

TEMA 3: ADQUISICIÓN DE DATOS

3.1.- INTRODUCCIÓN

Para llevar a cabo un control eficaz sobre un determinado proceso, se hace necesario captar una serie de datos para, posteriormente, analizarlos, tratarlos, almacenarlos y llevar a cabo una presentación clara y eficaz de la evolución de dicho proceso. Generalmente, los datos o variables que se han de captar tienen un carácter analógico, mientras que su tratamiento, almacenamiento y análisis son mucho más eficaces cuando se hace digitalmente. Esto implica una serie de *módulos electrónicos* que permitan llevar a cabo una transformación de los datos desde el campo analógico al campo digital, sin que por ello se deban perder aspectos fundamentales para el proceso que se desea controlar.

Al conjunto de los diferentes módulos electrónicos que permiten llevar a cabo la transformación anterior se le denomina Sistema de Adquisición de Datos (SAD), siendo su estructura general la mostrada en la figura 3.1.

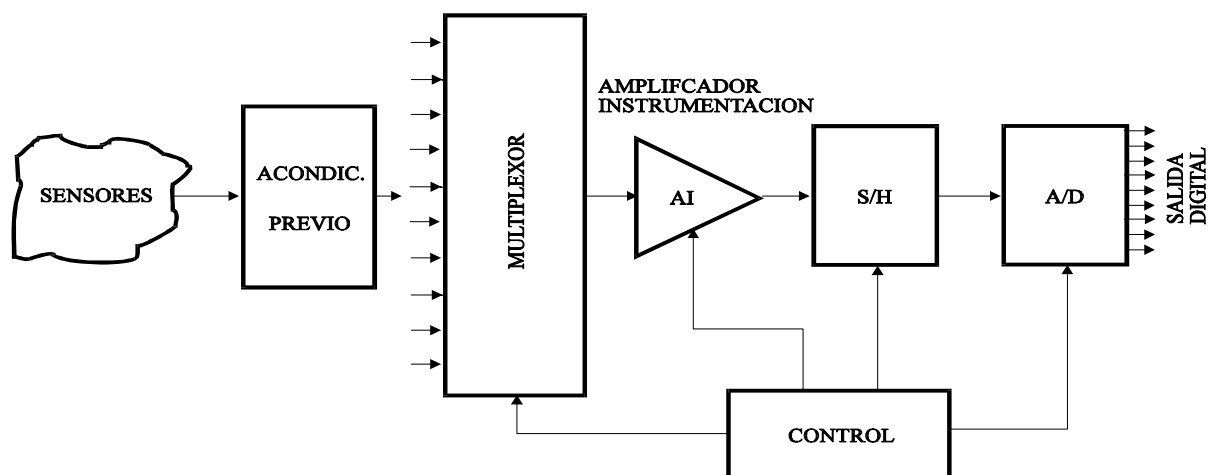


Fig. 3.1.- Diagrama de bloques de un SAD genérico.

Algunos de los elementos que forman el diagrama de bloques del SAD, han sido estudiados con profundidad en temas anteriores, aunque no está de más ofrecer a continuación un repaso a sus respectivas funciones.

- Sensores o transductores: Son los encargados de convertir la variable física a medir (temperatura, humedad, presión, etc.) en señal eléctrica. Esta señal eléctrica suele ser de muy bajo nivel, por lo que generalmente se requiere un *acondicionamiento previo*, consiguiendo así niveles de tensión/corriente adecuados para el resto de los módulos del SAD.

- Multiplexor: Este módulo o circuito se encarga de seleccionar la señal de entrada que va a ser tratada en cada momento. En el caso de que solamente deseáramos tratar con una única señal, este circuito no sería necesario.

- Amplificador de instrumentación: La función de este bloque es amplificar la señal de entrada del SAD para que su margen dinámico se aproxime lo máximo posible al margen dinámico del conversor A/D (ADC) consiguiéndose de esta forma máxima resolución. En SAD con varios canales de entrada, cada canal tendrá un rango de entrada distinto, con lo que será necesario que este amplificador sea de ganancia programable.

- S & H (Sample & Hold, Muestreo y Retención): Este circuito es el encargado de tomar la muestra del canal seleccionado (*sample*) y mantenerla (*hold*) durante el tiempo que dura la conversión. Este circuito será necesario siempre que la señal de entrada sufra variaciones apreciables durante el tiempo que dura la conversión. Si el ADC posee su propio circuito S & H, no será necesario añadirlo a su entrada.

- ADC (Conversor A/D): Se encarga de realizar la conversión analógico/digital propiamente dicha, proporcionando un código digital de salida que representa el valor de la muestra adquirida en cada momento. Es uno de los módulos fundamentales en cualquier SAD y sus características pueden condicionar al resto de los módulos/circuitos del sistema.

3.2.- CONFIGURACIONES DE LOS SAD

Atendiendo al número de canales de entrada, los SAD pueden clasificarse en monocanales y multicanales.

3.2.1.- SAD MONOCANALES

La configuración más general responde al diagrama de bloques de la figura 3.2.

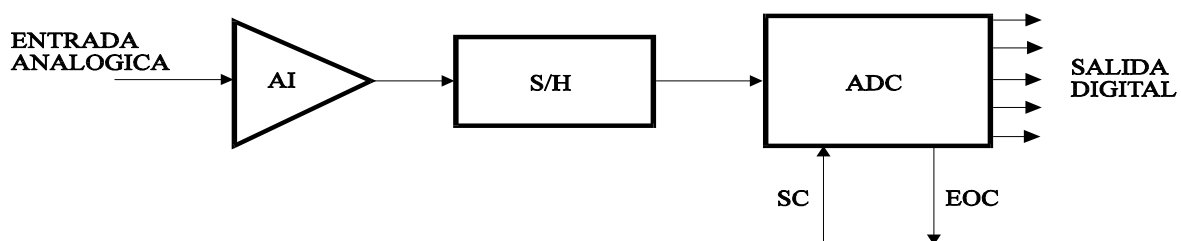


Fig. 3.2.- Diagrama de bloques de un SAD monocanal.

La señal procedente de la fuente de información (cuya obtención se realizará por medio de sensores apropiados con sus correspondientes acondicionadores de señal) es aplicada a la entrada del circuito amplificador de instrumentación, el cual adaptará el nivel analógico de la entrada al margen dinámico del ADC.

Como puede observarse en la figura, el SAD monocanal sólo permite la adquisición de una señal de entrada, lo que permite optimizar su configuración para un tipo concreto de entrada analógica.

3.2.2.- SAD MULTICANAL

Cuando se plantea la necesidad de realizar la conversión A/D de diversas señales (canales), los SAD pueden tener diferentes configuraciones. La configuración a utilizar vendrá, en muchos casos, impuesta por:

- Características de las señales analógicas de entrada (frecuencia, aperiodicidad, etc.).
- Información que se desee obtener de las señales.
- Velocidad de conversión.
- Costo, etc.

En un SAD multicanal pueden existir distintas configuraciones en función de como realicemos la distribución de los módulos en el sistema. Esta distribución dependerá de las necesidades de cada aplicación en particular.

3.2.2.1.- SAD multicanal con muestreo secuencial de canales

Es la configuración que menos componentes precisa y por consiguiente la más económica de entre los multicanal. Su estructura se muestra en la figura 3.3.

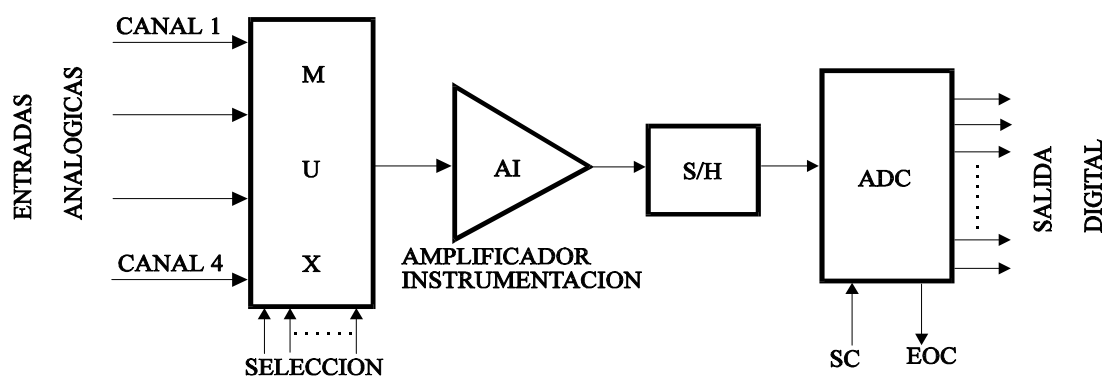


Fig. 3.3.- SAD multicanal con muestreo secuencial de canales.

El funcionamiento del circuito es bastante sencillo: una vez que se selecciona el canal de entrada del MUX y se fija la ganancia del AI, el circuito S/H pasa a modo *Sample* hasta que adquiere una muestra de la señal, momento en el que pasa al modo *Hold*, dándose al ADC la orden de inicio de conversión. Una vez transcurrido el tiempo de conversión el ADC lo indica mediante la señal fin de conversión, repitiéndose de nuevo todo el proceso anterior, bien con el mismo canal, bien con otro distinto.

Esta configuración permite que durante el tiempo de conversión de un canal, se puede estar seleccionando en el MUX, simultáneamente, el siguiente canal a muestrear. Así, el tiempo de establecimiento del MUX no influirá en la velocidad de adquisición final del SAD, siempre y cuando dicho tiempo sea menor que el tiempo de conversión del ADC.

3.2.2.2.- SAD multicanal con muestreo simultáneo de canales

Esta configuración presenta la ventaja de que todos los circuitos S/H de entrada conmutan simultáneamente a modo *Hold*, manteniendo el valor de la muestra de cada señal de entrada hasta que el ADC pueda realizar la conversión, cosa que no era posible en el modelo de muestreo secuencial.

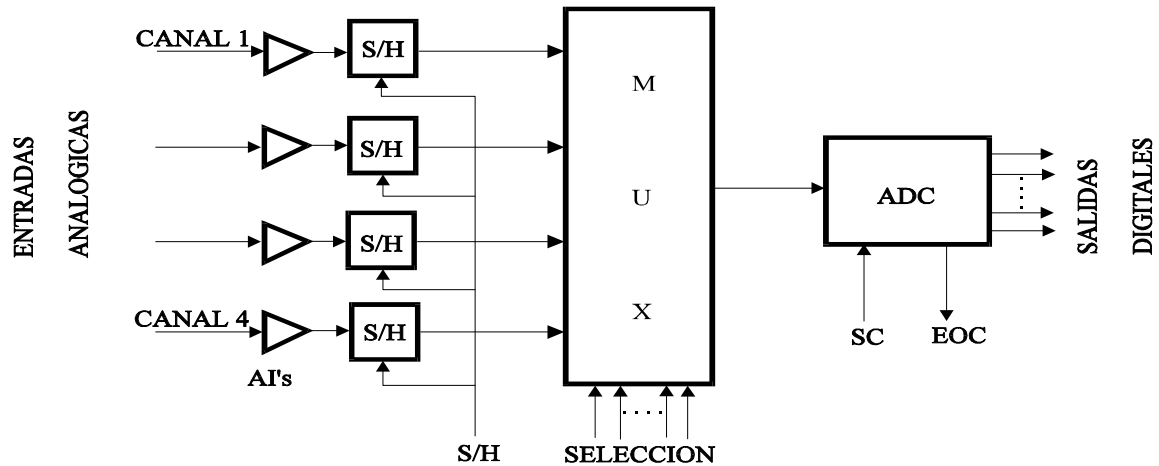


Fig. 3.4.- SAD multicanal con muestreo simultáneo de canales.

Algunas tarjetas de adquisición de datos comerciales que presentan esta configuración son las DT 2809, 2829, 2838 y 1498 de Data Translation.

3.2.2.3.- SAD multicanal paralelo

En este caso se puede decir que cada canal constituye un SAD independiente con todos los elementos necesarios para realizar una conversión A/D completa, con la salvedad de que al

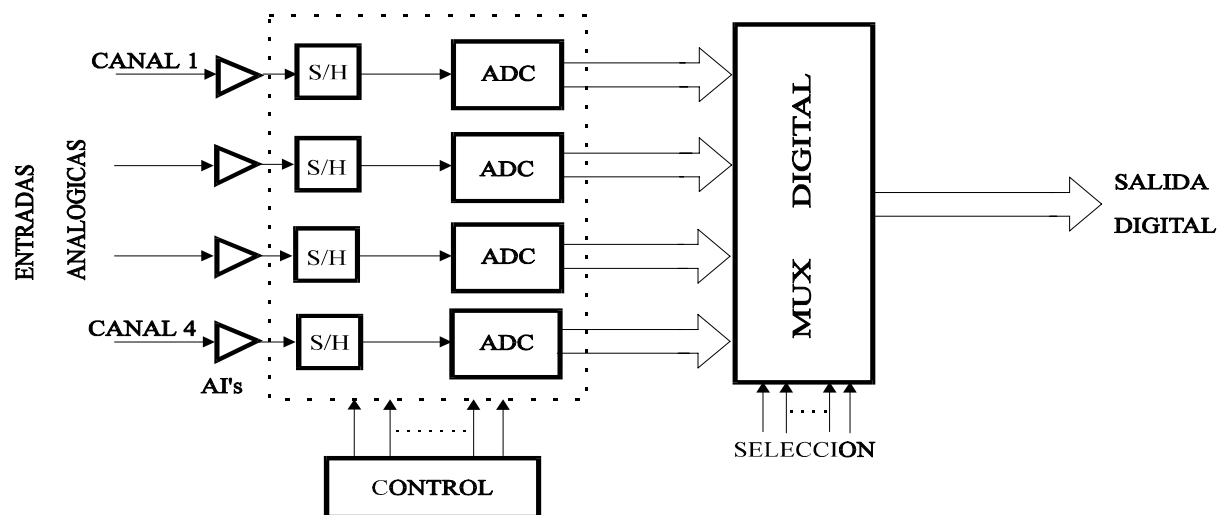


Fig. 3.5.- SAD multicanal paralelo

utilizar, generalmente, un sólo canal digital de salida es necesario incluir un MUX digital.

La configuración de un SAD multicanal paralelo es la mostrada en la figura 3.5 y en ella se puede observar que ofrece una gran flexibilidad, ya que cada canal puede ser adaptado de forma independiente, según las necesidades requeridas por la señal a adquirir (ganancia del amplificador, velocidad de adquisición, etc.). Otra ventaja adicional es que la velocidad del sistema se optimiza notablemente, ya que pueden realizarse simultáneamente varias conversiones. Como inconveniente fundamental cabe destacar su coste y el número de parámetros a controlar en cada instante.

3.3.- PARÁMETROS CARACTERÍSTICOS DE UN SAD

Los parámetros que caracterizan a un SAD son básicamente tres: número de canales, exactitud de la conversión y velocidad de muestreo (número de muestras por unidad de tiempo, *throughput rate*).

- Número de canales: depende del número de señales a adquirir.
- Exactitud de la conversión: viene impuesta por los circuitos utilizados, es decir, multiplexores, amplificadores, S/H y ADC, esencialmente. Así, a cada uno de estos circuitos o módulos le pediremos unos mínimos.
 - * *Multiplexor*: Baja resistencia de conducción (R_{ON}) y constante en el margen de variación de las señales de entrada. Tiempos de establecimiento pequeños.
 - * *Amplificador*: Mínimas tensiones y corrientes de offset, así como sus derivas. Tiempo de establecimiento pequeño, aún con altas ganancias. Amplio margen para programar la ganancia.
 - * *S/H*: Pequeña tensión offset y deriva de ésta. Máxima velocidad de caída en modo *Hold*, siempre y cuando la tensión a la salida del S/H esté constante el tiempo necesario para que el ADC la digitalice. Tiempos de apertura, de adquisición y de asentamiento mínimos.
 - * *ADC*: Alta resolución. Mínimo tiempo de conversión. Error de linealidad y de ganancia pequeños.
- Velocidad de muestreo (*throughput rate*): este parámetro especifica la velocidad a la que el SAD puede adquirir y almacenar muestras de las entradas. Las muestras pertenecerán a un único canal o a varios, según la configuración, por lo que es fundamental revisar cuidadosamente los datos suministrados por el fabricante. En general debemos identificar el *throughput rate* con el número de muestras por unidad de tiempo que pueden obtenerse de un canal. Los cuatro factores principales a tener en cuenta son:
 - * Tiempo de establecimiento del MUX
 - * Tiempo de establecimiento del amplificador
 - * Tiempo de adquisición del S/H
 - * Tiempo de conversión del ADC

Hasta que la muestra adquirida llega al conversor, pasa a través de un MUX, de un AI de ganancia programable y finalmente por un S/H. Cada elemento de esta cadena requiere un corto período de tiempo de establecimiento para conseguir su mejor precisión. Por tanto, cuando se calcula la máxima velocidad de muestreo del sistema, debemos incluir el efecto de todos estos elementos, y no sólo el tiempo de conversión del ADC.

3.4.- CLASIFICACIÓN DE LOS SAD Y SU CONEXIÓN A LOS EQUIPOS DE PROCESO

En muchos casos tiende a identificarse un SAD con las ya clásicas Tarjetas de Adquisición de Datos (TAD), al ser esta la opción más común por su simplicidad y bajo costo. Sin embargo, el campo de la *instrumentación* aporta una serie de *interfaces* estándar con sus correspondientes entornos de programación y protocolos. Dichos *interfaces* permiten una comunicación cómoda y elegante entre el equipo de control o procesador principal y el instrumento de medida correspondiente, que suministra o recibe la información necesaria.

De este modo, pueden establecerse dos clases genéricas de SAD:

- SAD basados en Tarjetas de Adquisición de Datos (TAD). Bus interno.
- SAD basados en *interfaces* estándar para instrumentación. Bus externo.

En algunas ocasiones y de forma muy esporádica, suelen utilizarse *circuitos integrados para adquisición de datos*. Estos chips encuentran su principal campo de utilización en diseños que se orientan hacia una aplicación específica y suelen incorporar la mayor parte de los subsistemas necesarios para realizar la adquisición de las señales analógicas: amplificador de instrumentación, S/H, ADC, temporizadores, etc. En este tema nos vamos a centrar en el estudio y análisis de los dos SAD citados inicialmente (bus interno y externo) ya que son los más difundidos, teniendo la posibilidad de un control fácil y cómodo desde un equipo de proceso tipo ordenador/estación de trabajo.

De este modo, una esquematización de los SAD de más uso en la actualidad sería la mostrada en la figura 3.6, en la cual podemos observar como el PC/Estación de trabajo está presente para cualquiera de los sistemas utilizados, convirtiéndose en la herramienta que más nos facilita la labor de Adquisición y Control sobre el proceso que se esté analizando.

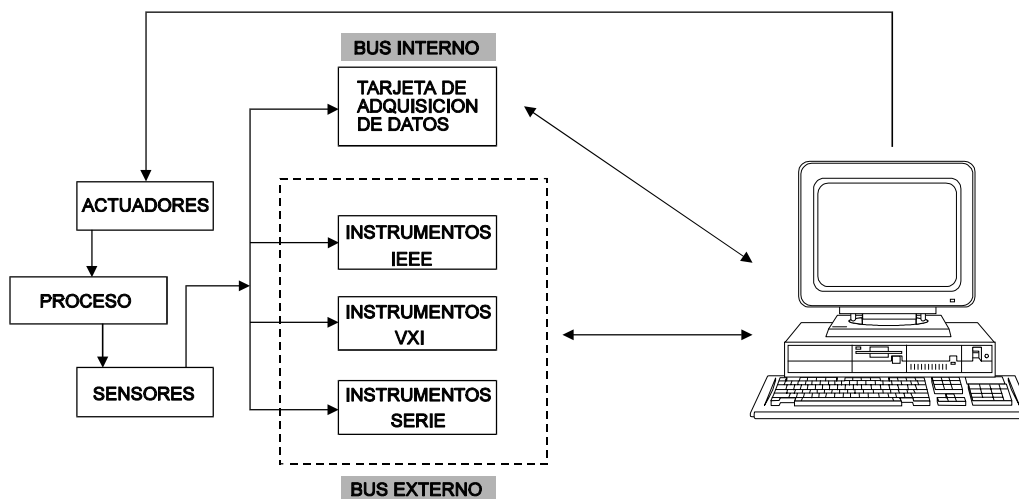


Fig. 3.6.- Métodos usuales para la adquisición de datos.

3.4.1.- SISTEMAS BASADOS EN TAD (BUS INTERNO)

Es el caso más simple y usual en los SAD, y se basa en la utilización de tarjetas conectadas al ordenador personal o estación de trabajo, mediante un slot de expansión o utilizando un conector que conduce a un rack de expansión. En la figura 3.7 podemos ver ambos casos, pudiéndose observar como en la figura 3.7 (a), la propia TAD se incorpora en una de las ranuras de expansión del equipo (conexión directa); en el caso de la figura 3.7 (b), lo que se conecta al slot del equipo es una tarjeta controladora a modo de *interface* y que permite establecer la comunicación entre el ordenador y el rack que verdaderamente contiene una o varias tarjetas para adquisición, presentación y control. Tanto si hablamos de la conexión directa de la TAD como de la conexión mediante rack, las tarjetas pueden estar mapeadas en memoria o como entrada/salida (E/S).

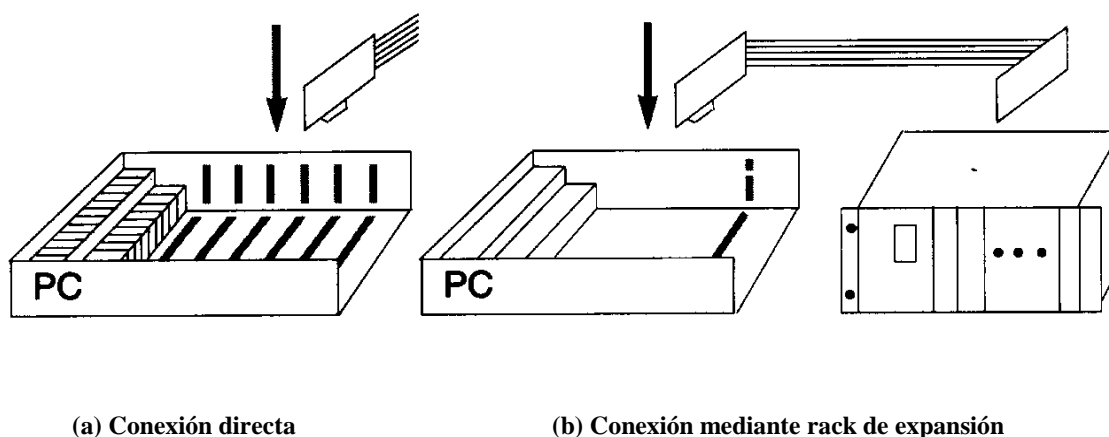


Fig. 3.7.- Sistemas de Adquisición de datos basados en tarjetas.

En el primer caso el hardware de adquisición de datos es tratado como memoria RAM leída y/o escrita por el procesador del sistema. De esta forma, se simplifica el software de acceso al hardware de adquisición y se reduce significativamente el tiempo de transferencia de datos. Cuando el número de canales analógicos y digitales crece, el mapeado de memoria puede resultar inabordable. En tal caso, la solución usual es mapear como E/S, por lo que las tarjetas son vistas por el ordenador como un periférico más.

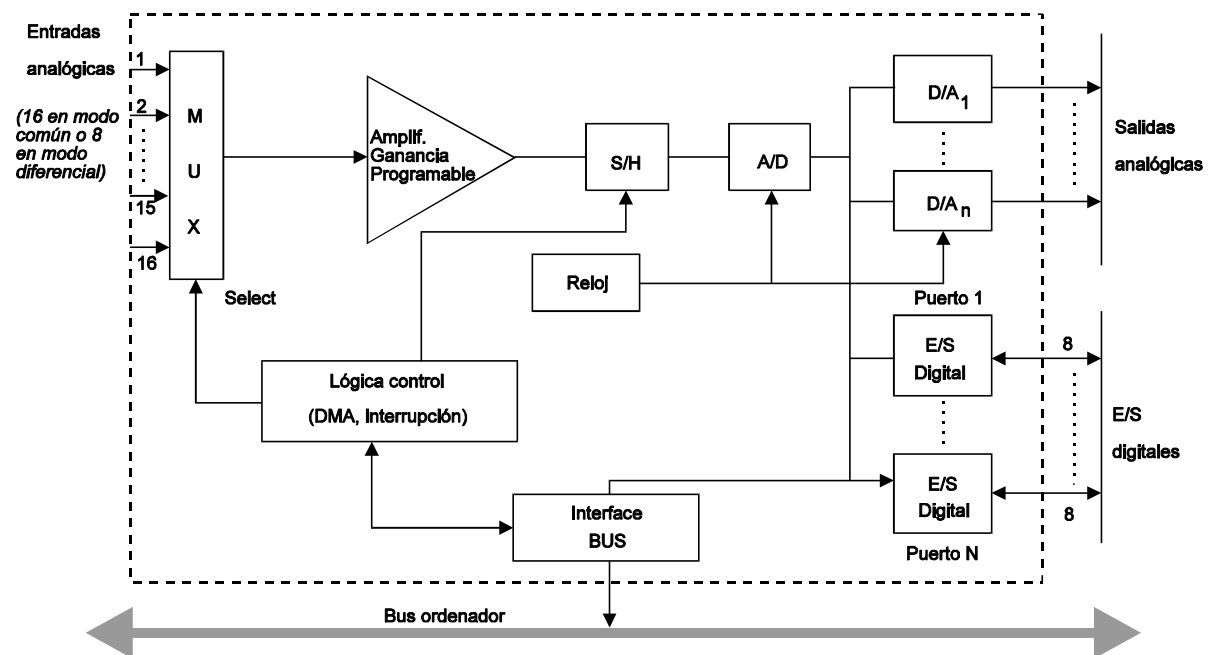
Un porcentaje muy elevado de los SAD instalados hoy en día, se basan en TAD asociadas a un ordenador monopuesto tipo PC. En un número inferior de aplicaciones, los sistemas utilizados consisten en arquitecturas abiertas sobre buses como el VME, Multibus o STD. Tanto los sistemas más simples y económicos como los más complejos, requieren de un análisis de los siguientes elementos:

- Tipo de ordenador
- Transductores
- Acondicionamiento de la señal
- Hardware de adquisición
- Hardware de análisis (p.ejemplo, procesadores digitales de señal)
- Software (a nivel driver y de aplicación)

Los sistemas basados en TAD son equipos *fuertemente acoplados* (por contraposición con los sistemas distribuidos), en el sentido de que tanto el proceso de adquisición de datos como el de gestión, se realizan bajo el control del procesador residente en el ordenador. Por este motivo, el tipo de ordenador determina esencialmente la velocidad de proceso y, consecuentemente, en las aplicaciones que requieren un procesado en tiempo real de señales cuyo ancho de banda es elevado, es aconsejable la utilización de ordenadores con procesadores de elevadas prestaciones y con alto rendimiento en operaciones en coma flotante. Si las exigencias de velocidad de proceso son aún mayores, una solución cada vez más habitual, consiste en incorporar una TAD con hardware dedicado, a partir de un procesador digital de señal (DSP).

En la figura 3.8, podemos observar el diagrama de bloques de una TAD estándar, la cual incluye los elementos básicos que suelen incorporarse en cualquier tarjeta.

Fig. 3.8.- Diagrama de bloques de una tarjeta de adquisición de datos estándar.



Hagamos a continuación una exposición de los elementos que caracterizan a una TAD estándar, así como sus posibilidades y limitaciones generales.

La gran mayoría de las TAD incorporan los ADC y DAC, así como las líneas de E/S. Respecto a las *entradas analógicas*, la calidad de la señal digitalizada puede venir afectada por el número de canales, la frecuencia de muestreo, la resolución, el rango de tensión que el ADC puede cuantificar, la precisión, el ruido y la no linealidad. Las entradas analógicas pueden conectarse en *modo común* (*single ended*) o *diferencial*. En el primer caso, las entradas están referenciadas a una masa común y se utilizan cuando las señales son del orden de 1v o mayores y la distancia entre la fuente de la señal y el hardware de adquisición es pequeña (inferior a 4 metros aproximadamente). En caso contrario se utiliza el modo diferencial, donde cada entrada tiene su propia referencia y el ruido puede ser eliminado/atenuado gracias a su elevado factor de rechazo en modo común (CMRR). El rango de tensión de entrada de la TAD puede ser *unipolar* (por ejemplo 0-10v) o *bipolar* ($\pm 10v$, por ejemplo), siendo recomendable la elección de una

tarjeta en la que, cubriendo el margen de tensión de la señal de entrada, el margen sobrante sea mínimo, a fin de obtener la máxima resolución.

En algunas aplicaciones resulta importante preservar la simultaneidad en la adquisición de las señales de entrada, sin ser por ello necesario que la conversión A/D sea simultánea. En tal caso, no es necesario adquirir una tarjeta con múltiples ADC sincronizados. Cabe plantearse dos soluciones: si se pretende muestrear, por ejemplo, 3 canales a 20 KHz, puede optarse por aumentar la velocidad a 60 KHz o, en el supuesto de no tolerar como *simultaneo* un retraso entre el muestreo del primer y último canal de 50 μ s, cabe considerar la multiplicidad de circuitos S/H previos al multiplexor de entrada.

Respecto al tipo de ADC utilizado frecuentemente en las TAD, aproximadamente un 75% de ellas utilizan ADC de *aproximaciones sucesivas*, por su buen compromiso entre coste, velocidad y precisión (un conversor rápido de este tipo puede efectuar una conversión simple en un tiempo de entre 750 ns y 1 μ s).

Algunas aplicaciones especiales, como la inspección visual automatizada, trabajan con señales cuyo ancho de banda es del orden de 10 MHz (100 ns de periodo). Con objeto de eliminar problemas de *aliasing*¹, dicha señal debería muestrearse a una frecuencia mínima de 20 MHz con el fin de cumplir el Teorema de Nyquist, por lo que el tiempo de conversión del ADC debería ser inferior a 50 ns. Este orden de magnitud tan sólo es alcanzable mediante conversores tipo FLASH. No obstante, en ciertos casos se reduce dicho requerimiento utilizando un filtro de paso bajo (previo al ADC) para reducir el ancho de banda de la señal, lo que permite aumentar el tiempo de conversión necesario (en otros casos, se asume “sin complejos” la presencia de *aliasing* en la digitalización).

La conversión A/D mediante ADC tipo FLASH, es adecuada si el ambiente no tiene un elevado contenido de ruido, ya que la rapidez de estos ADC los hace poco inmunes a perturbaciones indeseables. Cabe tener presente que el crecimiento no lineal de comparadores implicados en un ADC-FLASH respecto del número de bits de salida (n) es la razón básica para que habitualmente no se utilice este conversor para más de 8 bits. Recordemos que en este tipo de ADC, el número de comparadores internos viene dado por la expresión $2^n - 1$. Por el contrario, los ADC de aproximaciones sucesivas más comunes en TAD son de 12 bits, alcanzando sin problemas los 16 bits.

Una vez analizados los aspectos más significativos de las líneas de entrada y de los ADC de una tarjeta de adquisición de datos, pasemos ahora a ver las consideraciones más relevantes para los *conversores digital-analógicos* (DAC) en una Tarjeta de Adquisición de Datos (TAD) estándar, así como el método de *transferencia de datos*.

El parámetro más significativo de un DAC acostumbra a ser el tiempo de asentamiento (establecimiento). Este alcanza valores desde microsegundos hasta nanosegundos en los

¹ El efecto aliasing consiste en la superposición de dos señales, o parte de ellas, en el dominio de la frecuencia, de forma tal que ninguna de ellas podrá ser recuperada íntegramente sin verse afectada parcial o totalmente por la otra. Este efecto (no deseado) aparece cuando al muestrear una señal no se respeta el Teorema de Nyquist: $f_m \geq 2 f_s$, donde f_m es la frecuencia de muestreo y f_s la máxima frecuencia de la señal a muestrear.

convertidores rápidos. En las placas más completas se incorpora un circuito que elimina el problema de glitch, causado por las diferencias temporales en la conmutación de los distintos datos presentes a la entrada del convertidor (básicamente es un circuito mantenedor/retardador). Nos remitimos al Tema 2 de la asignatura para profundizar en los aspectos anteriores.

Las resoluciones usuales en los DAC son de 8, 10, 12 y 16 bits. A diferencia de los ADC, los DAC pueden adoptar dos tipos de señal analógica: salida en tensión o en corriente. Esta última resulta más conveniente en entornos con alto nivel de ruido (el margen de salida estándar es de 4-20 mA). Además de preservar la integridad de la señal a grandes distancias (con menor ruido por inducción), la salida en corriente no comporta pérdida de señal.

El método de *Transferencia de Datos* utilizado es un procedimiento decisivo en las aplicaciones que requieren alta velocidad de adquisición. En un elevado número de casos, el ancho de banda de las señales a almacenar permite su digitalización a frecuencias del orden de 20 a 40 KHz.

Para dichas aplicaciones es suficiente realizar la transferencia de datos entre la tarjeta y la memoria del ordenador mediante *E/S programada* (PIO). Desde el punto de vista de velocidad en las aplicaciones más modestas, es suficiente con *sondear por software* la TAD de forma continua (*polling*). En el caso de que el software realice otras operaciones que penalicen fuertemente la frecuencia de sondeo, resulta conveniente realizar la transferencia aprovechando el hardware de interrupciones del equipo. De esta forma, la CPU sólo es interrumpida por la tarjeta cuando realmente es necesaria la transferencia de datos, por lo que la frecuencia de digitalización puede alcanzar con facilidad 40-50 KHz. No obstante, vía interrupción se siguen manteniendo dos servidumbres importantes; por un lado, al ser un método de E/S, el resto de actividades de la CPU (teclado, disco, pantalla, etc.) se ven penalizados en cuanto a velocidad. Además, tan solo una palabra puede ser transferida a memoria en cada interrupción. La incorporación a las TAD de un controlador DMA permite obviar las limitaciones enunciadas, al liberar a la CPU de las tareas de adquisición. Las TAD con único canal de DMA proporcionan velocidad de transferencia del orden de 100 KHz. Para aplicaciones cuyos requerimientos de transferencia sean del orden de 250-300 KHz es necesario utilizar doble canal de DMA. De esta forma la TAD puede realizar transferencias de datos sin tiempos de espera, pues mientras un canal es reprogramado con un nuevo buffer de memoria, por el otro canal se realiza la transferencia de un dato. Por encima de los 300 KHz, la limitación proviene de la velocidad máxima a la cual los datos almacenados en memoria pueden ser procesados en tiempo real (dejando libre el espacio para nuevos datos), o bien de la velocidad de transferencia de datos al disco duro, a fin de evitar overflow en la RAM. En el primer caso, cabe pensar en la utilización de hardware dedicado (por ejemplo, DSP). Si la opción escogida es la del almacenamiento de datos para un posterior análisis, resulta conveniente escoger un disco duro de alta velocidad (tiempo de acceso pequeño), así como una *interface* hardware rápida (SCSI o superior).

Más allá de los 250-300 KHz hay que tener presente que el ancho de banda del bus del ordenador puede representar una limitación seria. En tal caso, tenemos dos posibles soluciones. La primera consistiría en realizar la transferencia de datos a una zona de memoria dedicada, cuya gestión deberá ser lo más eficiente posible por parte del software de adquisición. La segunda solución sería hacer uso de las tecnologías más avanzadas en buses de comunicación (PCI, USB,

IEC 1348, etc.), aprovechando el aumento de velocidad en la transferencia de datos que ofrecen estas posibilidades.

Lógicamente, las dos soluciones planteadas tendrán unos aspectos económicos a considerar, de forma que el par *TAD-Software de control* se adapte a nuestras características.

3.4.2.- SISTEMAS BASADOS EN INTERFACES ESTÁNDAR PARA INSTRUMENTACIÓN (BUS EXTERNO)

Estos sistemas, a los que normalmente se les denomina *sistemas de instrumentación*, pueden definirse como un conjunto de instrumentos conectados a través de un BUS que ejecutan un trabajo determinado de forma automática bajo el control de un equipo controlador, que en la mayoría de los casos es un ordenador (PC, MAC, Estación de trabajo), siendo el PC el más utilizado. Así, en la figura 3.9 podemos ver el diagrama de bloques de lo que sería la arquitectura de un Sistema de Instrumentación según la definición anterior. Obviamente, dependiendo de las necesidades, el sistema tendrá más o menos componentes, mostrándose en la figura 3.9 lo que sería una arquitectura prácticamente completa.

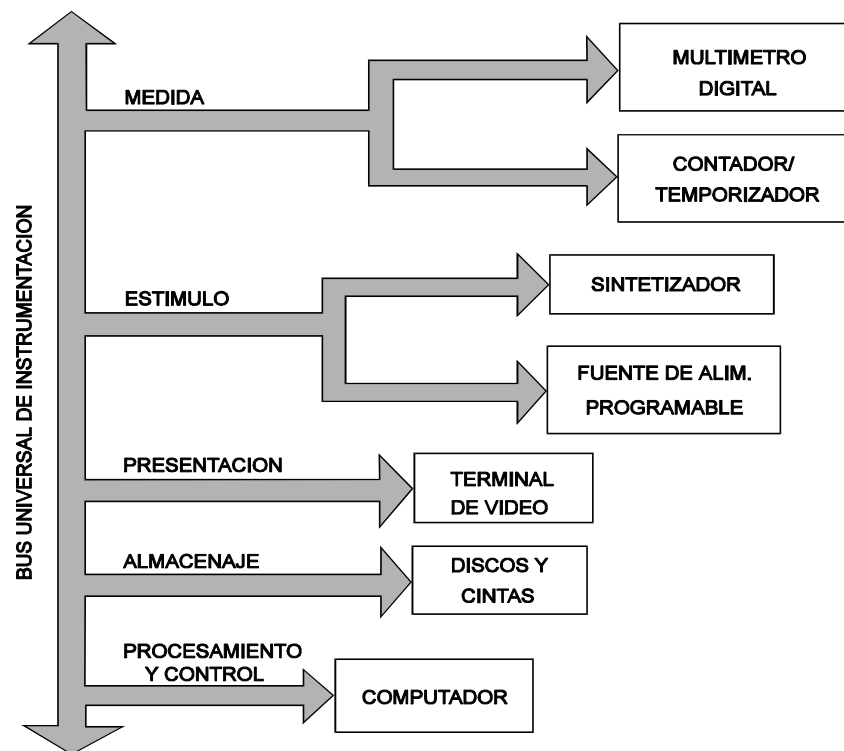


Fig. 3.9.- Diagrama de bloques de la arquitectura de un Sistema de Instrumentación.

Uno de los pioneros destacados en la elaboración de buses dedicados para la interconexión entre instrumentos programables y ordenadores fue la empresa Hewlett-Packard, quien en 1965 diseñó la *interface* HP-IB (Hewlett-Packard Interface Bus) para conectar su línea de instrumentos programables a sus ordenadores. Debido a su alta velocidad de transferencia nominal (1Mbyte/s), este bus incrementó su popularidad rápidamente. A los diez años de su creación, este bus fue aceptado como estándar (IEEE-488). Con el fin de mejorar el bus anterior, HP elaboró en 1987 un nuevo estándar, el ANSI/IEEE 488.2, que incorporaba una definición precisa del protocolo de comunicaciones entre instrumentos y ordenadores.

Hasta entonces, la utilización de instrumentos con el estándar IEEE 488 comportaba la adquisición de un ordenador HP. La empresa National Instruments decidió expandir el uso del bus IEEE 488 a ordenadores fabricados por otras firmas mediante la inclusión de la correspondiente tarjeta o módulo a modo de *interface* entre el ordenador y los instrumentos a controlar. Esta empresa otorgó al bus la denominación GPIB (General Purpose Interface Bus), siendo desde entonces más conocido por este nombre que por el de HP-IB ó IEEE-488.

Además del bus desarrollado en la norma IEEE-488, la adquisición de datos mediante bus externo, hace uso de otros dos buses: VXI y comunicación serie (RS-232, RS-422 y RS-485, por ejemplo). El primero de ellos está ganando terreno al GPIB, mientras que el segundo no está tan extendido en los SAD basados en bus externo.

La figura 3.10 sintetiza lo que sería un Sistema de Instrumentación, en el cual aparece el ordenador como equipo de control, teniendo en su interior conectada la correspondiente tarjeta que hará de *interface* entre el bus interno de éste y el bus externo al que se conectan los instrumentos.



Fig. 3.10.- Interfaces estándar de instrumentación.

3.4.2.1.- El estándar IEEE 488.2 y SCPI

En 1987 el estándar IEEE 488.2 eliminó las ambigüedades del IEEE 488.1, al definir los formatos de datos, el protocolo de intercambio de mensajes entre controlador e instrumento, las secuencias de control y la información de estado, entre otros requerimientos a los que todo instrumento IEEE 488.2 debe ajustarse. Este bus ha supuesto un avance notable en el campo de la adquisición de datos por lo que a compatibilidad, fiabilidad y simplicidad de programación se

refiere. En la figura 3.11 podemos observar gráficamente lo que ha supuesto cada *interface* para la unificación de criterios y la facilidad de uso en el campo del control de instrumentos programables mediante bus GPIB.

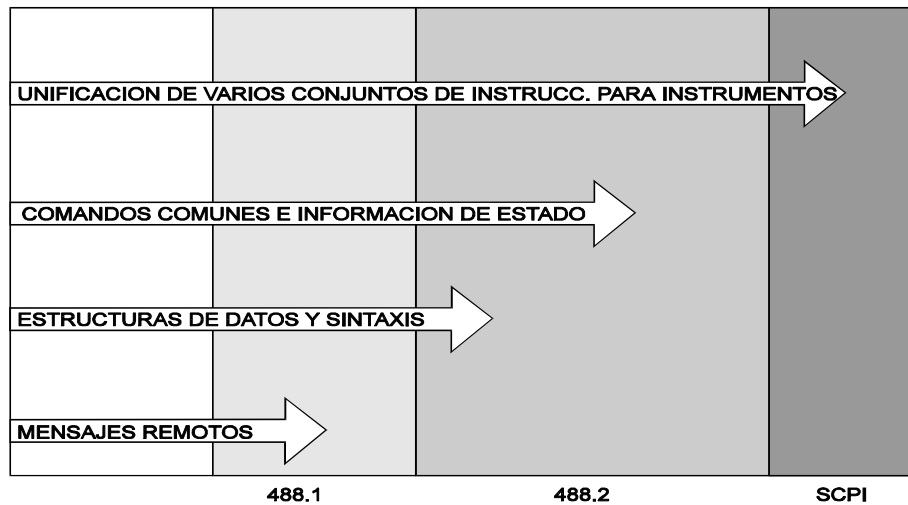


Fig. 3.11.- Evolución de los estándar de la instrumentación GPIB.

La *interface* GPIB va asociada a sistemas de interconexión de instrumentos cuyo *intercambio de información sea de naturaleza digital*. El número máximo de equipos interconectables es 15 y la longitud máxima de transmisión sobre cables de interconexión es de 20 metros, con un límite de 4 metros entre dos dispositivos cualesquiera (excepto cuando se utilicen técnicas especiales de amplificación de bus). La velocidad máxima de transmisión de datos, en cualquiera de las líneas que forman el bus, es de 1 Mbyte/s. La conexión de los equipos al bus GPIB se hace mediante una topología de *bus de línea compartida*, es decir, compartiendo las líneas de la señal. Así, respetando lo anterior, podemos encontrarnos con una configuración lineal (figura 3.12) o una configuración en estrella (figura 3.13) o una combinación de ambas.

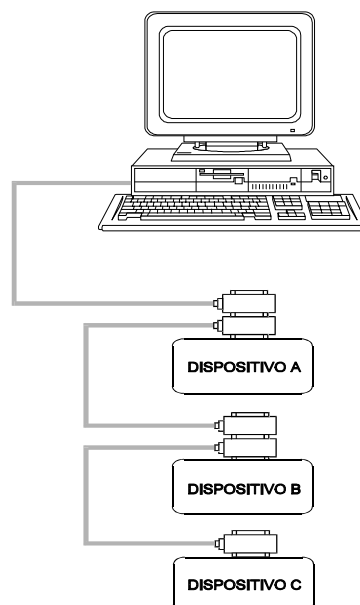


Fig. 3.12.- Configuración lineal para bus GPIB.

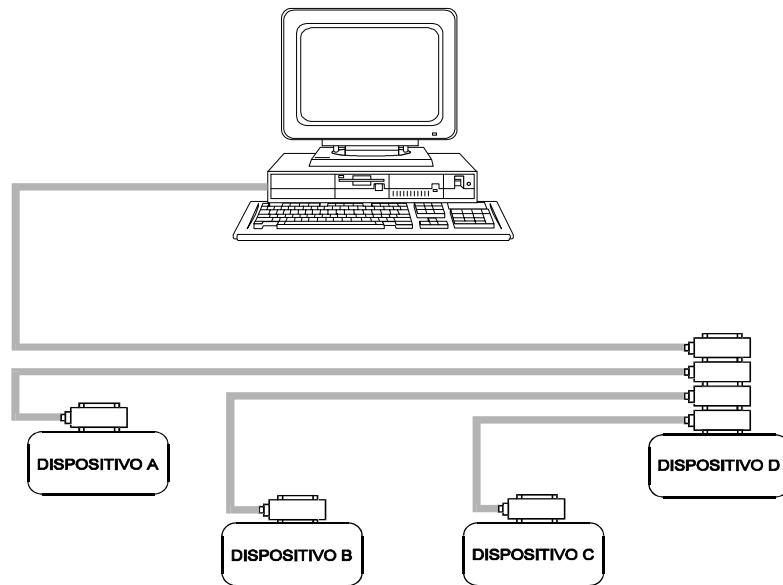


Fig. 3.13.- Configuración en estrella para bus GPIB.

Obsérvese como para cada uno de los dispositivos hay un elemento distribuidor de señal, el cual permite que todos los dispositivos puedan acceder a todas las líneas del bus, independientemente de los demás y de la topología utilizada.

El GPIB viene a ser un bus de ordenador excepto en el sentido de que éste último dispone de sus tarjetas interconectadas por medio de una *placa madre o backplane*. En cambio, GPIB consta de dispositivos interconectados vía cable. Los dispositivos GPIB pueden ser de tres tipos:

- *LISTENER* (Escucha, receptor): Equipo capaz de recibir datos a través del bus cuando se le direcciona. Puede haber hasta *catorce escuchas activos* simultáneamente. Algunos dispositivos que pueden actuar como *listeners* son: impresoras, equipos o dispositivos de visualización, fuentes de alimentación, generadores de señal, etc.
- *TALKER* (Habla, transmisor): Equipo capaz de transmitir datos a través del bus cuando se le direcciona. Únicamente puede haber un *talker activo* en el *interface* en todo momento. Algunos dispositivos que pueden actuar como *talkers* son: voltímetros, contadores, osciloscopios, analizadores de espectros, etc.
- *CONTROLLER* (Controlador): Dispositivo capaz de determinar o especificar la función que va a realizar cada equipo (*talker o listener*) en una transferencia de información en la que interviene él mismo. El controlador es una especie de centro de conmutación que controla toda la red de comunicación y cuyo papel podría asimilarse al de una CPU o, quizás más descriptivo, al de una centralita telefónica. Cuando el controlador detecta que un dispositivo quiere enviar datos o mensajes, realiza la conexión entre ese *talker* y el *listener* correspondiente. Después de que el mensaje ha sido transmitido, el *controller*, normalmente, desactiva (deja de direccionar) ambos dispositivos.

Según las definiciones anteriores, un *talker* puede enviar mensajes (resultados de medidas, ficheros de datos, instrucciones de programa, estado de la máquina, ...) a uno o más *listener*. El *controller* es quien gestiona el flujo de información, mediante órdenes enviadas a todos los dispositivos. La función de controlador la suele realizar normalmente un ordenador tipo PC al que se le incorpora una tarjeta de *interface*.

En ciertas configuraciones de sistema GPIB, puede prescindirse del controlador. Es el caso de sistemas en donde un dispositivo es siempre *talker* y el resto son siempre *listener*. En el caso de que algunos de los dispositivos sean reconfigurables, la función de controlador acostumbra a realizarse mediante un ordenador.

El GPIB permite conectar hasta 15 instrumentos a un solo controlador, proporcionando una comunicación paralela (8 bits) a una velocidad razonablemente elevada (como cifras orientativas: lecturas y escrituras a 1 Mbyte/s y control a 350 Kbytes/s).

Líneas del bus GPIB

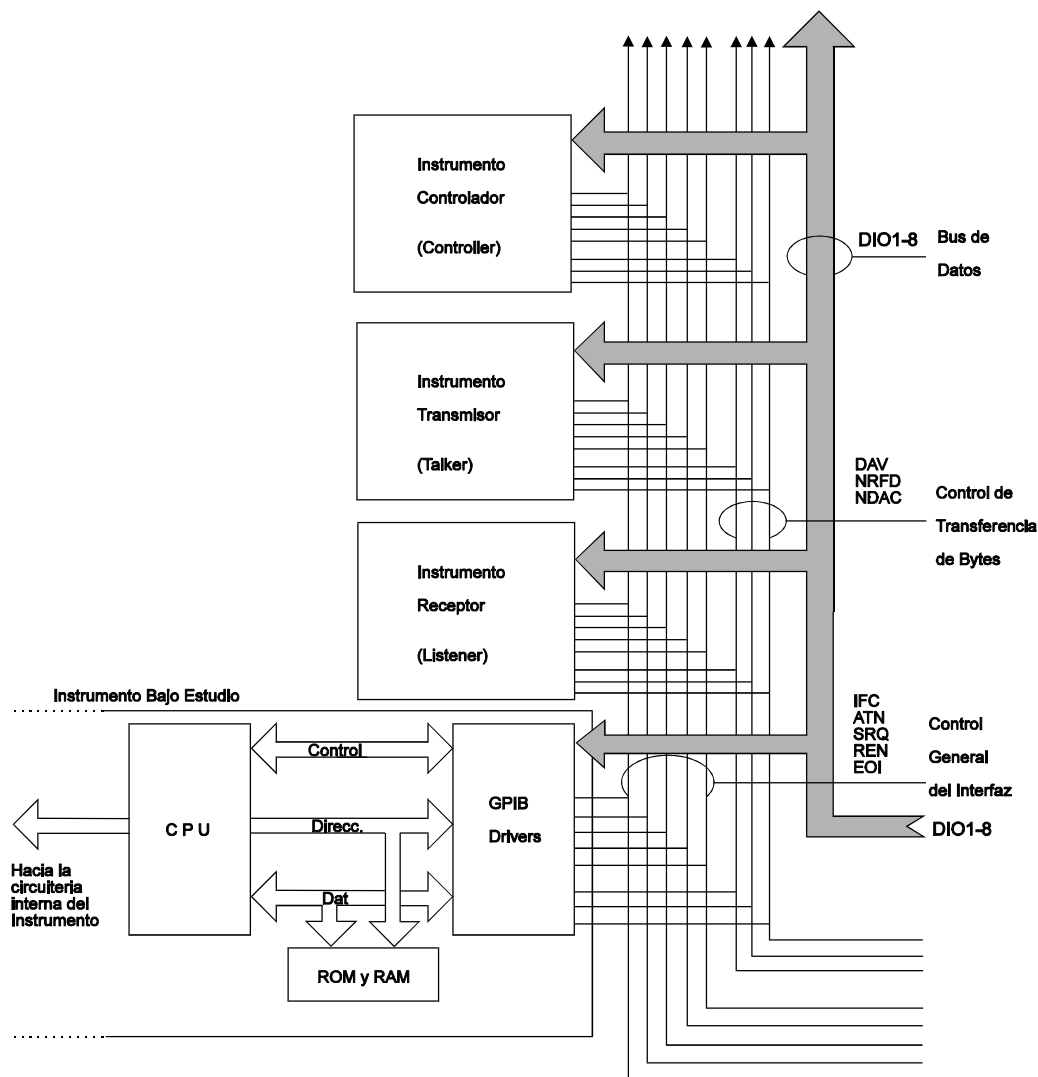


Fig. 3.14.- Conexión mediante GPIB. Detalle de las líneas del bus.

Los dispositivos GPIB se conectan, típicamente, mediante un cable apantallado que consta de 24 conductores, correspondientes a 16 líneas de señal y 8 tierras (6 pares trenzados para: DAV, NRFD, NDAC, IFC, SRQ y ATN; una masa analógica y otra digital). Las líneas de señal van asociadas a 8 líneas de datos, 3 líneas de *handshake* (protocolo) para el control asíncrono de la transferencia de bytes, a fin de obviar errores de transmisión y 5 líneas de gestión del flujo de información a través de la *interface* (para la comunicación entre el controlador y el resto de dispositivos). En la figura 3.14 podemos observar lo que sería una interconexión de dispositivos mediante bus GPIB. Las 16 líneas de señal utilizan la familia TTL pero con *lógica negativa*.

Veamos, por grupos, las características de las líneas que conforman el bus GPIB:

- *Líneas de datos (Data lines)*: Son ocho líneas bidireccionales que se utilizan para transferir información (control y datos) entre equipos a través del *interface*. Normalmente se utiliza un código ASCII de 7 bits, con el octavo bit disponible, si se desea, para la paridad. No obstante se pueden utilizar otras técnicas para codificar la información sobre estas ocho líneas. La información transferida incluye los comandos del *interface*, direcciones y datos, dependiendo de los equipos (medidas, instrucciones de programación, etc.).

- *Líneas de protocolo (Handshake lines)*: Son tres líneas utilizadas para coordinar el intercambio de información entre los equipos conectados al bus. Controlan asincrónicamente la transferencia de datos entre un dispositivo fuente (un *talker* direccionado o un controlador) y un dispositivo destino (uno o varios *listener*), utilizando el proceso denominado *protocolo de bloqueo a tres hilos*, para garantizar la transferencia íntegra de los datos entre ambos dispositivos. Esta técnica de transferencia de datos presenta las siguientes características:

- 1) La transferencia de datos es asíncrona.
- 2) La velocidad de transmisión se ajusta automáticamente a la velocidad del emisor y receptor, de forma que los datos se transmiten a la velocidad más baja de los dispositivos que intervienen en la transmisión.
- 3) Varios equipos pueden recibir datos simultáneamente.
- 4) El protocolo se realiza para todos los bytes transmitidos.

Con todo ello, este protocolo garantiza que los bytes enviados y recibidos estén libres de errores de transmisión. Las tres líneas de control de protocolo son:

* DAV (*Data Valid*)

Esta línea es utilizada para indicar que los datos que hay sobre el bus son válidos. Es controlada por el equipo fuente, bien sea el *controller* cuando envía mensajes de control, bien sea el *talker* activo cuando envía mensajes. La línea DAV se pone a nivel bajo (activa) cuando los datos son válidos y además se ha detectado que la señal NRFD está a nivel alto (no activa).

* NRFD (*Not ready for data*)

Cuando un dispositivo tiene su señal NRFD a nivel bajo (activa) está indicando que no está disponible para recibir datos. Cuando está disponible para recibir datos, pone la línea a nivel alto (no activa). La línea NRFD sólo puede pasar a nivel alto cuando *todos los listener* estén listos para recibir datos. La línea es controlada por todos los dispositivos cuando reciben

comandos del *controller* y por los *listeners* cuando reciben mensajes del *talker* activo en ese momento.

* NDAC (*Not data accepted*)

Esta línea es controlada por todos los dispositivos cuando reciben comandos (del *controller*) y por los *listeners* cuando reciben datos, para indicar si han aceptado o no la información previamente recibida a través del bus. Cuando la línea está a nivel bajo (activa) el dispositivo indica que no ha aceptado el dato. Cuando el dato o comando transmitido a través de las líneas de datos (DIO) es aceptado, la línea NDAC se pone a nivel alto (no activa). Esta línea permanece a nivel bajo hasta que el último (más lento) de los destinatarios participantes en la comunicación acepta el dato.

En la figura 3.15 se representa un diagrama de tiempos que ilustra el funcionamiento de estas líneas de control de protocolo. Cuando el *talker* ha situado el dato en el bus, lo indica enviando el mensaje DAV (dato válido). Los *listeners*, al detectarlo, ponen activa la línea NRFD y, tras tomarse el tiempo necesario, van indicando que han aceptado el dato, poniendo su línea NDAC a nivel alto. El bus realiza una operación AND-cableada con todas las NDAC, lo que no permite al *talker* recibir un nivel alto hasta que *todos* los *listeners* así lo indican.

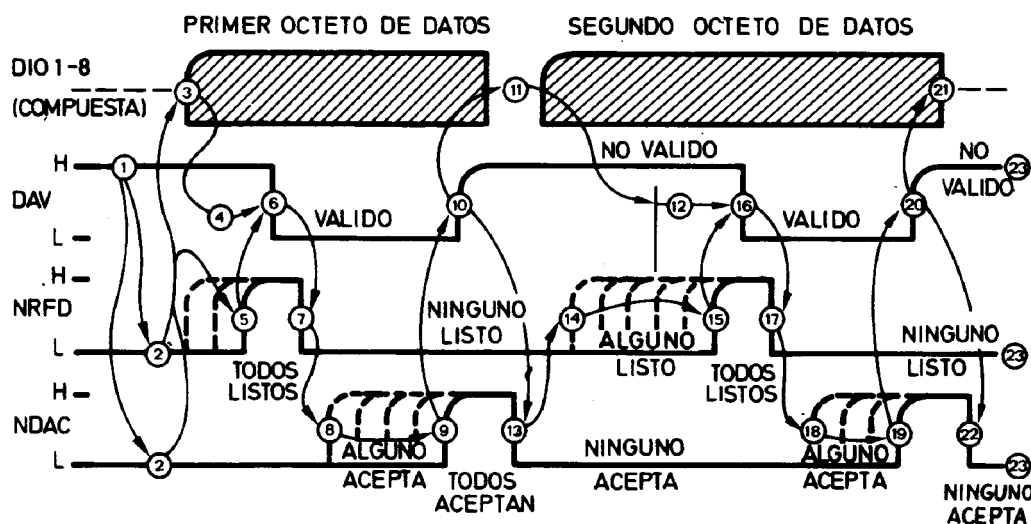


Fig. 3.15.- Cronograma de las señales empleadas en el protocolo de manipulación de datos cuando hay un *talker* y varios *listeners*.

Cuando el *talker* detecta el nivel alto en la línea NDAC (todos los datos aceptados), retira del bus el mensaje DAV (línea a nivel alto). Al detectarlo los *listeners*, ponen a nivel bajo la línea NDAC (ningún dato aceptado) y a nivel alto la línea NRFD (listos para recibir datos). Para que el *talker* detecte la línea NRFD a nivel alto, deben ponerla así *todos* los *listeners*, pues el bus hace una operación AND-cableada con esta señal. Cuando el *talker* recibe un nivel alto en NRFD, se inicia nuevamente el proceso.

Para comprender mejor el funcionamiento de las líneas de protocolo, también puede darse significado a su nivel alto. Así, si NDAC significa a nivel bajo "ningún dato aceptado", a nivel

alto puede significar “dato aceptado” (DAC). Lo mismo sucede con NRFD: a nivel bajo “no preparado para recibir datos”, a nivel alto “preparado para recibir datos” (RFD).

- *Líneas para control general del bus*: Se trata de cinco líneas utilizadas para manejar o controlar el flujo de información a través del *interface*. A continuación se hace una breve descripción de cada una de ellas.

* ATN (*Attention*)

A través del bus GPIB se envían dos tipos de mensajes:

- 1.- Mensajes dirigidos al *interface*: Mensaje de control de interés general.
- 2.- Mensajes dirigidos al instrumento: Comando específicos de cada instrumento.

La línea ATN indica el tipo de mensaje que se está transmitiendo en cada momento; cuando ATN está a nivel alto (no activa) se están transmitiendo datos al instrumento, mientras que si ATN está a nivel bajo, lo que se está transmitiendo son mensajes dirigidos al *interface* (bus).

Esta línea es activada y desactivada por el controlador del sistema. En la figura 3.16 podemos observar como el mensaje “penetra” más o menos en el dispositivo GPIB (instrumento) dependiendo del valor que tome la línea ATN.

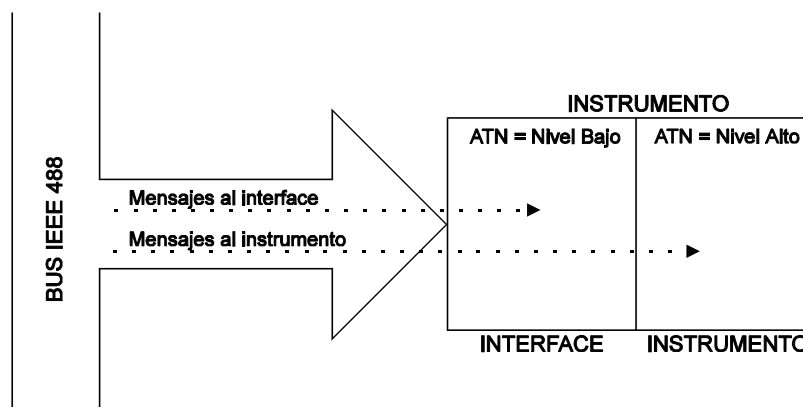


Fig. 3.16.- Actuación de la línea ATN.

* IFC (*Interface clear*)

Esta línea *solamente la puede activar el controlador del sistema* y la utiliza para paralizar todas las operaciones que se estén realizando en ese momento sobre el bus, es decir, inicializa el bus. Además todos los dispositivos dejan de ser direccionados. Todos los dispositivos deben controlar IFC constantemente y responder antes de 100 μ seg. Constituye el primer nivel de reset.

* REN (*Remote enable*)

Esta línea sólo la puede activar el controlador del sistema y la utiliza para poner a los dispositivos en modo de programación remota o local. Cuando la línea se pone a nivel bajo, todos los *listeners* que tengan la capacidad o función para trabajar en modo remoto serán activados para trabajar en dicho modo cuando se les direcciona. Cuando la línea pase a nivel alto, todos los dispositivos volverán a modo local, es decir, deben usar la información recibida del

panel frontal o local. Todos los dispositivos capaces de funcionar en ambos modos (remoto y local) deben controlar ésta línea constantemente y deberán responder antes de 100 μ seg.

* SRQ (*Service request*)

El controlador del sistema es el único dispositivo que puede iniciar la transferencia de mensajes por el bus GPIB. Ningún instrumento puede enviar un mensaje si anteriormente no ha sido requerido para ello por el controlador. A pesar de ello, hay muchos casos que requieren la iniciativa de un instrumento, por ejemplo, cuando se acaba de realizar una medida, o cuando se ha producido un error. En estos casos el instrumento activa (nivel bajo) la línea SRQ para comunicar al controlador estas situaciones.

Una vez detectada la activación de esta línea, el controlador realiza un sondeo (*polling*) para determinar cual ha sido el dispositivo que ha solicitado la interrupción y porqué. Al realizar el sondeo la línea SRQ se desactiva.

* EOI (*End or identify*)

Esta línea tiene dos propósitos:

1.- Cuando ATN está a nivel bajo (mensaje para la *interface*), EOI es utilizada por el controlador para realizar un sondeo paralelo, pidiendo a los dispositivos que han solicitado servicio que se identifiquen.

2.- Cuando ATN está a nivel alto (mensaje para el instrumento direccionado), la línea EOI es utilizada por el *talker* activo (emisor) para indicar que es el último byte del mensaje de datos. Es el modo de indicar el fin de la transmisión de un bloque de datos.

Direccionamiento

Como se ha comentado anteriormente, los equipos conectados al bus, deben ser capaces de, en unas ocasiones, transmitir la información (*talker*), y en otras, recibirla (*listener*). El encargado de indicar la función de los instrumentos en cada momento es el controlador (*controller*) del sistema a través del direccionamiento.

Todos los dispositivos GPIB tienen al menos una dirección primaria como *talker* y como *listener*, excepto que sean totalmente transparentes para el bus o que sean *talk* o *listen only*. El direccionamiento de equipos es utilizado por el controlador para especificar quien “habla” (mediante la dirección de un *talker*) y quien o quienes “escuchan” (mediante la dirección de uno o varios *listeners*).

Para codificar una dirección primaria GPIB se utilizan siete bits (1 - 7). El equivalente decimal de los cinco bits de menor peso determinan la dirección primaria del dispositivo y los bits sexto y séptimo se emplean para distinguir las direcciones de *listeners* y/o *talker* del mismo. Así cuando el sexto bit es “1” y el séptimo “0”, el dispositivo está siendo seleccionado como *listener* (escucha, oyente, receptor); en caso contrario (bit 7 = 1 y bit 6 = 0) el dispositivo se selecciona como *talker* (habla, emisor).

Cuando no se da ninguna de las dos combinaciones anteriores para los bits sexto y séptimo, o bien se trata de una dirección secundaria (1 1), o bien el mensaje no es de selección de dispositivo (como *talker* o *listener*).

Para seleccionar un dispositivo como *talker* o como *listener*, ya sea para iniciar un diálogo, ya para transmitirle a continuación un mandato selectivo, se emplean los *mensajes de dirección*.

Todos los dispositivos que tienen la función que les permite actuar de *talker* tienen una dirección de *talker* distinta. Todos los que tienen la función de *listener* tienen su propia dirección de *listener*. Los dispositivos que tienen las dos funciones, tienen la misma dirección en ambas (se distinguen por los bits sexto y séptimo).

El controlador puede enviar cuatro mensajes de dirección distintos y, lógicamente, al enviar alguno de ellos, ATN = 0 (mensajes dirigidos al *interface*).

* MLA (*My Listen Address*)

Cada uno de los dispositivos conectados al bus detectan, por los bits 6 y 7 de los datos, que la dirección es de oyente. La dirección propiamente dicha la constituyen los bits 1 a 5 del bus de datos.

Puede haber más de un dispositivo con la misma dirección de oyente.

* UNL (*Unlisten*)

La dirección de oyente más alta de las posibles, 11111, no corresponde a ningún dispositivo, y es interpretada por todos los dispositivos, que estaban seleccionados como oyentes, como una orden para que dejen de atender a los mensajes transmitidos por el bus.

* MTA (*My Talker Address*)

Cuando el controlador envía por el bus una dirección de locutor, el dispositivo al que corresponde esa dirección queda seleccionado como locutor para futuras comunicaciones a través del bus. El dispositivo que estaba actuando como locutor (*talker*) detecta que se ha transmitido otra dirección de locutor y deja de actuar como tal.

* UNT (*Untalk*)

La dirección de *talker* más alta de las posibles, 11111, es interpretada por los dispositivos capaces de hablar, como una orden para dejar de transmitir mensajes.

Esto sería una muestra de los mensajes que pueden circular por el bus ya que una relación completa de estos sería muy extensa, siendo conveniente, si se desea conocerla, remitirse a la norma IEEE 488.2 donde se desarrolla pormenorizadamente cada uno de ellos.

Hay varias firmas que fabrican chips que están especialmente diseñados para trabajar como *interface* con el bus GPIB (norma IEEE 488.2) y que permiten establecer de manera fácil y eficaz la comunicación entre el bus y el instrumento o dispositivo propiamente dicho. Esto evita diseños complicados y simplifica considerablemente la programación.

Quizás el más avanzado de estos chips sea el TNT 4882 de la firma National Instruments que ofrece funciones automatizadas que disminuyen el flujo de comandos a través del bus, así como drivers para que las señales mantengan el nivel adecuado para su correcta interpretación. Además ofrece compatibilidad con otros chips tales como el NEC μ PD7210 o el TI TMS9914A de amplio uso en el mercado de la instrumentación programable.

A fin de combinar el IEEE 488.2, el lenguaje de sistemas para test y medida (TMSL) de Hewlett-Packard y el formato de intercambio de datos analógicos (ADIF) de Tektronix, en 1990 se elaboró el último gran estándar de la industria de la instrumentación, el SCPI (*Standard Commands for Programmable Instruments*).

El nuevo lenguaje SCPI simplificó la programación de los instrumentos, tanto a fabricantes como a usuarios finales, al no tener que conocer diferentes conjuntos de instrucciones según el tipo de instrumento. Estos son intercambiables entre líneas de producción y entre fabricantes, por lo que el mantenimiento también es mucho más simple. En la figura 3.11 podemos ver como el estándar SCPI está soportado sobre la norma IEEE 488.2, por lo que llega más allá que ésta última y supone una mejora generalizada en la instrumentación programable.

3.4.2.2.- El estándar VXI

Una alternativa interesante al GPIB, como gran estándar entre los sistemas de instrumentación, es su coetáneo (al igual que el IEEE 488.2 también fue presentado oficialmente en 1987) conocido por VXI.

Básicamente, la *interface* VXI se presenta como una plataforma que a priori combina lo mejor de los sistemas basados en tarjetas de adquisición de datos (velocidad) y de los sistemas GPIB (facilidad de uso, entre otros) incorporando nuevos alicientes tales como menor coste que GPIB, menor tamaño, mayor precisión en la temporización y sincronización, utilización de un *backplane* (placa madre) basado en el bus de Motorola VME (cuyos 32 bits proporcionan mayor ancho de banda que GPIB y TAD sobre bus PC/AT), así como mayor facilidad de programación.

A tenor de las ventajas mencionadas, no ha de sorprender que algunos estudios de mercado estimen que en la actualidad el 50% de los sistemas de instrumentación utilizan VXI.

La definición del bus VXI fue realizada a partir de la creación de un consorcio que integra a 10 de los fabricantes líderes en instrumentación (National Instruments, Tektronix y Hewlett-Packard, entre otros).

Si bien las aplicaciones impulsoras del VXI han sido el test y la medida (mediante instrumentos como voltímetros o generadores de funciones), los entornos de adquisición de datos y control aún han de experimentar la consagración definitiva. La clara competencia del bus GPIB no ha permitido al VXI hacer valer de forma clara sus ventajas, a pesar de posibilitar una migración sin traumas desde el estándar GPIB. La arquitectura multiprocesador, modular y fácilmente reconfigurable, permite etiquetar al VXI como un *super instrumento* programable con herramientas estilo GPIB, con velocidades de transferencia elevadas y capacidad de memoria compartida, de forma que facilita la adquisición y procesamiento de múltiples canales en tiempo real.

La configuración de un sistema VXI obedecería a una estructura similar a la que se mostró en la figura 3.7. (b), es decir, el VXI usa un chasis principal (rack de expansión) con un máximo de 13 slots para albergar instrumentos modulares sobre tarjetas. Al estar soportadas todas las tarjetas-instrumentos en un *backplane* (placa madre) VME, también se pueden usar tarjetas o módulos VME en los sistemas VXI.

El ordenador en este caso, estará dotado de una placa a modo de *interface* que permitirá comunicarse con el rack que contiene las tarjetas VXI para seleccionar un instrumento, asignarle su función (emisor, receptor) o simplemente, conocer su estado.

En la figura 3.17 se ha esquematizado la unión de estándares y nuevas características que permiten considerar al VXI como el futuro en lo que a instrumentación programable se refiere.

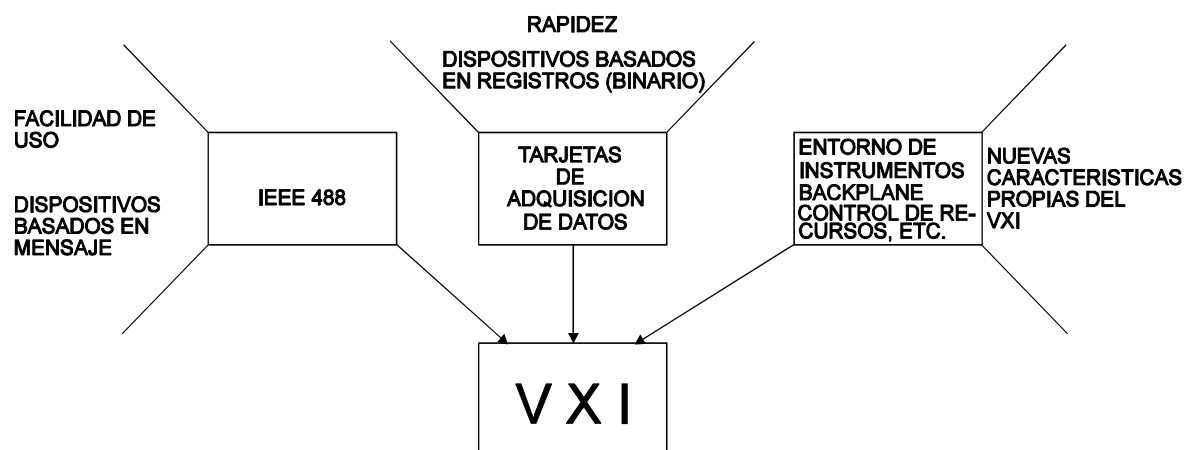


Fig. 3.17.- La tecnología VXI combina lo mejor de otras tecnologías.

3.4.2.3.- Comunicación serie

A pesar de la popularidad de la transmisión serie (entre ordenadores e impresoras, plotters o ratones, por ejemplo), éste estándar de comunicación no está tan extendido como los sistemas GPIB, VXI o basados en tarjetas de adquisición de datos (TAD), entre los equipos de test y medida. Las razones básicas son dos: la E/S serie realiza la transmisión de señales de datos y control bit a bit (de forma asíncrona) por lo que resulta más lenta que los anteriores sistemas. Por otro lado, cada instrumento serie comporta un puerto serie en el ordenador, lo que limita el número máximo de instrumentos integrables en el sistema.

Al igual que el GPIB, los diferentes protocolos de comunicación serie (RS-232, RS-422 y RS-485, por ejemplo) definen un número de líneas para datos, protocolo (*handshaking*) y gestión de flujo de información.

La utilización más habitual de instrumentos programables vía línea serie (fundamentalmente RS-232) se lleva a cabo en el contexto de la analítica, especialmente en equipo de laboratorio y ensayo (cromatógrafos, espectro-fotómetros, etc.).

3.4.3.- SELECCIÓN DEL SISTEMAS DE ADQUISICIÓN DE DATOS (SAD)

Desde hace ya bastantes años, la implantación de sistemas de adquisición de datos y control es un hecho en nuestro país en ámbitos tales como la monitorización y control industrial, la medida y test automáticos o los laboratorios de aplicaciones. Aún así, la adecuación de los equipos adquiridos a las necesidades reales de las aplicaciones dejan mucho que desear, en un porcentaje considerable de casos. La pobreza de criterios en la selección de equipos o la confianza depositada en los suministradores de SAD se ha traducido, en ocasiones, en equipos que han quedado infradimensionados para las aplicaciones que debían asumir en pocos años o bien en sobrecostes injustificados.

La selección rigurosa de esta clase de equipos requiere (atendiendo a la aplicación) un conocimiento diáfano en aspectos relacionados con áreas tales como transductores y actuadores, acondicionamiento de señales, hardware de adquisición y control, *interfaces* de comunicaciones y software.

Por otra parte, deben concretarse a priori las especificaciones que demanda la aplicación, las ampliaciones presumibles que deberá atender el sistema a corto y medio plazo, el nivel de integración que puede haber con otros equipos ya instalados y el perfil del operador previsto para el sistema.

Desde las aplicaciones más simples, usualmente resueltas mediante TAD conectadas directamente al bus del ordenador (equipos fuertemente acoplados) hasta las aplicaciones basadas en instrumentos programables (sistemas distribuidos) tipo GPIB, VXI o serie, merecen un análisis riguroso que redundará en un ahorro económico final a partir de soluciones ajustadas a las necesidades reales.

En general, la especificidad de cada gama de aplicaciones conlleva un grupo de condicionantes que acotan el tipo de SAD a escoger. Las características que delimitan el sistema pueden ser de tipo *físico*: equipo estático o portátil; *eléctrico*: alimentación a red o con batería; *hardware*: tipo de ordenador predeterminado; *software*: sistema operativo o software de aplicación definidos; *empresarial*: tipo de ordenador que deberá manejar el equipo, disponibilidad económica, etc.

En cualquier caso, cabe plantearse desde un principio los grados de libertad existentes en la especificación del ordenador, del hardware de adquisición y del software, a fin de afrontar las aplicaciones presentes y futuras con exactitud y rigor, pero sin sobredimensionar innecesariamente el SAD.

Igualmente, deben concretarse o prever el número y tipo de señales de entrada que vayan a ser objeto, en su caso de medida, así como la posible necesidad de acondicionamiento de señal, el número de canales de E/S digitales necesarios para el conjunto de aplicaciones, la posibilidad de incorporar slots de expansión y la velocidad de proceso, entre otros aspectos. En cualquier caso, no resulta extraño que la solución óptima en la elección de un SAD cuyos requerimientos no son diáfanos se haga «por aproximaciones sucesivas», realizando un refinamiento progresivo de las partes del equipo a medida que las aplicaciones van acotando sus requerimientos. En tal

caso, un margen de sobredimensionamiento del equipo (especialmente en el ordenador y en el hardware de adquisición) no es una medida desdeñable, siempre que sea razonable.

3.5.- ESTUDIO DE UNA TARJETA DE ADQUISICIÓN DE DATOS

La amplia difusión de los ordenadores personales que se ha alcanzado en la actualidad, unido a las posibilidades que estos ofrecen (capacidad de operación, posibilidad de expansión, interrupciones, acceso DMA, facilidad de programación prácticamente en cualquier lenguaje, velocidad de trabajo, etc.), hacen que las tarjetas de adquisición de datos sea la forma más extendida actualmente de construir un sistema de instrumentación o una estación de adquisición de datos y control.

Son varios los fabricantes de TAD y muchos los modelos de tarjetas que ofrece cada uno. Así, entre los que específicamente se dedican a la fabricación de TAD o éstas son su principal campo industrial se encontrarían National Instruments, Intelligent Instrumentation, Iotech, Data Translation y Advantech. Todos estos fabricantes ofrecen tarjetas con enormes posibilidades y muy sofisticadas, así como aquellas que siendo más simples cubren un vasto campo para aplicaciones de laboratorio y enseñanza. En cualquier caso, algunos de los aspectos que pueden servirnos para clasificar una TAD o valorar sus posibilidades serían:

- Tipo de bus de ordenador para el que está diseñada (XT, AT, EISA, NuBUS, PCI, etc.).
- Número de canales analógicos de entrada (modo común y diferencial) y de salida.
- Número de canales digitales de entrada y de salida.
- Velocidad de muestreo (global o por canal).
- Resolución de los DAC/ADC.
- Rango dinámico para los ADC.
- Ganancia variable para los amplificadores de entrada.
- Contadores y temporizadores.
- Buses auxiliares para sincronización con otras TAD.
- Programación a bajo y/o alto nivel.
- Incorporación de DSP (Procesador de señales digitales).
- Otros circuitos para aplicaciones específicas (ASIC) que faciliten aspectos de uso y explotación.
- Consumo.
- Entorno de programación (Windows, Mac, Linux, etc.).
- Etcétera.

Como el simple hecho de analizar una TAD de cada uno de los fabricantes supondría salirse de la idea que este tema pretende ofrecer, se ha elegido para estudiar como modelo de tarjeta de adquisición de datos la denominada Lab-PC+ del fabricante National Instruments.

Haciendo una descripción de carácter general, podemos decir que la TAD Lab-PC+ es una tarjeta de bajo costo y multifunción con I/O analógicas, digitales y temporizadores para PC. Tiene un ADC de 12 bits con 8 entradas analógicas configurables como 8 entradas en modo común o 4 entradas diferenciales, dos DAC de 12 bits con salida en tensión, 24 líneas de entrada/salida digital compatibles TTL, y tres canales a disposición del usuario que pueden ser

utilizados cada uno como contador/temporizador de 16 bits para temporización de entrada y salida. La *interface* con el bus del PC tiene un bus de datos de 8 bits con circuitería para interrupciones y DMA.

En cuanto a las aplicaciones de esta tarjeta, se puede decir que la Lab-PC+ es ideal para trabajar en laboratorio y en entornos industriales y académicos. La entrada analógica multicanal es muy útil en análisis de señal, registro de datos, medida de temperatura y medida de tensiones continuas. Por otro lado, los canales de salida analógica pueden generar señales de estímulo experimentales, señales para control de máquinas y procesos, además de actuar como generador de funciones. Las 24 líneas de I/O compatibles TTL, pueden utilizarse para la conmutación de dispositivos externos tales como transistores y relés de estado sólido, para lectura del estado de circuitos digitales externos y para generar interrupciones. Los tres canales temporizador/contador se pueden usar para sincronizar eventos, generar pulsos y medir frecuencia y tiempo. Todo este cúmulo de posibilidades permite decir que un PC equipado con la TAD Lab-PC+ es una plataforma versátil con buena relación precio/prestaciones para test de laboratorio, medida y control.

3.5.1.- ANÁLISIS POR MÓDULOS DE LA TAD LAB-PC+

La figura 3.18 muestra el diagrama de bloques de los elementos que forma la Lab-PC+. Para estudiar los distintos elementos que forman esta tarjeta se ha dividido en 5 partes atendiendo a su funcionalidad:

- 1.- Módulo de *interface* con el bus del PC.
- 2.- Módulo para adquisición de señales analógicas.
- 3.- Módulo para salidas analógicas.
- 4.- Módulo para I/O digital.
- 5.- Módulo de temporización.

La numeración dada en la lista anterior coincide con la asignada a cada módulo en la división que se ha hecho en el diagrama de bloques de la figura 3.18.

Para interconectar los diferentes elementos, la tarjeta dispone de dos buses internos, uno de datos y otro de control. Pasemos a continuación a realizar un análisis más detallado de cada una de las partes en las que se ha estructurado el estudio de la tarjeta.

3.5.1.1.- Módulo de *interface* con el bus del PC

Los circuitos que forman este módulo se encargan de gestionar la conexión de la Lab-PC+ con los buses del PC (datos, direcciones y control) y está formado por los siguientes circuitos:

- Latches y decodificador de direcciones.
- Buffer de datos.
- Circuitos de temporización.
- Circuito para control de interrupciones.

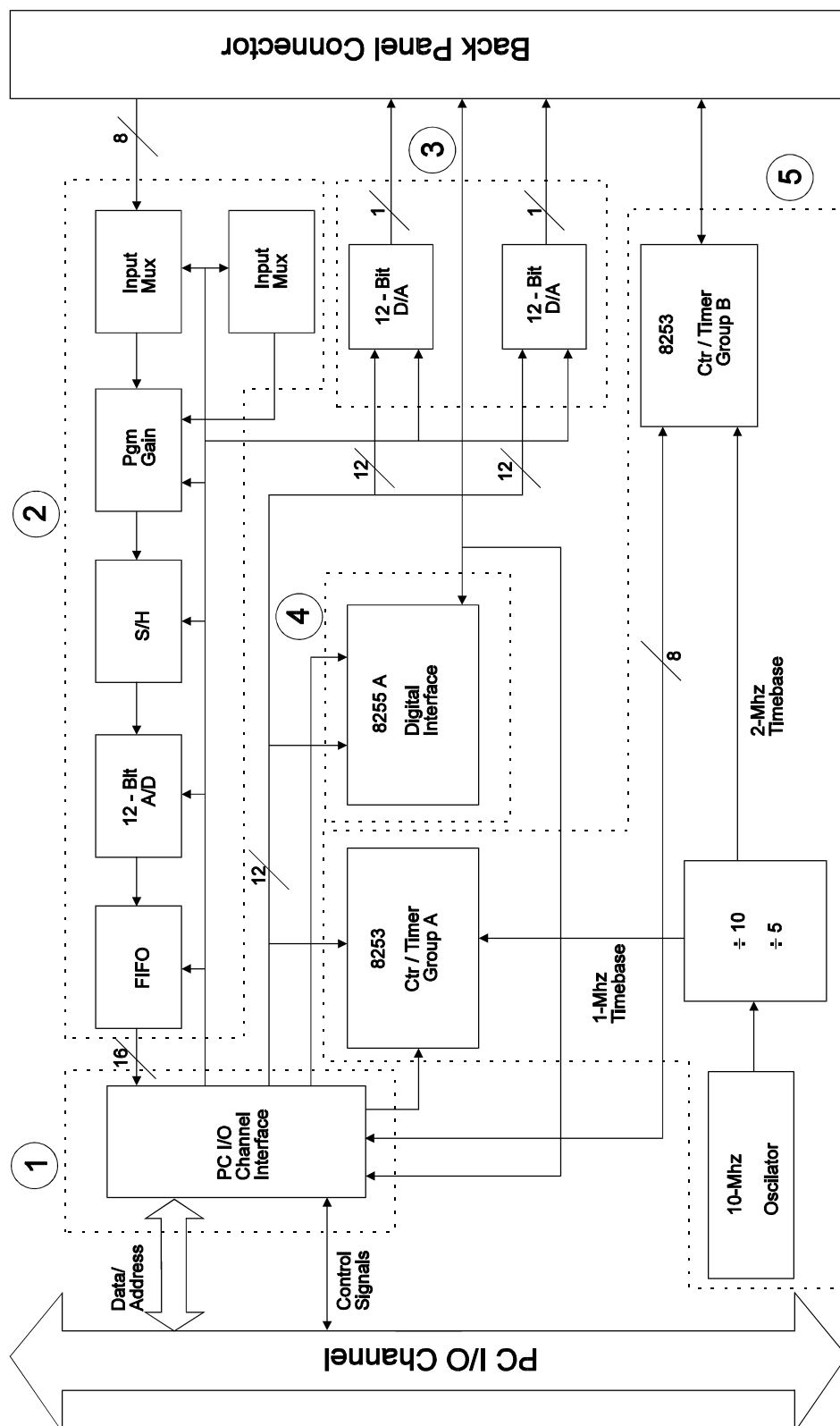


Fig. 3.18.- Diagrama de bloques de la TAD Lab-PC+

- Circuito para control DMA.

El circuito de direcciones monitoriza las líneas de dirección SA5 a SA9 del PC para generar la señal de selección de tarjeta; las líneas SA0 a SA4 junto con las señales de temporización se utilizan para generar las señales de selección y lectura/escritura de los distintos registros de la tarjeta.

Los buffers de datos controlan el sentido de estos en las transferencias bidireccionales, dependiendo de si dicha transferencia es una lectura o una escritura.

El circuito de control de interrupciones se encarga de encaminar hacia el PC la petición de interrupción que puede generar la tarjeta. La Lab-PC+ puede utilizar seis líneas de interrupción, desde IRQ3 hasta IRQ9 (excepto IRQ8). La interrupción que viene prefijada es la IRQ5, pudiéndose cambiar por cualquiera de las anteriores mediante un *jumper* incorporado en la tarjeta. Se debe tener cuidado especial con la IRQ6, ya que la mayoría de los PC la utilizan para la controladora de la disquetera.

Algunas de las causas por las que la tarjeta realiza una petición de interrupción son:

- Cuando el resultado de una conversión está disponible.
- Cuando se produce un overflow.
- Cuando alguno de los puertos de I/O digitales está preparado para realizar una transferencia de datos.

Cada una de las “causas” de interrupción puede activarse y desactivarse individualmente a través de un registro interno de la tarjeta.

El circuito de control DMA genera una petición de servicio DMA si el resultado de una conversión A/D está disponible en la memoria FIFO. La tarjeta Lab-PC+ soporta transferencias DMA de 8 bits. Los canales DMA 1, 2 y 3 del PC están disponibles para estas transferencias, viniendo preseleccionado el 3. Igual que para las interrupciones, el canal DMA puede cambiarse mediante un *jumper*.

3.5.1.2.- Módulo para adquisición de señales analógicas

Como puede observarse en la figura 3.18, este módulo es fundamentalmente una cadena de medida: multiplexores analógicos, amplificador de instrumentación de ganancia programable, S/H, conversor A/D y memoria. Todos estos elementos irán sincronizados convenientemente, de modo que la adquisición se lleve a cabo en los términos de máxima precisión y rigor. Analicemos las principales características de los circuitos que forman la cadena:

- a) Multiplexores analógicos: La entrada del circuito de adquisición de señales analógicas está formada por dos multiplexores analógicos CMOS (74HC4051) de 8 bits, con la finalidad de proporcionar ocho canales de entrada simple, *single-ended*, (entradas 0-7 del primer multiplexor) o bien cuatro diferenciales (entradas 1-3-5-7 del segundo

multiplexor). Las entradas de los multiplexores están protegidas contra sobretensiones de hasta $\pm 45\text{V}$, independientemente de que el sistema esté o no alimentado.

b) Amplificador de instrumentación de ganancia programable: La finalidad de este circuito es amplificar la señal de entrada de forma que el margen dinámico de su señal de salida se asemeje lo máximo posible al margen dinámico del ADC, aumentando de este modo la resolución y precisión de la conversión. Así mismo, este amplificador proporciona un elevado rechazo al modo común y una alta impedancia de entrada. La ganancia del amplificador es seleccionable mediante software, pudiendo tomar uno de los siguientes valores: 1, 1.25, 2, 5, 10, 20, 50 y 100.

c) Circuito de muestreo y retención (S/H): El circuito S/H, como se sabe, tiene la finalidad de tomar una muestra de la señal y mantenerla durante el tiempo que dura la conversión A/D. Este circuito es necesario siempre que la señal de entrada sufra variaciones considerables durante el tiempo que dura la conversión.

d) Conversor A/D: La tarjeta utiliza un ADC de 12 bits de aproximaciones sucesivas (ADC 774). El margen dinámico de este conversor es de 0-5V, aunque mediante hardware adicional se pueden conseguir márgenes de $\pm 5\text{V}$ o de 0-10V.

El ADC proporciona su salida en dos códigos: binario y complemento a dos (C-2), dependiendo de si se ha seleccionado modo de entrada unipolar o bipolar. En modo unipolar, el código de salida es binario de doce bits (0-4095), mientras que si se ha seleccionado modo bipolar la salida es en C-2 (-2048, +2047).

e) Memoria FIFO: Cuando se completa una conversión A/D, el resultado se almacena en una memoria FIFO (*First Input, First Output*). La capacidad de la memoria FIFO es de 16 palabras de 16 bits. La FIFO sirve de buffer al ADC y proporciona dos ventajas importantes:

- * Una vez que la conversión A/D ha finalizado, el resultado es almacenado en la FIFO, pudiendo de esta forma comenzar una nueva conversión.

- * En la FIFO pueden almacenarse hasta 16 muestras antes de que se pueda empezar a perder información, así que el software puede, durante este tiempo (16 veces el intervalo de muestreo), realizar otras tareas.

Si antes de leer la FIFO se realizasen más de 16 conversiones, se produciría un error de *overflow*, indicándonos que se ha sobrepasado la capacidad de la memoria con la consiguiente pérdida de información. Si en el registro correspondiente de la tarjeta está “habilitada” la interrupción de *overflow*, ésta pasará al PC, avisando de lo sucedido.

La FIFO genera una señal para indicar que contiene información y el estado de esta señal puede conocerse consultando un registro interno de la tarjeta.

f) Circuito de temporización: Una operación de adquisición consiste en realizar sucesivas conversiones a intervalos de tiempo (periodo de muestreo) cuidadosamente sincronizadas. Por otra parte, la TAD permite dos tipos de adquisición: adquisición de

un único canal y adquisición de múltiples canales (modo secuencial o escaneado), pudiéndose a su vez realizar ambas en dos modos diferentes, modo continuo o a intervalos. Cuando se realiza una adquisición de múltiples canales debe haber alguna señal que indique cuando se debe conmutar de canal, y si además la adquisición se hace en modo intervalo también será necesario indicar el comienzo y final de cada intervalo.

Como vemos, la temporización en este módulo es fundamental, y de ella depende en gran medida que el sistema de adquisición funcione con precisión. Los relojes necesarios para la temporización de la Lab-PC+ se obtienen a partir de dos *timers* programables 82C53 de Intel. Concretamente, y para el módulo de la adquisición de señales analógicas, el utilizado es el denominado Group A en la figura 3.18.

3.5.1.3.- Módulo para salidas analógicas

La Lab-PC+ proporciona dos canales analógicos de salida basados en dos DAC de 12 bits (DAC 813 de Burr-Brown), tal como puede observarse en el diagrama general de la tarjeta (figura 3.18).

Cada DAC genera una tensión analógica de salida proporcional a la tensión V_{REF} y al código digital aplicado en su entrada. El código digital de entrada de cada conversor se carga en cuatro registros (DAC0L, DAC0H, DAC1L, DAC1H) y estos pueden ser actualizados de varios modos: sincronizados con señales internas, externas o por simple carga de nuevos datos.

Las tensiones de salida en cada canal analógico pueden ser bipolares ($\pm 5V$) y unipolares (0-10V), eligiéndose el margen deseado mediante jumpers.

3.5.1.4.- Módulo para I/O digital

El módulo para I/O digital se basa en el circuito integrado 8255 de Intel. Este chip es un PPI (*Programmable Peripheral Interface*) de propósito general que entre otras cosas, posee 24 pins de I/O digitales programables.

Además, este versátil circuito es completamente compatible TTL, posee unas excelentes características para temporización y tiene la posibilidad de hacer set/reset sobre un bit directamente. Todo ello hace del 8255 uno de los *interfaces* más utilizados en múltiples aplicaciones de electrónica.

3.5.1.5.- Módulo de temporización

Tal y como se comentó anteriormente, la TAD Lab-PC+ dispone de dos *timers* programables 82C53. Uno de ellos (Group A), lo utiliza para realizar toda la temporización necesaria en las operaciones de adquisición, mientras que el otro, Group B, queda disponible para operaciones de propósito general, aunque también se utiliza para realizar alguna operación complementaria en la temporización de la tarjeta.

Cada 82C53 está formado por tres contadores independientes de 16 bits más un registro de control de 8 bits. Posee varios modos de trabajo programables y puede contar en BCD o en binario.

El conector que pone en contacto a la Lab-PC+ con el mundo exterior y que permite la conexión con los elementos sensores, actuadores u otros circuitos externos, está dotado de 50 conexiones y en él aparecen tanto las entradas y salidas citadas anteriormente (analógicas y digitales) como sus respectivas masas. Además cuenta con entradas de disparo, de sincronismo, etc.

En definitiva, la Lab-PC+ es una tarjeta de adquisición de datos con múltiples posibilidades y cuyo conocimiento requiere algo más que las páginas aquí dedicadas, aunque con lo expuesto se puede formar una idea bastante aproximada de lo que es capaz e incluso de lo que se puede llegar a conseguir.

3.5.2.- PROGRAMACIÓN DE LA TAD LAB-PC+

La programación de la tarjeta puede realizarse, en general, de dos formas: programación a *bajo nivel* y programación a *alto nivel*.

- Programación a bajo nivel: Consiste en programar la tarjeta accediendo directamente a su estructura de registros internos (un total de 28). Cada uno de estos registros controla distintos aspectos de los circuitos de la tarjeta. Si por ejemplo deseamos seleccionar una ganancia de 2 para el amplificador de instrumentación, lo único que debemos hacer es acceder al registro denominado “Command Register 1” y escribir en los bits 4-5-6 el código correspondiente según indica el fabricante, y que en este caso (ganancia 2), sería 0-1-0.

Este método de programación tiene la ventaja de que permite utilizar todos los recursos hardware de la tarjeta según creamos conveniente, disponiendo por tanto de una gran flexibilidad a la hora de realizar nuestros programas. Se puede decir por tanto, que este método permite obtener el máximo rendimiento de la tarjeta. Por otro lado, presenta el inconveniente de que para realizar la programación, es imprescindible tener un perfecto dominio del hardware de la tarjeta y de su funcionamiento, ya que cada registro tiene asignada una dirección I/O a partir de la dirección base de la tarjeta, que por defecto es la 260H. Por tanto, los registros están mapeados como puertos de I/O y en los compiladores de C habrá que utilizar funciones del tipo `inp()`, `outp()`, `inpw()`, `outpw()`, etc.

- Programación a alto nivel: Al adquirir la tarjeta, el fabricante proporciona unas librerías que incorporan una serie de funciones para distintos lenguajes de programación, normalmente C, BASIC y Pascal. Pues bien, este método consiste en programar la tarjeta utilizando las funciones proporcionadas por el fabricante.

Este método presenta la ventaja de que el hardware pasa a ser transparente al usuario. Así, por ejemplo, no es necesario conocer en que dirección se encuentra un determinado registro, ni la función o significado de cada uno de sus bits. Con tener unos

conocimientos básicos sobre los parámetros que intervienen en una adquisición es más que suficiente para poder realizar el programa de adquisición de datos. Tales parámetros serían: número de muestras a adquirir, velocidad de muestreo (muestras por segundo), número de canales, etc.

Este método presenta el inconveniente de que la programación es menos flexible que en el caso anterior ya que las operaciones que podemos realizar vienen impuestas por las funciones. Así, pudiera darse el caso de que deseáramos realizar alguna operación algo especial y no disponer de la función adecuada para ello, cosa difícil si tenemos en cuenta que el conjunto de funciones que suelen proporcionar los fabricantes es muy amplio y permite realizar, prácticamente, cualquier tipo de operación.

3.6.- SOFTWARE PARA LA ADQUISICIÓN DE DATOS

La elección del software asociado al sistema físico (tarjeta, bus de instrumentación, comunicación serie, etc.) comporta tres niveles de decisión básicos: *sistema operativo*, *software a nivel driver* y *software de aplicación*. Cada uno de ellos depende, en gran manera, del tipo o la gama de aplicaciones que van a ser resueltas con el SAD. El enfoque resulta muy diferentes si la aplicación es tipo SCADA (Simulación, Control y Adquisición de Datos), en la cual se trabaja en tiempo real, o de simple monitorización de datos con posibilidad de un análisis off-line.

Por lo que respecta a las soluciones basadas en PC (un elevado porcentaje del total), si bien un tanto por ciento elevado del software de adquisición es únicamente compatible con los sistemas operativos de la familia Windows, la elección final del mismo puede limitar de antemano las prestaciones del SAD, fundamentalmente en sus posibilidades multitarea, en la conexión con otras aplicaciones y en la proximidad que proporcione la *interface* gráfica de usuario (GUI), aspectos cada vez más decisivos tanto en el campo de la automatización industrial como en el contexto de la investigación.

La utilización de sistemas operativos de Microsoft como plataforma sobre PC es, cada vez más, un hecho al ofrecer globalmente mayores prestaciones, tanto en lo referente a la conectividad del software con otras aplicaciones como en la potencia de la *interface* hombre-máquina. No obstante, en aplicaciones de control cuyo requerimiento de velocidad de proceso sea elevado, cabe coordinar eficientemente la opción Windows con un software a nivel *driver* y, obviamente, un hardware adecuados. Se debe tener presente que el entorno Windows está sometido a una penalización apreciable (en tiempo) sobre las operaciones E/S vía DMA y por interrupción. Por este motivo, si la transferencia de datos se realiza mediante una de las dos técnicas anteriores tratando, precisamente, de conseguir una velocidad elevada, la limitación que introduce la plataforma Windows puede paliarse mediante un software a nivel driver (controlador de dispositivo) que realice las funciones de dispositivo virtual Windows durante las tareas de interrupción y de gestión de memoria y bufferización durante los servicios de DMA (como se verá posteriormente, un software que realiza tal función es el NI-DAQ de National Instruments).

Por lo que respecta al software de aplicación, cabe decidir de antemano la alternativa “paquete integrado” versus “paquete a medida”. En la elección influyen, básicamente, los

requerimientos de facilidad de uso, de adaptabilidad a las aplicaciones presentes y futuras, así como la velocidad de proceso. Los paquetes integrados se ajustan a un uso intuitivo, a fin de proporcionar un tiempo mínimo de programación y puesta en funcionamiento. Algunos de los paquetes integrados comerciales permiten adquirir, analizar y monitorizar datos, así como controlar instrumentos. Por el contrario, la facilidad de uso de los mismos acostumbra a llevar asociada una limitación importante en flexibilidad y adaptabilidad.

Los paquetes a medida permiten generar aplicaciones de adquisición de datos y/o control de instrumentación mediante entornos de programación relativamente cómodos (Visual Basic, Visual C++, etc.), aprovechando de forma más eficiente la potencia de una configuración hardware concreta, a la vez que permite adaptarse plenamente a las características de la aplicación, a costa de una mayor exigencia en el nivel del personal encargado de desarrollar (programar) el sistema.

En cualquier caso, en la elección del software de aplicación conviene tener presente que éste debe incluir todos los componentes necesarios para diseñar y desarrollar las aplicaciones, debe soportar plenamente el hardware del sistema, así como permitir crecer la aplicación a medida que las necesidades lo requieran, respetando la operación en tiempo real, en el caso SCADA y, finalmente, proporcionar una *interface* de usuario que se ajuste al tipo de personal que deberá operar el SAD.

3.6.1.- SOFTWARE DE NATIONAL INSTRUMENTS

National Instruments ha desarrollado un software estructurado de tal forma, que permite la integración de una amplia variedad de instrumentos de medida y control electrónicos. Ofrece productos software a varios niveles (software de aplicación, utilidades, drivers de dispositivos, etc.), de modo que forman una arquitectura abierta, en donde se puede elegir el software que mejor se adapte a las necesidades de una determinada aplicación.

La base de todo el software de National Instruments son los controladores de dispositivo (drivers). Este software constituye el nivel más bajo en cualquiera de las cuatro opciones que se ofrecen para el hardware de la adquisición de datos y cada driver incluye librerías de funciones de programar el hardware que se ha de controlar. Tal como puede observarse en la figura 3.19, el hardware que se va a utilizar pertenecerá a uno de los cuatro grupos indicados: Tarjeta de adquisición de datos, GPIB, VXI y serie RS-232. Para cada grupo, National Instruments ofrece su driver respectivo:

- NI-DAQ: Es el software (driver) que actúa de *interface* estándar en la programación de tarjetas de adquisición de datos.
- NI-488.2: Este driver permite el control desde un PC de todos aquellos instrumentos programables que poseen una *interface* GPIB.
- NI-VXI: Mediante este driver se consigue vincular el software de aplicación con la instrumentación VXI.

Estos drivers se ejecutan a nivel del sistema operativo del ordenador, por lo que se puede programar y controlar cualquiera de los productos hardware anteriores mediante lenguajes de programación estándar tales como C o Basic o bien, mediante un paquete de aplicación. Cada uno de los drivers de dispositivo, está diseñado para maximizar la flexibilidad en la programación y la velocidad en la transferencia de datos. La *interface* de programación de cada uno de los drivers es común, por lo que cualquier programa de aplicación que se escriba, será completamente portable, independientemente del tipo de ordenador y del sistema operativo utilizado por la plataforma.

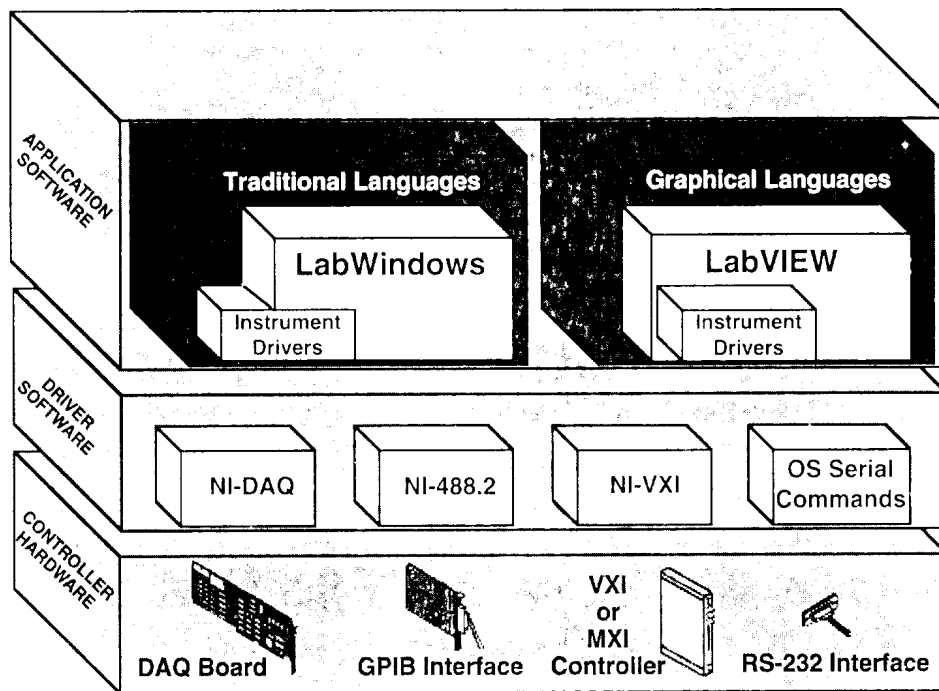


Fig. 3.19.- Arquitectura del software para la instrumentación virtual y la adquisición de datos de National Instruments.

En la parte superior de esta figura, se sitúa el denominado *software de aplicación* y que está constituido por los paquetes denominados LabWindows y LabVIEW, los cuales permiten, además de controlar el hardware de adquisición y control, desarrollar sistemas de *instrumentación virtual*, es decir, organizar un conjunto de instrumentos que sólo existen en la pantalla del ordenador, pero que permiten simular con gran realismo cualquier situación que pudiese darse en un sistema físico de instrumentación, adquisición o control.

LabVIEW y LabWindows ponen a disposición del usuario herramientas para control de instrumentos (incluyen los drivers), adquisición, análisis y presentación de datos. La presentación de datos puede hacerse tanto por pantalla como por una gran variedad de impresoras y plotters. Usando estos dos paquetes de software, se reduce considerablemente el tiempo de desarrollo, permiten crear programas de una elevada calidad y hace que el sistema (real o virtual) sea fácil de modificar y mantener. Según vemos en la figura 3.19, ambos paquetes de software están situados al mismo nivel, es decir, son funcionalmente equivalentes, entonces ¿por qué dos productos?. La respuesta está en el perfil del usuario que vaya a trabajar al frente del sistema de adquisición de datos y control.

Así, la diferencia básica entre LabView y LabWindows reside en la metodología de la programación. El primero de ellos ofrece una programación gráfica que combina lo más reciente de los sistemas operativos (entornos más amigables) con la programación orientada a objetos, permitiendo obtener el compromiso entre la flexibilidad de la programación estándar (C, Pascal) y la mayor rigidez (pero también mayor simplicidad) de los sistemas exentos de programación (combinación de objetos predefinidos).

El segundo, está diseñado para la gran cantidad de programadores en C o Basic y ofrece una mejora en el desarrollo de los programas así como unas herramientas de depuración diseñadas específicamente para la *programación de instrumentos*. De este modo, los programadores de C o Basic pueden desarrollar sistemas de instrumentación completos en poco tiempo, menor que el que necesitarían con entornos de compilación y programación genéricos.

3.6.2.- SOFTWARE DE DATA TRANSLATION

El software de desarrollo de Data Translation se basa en un lenguaje de programación gráfico optimizado para la adquisición de datos, denominado DT VEE y que corre bajo Windows.

Los elementos básicos de cualquier aplicación gráfica creada con DT VEE son los iconos, que representan elementos de entrada, salida, análisis, visualización, etc., y que permiten crear aplicaciones para adquirir, analizar, presentar y compartir datos. En la figura 3.20 podemos observar lo que podría ser un simple ejemplo para adquisición de datos mediante el software DT VEE.

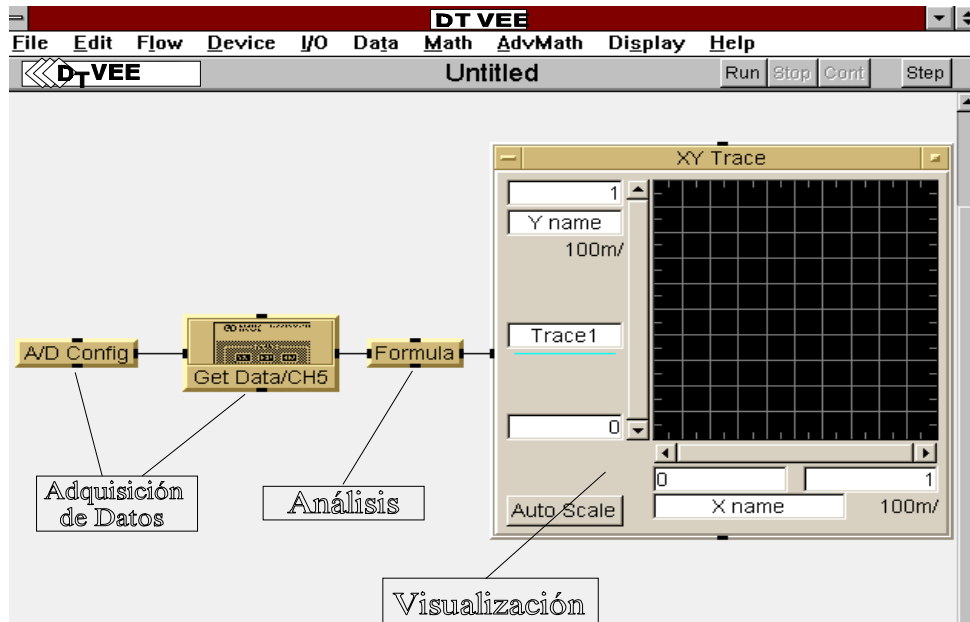


Fig. 3.20.-Ejemplo sencillo de programa de adquisición de datos desarrollado mediante DT VEE.

En dicho ejemplo se pueden distinguir los siguientes pasos:

- Adquisición: mediante un conversor A/D configurable por el usuario. Esta adquisición se dirige a un canal concreto del sistema, el número cinco.
- Análisis: el análisis de los datos se hará mediante la aplicación de un algoritmo o fórmula matemática que obtenga la información deseada de dichos datos.
- Presentación: la presentación permite visualizar sobre un instrumento (un trazador X-Y en este caso) adecuado parte o la totalidad de los datos obtenidos en la entrada del sistemas de adquisición.

La idea del software DT VEE está basada en un *tiempo de aprendizaje nulo*, y aunque ésta idea pueda resultar ambiciosa, en muchos aspectos lo consigue ya que el desarrollo de cualquier programa sigue un procedimiento natural: de donde se adquieren los datos, a través de qué, por medio de qué, como se analizan, sobre qué se representan. Esta orientación del software unido al hecho de utilizar un lenguaje de programación gráfico, permite al usuario una total concentración en el proceso, al contrario que el software basado en lenguajes tradicionales, que requiere también la atención al texto del programa.

DT VEE combina el lenguaje de programación visual HP VEE con la *interface* de programación visual DT VPI. De este modo se puede trabajar con un lenguaje visual muy extendido en el mundo de la instrumentación tal como el HP VEE de Hewlett Packard. La *interface* DT VPI será la encargada de convertir el programa en pantalla a funciones que puedan implementarse sobre las tarjetas de adquisición de datos de Data Translation, permitiendo de este modo un control a bajo y alto nivel de la funcionalidad de la TAD.

Con el fin de independizar el programa visual implementado de la TAD que se esté utilizando, Data Translation establece aún otra capa más en su software: DT-Open Layers. Tal como puede verse en la figura 3.21 esta nueva *interface* software es la que vincula realmente el hardware de adquisición con el software de programación.

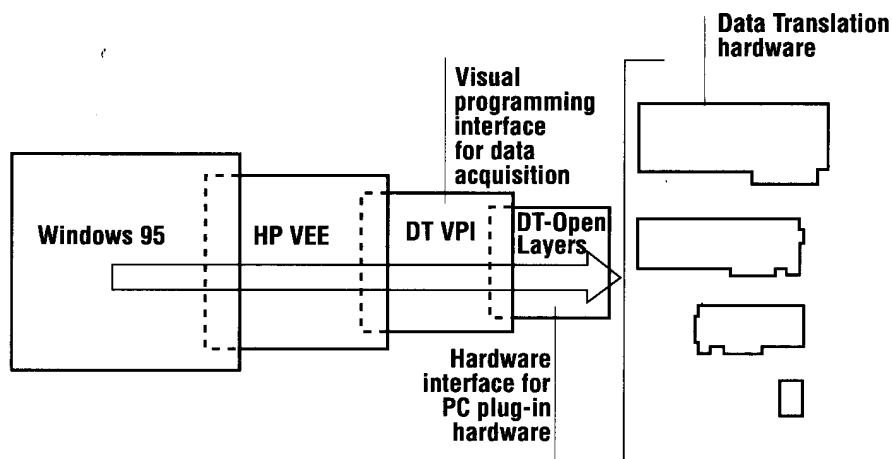


Fig. 3.21.- Estructura y vínculos de los distintos niveles que forman el software de programación DT VEE.

El DT-Open Layers aporta una ventaja esencial para cualquier usuario de las tarjetas de adquisición de Data Translation: asegura la ejecución de un programa sobre cualquiera de las TAD soportadas por DT VEE. Efectivamente, por un lado DT-Open Layers tiene todos los drivers necesarios para profundizar en los registros de la tarjeta de adquisición utilizada y por el otro es capaz de comunicarse con el resto del software. Esto permite al usuario utilizar tarjetas que se adapten a sus necesidades y probar con varios modelos de ellas sin necesidad de tener que cambiar el software desarrollado inicialmente.

Una vez hecha esta pequeña incursión en la estructura del software de adquisición y presentación DT VEE, volvamos al análisis general de lo que sería un programa hecho con este lenguaje visual.

Cualquier función o icono que se necesite, viene (su nombre o etiqueta) incluido en el menú principal y al elegirlo mediante el ratón aparece en la zona de trabajo. Las etiquetas de los iconos pueden ser cambiadas si se desean. Cada icono consta de una serie de conectores repartidos a su alrededor y cuya función depende del lugar que ocupen (figura 3.22). Así, los conectores situados a los lados permiten establecer el sentido del flujo de los datos y los situados en la parte superior e inferior definen la secuencia de operaciones. En el ejemplo mostrado en la figura 3.22 (izqda.), el generador de funciones sólo tiene un terminal para el flujo de datos situado a su derecha y que, obviamente, será de salida. Si dicho generador hubiese de ser incluido en un organigrama que definiese la secuencia de operaciones en el sistema, utilizaríamos los terminales superior e inferior para vincularlo con el elemento anterior y posterior de dicho organigrama.

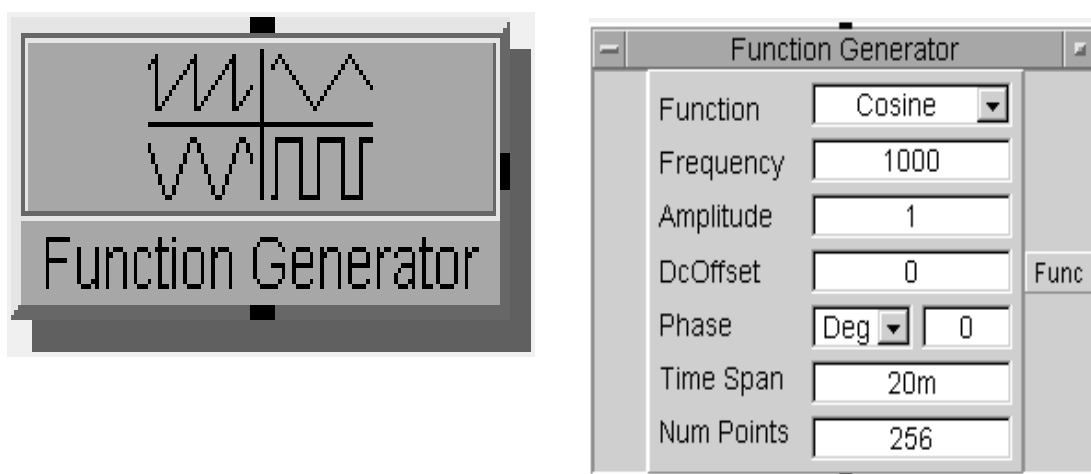


Fig. 3.22.- Un objeto puede ser un simple icono que lo represente (izqda.) o una ventana que permita definir sus parámetros (dcha.).

Hay objetos (iconos) cuyos parámetros pueden ser modificados por el usuario. En este caso, el icono puede ser editado como una ventana que muestra todos los detalles y parámetros que pueden ser alterados. En la figura 3.22 (dcha.) se puede observar como la edición del generador de funciones permite seleccionar el tipo de función, la frecuencia, la amplitud, etc., es decir, todos los parámetros que caracterizan a un instrumento de estas características de cualquier laboratorio en el mundo real.

Si a todas las posibilidades resumidas en los párrafos anteriores le sumamos otras, tales como: facilidad para documentar las aplicaciones, creación de cuadros de diálogo personalizados, diseño de nuevos instrumentos e iconos, etc., podemos sacar la conclusión de que estamos ante un software potente y de fácil uso, que no requiere programadores expertos para sacarle un buen rendimiento a cualquier sistema de adquisición de datos, virtual o real.

El paquete DT VEE soporta, como es natural, la mayoría de las tarjetas de adquisición de la firma Data Translation sin necesidad de añadir drivers especiales. Mediante el uso de este software y de la TAD adecuada, se puede obtener un sistema de adquisición y control de muy fácil manejo, con un tiempo empleado mínimo y con la ya citada ventaja de una alta portabilidad de los programas y un fácil mantenimiento de éstos.