preparar puerto
preparar_pto
'Espera una "R" como respuesta de la interface
'Para saber si se pudo establecer la comunicacion
If puerto.Input <> "R" Then 'No se pudo
    MsgBox mensaje, estilo, titulo
    GoTo alla
Else 'Si se pudo
    adelante
'Se envia configuracion de cada canal
cadena_config
'Se establecen condiciones de adquisicion
condiciones_adq
'Se indica que se estan adquiriendo datos
soloadq
'Se escoge el eje para graficar cada canal
escoger_eje
'Se abre el archivo donde se guardan los datos
directorio = CurDir
directorio = directorio + "\"
numarch = FreeFile
Open directorio + "monitor1.txt" For Output As numarch
Print #numarch, Str(activo) 'No. de canales activos
Print #numarch, Variables 'Titulos de canales
End If
'Finalizar
Else
'Se cierra el archivo abierto para guardar datos
Close #numarch
On Error Resume Next
directorio = CurDir
directorio = directorio + "\"

```

```

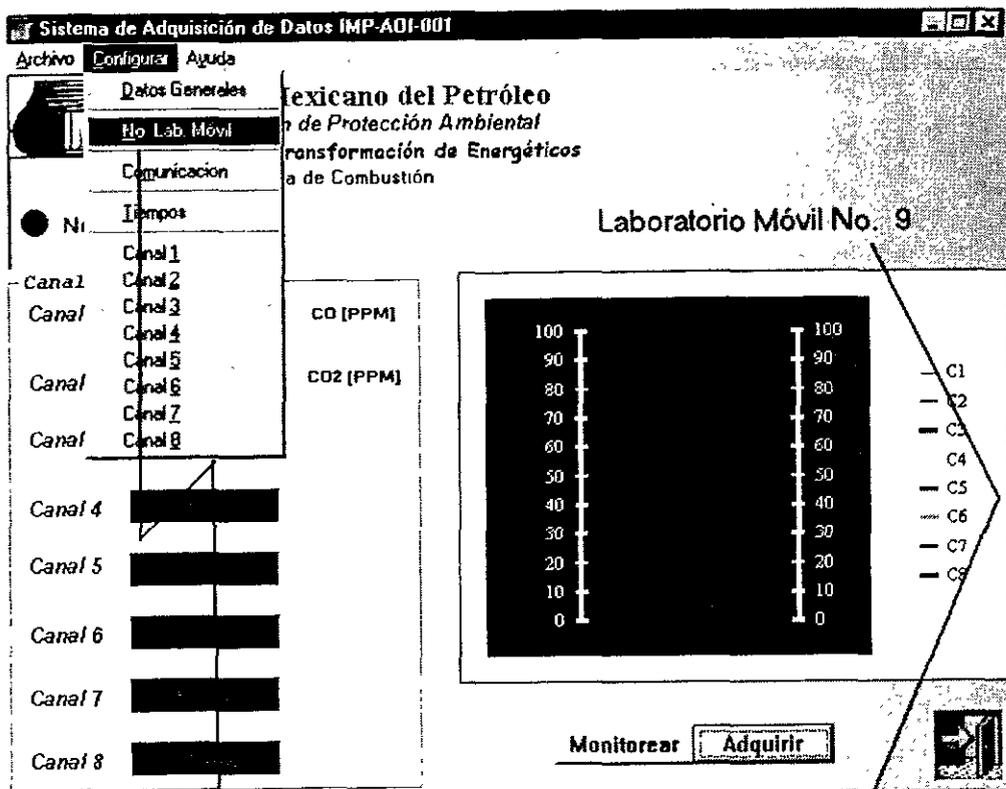
    salvar.DialogTitle = "Guardar Datos Como..."
    salvar.Filter = "adqdatos (*.adq)|*.adq|"
de_nuevo:
    salvar.ShowSave
    archivo = salvar.filename
    archivo = Mid(archivo, 1, Len(archivo) - 4)
    Name directorio + "monitor1.txt" As archivo + ".adq"
    If Err.Number = 58 Then
        Err.Clear
        boton = MsgBox("El archivo ya existe." + vbCrLf + "¿Desea
sustituirlo?", vbExclamation + vbYesNo, "CUIDADO;")
        If boton = 7 Then GoTo de_nuevo 'NO oprimido
        'Kill archivo + ".adq"
        SetAttr archivo + ".adq", vbNormal
        SetAttr archivo + ".dat", vbNormal
        Kill salvar.filename
        Kill Mid(salvar.filename, 1, Len(salvar.filename) - 4) + ".dat"
        Name directorio + "monitor1.txt" As archivo + ".adq"
    End If
    SetAttr archivo + ".adq", vbReadOnly
    numarch = FreeFile
    Open archivo + ".dat" For Output As numarch
    Print #numarch, lugar
    Print #numarch, planta
    Print #numarch, equipo
    Print #numarch, signatario
    Print #numarch, Format(Now, "long date")
    Close #numarch
    SetAttr archivo + ".dat", vbReadOnly
alla:
    'Se finaliza
    terminar_adq
    'Se sale de la adquisicion de datos
    saliradq
alli:
End If
End Sub

```

Después de finalizar la adquisición y salvar los datos en un archivo el usuario tiene la posibilidad de revisar los resultados e incluso imprimirlos, esto se explicará en secciones posteriores.

4.9 CONFIGURAR EL No. DE LABORATORIO MÓVIL

La aplicación permite configurar el número económico del laboratorio móvil donde esta instalada, por medio de una pequeña ventana como se muestra a continuación. Esto con el fin de hacer más congruente toda la información que el sistema maneja debido a que como ya se mencionó antes, existe la posibilidad de instalarlo en alguno de los existentes o que en un futuro se adquieran.



Menú para configurar el
No. de laboratorio

No. de laboratorio

Figura 4.15. Menú para configurar No. de laboratorio móvil.

Para poder configurar el número de laboratorio móvil que aparece en la ventana principal se necesita hacer uso del menú *configurar* y del submenú titulado "*No. Lab. Móvil*". Esto se puede apreciar en la figura 4.15.

La configuración del número económico del laboratorio móvil se puede realizar por medio de una ventana dispuesta expresamente para ese fin. Como se puede apreciar en la figura siguiente, el usuario utiliza un control de tipo *Combo box* y llamado *no lab*. Este control tiene una lista de números consecutivos de 1 a 10 y el usuario tiene que escoger el correspondiente al laboratorio móvil donde el sistema se encuentre *operando*. No es necesario tener una lista más amplia debido a que en la actualidad el Instituto Mexicano del Petróleo solo cuenta con 5 unidades de este tipo y no crecerá su número significativamente en un mediano plazo.

La pantalla de configuración del número económico del laboratorio móvil se muestra a continuación en la figura 4.16.

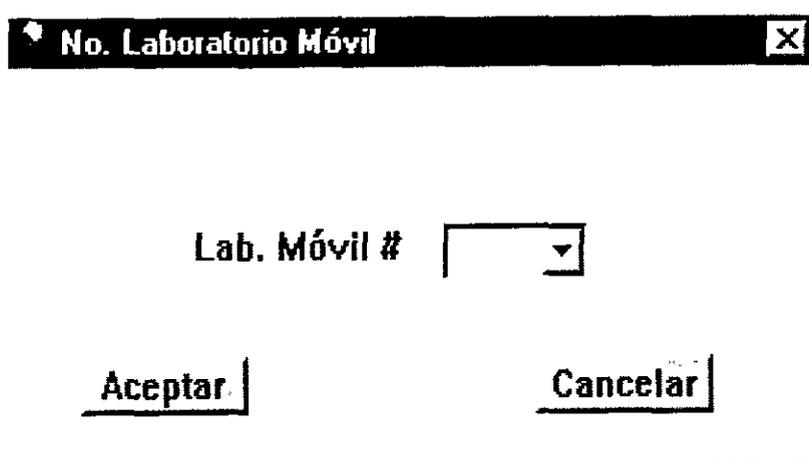


Figura 4.16. Ventana de configuración de No. de laboratorio

Para poder tener acceso a esta ventana solo se necesita hacer clic en el submenú *No. Lab. Móvil*, mencionado en párrafos anteriores, y automáticamente se ejecuta el código siguiente.

```
Private Sub labmov_Click()  
    configlab.Show  
End Sub
```

Al aparecer la ventana de configuración el usuario solo tiene dos opciones, una es el realizar alguna modificación y confirmarla por medio del botón *Aceptar* o solo revisar la información y salir de la ventana sin modificarla utilizando el botón *Cancelar*.

Cuando se hace clic sobre el botón *Aceptar* se ejecuta el código siguiente. En el se puede apreciar que la información es salvada en los registros de Windows para poder recuperarla cada que la aplicación es corrida con el fin de que el usuario no tenga que volver a configurar al sistema, para ello se utiliza la instrucción *SaveSetting App.Title* de Visual Basic. Para poder utilizar esta instrucción es necesario antes declarar el parámetro *App.Title*, ya que en caso de no hacerlo se tendrá un error en tiempo de ejecución.

En el caso nuestro, ese parámetro es declarado justo cuando la aplicación es arrancada, específicamente cuando la ventana principal es cargada en memoria. *App.Title* es el nombre con el cual se va a identificar a la aplicación dentro de los registros de Windows. El nombre de nuestra aplicación es *adqdatos*.

```
Private Sub labacep_Click()
    labmovil = nolab.ListIndex + 1
    SaveSetting App.Title, "configuracion", "Nolab", nolab.ListIndex +
    1
    Form1.laboratorio.Caption = "Laboratorio Móvil No. " _
    + Str(labmovil)
    configlab.Hide
    Form1.Show
End Sub
```

Cuando se hace clic sobre el botón *Cancelar* se debe salir de esta ventana sin registrar algún cambio aunque el usuario lo haga en la ventana, eso se logra ejecutando simplemente el código que a continuación se enumera.

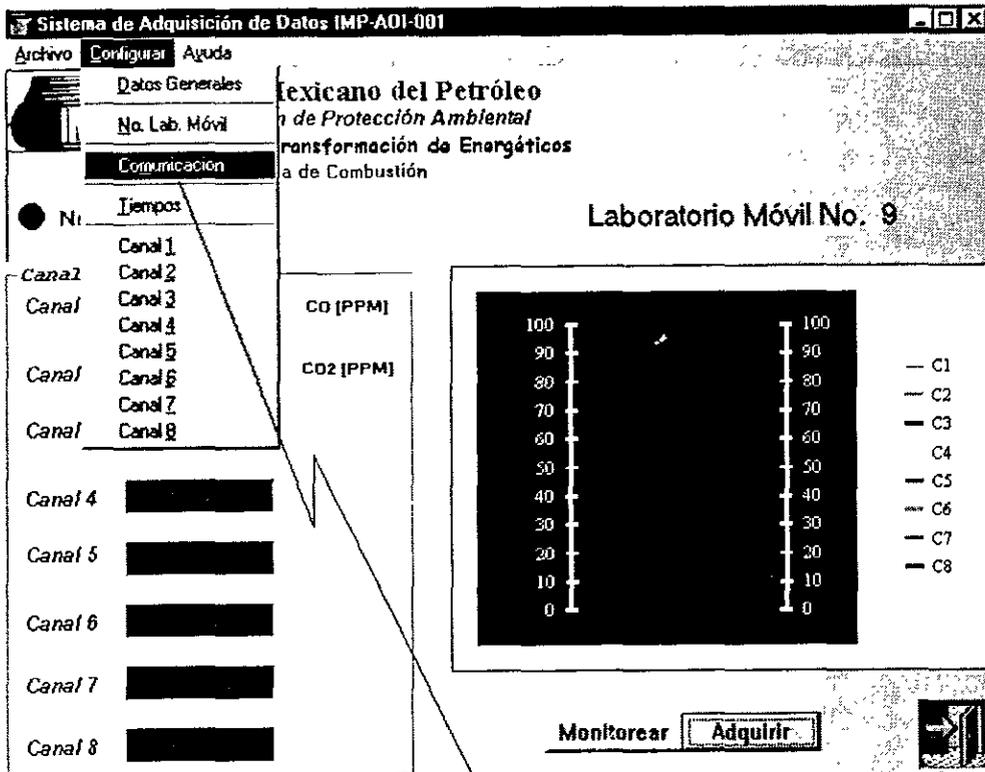
```
Private Sub labcancel_Click()
    configlab.Hide
    Form1.Show
End Sub
```

Como se puede ver, únicamente se oculta la ventana de configuración llamada *configlab* y se hace aparecer nuevamente a la ventana principal.

A continuación se explica la manera de configurar el puerto de comunicaciones.

4.10 CONFIGURAR EL PUERTO DE COMUNICACIONES

Para configurar el puerto de comunicaciones se creó la ventana llamada *Configpto*. Esta ventana contiene las opciones que deben ser configuradas para que la comunicación entre la computadora y la tarjeta de adquisición de datos se establezca correctamente.



Menú para entrar a la ventana de configuración de comunicaciones.

Figura 4.17. Menú para configurar las comunicaciones

Para nuestro caso, la tarjeta de adquisición de datos puede transmitir bajo las siguientes condiciones:

- velocidad = 9600 bauds
- paridad = N

- bits por dato = 8
- bits de parada = 1

Las anteriores condiciones establecerse sin tener que crear una ventana de configuración, pero se penso en que en algún momento la aplicación pudiese trabajar con otras tarjetas de adquisición de datos que manejaran otras características de comunicación por lo cual era necesario dejar abierta la posibilidad de adecuarlas.

En la figura 4.17 se muestra el menú con el cual se puede tener acceso a la ventana de configuración de comunicaciones desde la ventana principal.

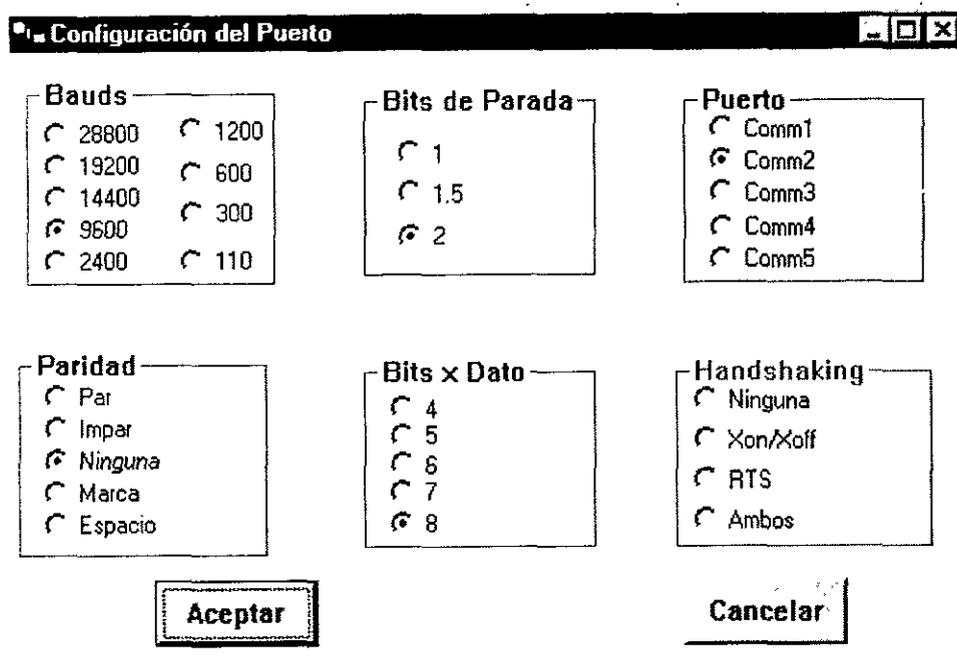


Figura 4.18. Ventana de configuración de comunicaciones

Para entrar a la ventana de configuración de comunicaciones solo se tiene que hacer clic en el submenú *Comunicación* del menú *Configurar* para ejecutar el siguiente código.

```
Private Sub com_Click()  
    configpto.Show  
End Sub
```

Este código solo hace aparecer dicha ventana sobrepuesta a la ventana principal. La ventana de configuración es mostrada en la figura 4.18.

Esta ventana es bastante diferente a todas las demás utilizadas para configurar alguna parte del sistema. La ventana esta construida básicamente de controles llamados *marcos (frames)* y *botones de opción (OptionButton)* contenidos en ellos. En la ventana el usuario solo tiene que hacer clic en la opción correcta para seleccionarla y automáticamente todas las demás quedan no seleccionadas.

Como es obvio, cada *marco* solo contiene las opciones que el estándar de comunicación permite.

El primer marco titulado *Bauds* se utiliza para configurar la velocidad de intercambio de información entre la tarjeta de adquisición de datos y la computadora a través del puerto serie. Aunque en la actualidad ya se manejan velocidades superiores a las mostradas allí, *Visual Basic* recomienda solo trabajar con estas debido a que el estándar pudiese no soportarlas si no es implementado adecuadamente. Como ya se mencionó antes, la tarjeta de adquisición de datos solo puede trabajar a una velocidad de 9600 bauds por lo que se recomienda al usuario no intentar trabajar con otra diferente mientras la tarjeta no tenga esa opción.

Otro marco es titulado *Bits de Parada* y en el se puede configurar este parámetro utilizado en el estándar RS 232, el cual sirve como un método simple de protección contra errores de transmisión. Como dato curioso se puede decir que con este método se pueden detectar el 100% de los errores que afectan a un solo bit pero la probabilidad de detectar errores que involucren a más de un bit disminuye al 50% por lo que este método es bastante útil y practico en sistemas que utilizan medios de comunicación muy seguros.

La paridad consiste en que cada carácter transmitido incluye un bit llamado bit de paridad; este bit de paridad puede tomar el valor de 0 ò 1 dependiendo de sí el numero total de 1's en el carácter es par (paridad par) o impar (paridad impar). Para el puerto serie se tienen las siguientes opciones:

- SIN PARIDAD: No se utiliza bit de paridad
- PARIDAD PAR: Se fija el valor de bit de paridad de tal forma que el número total de bits con valor 1, incluyendo al bit de paridad, sea par.
- PARIDAD IMPAR: El bit de paridad es fijado de tal forma que el numero total de bits con valor 1, contando al bit de paridad, sea impar.
- MARCA: El bit de paridad siempre es 1.
- ESPACIO: El bit de paridad es siempre 0.

En el caso nuestro, no se utiliza bit de paridad por restricciones de la tarjeta de adquisición de datos, aunque es importante señalar que durante las pruebas realizadas no se detectaron errores de transmisión por lo que puede considerarse que el sistema cuenta con un medio seguro para transmitir, consecuentemente no es necesario incluir en cada carácter transmitido un bit de paridad para detectar errores.

Otro marco es titulado *Bits de Parada* en el cual se programa el número de bits de parada que contendrá un carácter cuando es transmitido. Estos bits de parada sirven para marcar el final de un carácter por lo que no deben considerarse como bits normales de información. Los valores que puede tomar son 1, 1.5 y 2 para indicar el equivalente en tiempo de duración de un bit normal, que este bit debe permanecer en 1.

Lo anterior tiene su origen en las telecomunicaciones que utilizaban sistemas mecánicos que debían restablecerse después de haber transmitido un carácter. En la actualidad casi todos los sistemas de comunicaciones utilizan el valor de 1 tal como es el caso del trabajo descrito aquí.

El marco titulado *Bits x Dato* debe ser utilizado por el usuario para configurar el número de bits de datos por cada carácter transmitido. Por razones históricas se sigue utilizando palabras desde 5 bits de datos pero en la actualidad el código más utilizado es el conocido como ASCII (American Standard Code Interchange Information) el cual solo maneja datos de 7 u 8 bits.

Finalmente este último código es el que se utiliza en este trabajo para codificar la información, en su modalidad extendida (8 bits), debido principalmente a que permite manejar la información de manera más rápida ya que no se tienen que hacer operaciones adicionales para descifrarla pues toda la información es trabajada con 8 bits.

Otro de los marcos es titulado *Puerto* y aquí es donde el usuario escoge el puerto serie de su computadora personal a donde se conecta el cable que interfaza a la tarjeta de adquisición de datos con ella.

Tradicionalmente una computadora personal es ensamblada con dos puertos seriales, *Com1* y *Com2*, pero pueden añadirse hasta tres más para tener un total de cinco. Se pensó que era necesario que el usuario pudiese escoger el puerto para darle flexibilidad al sistema pues de otra manera la tarjeta tendría que estar atada al mismo puerto siempre, lo cual es un problema potencial al momento de intentar conectar otros dispositivos (módems, escaners, etc.) a este tipo de puertos.

Es importante aclarar que aunque se tienen cinco posibles puertos para escoger, es necesario que físicamente estén instalados en la computadora ya que de otra manera se presentará un error al intentar

escoger un puerto no presente, por lo que es necesario que el usuario sepa cuantos puertos de este tipo tiene su computadora y a cual de ellos conectará la tarjeta de adquisición de datos.

El último marco presente en esta ventana es el titulado *Handshaking* que en el caso del presente trabajo no es utilizado debido a que la cantidad de datos provenientes de la tarjeta de adquisición de datos no es grande, pero se decidió ponerlo para futuras mejoras a la tarjeta que requirieran esta característica.

Dado que esta es la primera versión del sistema se piensa en un futuro mejorar las características tanto de hardware como de software con el fin de poder abarcar otros campos de aplicación que lo requieran.

Después de haber entrado a esta ventana el usuario tiene dos opciones nuevamente, hacer modificaciones y salvarlas o simplemente salir sin que la configuración tenga algún cambio.

Para salir sin modificar la configuración solo se tiene que hacer clic en el botón *Cancelar* y al instante se ejecutará el siguiente código para ocultar esta ventana y mostrar a la principal sin salvar alguna modificación manteniendo la que se tenía al momento de entrar.

```
Private Sub configptocan_Click()
    configpto.Hide
    Form1.Show
End Sub
```

Si el usuario desea realizar algunas modificaciones a la configuración y desea salvarlas solo tiene que hacer clic en el botón *Aceptar* para ejecutar el código siguiente.

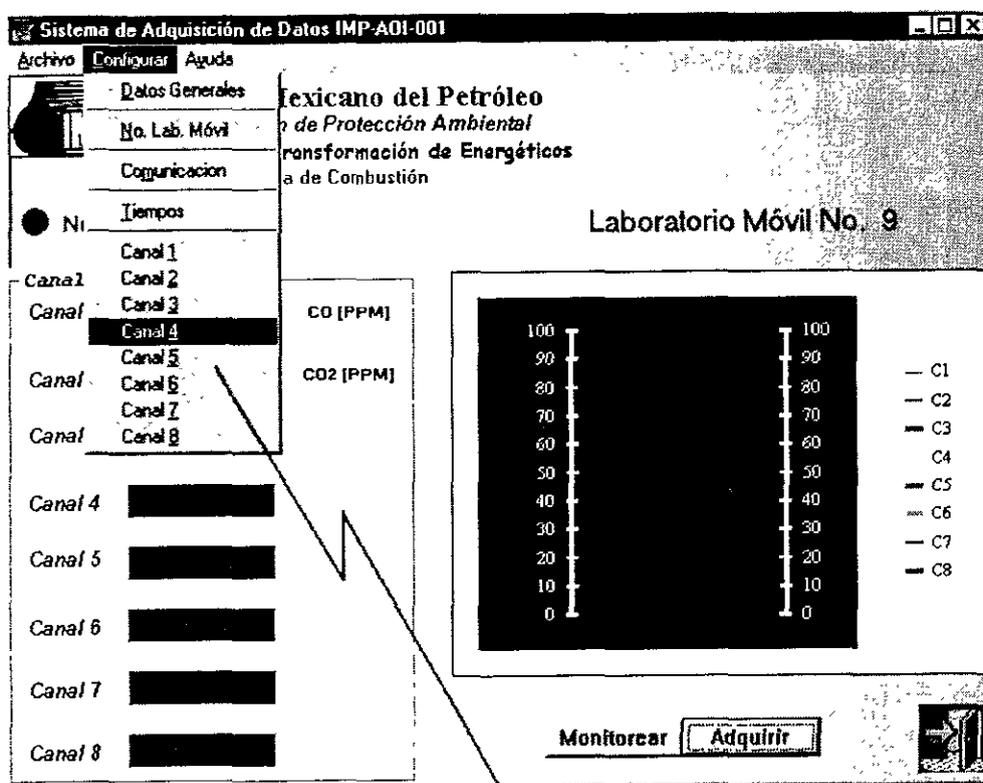
```
Private Sub configptook_Click()
    Form1.puerto.CommPort = nvopto
    Form1.puerto.Settings = nvelocidad & "," & _
    nparidad & "," & nbitsdato & "," & nbitsparada
    SaveSetting App.Title, "configuracion", _
    "ajustes", Form1.puerto.Settings
    SaveSetting App.Title, "configuracion", _
    "numeropto", Form1.puerto.CommPort
    configpto.Hide
    Form1.Show
End Sub
```

Aquí nuevamente se hace uso de los registros de Windows para salvar la configuración del puerto y poder recuperarla cuando se requiera, especialmente al arrancar la aplicación.

4.11 CONFIGURAR CADA CANAL ANALÓGICO

Esta de configuración es una de las más importantes, ya que en ella se pueden configurar las características de trabajo de cada uno de los ocho canales analógicos.

Como es evidente, no es practico intentar configurar varios canales de manera simultanea ya que se puede prestar a confusión, por lo que se decidió utilizar una ventana para configurar a cualquiera de los ocho canales, pero solo uno a la vez.



Menú para entrar a la ventana de configuración de cada canal.

Figura 4.19. Menú para configurar

En la figura 4.19 se muestra el menú que se utiliza para entrar a la ventana de configuración de cada canal. Aunque pareciera que existe una ventana de configuración por cada canal analógico en la practica solo es una. En esta ventana se despliegan los parámetros de operación del canal que específicamente se quiera configurar.

El poder utilizar un solo formulario para configurar a cualquiera de los canales es una de las ventajas de la programación en Visual Basic ya que el código es reutilizable y, estructurando el programa de manera adecuada se puede lograr fácilmente lo anterior.

Figura 4.20. Ventana de configuración, canal inactivo.

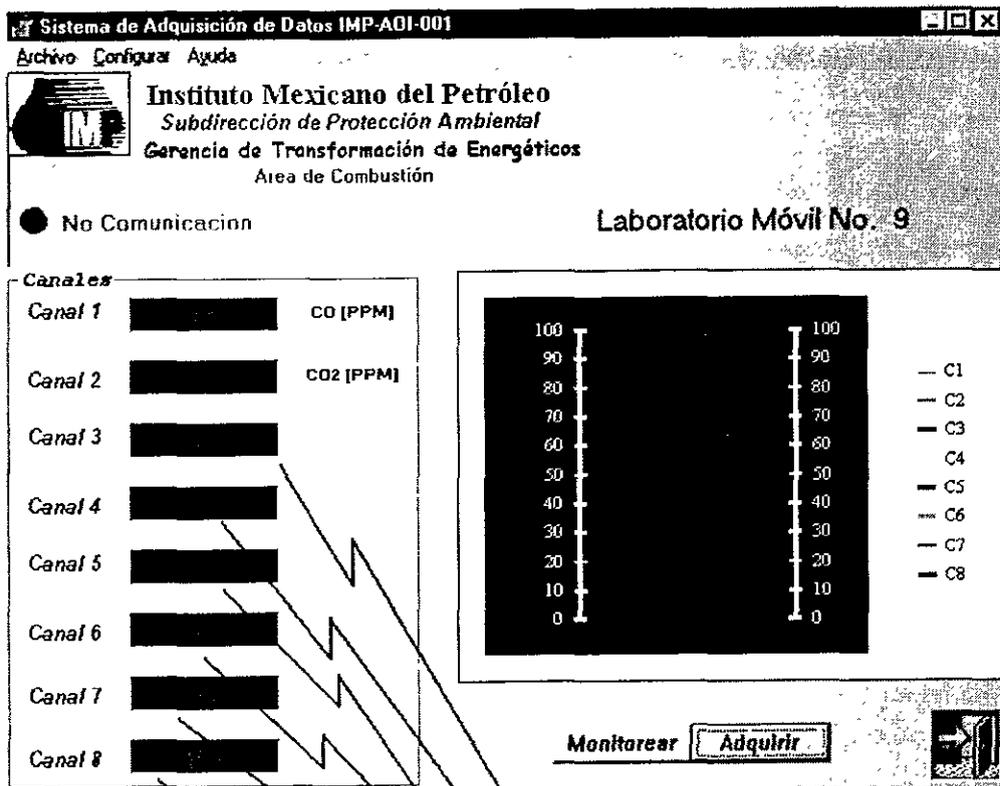
Al momento de hacer clic en el menú que corresponde al canal que en particular se quiere configurar aparece la ventana de configuración para ese canal debido a que se ejecuta el siguiente código.

```
Private Sub c4_Click()
```

```

configurarcanal (4)
Form1.Hide
configadq.Show
End Sub
    
```

En el código anterior, la primera instrucción hace un llamado a una subrutina llamada *configurarcanal(i)* donde *i* es el número de canal que se quiere configurar. Esta subrutina se encuentra en el módulo de código estándar llamado *Canales* y el código de ella se mostrará después, aunque a grosso modo esta subrutina se encarga de recuperar de los registros de Windows la configuración del canal para ponerla en la ventana antes de mostrarla.



Canales deshabilitados.

Figura 4.21. Canales deshabilitados

Como ejemplo, en la figura 4.20 se muestra la ventana de configuración para el canal cuatro el cual se encuentra deshabilitado, es decir, el usuario no lo esta utilizando para monitorear la señal proveniente

de algún equipo de análisis, y por lo tanto no es necesario presentar su valor en la ventana principal al momento de realizar una adquisición de datos o un monitoreo.

Cuando algún canal se encuentra deshabilitado no se puede realizar ningún cambio en cualquiera de sus parámetros, tan solo se puede cambiar el estado del canal. Esto se nota porque el color de los controles tiende a ser más opaco de lo normal y al momento de hacer clic en cualquiera de ellos no responden a él.

Un canal deshabilitado conserva la última configuración que tuvo, debido a que en la práctica en un buen numero de ocasiones se deshabilita porque el analizador cuya salida analógica se encuentra conectada a él esta fuera de servicio por mantenimiento preventivo o correctivo y volverá a ser conectado en cuanto se encuentre listo, de esta manera el usuario no tendrá que volver a configurar el canal.

Configuración de Canal 2

Canal 2

Variable: CO2

Unidades: PPM

Esc. Max: 5000

R. Entrada: 0 - 10 V

Activar Canal

Si

No

Aceptar Cancelar

Figura 4.22. Canal habilitado

Todos los canales deshabilitados muestran la leyenda "Libre" en el cuadro de texto que le corresponde en la pantalla principal, para indicarle al usuario que estos canales no se encuentran

ocupados para monitorear alguna señal y que puede utilizarlos si es necesario. En la figura 4.21 se muestra un ejemplo de la pantalla principal con los canales 3 a 8 deshabilitados.

Cuando un canal es habilitado para ser utilizado es posible realizar cambios a su configuración para adecuarlo a las características de la entrada analógica. A continuación se muestra, en la figura 4.22, como ejemplo de un canal activo la ventana de configuración correspondiente al canal 2.

Aparte de activar o desactivar el canal es necesario configurar otros cuatro parámetros (en el caso de que se active) para que la adquisición de datos se realice de manera correcta.

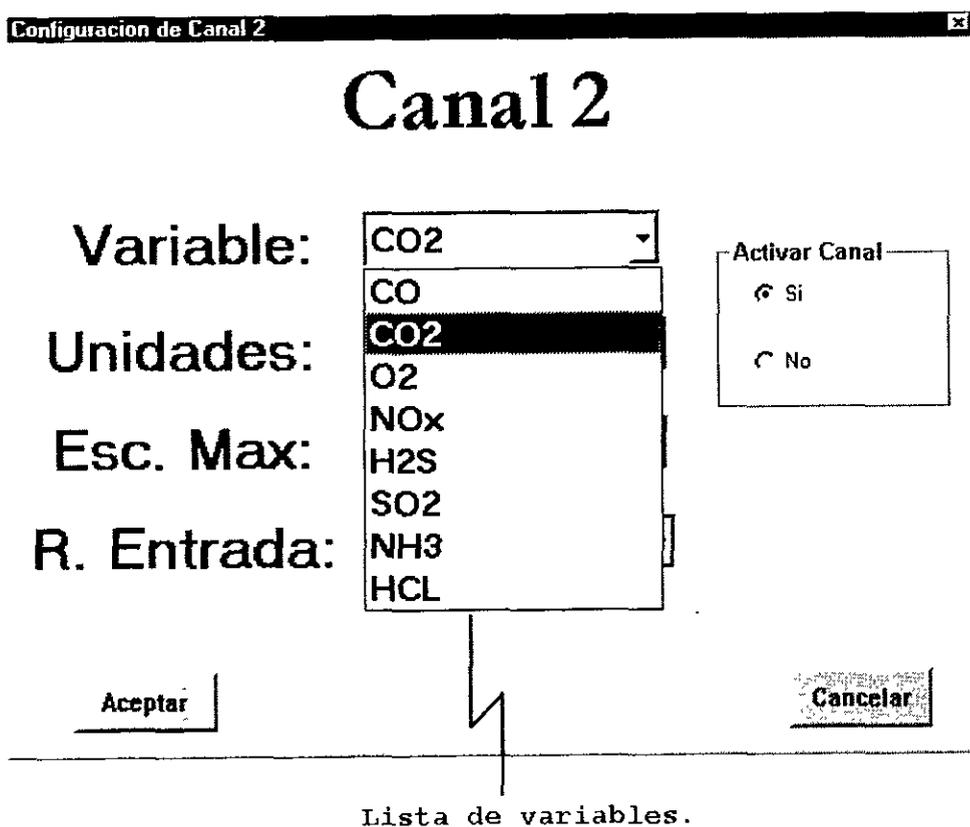


Figura 4.23. Desplegado de la lista de variables

El primer parámetro a configurar es la variable que va a manejar el canal. Como ya se ha mencionado antes los laboratorios móviles para los cuales fue diseñado el sistema, pueden manejar hasta 8 distintos analizadores, cada uno para distintas variables. Esas 8 variables que se analizan son:

- Monóxido de Carbono, CO.

- Bioxido de Carbono, CO₂
- Oxigeno, O₂.
- Oxidos Nitrosos, NO_x.
- Acido Sulfidrico, H₂S.
- Bióxido de Azufre, SO₂.
- Amoniac NH₃.
- Acido Clorhídrico, HCl.

Los laboratorios móviles pueden contar con un analizador para cada variable pero no necesariamente van a estar todos en uso simultáneamente al momento de capturar los datos de alguna emisión, esto debido a que las variables a analizar dependen de la fuente de emisiones bajo estudio.

Es importante señalar que estas y otras opciones fueron señaladas por los usuarios al momento de que ellos nos indicaron sus requerimientos del sistema.

Para configurar la variable que el canal va a manejar, el usuario solo debe escoger la adecuada de entre la lista de ellas como se muestra a continuación. Es importante también señalar que el usuario no tiene la posibilidad de introducir otras opciones por indicación de ellos mismos, debido a que según su experiencia hay más probabilidad de configurar erróneamente el sistema.

Lo anteriormente señalado también aplica para los otros tres parámetros.

En la figura 4.23 se puede observar la lista de opciones de variable al momento de intentar ser configurada.

El siguiente parámetro a configurar son las unidades en que se expresarán las lecturas obtenidas en cada canal. El usuario solicitó que para el caso particular del sistema de adquisición de datos de los laboratorios móviles solo son necesarias dos tipos de unidades:

- Partes Por Millón, PPM y
- Porcentaje de volumen, % Vol.

Esas unidades son mostradas, en la ventana principal al momento de estar monitoreando o adquiriendo datos, junto al nombre de la variable que monitorea cada canal.

En la figura 4.24 se muestra el desplegado de la lista de unidades para la ventana de configuración del canal 2, como un ejemplo. Las listas son las mismas para la configuración de cualquier canal.

Las unidades de porcentaje volumen (% vol) son utilizadas para expresar la concentración de oxígeno, mientras que las partes por millón (PPM) para la concentración de los otros gases.

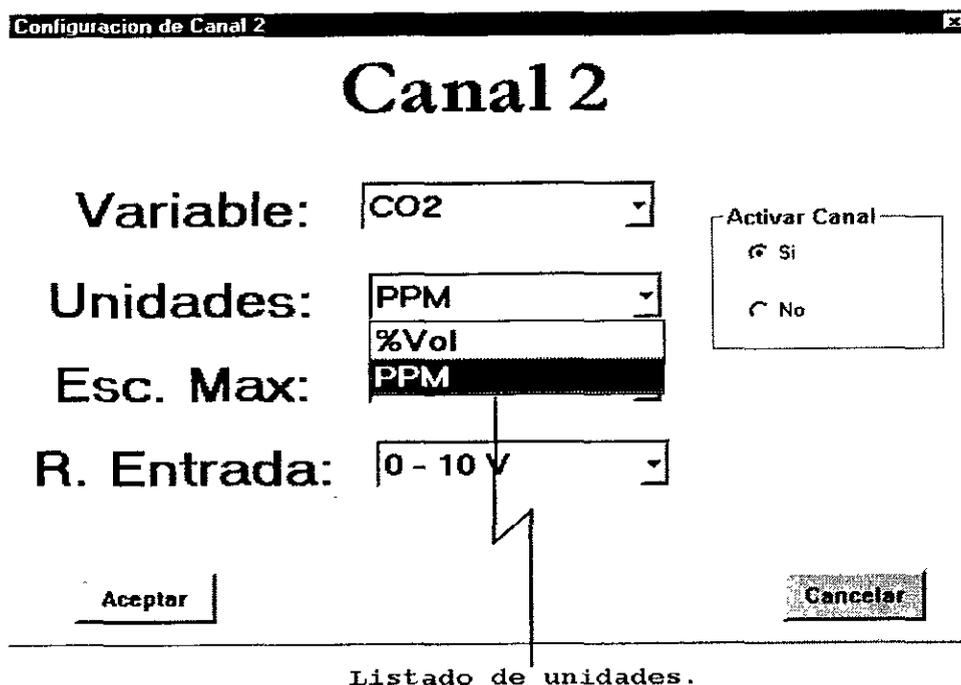


Figura 4.24. Desplegado de unidades

Otro de los parámetros a configurar es el relacionado con la escala máxima del canal analógico, la cual debe ser igual a la del analizador cuya señal se este monitoreando. Esto significa que si el analizador esta trabajando en una escala de 0 – 5000 PPM el canal que lo monitorea debe tener configurada una escala igual o de otra forma se obtendrán resultados erróneos.

Por ejemplo, si el equipo de analisis tiene una salida analógica de 0 –10 volts y una escala de trabajo de 0 – 1000 PPM y esta bien calibrado y ajustado, cuando este leyendo una concentración de 500 PPM, en su salida analógica se tendrán 5 volts. Por otro lado, si el canal analógico del sistema de adquisición de datos que lee esa señal, tiene configurada una escala de trabajo de, por ejemplo, 2000 PPM, y un rango de 0 – 10 volts, cuando lea un voltaje de 5 volts, el sistema de adquisición de datos lo interpretará como 1000 PPM, lo cual es evidentemente erróneo.

Las doce escalas que se manejan fueron también sugeridas por los futuros usuarios y abarcan todas las posibilidades con que ellos trabajan. El número de escalas es grande debido a que un mismo equipo de análisis de emisiones puede manejar varias de ellas, pero solo una a la vez. Al realizarse monitoreos en diferentes lugares es posible que la concentración de los gases emitidos varíe y por lo tanto es necesario cambiar las escalas de trabajo para obtener la mejor resolución.

Las doce escalas de trabajo son:

- 0 -10 000
- 0 – 5 000
- 0 – 1 000
- 0 – 500
- 0 – 200
- 0 – 100
- 0 - 50
- 0 – 20
- 0 – 10
- 0 – 5
- 0 - 2
- 0 – 1
- 0 – 0.1

Esta lista de escalas de trabajo es fija y no puede modificarse ya que también por recomendación de los usuarios, no pueden introducir manualmente otra escala que no este en la lista anterior.

La combinación de escala máxima de trabajo y las unidades es algo que la persona que esta configurando debe tomar en cuenta. Por ejemplo, si se necesitan unidades de PPM se puede utilizar *cualquier* escala, pero si se utilizan unidades de % Vol, no es lógico utilizar escalas mayores a 100 debido estas unidades se refieren a una porción de un todo, es decir, a porcentajes menores a 100%. Es obvio que el usuario debe tener cuidado en ese aspecto.

No se utilizan escalas que manejen cantidades negativas debido a que los analizadores de emisiones contaminantes son calibrados y ajustados con muestras patrón para ajustar su cero y su escala máxima. Para analizar la muestra esta es introducida al sistema por medio de una pequeña bomba y su respuesta

es comparada contra la perteneciente a la referencia, que no es otra cosa que la respuesta del patrón con que se ajusto el cero.

Configuración de Canal 2 [X]

Canal 2

Variable:

Unidades:

Esc. Max:

R. Entrada:

Escalas: 10000, 5000, 1000, 500, 200, 100, 50, 20

Activar Canal: Si No

Escalas

Figura 4.25. Desplegado de la lista de escalas

En la realidad, si se obtuvieran lecturas negativas significaría que la muestra a analizar tiene menos concentración que el patrón de cero o que en lugar de absorber la muestra el analizador esta emitiendo el gas contaminante. Ambos casos no son posibles debido a que por un lado los gases patrón son embazados con estrictos controles de calidad y deben contar con un certificado y por el otro lado, tampoco es posible que el analizador este produciendo el gas contaminante y lo este enviando a la atmósfera.

Cuando se obtienen lecturas negativas se debe básicamente al ruido inducido en los cables que transmiten las señales analógicas de cada analizador a la tarjeta de adquisición de datos

En la figura 4.25 se muestra el desplegado de la lista de escalas que el usuario tiene al momento de hacer clic para seleccionar alguna.

El último parámetro a configurar es el rango de entrada de la señal analógica proveniente de la salida del mismo tipo de los equipos analizadores de emisiones contaminantes.

Como ya se ha explicado antes, no existe un estándar que regule las características de dichas señales y por lo tanto los fabricantes de esos equipos tienen la libertad de escogerlas a su libre arbitrio.

A pesar de la falta de estándares las salidas analógicas de los equipos de análisis de emisiones contaminantes caen por lo regular en las siguientes categorías:

- 0 – 1 volts
- 0 – 5 volts
- 0 – 10 volts

De los tres rangos, en la actualidad el más común es el último debido a que presenta mejores características para su manejo en sistemas de adquisición de datos. El segundo es menos frecuente y aunque no es muy común en los equipos de origen americano si lo es en los de origen europeo [86]. El primer rango esta prácticamente en extinción pero muchos equipos, sobre todo antiguos, todavía lo manejan. Los equipos de los laboratorios móviles pueden tener alguno de los tres rangos antes mencionados, es decir, unos tienen salida de 0 – 1 volt, otros de 0 – 5 volts y los restantes de 0 – 10 volts.

Pensando en esta situación la aplicación fue desarrollada para permitir que cualquier canal pueda manejar cualquiera de los tres rangos antes mencionados, lo cual le brinda una buena flexibilidad.

En la figura 4.26 se puede ver el desplegado de la lista de los rangos de entrada de la señal analógica.

Originalmente se tenía pensado incluir en la lista de rangos al estándar de salidas analógicas de 4 – 20 mA, pero debido a que, por un lado ninguno de los analizadores de los laboratorios móviles tiene en la actualidad dicha salida y por otro, es muy poco probable que después se adquieran algunos que la tengan se consideró innecesario dejarla como una opción e inclusive podría ser causa de confusiones.

El estándar 4 – 20 mA es más frecuentemente utilizado en equipos analizadores situados en estaciones bastante distantes del cuarto de control, caso contrario al de los laboratorios móviles.

Figura 4.26. Desplegado de la lista de rangos

Para salir de la ventana de configuración de canales sin salvar las modificaciones que se hayan hecho, es decir, dejando intacta la configuración que existía al momento de entrar a dicha ventana el usuario solo tiene que hacer clic en el botón *Cancelar*. El siguiente código se ejecutará inmediatamente.

```
Private Sub cancelar_Click()
    configadq.Hide
    Form1.Show
End Sub
```

Este código solo ocultará la ventana de configuración y mostrará a la ventana principal sin salvar alguna información en los registros de Windows.

Si el usuario necesita realizar cambios en la configuración de algún canal debe salvarlos para que tengan efecto ya que de otro modo la configuración original no será modificada.

Para salvar los cambios realizados en la configuración de algún canal el usuario solo tiene que hacer clic en el botón *Aceptar*. A continuación se muestra el código que se usa para salvar dichas modificaciones.

```

Private Sub aceptar_Click()
    If configdq.noactivar.Value = True Then
        A = 2 'Desactivado
    Else
        A = 1 'Activado
    End If

    'b es la variable a monitorear
    b = configdq.Combovar.ListIndex + 1
    'c representa las unidades de medición
    c = configdq.Combouni.ListIndex + 1
    'd representa la escala máxima para el canal
    d = configdq.Comboescmax.ListIndex + 1
    'E es el rango de entrada del canal
    E = configdq.Comboran.ListIndex + 1

    'Se obtiene la cadena con la configuracion a salvar
    canalconfig = (A * 10000) + (b * 1000) + (c * 100) + _
        (d * 10) + E

    'Se escoge el canal cuya configuracion se salvara
    canal = Mid(titulo.Caption, 7)
    SaveSetting App.Title, "configuracion", "canal" + _
        canal, canalconfig
    ncanal = Val(canal) - 1
    If configdq.noactivar.Value = True Then
        Form1.entrada(ncanal).Caption = "Libre"
        Form1.nomvar(ncanal).Caption = ""
    Else
        Form1.entrada(ncanal).Caption = "0.0000"
        Form1.nomvar(ncanal).Caption = Combovar.Text + _
            " [" + Combouni.Text + "]"
    End If
    configdq.Hide
    Form1.Show
End Sub

```

La configuración de cada canal es guardada utilizando una cadena de seis caracteres donde cada uno de ellos representa a alguno de los parámetros. La cadena tiene el formato "ABCDDDE" y el significado es el siguiente:

- **A** es el estado del canal. 1=habilitado, 2=habilitado.
- **B** es la variable a monitorear. 1 – CO, 2 – CO₂, 3=O₂, 4=NO_x, 5=H₂S, 6=SO₂, 7=NH₃, 8=HCl.
- **C** representa a las unidades en que se expresan las lecturas. 1=%Vol, 2=PPM.
- **DD** es la escala de trabajo. 01=10 000, 02=5 000, 03=1 000, 04=500, 05=200, 06=100, 07=50, 08=20, 09=10, 10=5, 11=1, 12=0.1.
- **E** es el rango de la entrada analógica. 1=0-1 Volts, 2=0-5 Volts, 3=0-10 Volts.

El primer paso para salvar la configuración es obtener dicha cadena a partir de los datos en la ventana de configuración. Después de tener dicha cadena, el siguiente paso es obtener el nombre del canal al cual se le están salvando las modificaciones de su configuración. Por último, se salva toda la información en los registros de Windows.

Los cambios hechos a la configuración del canal son reflejados también en la ventana principal para su actualización.

Es importante señalar que los cambios de configuración en cualquiera de las ventanas vistas hasta aquí solo pueden realizarse cuando no se está realizando ninguna acción de captura de datos, es decir, no se puede realizar cambios a la configuración del sistema cuando se están adquiriendo datos.

4.12 CONFIGURACIÓN DE TIEMPOS

Esta sección se refiere a los tiempos que el usuario debe manejar para obtener los datos en la forma correcta. Dichos tiempos son solo dos y que anteriormente ya fueron mencionados en la sección de adquisición de datos de este capítulo, uno de ellos es la frecuencia de muestreo y el otro el periodo de promedio.

La frecuencia de muestreo no es otra cosa que el periodo de tiempo que debe transcurrir entre dos solicitudes de conversión consecutivas que el sistema le hace a la tarjeta de adquisición de datos.

El periodo de promedio se refiere al tiempo que debe transcurrir para que el sistema guarde el promedio de las muestras tomadas durante ese lapso de tiempo. Como ya se mencionó antes, a los usuarios del sistema solo les interesa obtener los promedios de las muestras tomadas.

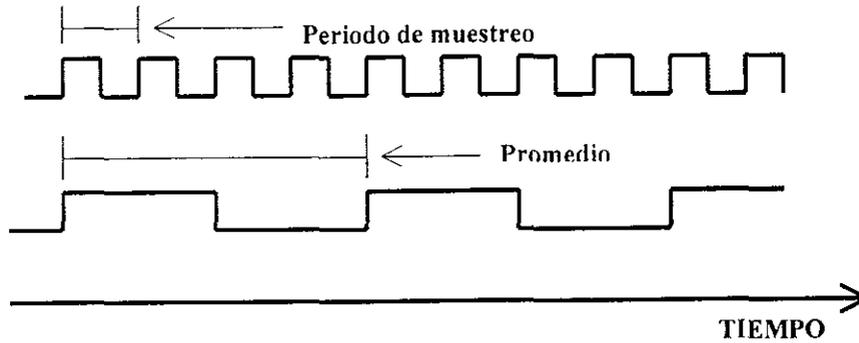


Figura 4.27. Tiempo de muestreo y de promedio.

Es obvio que ambos tiempos están íntimamente relacionados y que deben guardar una proporción ya que el tiempo de muestreo no puede ser mayor que el tiempo para obtener los promedios de los datos. La relación entre ambos tiempos se muestra en la figura 4.27.

Cabe aclarar que el periodo de muestreo no es el tiempo que la tarjeta de adquisición de datos tarda en convertir las señales de los 8 canales y transmitirlos a la computadora, sino la periodicidad con que esta se los solicita.

A solicitud de los usuarios los tiempos de muestreo, en segundos, que el usuario puede seleccionar son 1, 2, 5, 10, 20, 30, 60.

Por otro lado, los posibles periodos de muestreo, en minutos, son 1, 2, 3, 4, 5, 10, 20, 30 y 60.

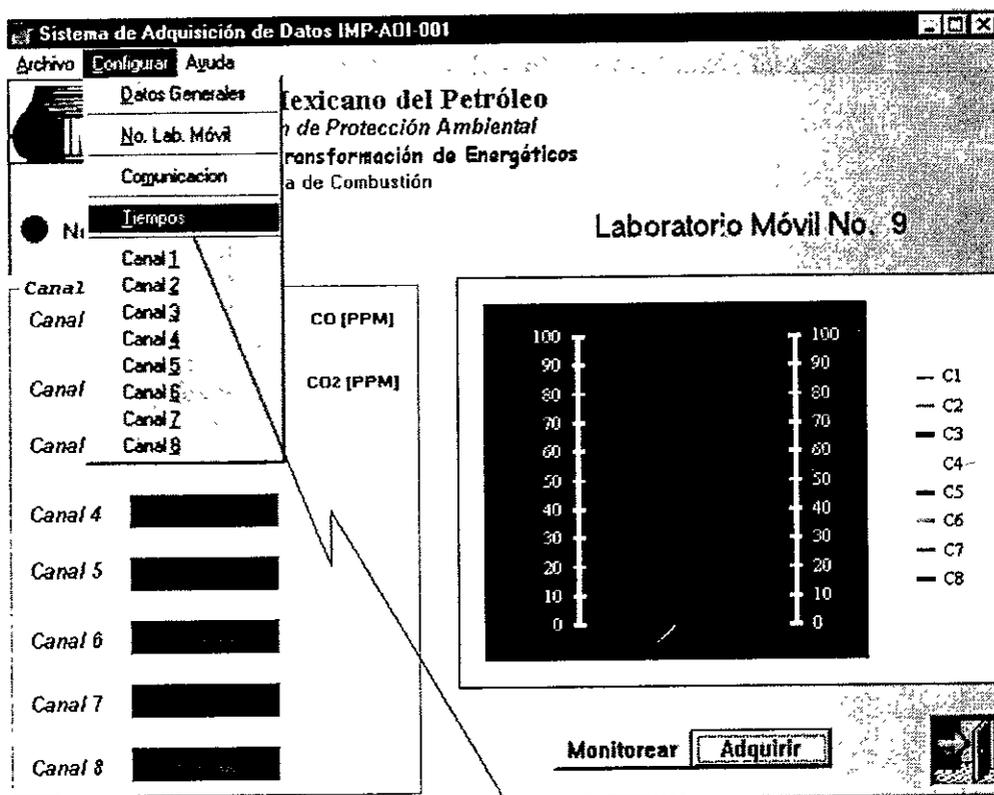
En el capítulo 1 se hizo referencia a que la frecuencia de muestreo debe tener un mínimo valor de tal forma que los datos obtenidos sean representativos. La frecuencia de muestreo debe ser como mínimo del doble de la del componente de mayor frecuencia de la señal.

Las variables que se van a monitorear, concentraciones de ellas en emisiones a la atmósfera, no tienen componentes de altas frecuencias por lo que los periodos de muestreo enumerados anteriormente son adecuados para obtener datos válidos además de que están perfectamente probados pues son los que se han venido utilizando cotidianamente en los laboratorios móviles.

Los periodos de muestreo y promedio que se utilizan cotidianamente en los análisis que realizan los laboratorios móviles son 10 segundos y 5 minutos respectivamente y los muestreos duran como máximo 2 horas.

Los otros valores para tiempos de muestreo y promedio se utilizan esporádicamente o están contemplados a utilizarse cuando la norma oficial mexicana cambie y sea más estricta.

En la figura 4.28 se puede observar el menú con el cual se puede acceder a la pantalla de configuración de tiempos.



Menú para configurar tiempos

Figura 4.28. Menú para configurar tiempos.

Cuando el usuario hace clic en el menú mostrado en la figura 4.28, aparece la ventana de configuración de tiempos. En dicha ventana el usuario escoge los valores que requiere. Los tiempos aquí programados sirven para actualizar dos variables globales, es decir dos variables cuyos valores están

disponibles para todas las partes del programa, esas variables son llamadas *muestra* y *media* y sirven para almacenar los tiempos de muestreo y promedio respectivamente.

Los últimos valores de *muestra* y *promedio* son guardados en el registro de Windows para recuperarlos cada vez que la aplicación es arrancada. Esos valores también son actualizados cada vez que el usuario los modifica. En la figura 4.29 se aprecia la pantalla de configuración de tiempos con los menores valores posibles para el tiempo de muestreo y de promedio.

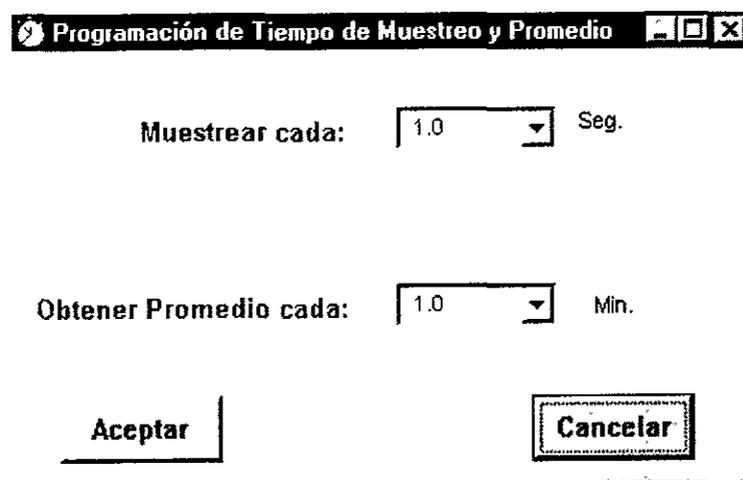


Figura 4.29. Pantalla de configuración de tiempos.

4.13 CONFIGURACIÓN DE DATOS GENERALES

El último submenú del menú configuración es el relacionado con los datos generales y se puede ver en la figura 4.30. Esta información es necesaria para cuando se va a realizar una adquisición de datos, debido a que es guardada en un archivo que identifica a la información recabada.

Son cuatro datos que el usuario debe introducir, el nombre del lugar donde se realiza la adquisición de datos, el nombre de la planta, el nombre del equipo al que se esta analizando y por ultimo el nombre del signatario u operador del sistema.

Estos cuatro datos, más la fecha en que se esta realizando el monitoreo y que es tomada del sistema operativo, son guardados en un archivo cuyo nombre es igual al que el usuario le asigna al archivo donde se salvan los datos capturados, pero con la extensión *.dat*.

La creación de este archivo es transparente para el usuario por lo que el nunca tiene que saber que existe. A este archivo se le crea con el atributo de solo lectura para que no pueda ser modificado con el fin de proteger la información.

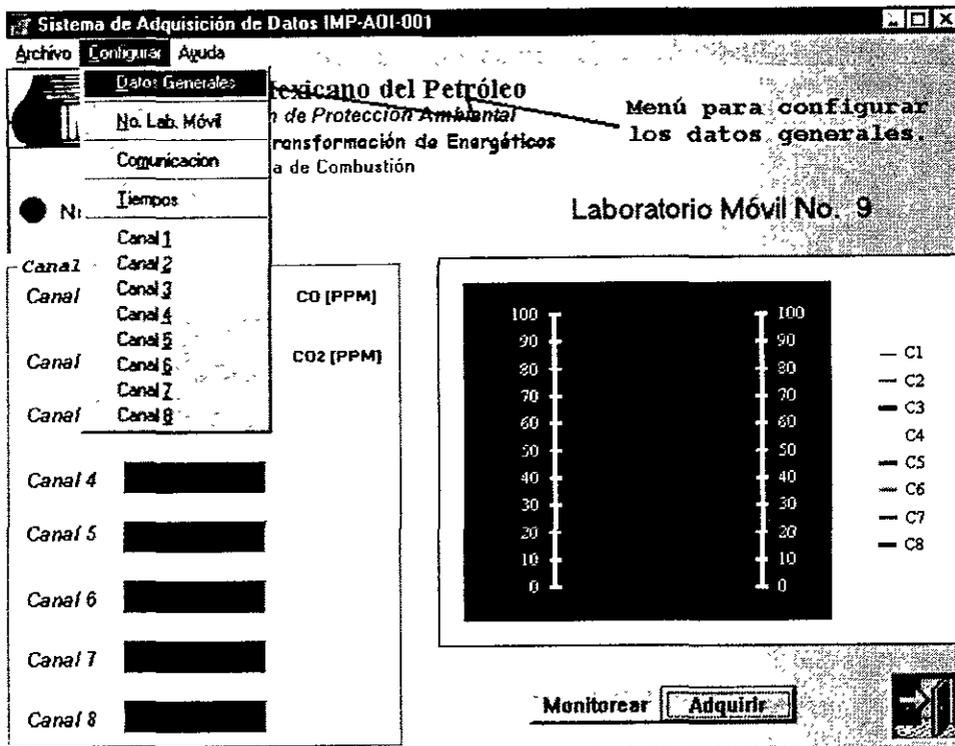


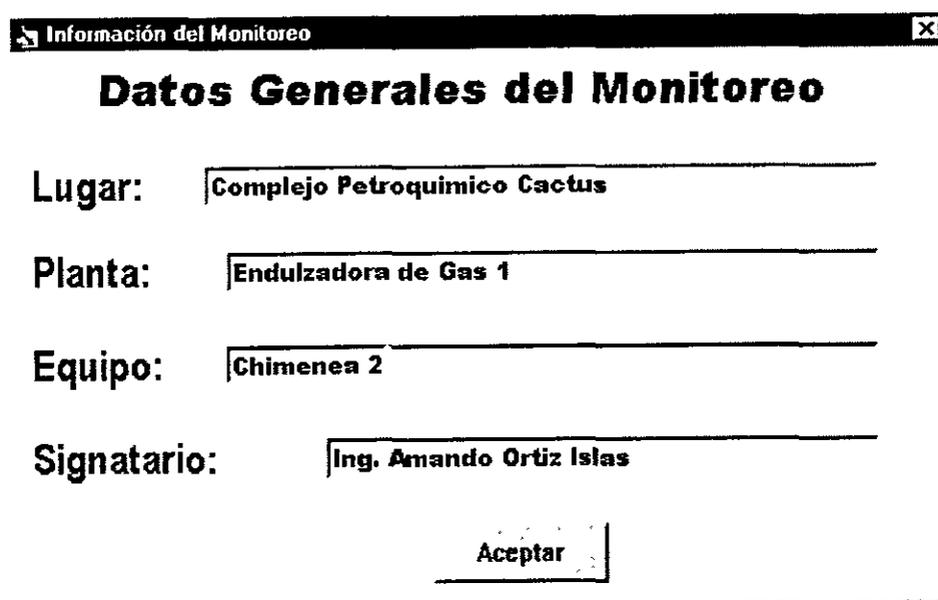
Figura 4.30. Menú de configuración de datos generales.

La información de este archivo es utilizada para desplegarla al momento de que el usuario revisa los datos capturados como se explica más adelante. Es evidente que esta información es importante para identificar los datos capturados y para evitar confusiones futuras.

La información obtenida por medio del *datalogger*, como se explicó anteriormente, no tiene ninguna identificación de este tipo, por lo que la posibilidad de confundirlos es bastante grande y ha sucedido en

la realidad. Por este motivo, los usuarios del sistema recomendaron que esta información pueda ser capturada al momento de iniciar la adquisición.

Los datos generales deben ser configurados, es decir introducidos, antes de que una adquisición pueda ser iniciada, de otro modo el sistema no lo permitirá. Esta característica se implementó así con el fin de que ninguna adquisición se realice sin antes tener la información que la identifique.



The image shows a screenshot of a software window titled "Información del Monitoreo". The window contains a form with the following fields and values:

Field	Value
Lugar:	Complejo Petroquimico Cactus
Planta:	Endulzadora de Gas 1
Equipo:	Chimenea 2
Signatario:	Ing. Amando Ortiz Islas

At the bottom right of the form is a button labeled "Aceptar".

Figura 4.31. Ventana de configuración de datos generales.

En la figura 4.31 se puede apreciar la pantalla de configuración de datos generales. En dicha pantalla el usuario solo debe teclear la información solicitada y no puede dejar ninguna en blanco. Como es obvio, el sistema no puede comprobar que los datos introducidos sean correctos ya que eso es responsabilidad directa del usuario.

Los datos mostrados en la figura 4.31 no son necesariamente correctos ya que fueron introducidos tan solo para ilustrar.

4.14 REVISIÓN DE RESULTADOS

Después de que una adquisición de datos ya fue finalizada por el usuario, este puede desear revisar la información obtenida, para hacerlo tiene que entrar a una ventana donde podrá no tan solo revisar el archivo que contenga la ultima información capturada, sino cualquiera que hubiese obtenido antes.

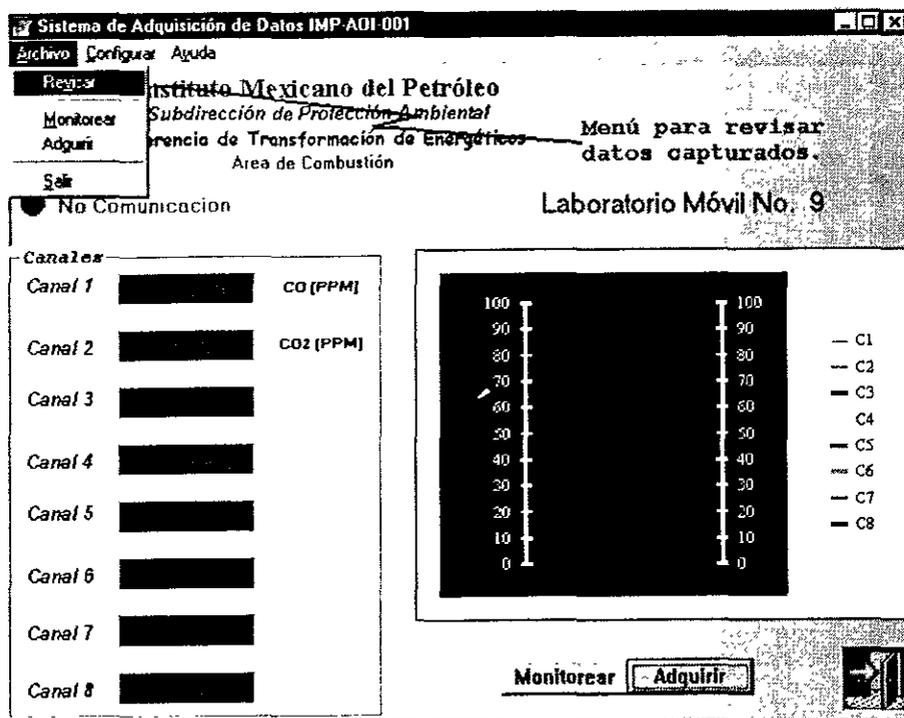


Figura 4.32. Menú para revisar datos capturados.

Para entrar a dicha ventana, el usuario tiene que usar el menú mostrado en la figura 4.32. Al hacer clic en dicho menú inmediatamente aparece la pantalla de revisión de resultados de la figura 4.33.

La información que se puede revisar en esta pantalla se resume básicamente en dos cosas, los promedios de cada variable activa relacionados con la hora del día en que fueron tomados y los datos generales que relacionan a esos promedios con el sitio a que corresponden.

En esta pantalla el usuario puede revisar los resultados de monitoreos realizados con este sistema en fechas anteriores, es decir, si hoy se genera un archivo correspondiente a una adquisición, en un futuro cualquier persona puede revisarlos nuevamente.

Como se mencionó antes, los archivos que contienen los datos capturados tienen una extensión *.adq* y son los únicos que el usuario puede revisar. Para abrir un archivo de ese tipo, el usuario solo tiene que hacer clic en el menú mostrado en la figura 4.33.

Al hacer clic en el menú mencionado, aparece un control llamado de dialogo común, que le permite al usuario navegar por todos los directorios de la computadora para abrir el archivo con extensión *adq* que desee.

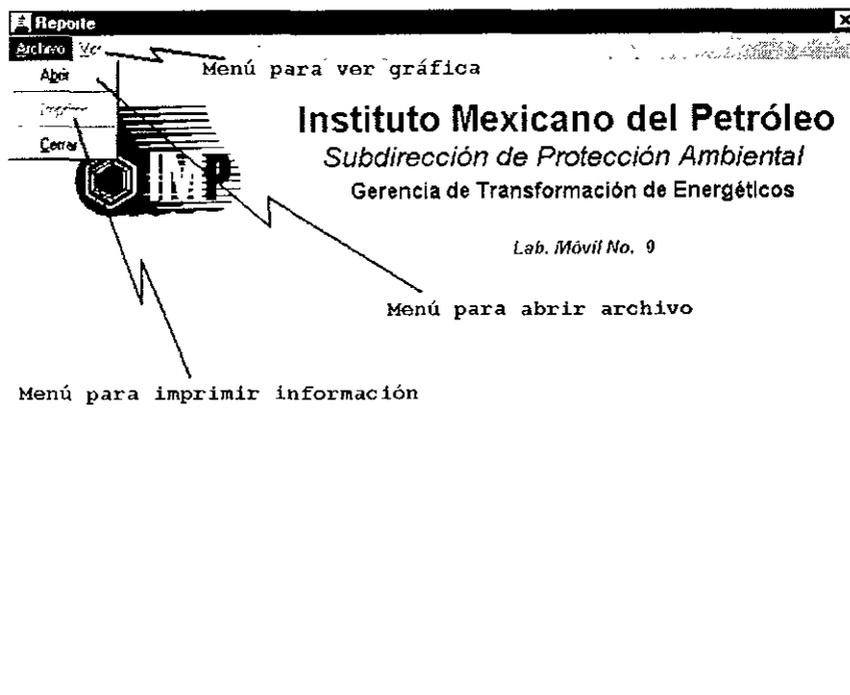
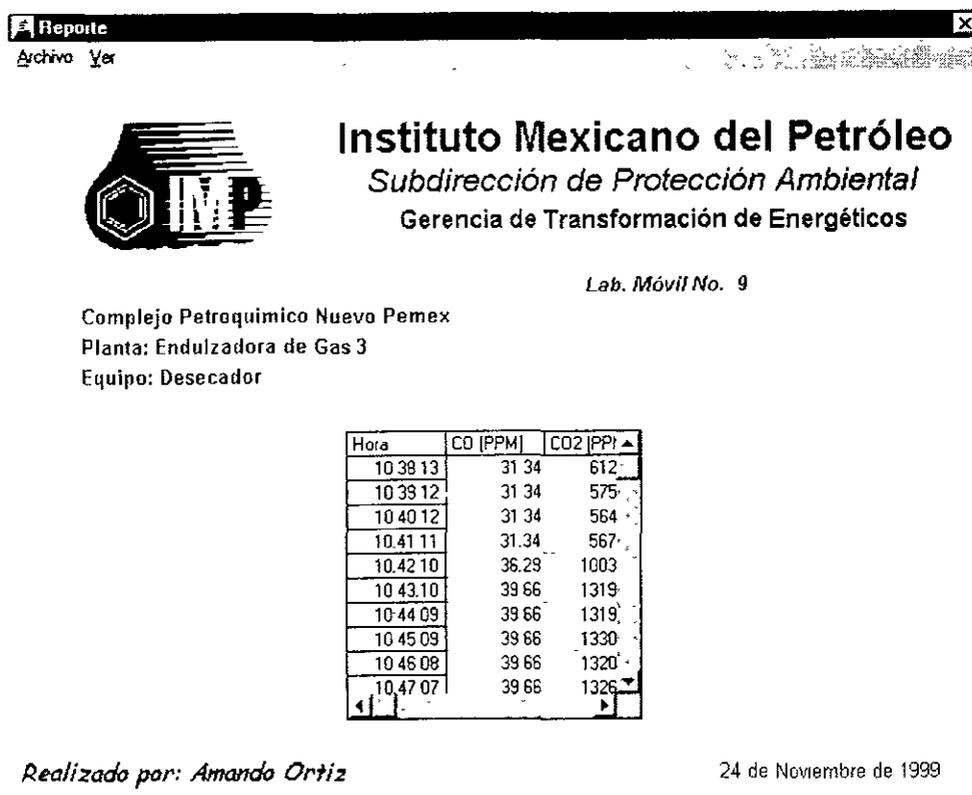


Figura 4.33. Pantalla para revisar resultados.

Después de que el usuario abre el archivo, los datos que contiene son mostrados en una tabla con los datos que lo identifican. En la figura 4.34 se muestran los datos de un archivo abierto por el usuario.

Cuando el número de datos rebasa el tamaño de la pantalla al momento de desplegarlos, la tabla automáticamente aparece con las barras de desplazamiento de tal forma que el usuario pueda moverse a través de ella.

La tabla que contiene los datos, relaciona a cada variable con su promedio para un tiempo específico y, como se puede apreciar en la figura 4.34, también se pueden visualizar los datos de ubicación, fecha y signatario de los mismos.



Instituto Mexicano del Petróleo
 Subdirección de Protección Ambiental
 Gerencia de Transformación de Energéticos

Lab. Móvil No. 9

Complejo Petroquímico Nuevo Pemex
 Planta: Endulzadora de Gas 3
 Equipo: Desecador

Hora	CO (PPM)	CO2 (PPM)
10:38:13	31.34	612
10:39:12	31.34	575
10:40:12	31.34	564
10:41:11	31.34	567
10:42:10	36.29	1003
10:43:10	39.66	1319
10:44:09	39.66	1319
10:45:09	39.66	1330
10:46:08	39.66	1320
10:47:07	39.66	1326

Realizado por: *Amando Ortiz* 24 de Noviembre de 1999

Figura 4.34. Desplegado de los datos de un archivo.

En esta pantalla se puede realizar una impresión de la tabla de datos, haciendo clic en el menú mostrado en la figura 4.33, con el fin anexarla a algún reporte o informe técnico, ya que esta información puede ser utilizada para varios fines.

Por un lado, las emisiones en fuentes fijas deben ajustarse a los límites marcados en las normas que los rigen por lo que es necesario que exista un organismo que pueda verificar y certificar los niveles de esas emisiones. El Instituto Mexicano del Petróleo cumple con esa función.

Por otro lado, los niveles de ciertos contaminantes emitidos a la atmósfera pueden ser reducidos optimizando el proceso del cual son resultado, por lo que la información capturada por el sistema de adquisición de datos también sirve para apoyar estudios y recomendaciones relacionados con estos procesos.

Por último, el Instituto Mexicano del Petróleo utiliza esta información para cuantificar las cantidades de contaminantes emitidas a la atmósfera y evaluar su posible impacto ambiental.

Como es evidente, es muy importante que la información recopilada esté completamente libre de posibles alteraciones. Con el fin de proteger la integridad de la información, la tabla no puede ser manipulada por el usuario, de esta forma no se pueden copiar ni pegar datos a la misma, y como se mencionó anteriormente, los archivos que la contienen tampoco pueden ser alterados.

Si alguna persona tiene la intención de alterar los resultados guardados en los archivos tiene que hacerlo de otra forma y será su completa responsabilidad.

Esta característica es importante debido a la costumbre de algunos usuarios de adecuar los datos capturados a ciertos requerimientos del cliente con el fin de obtener algún certificado de buenos niveles en sus emisiones. Existen sistemas de adquisición de datos comerciales que permiten editar, es decir, alterar los datos capturados.

Después de que un archivo es abierto por el usuario y la información que contiene desplegada en pantalla, el usuario puede ver una gráfica de tendencia en el tiempo de los mismos. Para poder ver la pantalla donde se muestra la gráfica solo tiene que hacer clic en el menú mostrado en la figura 4.33.

Al hacer clic en el menú mencionado, aparece una pantalla como la mostrada en la figura 4.35. Es importante volver a aclarar que tanto la tabla de datos como su correspondiente gráfica son el resultado de pruebas realizadas en el laboratorio de electrónica simulando los señales provenientes de los equipos analizadores de emisiones con fuentes variables de voltaje, por lo que las concentraciones y tendencias mostrados tanto en la tabla como en la gráfica pueden no corresponder a la realidad.

En este ejemplo solo se adquirieron datos de dos canales, uno para CO y otro para CO₂ pero como ya se ha mencionado antes, el sistema puede capturar la información de hasta 8 canales por lo que ambos, la tabla y su correspondiente gráfica, también pueden contener la información de todos esos canales.

La gráfica también puede ser impresa utilizando el menú que se aprecia en la figura 4.35. La posibilidad de imprimir esta gráfica es con el mismo fin que el caso de tabla, que puedan ser incluidas en algún reporte o informe técnico.

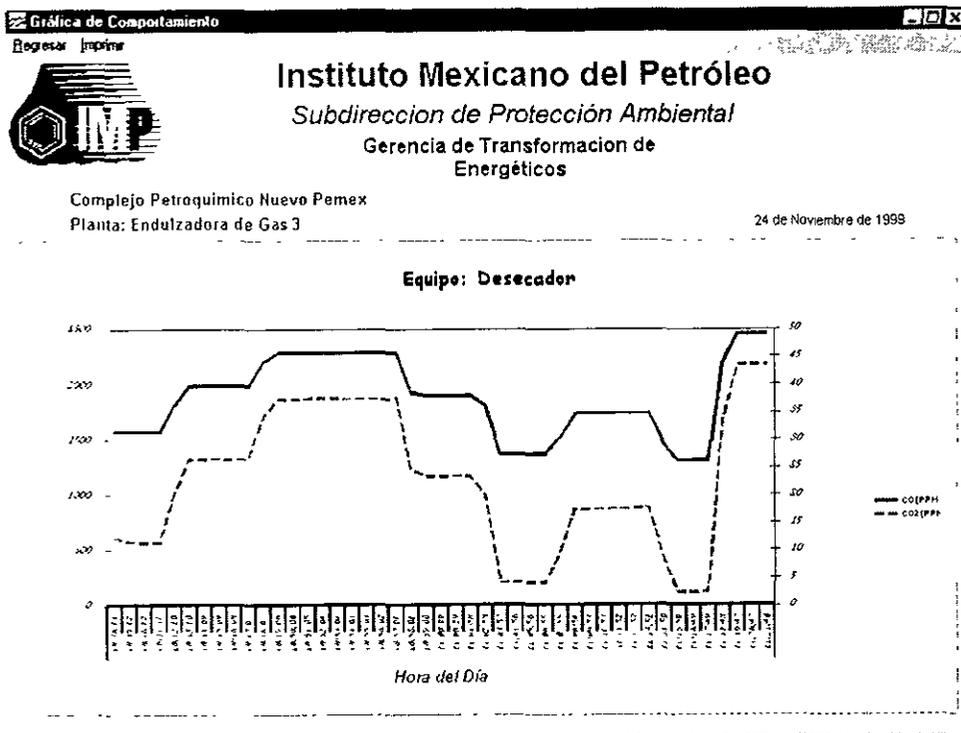


Figura 4.35. Gráfica de datos.

La impresión en *Visual Basic* es una de las tareas más complicadas debido a que, por medio del código correspondiente, el programador tiene que dibujar cada una de las líneas y círculos que aparecen en la impresión, además de posicionar una a una las imágenes y textos. Debido a esto el código resultante es bastante extenso y el proceso de impresión un poco lento pero se pueden obtener bastantes buenos resultados.

Otra opción para imprimir es el utilizar la instrucción *print.form* la cual imprime toda una pantalla incluyendo botones, menús y otros controles que aparecen en ella pero que normalmente no se desea que aparezcan en la impresión. Tampoco se pueden controlar otras características importantes de impresión por lo que las impresiones realizadas con esta instrucción dejan mucho que desear.

Probablemente la presentación de las impresiones de la tabla de datos y de la gráfica en un momento dado, *no satisfaga las necesidades de las características con que el reporte o informe técnico deba ser emitido y exista la necesidad de adecuarlo*. Como el sistema es inflexible en ese aspecto para proteger la integridad de los datos, debía existir la posibilidad de que el usuario pudiese manejar la información con otra aplicación que brindara esa posibilidad.

Para salvar esta situación, los datos son guardados en el archivo en forma tabular de tal manera que el numero de datos guardados en cada renglón corresponde al numero de canales activos, tan solo son separados por una simple coma.

Al momento de iniciar la adquisición y abrir el archivo para guardar los datos lo primero que se salva en él, es el numero de canales activos, para después en el siguiente renglón salvar el encabezado de cada columna que no es otra cosa que el nombre de la variable correspondiente a cada canal activo. A partir de aquí se comienzan a salvar los datos siempre en el mismo orden, por lo que en todos los renglones, los datos correspondientes a cada canal siempre ocupan la misma posición.

Este formato es bastante útil debido a que de esta manera el archivo puede ser abierto desde cualquier aplicación del tipo de Excel para manejar los datos en la forma que mejor convenga. Es obvio que a partir de aquí, la integridad de la información ya no puede ser garantizada debido que esta puede ser modificada, aunque los datos originales permanecen intactos en el archivo que los contiene.

Como se mencionó antes, el sistema de adquisición de datos no puede vigilar que el usuario reporte *los datos que realmente se capturaron, pero si puede salvaguardar los datos originales*. Si el reporte contiene datos diferentes a los reales, eso es mera responsabilidad de las personas que lo hacen.

4.15 LOS MODULOS ESTANDAR QUE CONTIENEN CÓDIGO.

Cada formulario del proyecto desarrollado en Visual Basic tiene asociado su propio código, el cual se ejecutará de acuerdo a los eventos que el usuario provoque. Dentro de ese código se encuentran variables y funciones que solo son útiles para algún procedimiento específico del formulario y que por lo tanto solo ocupan memoria en el momento en que el procedimiento se ejecuta.

Existen otras variables y funciones que pueden ser utilizadas en cualquier código relacionado con algún evento del formulario. Estas variables y funciones ocupan memoria todo el tiempo que el formulario permanezca cargado en ella.

Por último, existen variables y funciones que están disponibles para cualquier parte del proyecto, es decir, cualquier evento de cualquier formulario que compone al proyecto puede tener acceso a ellas. Estas variables y funciones ocupan memoria todo el tiempo que la aplicación este ejecutándose.

Como ya se explicó antes, las variables y funciones descritas en el párrafo anterior se deben colocar en los llamados módulos estándar

En el caso nuestro, el colocar ciertas funciones en estos módulos obedece a se piensa reutilizarlas en un futuro, es decir, estos módulos podrán ser agregados en proyectos futuros para utilizar el código que contienen. Esta es una gran ventaja de los programas desarrollados con *Visual Basic*.

Con relación al trabajo presentado aquí se decidió crear 4 módulos de este tipo para agrupar a las funciones que tengan algo en común. El primer modulo es el llamado *Canales* y contiene básicamente funciones utilizadas en la configuración de cada uno de los canales analógicos del sistema.

El segundo modulo se llama *Comunicaciones* y es utilizado para guardar las funciones relacionadas con el puerto de comunicación.

El tercer modulo se llama *Utilerias* y como su nombre lo indica esta formado por funciones de utileria tales como retardos, inicializaciones, etc.

Por último esta el modulo llamado *Variables*, en el cual están contenidas todas las variables globales del proyecto.

Tal vez la primera conclusión que se puede obtener al haber desarrollado este trabajo, es el hecho de que a pesar de que el campo de los sistemas de adquisición de datos ha sido ampliamente abordado, no está agotado ni lo estará en el futuro.

Por un lado, el advenimiento de nuevas tecnologías que ayudan a mejorar las características de dichos sistemas propicia que estos se encuentren en constante evolución. Un ejemplo claro lo constituyen los convertidores analógico/digital, los cuales continuamente están siendo mejorados en varias de sus características más vitales como son la velocidad de conversión y su resolución (número de bits por dato). Otro ejemplo lo constituye la rápida evolución de las computadoras que ha permitido que se desarrollen sistemas más rápidos, más confiables y más amigables.

Por el otro lado, cada día los sistemas de adquisición de datos abarcan nuevas áreas de aplicación, cada una con características propias y especiales. Este hecho significa que es sumamente difícil que un sistema de los llamados de propósito general puede ser lo suficientemente útil para todas las áreas de aplicación. Por tal motivo los sistemas desarrollados para una aplicación específica, como el presentado en este trabajo de tesis, tienen ventaja sobre los antes mencionados debido a que cumplen con todos los requerimientos del usuario y son, por lo general, de precio mucho más bajo.

La computadora ha jugado un papel muy importante en la evolución en los sistemas de adquisición de datos. Los primeros sistemas que utilizaron una computadora estaban restringidos a campos meramente científicos e industriales, los cuales eran capaces de pagar o justificar el costo monetario de dichos sistemas ya que las computadoras utilizadas no estaban al alcance de cualquiera, a pesar de que no eran muy potentes si las comparamos con las actuales.

La evolución de la computadora, que dio como resultado que se desarrollaran las de tipo personal de nuestros días, ha permitido que más gente y más campos de aplicación de los sistemas de adquisición de datos tengan acceso a ellos. El hecho de que una computadora personal pueda ser transportada a cualquier lugar y funcione sin ningún problema ha permitido el desarrollo de sistemas como el presentado aquí.

En nuestro caso, la computadora viaja con los demás instrumentos del laboratorio móvil y es llevada al sitio donde se requiera realizar un monitoreo, situación impensable en los primeros sistemas que dependían de una computadora voluminosa y delicada.

Las computadoras de tipo personal ya son parte de muchos sistemas de adquisición de datos principalmente por dos razones, por un lado la potencia en cuanto a velocidad de procesamiento y capacidad de almacenamiento y por el otro, su costo está al alcance de mucha gente. En consecuencia, existen ya sistemas de adquisición de datos que utilizan una computadora personal para aplicaciones estrictamente domésticas.

Otra de las áreas relacionadas con las computadoras y que tiene un impacto importante en los sistemas de adquisición de datos es el software. Por un lado, los sistemas operativos han evolucionado de tal forma que actualmente se han impuesto los ambientes visuales tales como Windows. Este tipo de sistemas tiene la característica de que son bastante atractivos para los usuarios, es decir, se les puede dar una apariencia agradable a la vista sin que el sistema pierda su funcionalidad, sino por el contrario.

Por otro lado, se han producido muy buenas herramientas especiales de software para desarrollar aplicaciones que involucren adquisición de datos. Esas herramientas permiten desarrollar aplicaciones bastante poderosas y confiables, un ejemplo de ese tipo de software es el ya mencionado *LabView* de National Instruments.

Lo anterior no implica que no se puedan utilizar lenguajes de programación de propósito general para desarrollar estas aplicaciones, ya que también este tipo de software ha evolucionado de tal forma que su campo de aplicación es muy variado y casi ilimitado y lo más importante, permite implementar aplicaciones con bastante calidad.

En el caso del trabajo presentado aquí, se utilizó esta última opción con buenos resultados. Aunque probablemente se invirtió más tiempo en el desarrollo del sistema, el costo del software utilizado para este fin, la versión 5 de *Visual Basic* Profesional, es mucho menor que la de *LabView*. Mientras el primero tiene un costo de 800 dólares para el segundo es de aproximadamente 8000.

Algo importante de mencionar, es el hecho de que en la actualidad el desarrollo de aplicaciones debe ser rápido debido a que en un lapso muy corto de tiempo pueden pasar a ser obsoletas pues las grandes compañías productoras software, Microsoft por ejemplo, con intereses meramente mercantiles actualizan sus programas arrastrando con ellos a los demás.

Otra conclusión que se puede obtener es que es difícil hablar de sistemas de adquisición de datos de propósito general, ya que cada una de las señales de entrada necesita ser tratada e interpretada de manera diferente, dependiendo del instrumento de donde provenga, es decir un nivel de voltaje de 500 mV no significa lo mismo si proviene de la salida analógica de un espectrofotómetro, de un potenciómetro o de un analizador de emisiones contaminantes.

La captura de datos de provenientes de diferentes tipos de instrumentos implica que cada uno de ellos deba ser tratado de diferente manera tanto en la frecuencia de muestreo como en el manejo matemático de los datos mismos.

Además de que el software debe ser capaz de manejar la información capturada para que sea útil al usuario, el hardware juega un papel importante ya que debe tener la posibilidad de trabajar de la manera más óptima todos los aspectos de la señal que esta capturando y manipulando.

Se han desarrollado sistemas de adquisición de datos comerciales que intentan tener la característica de ser de aplicación general pero la mayoría de ellos no ha tenido éxito debido a sus limitaciones, de tal forma que no se pueden obtener versiones actualizadas de ellos, un buen ejemplo es el sistema que incluye una tarjeta de adquisición de datos de 8 bits y el software llamado *LabTech Notebook* con un costo de catálogo de aproximadamente 3000 dólares.

Lo expuesto en estos últimos párrafos son algunas razones importantes de que, como se apuntaba antes, a pesar de que los sistemas de adquisición de datos son un tema extensamente tratado, aun se siguen, y seguirán siendo desarrollando trabajos al respecto.

Con relación a la situación del desarrollo de sistemas de adquisición de datos en México, la realidad es que prácticamente la oferta de estos en el mercado es nula, es decir, prácticamente no existen sistemas de adquisición de datos (Hardware y Software) desarrollados en el país y que se oferten comercialmente. Todos los sistemas que son ofertados en México son sistemas de origen extranjero, principalmente estadounidense.

Como se apunta en el capítulo de introducción, en la UNAM existen poco más de 60 trabajos relacionados con la adquisición de datos, pero solo los más recientes incluyen una computadora en el sistema. Todos estos trabajos de tesis son de una aplicación específica [14-21].

En otras instituciones educativas del país también se están desarrollando sistemas de adquisición de datos, en el Instituto Politécnico Nacional [11] y en la Universidad de las Américas [8], pero a un nivel meramente experimental y con fines pedagógicos.

Con respecto a sistemas de adquisición de datos para el monitoreo de emisiones como el desarrollado en este trabajo, en el mercado nacional e internacional prácticamente no existen, debido a que por un lado este campo de trabajo es relativamente nuevo y por el otro, en el mundo los clientes potenciales no son muchos. La tendencia con este tipo de sistemas es desarrollarlo junto con la implementación del laboratorio móvil o del sistema de equipos analizadores que un cliente quiera implementar.

Como ejemplo, esta es la situación de los nuevos laboratorios móviles que se adquirieron recientemente en el Instituto Mexicano del Petróleo. Estos laboratorios ya cuentan con un sistema de adquisición de datos desarrollado por la misma compañía que implementó los laboratorios, pero el costo del sistema de adquisición de datos incluido, hardware y software, fue bastante elevado, \$ 200 000.00 M.N. por cada uno. Estos sistemas ni siquiera fueron desarrollados por alguna compañía mexicana sino que se desarrollaron en España. Como es obvio, la experiencia y la ganancia económica no impactan positivamente en nuestro país.

El costo para desarrollar el prototipo del sistema que se explica en este trabajo de tesis fue de aproximadamente \$ 3 000.00 M.N. para el material necesario para el hardware, más las horas hombre necesarias para desarrollar la aplicación y la tarjeta de adquisición de datos. Es obvio que los subsecuentes sistemas tendrían un costo mucho más bajo, unos \$ 5 000.00 por cada sistema a precios actuales.

Como valor agregado quedara la experiencia de desarrollar sistemas de este tipo lo cual permitiría que en un futuro se pueda reutilizar tanto el software como el hardware para otras aplicaciones similares con sus respectivas modificaciones.

También se puede concluir que es necesario que en México se siga explorando en esta área para no seguir dependiendo de la tecnología desarrollada en el extranjero tal como en el ejemplo anterior.

En relación con el trabajo de tesis presentado aquí, este puede servir como base para otros sistemas cuya orientación sea diferente a la captura de datos de tipo ambiental.

Uno de los aspectos que pueden ser mejorados en cuanto a la tarjeta de adquisición de datos es el relacionado con los rangos de trabajo. Si se necesitara manejar otros rangos diferentes a los que maneja la tarjeta, lo ideal es implementar una ganancia variable y programable.

Esto significa que debido a que no existe un estándar, nacional ni internacional, con relación a las características que deben de tener las salidas de voltaje analógicas de los diferentes equipos de análisis químico, pero que la gran mayoría de ellas cae en el rango 0 – 10 volts, se puede hacer una tarjeta de adquisición de datos que pueda escalar cualquier salida de voltaje analógico que se encuentre dentro de ese rango. Por ejemplo la mayoría de potenciómetros tienen una salida de ± 200 mV, entonces una futura tarjeta debe ser capaz de escalar esa señal de tal forma que -200 mV sea la escala mínima y $+200$ mV sea la escala máxima para obtener la mejor resolución.

Otro rango que puede implementarse en la tarjeta de adquisición de datos es el muy conocido 4 – 20 mA el cual, como se menciono en otros capítulos, es muy utilizado en equipos de análisis que se encuentran localizados en lugares remotos debido a su alta inmunidad al ruido en distancias grandes.

La tarjeta de adquisición de datos presentada en este trabajo solo maneja señales analógicas de voltaje mientras que la mencionada en el párrafo anterior es una señal de corriente.

Es evidente que es necesario que exista un estándar que regule las salidas analógicas de los equipos analíticos, probablemente en un futuro no muy lejano alguna institución lo creará. Cuando ese estándar exista muchos problemas relacionados con el escalamiento desaparecerán.

Cabe mencionar que recientemente se ha creado la *Open Data Acquisition Association* (ODAA por sus siglas en inglés) con la finalidad de encontrar la forma de que exista interoperabilidad real entre el hardware y software de los sistemas de adquisición de datos basados en PC, lo cual resolvería el problema mencionado en el párrafo anterior.

También en cuanto al hardware, es posible mejorar otras características, por ejemplo se puede experimentar con velocidades de conversión mayores lo cual permitiría monitorear señales de otro tipo.

Las entradas analógicas estan configuradas en un modo no diferencial, pero en versiones posteriores se puede llegar a implementar el modo diferencial el cual tiene una mayor inmunidad al ruido.

Se puede implementar también una etapa de aislamiento ya sea óptico o magnético que permita trabajar en ambientes con un nivel de voltaje de modo común alto.

En relación con el software, se puede implementar un control que permita mostrar las gráficas de tendencia de las variables de manera más eficiente y clara. Un control de este tipo que maneje de manera más eficiente las gráficas, especialmente en tiempo real, permitirá que la aplicación corra más rápido cuando este monitoreando todos los ocho canales.

También se puede implementar un filtro que permita convertir directamente los archivos generados por la aplicación y que contienen los datos tabulados para cada variable, a archivos de tipo Excel o cualquier hoja electrónica para su mejor manejo y presentación. Como es obvio, el tener esta opción crea la posibilidad de que los datos puedan ser manipulados y que los reportes finales contengan información no correcta de tal forma que no se realicen las correcciones adecuadas para mejorar la calidad de las emisiones.

Otra opción que puede desarrollarse en un futuro es el enlazar los datos obtenidos en los monitoreos con una base de datos que permita en un momento dado manejar la información de manera más eficiente y global.

EL REGISTRO DE WINDOWS 95 Y 98

El registro puede considerarse el corazón de Windows 95/98 debido a que todo el software y hardware que se instala en la computadora lo actualizan o modifican. Las rutinas de instalación de las aplicaciones, son las encargadas de realizar esos cambios.

El registro de Windows 95/98 puede ser modificando utilizando el programa *regedit.exe* pero no es recomendable que el usuario lo haga sin conocimiento, debido a que el resultado puede ser bastante catastrófico pues la computadora puede resultar incluso inoperante.

El registro es utilizado a partir de Windows 95 [87] en lugar de los archivos *INI* para guardar la información relacionada con la configuración del sistema, del hardware y de las aplicaciones de 32 bits.

Los archivos del registro son ocultos y por tal razón no aparecen en la carpeta de Windows. Para acceder a ellos, como ya se menciono antes, es necesario correr un programa especial.

El registro está formado por dos archivos ocultos que se encuentran en el directorio *Windows*, esos archivos son llamados *USER.DAT* y *SYSTEM.DAT*. El primero contiene información que es única para cada persona que utiliza el sistema. El segundo contiene la información relacionada con la configuración global de la computadora. Ambos archivos son de solo lectura, ocultos y de sistema y son puestos aparte de los archivos normales como una forma de protegerlos de un posible borrado.

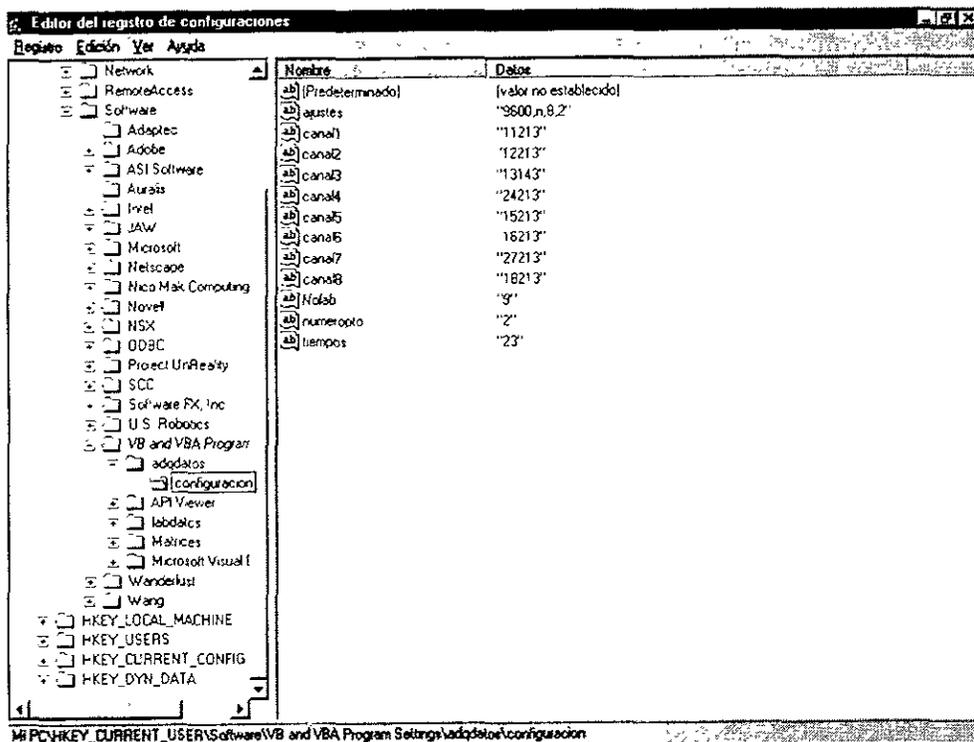


Figura A. El registro de Windows 95/98.

El registro esta estructurado jerárquicamente tal como los directorios de un disco duro. Cada rama es marcada, en la ventana de edición del registro, con el icono de una carpeta y se le conoce como Clave(Key). Cada clave puede contener a otras claves llamadas Valores(Values). Cada valor contiene la información actualizada que se ha guardado en el registro. Esto se puede apreciar claramente en la figura A.

Los Valores pueden ser de tres tipos: cadena (*string*), binario (*binary*) y *DWORD*.

En el registro existen seis ramas principales cada una de las cuales contiene información específica guardada en él. Esas ramas principales son:

- **HKEY_CLASSES_ROOT** la cual contiene todos los tipos de archivos presentes en la computadora además de la información *OLE* para las aplicaciones que lo utilicen.
- **HKEY_CURRENT_USER** esta rama apunta a la parte de *HKEY_USERS* apropiada para el usuario actual.

- **HKEY_LOCAL_MACHINE** contiene la información relacionada con todo el hardware y software instalado en la computadora. Debido a que en un momento dado se pueden especificar varias configuraciones de hardware, la configuración de hardware actual se encuentra en la rama **HKEY_CURRENT_CONFIG**.
- **HKEY_USERS** en esta rama se guardan ciertas preferencias de los usuarios con relación a colores y otras características del panel de control para cada uno de ellos. Aunque Windows 95 puede ser configurado para varios usuarios (el usuario actual es especificado en **HKEY_CURRENT_USER**), usualmente solo hay un usuario llamado **default**.
- **HKEY_CURRENT_CONFIG** esta rama apunta a la parte de **HKEY_LOCAL_MACHINE** que contiene la configuración de hardware actual.
- **HKEY_DYN_DATA** apunta a la parte de **HKEY_LOCAL_MACHINE** utilizada con la característica de Plug & Play de Windows 95/98.

Debido a la importancia de los archivos *USER.DAT* y *SYSTEM.DAT* siempre que Windows 95/98 es apagado el sistema crea dos archivos de respaldo automáticamente. Esos archivos llamados *USER.DA0* y *SYSTEM.DA0* sirven para que Windows 95/98 reemplace a los originales en caso de que se encuentren corrompidos.

Aunque Windows 95 hizo a un lado los archivos *INI* por el registro, todavía soporta aplicaciones que utilizan esos archivos, particularmente las aplicaciones basadas en Windows de 16 bits. Los archivos *AUTOEXEC.BAT* y *CONFIG.SYS* aun existen por compatibilidad con componentes del sistema en modo real y para permitirle al usuario poder cambiar ciertas configuraciones que originalmente no puede hacerse.

El uso de los registros por Windows 95/98 proporciona los siguientes beneficios:

- Los datos para enumerar y configurar el hardware, las aplicaciones, los manejadores de dispositivos y parámetros de control del sistema operativo provienen de una sola fuente. La configuración puede ser recobrada fácilmente en un evento que produzca una falla del sistema.
- Los usuarios y administradores pueden cambiar la configuración de la computadora utilizando las herramientas del panel de control estándar y otras herramientas administrativas reduciendo la probabilidad de errores de sintaxis en la información de configuración.
- Un conjunto de funciones independientes de red puede ser utilizado para poner e indagar información relacionada con la configuración, permitiendo a los administradores del sistema examinar los datos de configuración por medio de computadoras remotas conectadas en red.

- El sistema operativo automáticamente respalda la última información correcta utilizada para inicializar la computadora.

Aprovechando las características de Windows 95/98 en lo relacionado a su registro, en el software aquí desarrollado se utilizaron las instrucciones de *Visual Basic* que permiten guardar la configuración del sistema de adquisición de datos. En la figura A se muestra, como ejemplo, la información ya guardada en el registro.

1. CMB Control SA de CV
Ejercito Nacional 1112, despacho 1102
México, D.F.
Tel 5 580 6959
2. Ludy, Timothy
Interoperability comes to data acquisition
Sensors
October 1999
3. Derenzo, Stephen E.
Interfacing: A laboratory approach using the microcomputer for instrumentation, data analysis and control.
Prentice Hall 1990
4. Miner, Gayle F.
Physical data acquisition for digital procesing.
Prentice Hall 1992
5. Carr, Joseph J.
Microcomputer interfacing: A parctical guide for technicians, engineers and scientist.
Prentice Hall 1991
6. *Avionics: Airborne Application Employs Lightweight, Notebook-PC-Based Data Acquisition System.*
Application Note # 12
Iotech Inc
August 1993
7. Kroening, Mike and Hite, Jacob
A Velocity sensor for rotating shafts.
Sensors Magazine
September 1998
8. Castañón Niconoff, Iván y Alarcón Aquino, Vicente
Sistema de adquisición de datos a través de internet usando Java.
Universidad de las Americas campus Puebla
<http://www.pue.udlap.mx/~electro/REDES/sad/representacion/index.htm>
9. Gonzalez S, Antonio
Estudio funcional de tres diferentes sistemas valvulares y diseño de una nueva valvula de derivación de liquido cefalo-raquideo en el tratamiento de la hidrocefalia.
Universidad Pontificia Bolivariana
<http://www.goecites.com/HotSprings/Falls/3749/index.html#m4>
10. *SITES-Sistema Integral para Telecomando de Edificios y Servicio.*
Instituto de Microelectrónica de Sevilla
Centro Nacional de Microelectrónica
<http://www.imse.cnm.es/projects.htm>
11. *Desarrollo de un sistema de captura de señales fisiológicas.*
Escuela Superior de Computo
Instituto Politecnico Nacional
<http://www.escom.ipn.mx/proyecto/gener2.html>

12. *Online data acquisition and remote control*
Illinois Institute of Technology
<http://weblab1.uit.edu>
13. Lee, Ho-Young; Hyun, Byung_Koo; Kong, Young-Sae.
PC based acquisition and processing of high resolution marine seismic data.
Geophysics
Vol 61, No. 6 (Nov-Dec 1996)
14. Portugal Antunduaga, Juan Gonzalo
Diseño del equipo de adquisición de datos y controles de servos para una aeronave no tripulada.
UNAM Facultad de Ingeniería 1992
15. Cervantes Franco, Alejandro
Desarrollo de un sistema de adquisición de datos para un espectrofotómetro.
UNAM Facultad de Ciencias 1993
16. Vázquez Angeles, José Antonio y Reséndiz Martínez, Juan José
Adquisición y tratamiento de datos experimentales de fenómenos físicos por medio de la microcomputadora.
UNAM FES-Cuautitlán 1995
17. Sanchez Terreros, Jorge
Automatización de la adquisición de datos de la cámara de curado seco del Instituto de Ingeniería.
UNAM Facultad de Ingeniería 1996
18. Feria Martínez, Consuelo
Diseño e implantación de una red de area local para el control estadístico de proceso con adquisición de datos en tiempo real.
UNAM Facultad de Ingeniería 1997
19. Piedragil Rojas, Alma Judith
Desarrollo de un programa para la adquisición directa de datos para análisis de gasolina por componentes.
UNAM ENEP Aragón 1999
20. Miranda Centeno, Victor Manuel
Diseño y construcción de un sistema de adquisición de datos para detección de orientación y control de estabilización de satélite.
UNAM FES Cuautitlán 1999
21. Garmilla Rojon, José Arturo
Integración del sistema de control supervisorio y adquisición de datos (SCADA) y el sistema de información geográfica (SIG).
UNAM FES Cuautitlán 1999
22. Grier, Richard.
Visual Basic programmer's guide to serial communication
Mabry 1997
23. Kruglinski, David J.
Programación avanzada con Visual C++ 5.
Mc Graw Hill 1998

24. *Communication-Based data acquisition systems.*
Complete data acquisition & computer interface handbook & encyclopedia.
Vol 28
Omega Engineering
25. Mitchell, Ross
Introduction to the HC16 for HC11 users
Application Note 461
Motorola Semiconductor
26. *Designing a data acquisition and control system*
Tech Tutorial
CyberResearch Inc. 1998
27. Putnam William and Knapp, R. Benjamin
Sensors course
Introduction
Stanford University 1996
28. Potter, David
Entering to age of smart distributed I/O.
Sensors Magazine
Mayo 1998
29. Logan, Leslie
Are you on the right bus?
Sensors Magazine
Agosto 1997
30. Baker, L. and Wong, C.A.
Understanding the basics of portable data acquisition
Sensors Magazine
Noviembre 1997
31. Ramsden, Ed
The sensors glossary of data acquisition and signal processing.
Sensors Magazine
Data acquisition special issue 1998
32. *Input resolution*
Analog Inputs
Introduction to PC-Based data acquisition
Data acquisition catalogue
Vol 28
Keithley Metrabyte
33. *Input accuracy*
Analog Inputs
Introduction to PC-Based data acquisition
Data acquisition catalogue
Vol 28
Keithley Metrabyte
34. Sraff, Fred R.
Defining data accuracy
Sensors Magazine

Agosto 1998

35. *Accuracy*
Chapter 2: Analog to digital conversion.
Signal Conditioning & PC-Based Data Acquisition Handbook
Iotech Inc.
36. *Maximum sampling rate*
Analog Inputs
Introduction to PC-Based data acquisition
Data acquisition catalogue
Vol 28
Keithley Metrabyte
37. Wilson, Dave.
Data acquisition in the fast lane.
Sensors Magazine
Data acquisition special issue 1998
38. Pang, Chao-Sun.
Technology trends in digital instrumentation recorders.
Scientific Computing & Instrumentation
Enero 1999
39. *Input range.*
Analog Inputs
Introduction to PC-Based data acquisition
Data acquisition catalogue
Vol 28
Keithley Metrabyte
40. *TU-Transmitter DAT12*
Technical reference
Vaisala 1996
41. *Wind Transmitter*
Technical reference
Vaisala 1996
42. Stitt, R. Mark and Kunst, David.
4-20mA to 0-20mA converter and current summing current to current converter.
Application bulletin
Burr-Brown
43. Stitt, R. Mark and Kunst, David
0 to 20mA receiver using RCV420
Application bulletin
Burr-Brown
44. Shannon, Claude E. and Weaver, Warren
Mathematical Theory of communication.
University Illinois Press 1963
45. Smith, Strether
The DA time bomb
Sensors Magazine

Data acquisition special issue 1998

46. Lankford, Dale.
Defusing de DA time bomb.
Sensors Magazine
Data acquisition special issue 1998
47. *Aliasing*
Analog Inputs
Introduction to PC-Based data acquisition
Data acquisition catalogue
Vol 28
Keithley Metrabyte
48. Gaddy, Larry
Selecting an AD converter
Application bulletin
Burr-Brown
49. Baker, Bonnie C.
Consider the options before tackling the mixed signal design.
Sensors Magazine
Data acquisition special issue 1998
50. Ramsden, Ed
Data acquisition alphabet soup.
Sensors Magazine
Data acquisition special issue 1998
51. *Data transfer modes.*
Analog Inputs
Introduction to PC-Based data acquisition
Data acquisition catalogue
Vol 28
Keithley Metrabyte
52. Schraff, Fred.
Choosing differential or single-ended measurements for data acquisition systems.
Sensors Magazine
Diciembre 1997
53. Nash, Eamon.
A practical review of common mode amplifiers.
Sensors Magazine
Julio 1998
54. Cooper, Dave.
Data acquisition tips.
EDN products edition
Septiembre 1996
55. *The ins and outs of isolation: a guide to selecting the right isolator.*
Technical tips
Acromag 1998

56. *Introduction to data acquisition systems.*
Complete data acquisition & computer interface handbook & encyclopedia.
Vol 28
Omega Engineering
57. *RS 232/RS 485/RS 422 specifications and wirings.*
Integrity Instruments 1999
58. *Parallel port background.*
Introduction to the IEEE 1284 parallel port standard
Warp Nine Engineering 1998
59. Goldie, John
Summary of well known interface standards.
Application note 216
National Semiconductor Corp 1996
60. Carballar, José A.
El libro de las comunicaciones del PC.
Computec-Rama 1997
61. *Fundamentals of RS-232 serial communications.*
Application note 83
Dallas Semiconductor
62. *RS422, RS423, RS449 – A compatible improvement of RS232C.*
RAD Communications Ltd. 1998
63. *Guide to RS422 and RS485*
KK Systems Ltd. 1998
64. *RS-422, RS-485, V.11 and other balanced interfaces.*
Sangoma Technologies Inc 1999
65. Norma oficial mexicana NOM-085-ECOL-1994
Norma oficial mexicana para fuentes fijas que utilizan combustibles fósiles líquidos o gaseosos o cualquiera de sus combinaciones.
66. Norma oficial mexicana NOM-AA-35-1978
Determinación de bioxido de carbono, monóxido de carbono y oxígeno en los gases de combustión.
67. Norma oficial mexicana NOM-034-ECOL-1993
Métodos de medición para determinar la concentración de monóxido de carbono en el aire ambiente y los procedimientos para la calibración de los equipos de medición.
68. Norma oficial mexicana NOM-036-ECOL-1993
Métodos de medición para determinar la concentración de ozono en el aire ambiente y los procedimientos para la calibración de los equipos de medición.
69. Norma oficial mexicana NOM-037-ECOL-1993
Métodos de medición para determinar la concentración de bióxidos de nitrógeno en el aire ambiente y los procedimientos para la calibración de los equipos de medición.
70. Norma oficial mexicana NOM-038-ECOL-1993
Métodos de medición para determinar la concentración de bióxidos de azufre en el aire ambiente y los procedimientos para la calibración de los equipos de medición.

71. *M68HC11 Reference manual.*
Motorola Inc. 1991
72. *MC68HC11F1 Technical data.*
Motorola Inc. 1991
73. *ADC1241 Self-Calibrating 12-Bit plus sign μ P-Compatible A/D converter with sample and hold.*
National data acquisition databook.
National Semiconductor 1995
74. *DG506A, DG507A, DG508A, DG509A CMOS analog multiplexers.*
Data Sheet
Intersil Corp.
Noviembre 1999
75. *LF155/LF156/LF157 Series monolithic JFET input operational amplifiers.*
National Operational Amplifiers Databook.
National Semiconductor 1995
76. *OP07 Low offset voltage operational amplifier.*
Data sheet
Maxim Integrated Products 1995
77. *Simultaneous Sample and Hold (SSH)*
Analog Inputs
Introduction to PC-Based data acquisition
Data acquisition catalogue
Vol 28
Keithley Metrabyte
78. Clarkson, Mark.
The changing role of software
Sensors Magazine
Marzo 1998
79. *Instrumentation Reference and Catalogue.*
National Instruments 1997
80. *Data acquisition & communication Catalog.*
Iotech Inc. 1998
81. Cornell, Gary.
Manual de Visual Basic 4 para Windows 95.
Osborne Mc Graw Hill 1996
82. *Visual Basic 5 Manual del Programador.*
Microsoft 1997
83. Brown, Kenion.
Como Usar Visual Basic
Megabyte 1992
84. *Visual Basic 5 Guia de herramientas y componentes.*
Microsoft 1997

- 85. Parthasarathy, Murali.
Using ActiveX to share data among applications.
Sensors Magazine
Febrero 1998
- 86. *Oxygen Analysers, 7000 series.*
Operating Manual
Telegan Gas Monitoring Limited.
- 87. Prosize, Jeff.
Exploring the Windows 95 Registry
PC Magazine
21 de noviembre de 1995

OTRAS REFERENCIAS

- 88. C. D. Spencer and S. R. Paul
Hardware and software for a pulse height analyzer linked to a personal computer.
Computers in physics
Vol 11, No. 1, Jan/Feb 1997
- 89. Patrick H. Garret
Advanced instrumentation and computer I/O design.
Real-time systems computer interface engineering.
IEEE Press 1994
- 90. J. Cambell
C programmer's guide to serial communications.
SAMS PUBLISHING
- 91. T. S. Monk
Windows Programmer's: guide to serial communications.
SAMS PUBLISHING 1993
- 92. Michel Holmes and Bob Flanders
C++ communication utilities
PC Magazine
- 93. Floy, Michael.
Developing Visual Basic 4 communications applications.
Coriolos 1996
- 94. Hebert, Douglas A.
Visual Basic Programming with applications.
Bantam 1991
- 95. Brown, Kenyon
Programmer's introduction to Visual Basic.
SYBEX 1992
- 96. Brown, Kenyon
Como usar Visual Basic.
Megabyte 1992

97. Halvorson, Michel.
Microsoft Visual Basic 5.
Mc Graw Hill 1997
98. Mc Keown, Patrick G.
Learning to program with Visual Basic.
J. Wiley 1998
99. Amundsen, Mike.
Visual Basic 5 Fundamentals, unleashed.
SAMS PUBLISHING 1997
100. Ceballos Sierra, Francisco Javier.
Enciclopedia de Visual Basic
RA-MA
101. Schildt, Hebert.
C: Guía para usuarios expertos.
Mc Graw Hill 1989
102. Chesney, Chad.
OPC vs. ODAS
Sensors Magazine
Noviembre de 1999
103. Putnam, Frederick A.
Internet –Based data acquisition and control.
Sensors Magazine
Noviembre de 1999
104. Clarkson, Mike.
Designing the noise out of DA boards.
Sensors Magazine
Data acquisition special issue 1998
105. Khalid, Muneeb.
Working at high speed: multimegahertz 16-bit A/D conversion.
Sensors Magazine
Mayo 1998
106. Fuller, Bruce.
Data loggers: the next generation.
Sensors Magazine
Febrero 1998
107. Howarth, David.
Performing data acquisition over the internet.
Sensors Magazine
Enero 1998