

APÉNDICE B

INTERFASE DE PROPÓSITO GENERAL PARA INSTRUMENTACIÓN, GPIB¹

B.1 Introducción

La interfase GPIB ha sido normalizada en América como IEEE-488. Este standard internacional surgió ante la necesidad de unificar el interconexión y la comunicación de varios instrumentos provenientes de distintos fabricantes. A medida que las mediciones van tomando complejidad, ya sea por la necesidad del registro de los datos obtenidos así como para el control y la observación de otros parámetros, se impone la condición de intercomunicar instrumentos utilizando un lenguaje en común, así como de implementar un sistema automático de adquisición, sin supervisión humana.

Como consecuencia de la normalización, la técnica de mediciones automatizadas se ha ido incrementando rápidamente en los últimos años. Hasta hace poco tiempo atrás el hecho de combinar diferentes instrumentos de medida en un sistema solía traer aparejados diversos problemas prácticos, ya que cada fabricante tenía sus propias convenciones para el tratamiento remoto de señales de control y datos, así como sus propios conectores.

En una aplicación dada puede ser necesario implementar un sistema versátil, que pueda ser utilizado en diferentes proyectos. Los instrumentos para ese “amplio uso” deben ser flexibles y fáciles de interconectar y desconectar. Si se decidiera el uso del instrumento más apropiado para cada tipo de medida no sería extraña la necesidad de recurrir a diferentes fabricantes.

¹ General-Purpose Interface *Bus*

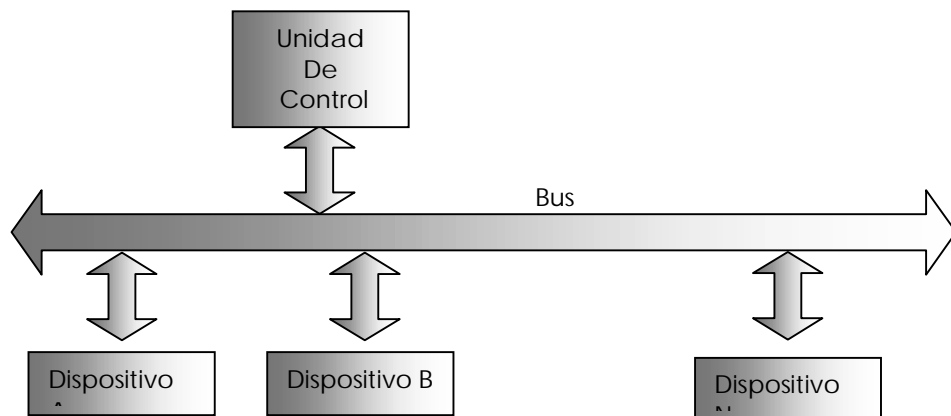
También existen denominaciones equivalentes: HP-IB = Interfase de *Hewlett Packard*; IEEE-488 = Standard Americano; IEC 625 = Standard Europeo

La automatización de un sistema de tales características sería harto complicada ya que habría serios problemas de interfase y resolverlos consumiría mucho tiempo y dinero. Estos problemas aumentan si además se pretende conectar y comunicar computadoras e instrumentos.

La única manera de franquear estas dificultades es la introducción de una interfase normalizada internacionalmente. Para hacer efectiva esta solución todos los fabricantes de instrumentos de medida, registro y control deben adaptarlos a dicha interfase normalizada.

El IEC (*International Electrotechnical Commission*) define la interfase normalizada para instrumentos de medida. La base o punto de partida fue la interfase ASCII que *Hewlett-Packard* diseñó para sus propios instrumentos. La misma se elaboró con el agregado de algunos detalles y precisando su definición.

El resultado de este acuerdo internacional fue la publicación 625-1 del IEC “*Standard Interface Systems for programmable measuring equipment*”. Este documento describe una interfase de conexión de varios instrumentos, programables o no con el fin de integrar un sistema de medida. Los requerimientos mecánicos, eléctricos y funcionales en los instrumentos habilitados para su interconexión en esta interfase están definidos en dicho documento como convenciones de código y formato normalizado.



- Fig. 1 Principio de funcionamiento de la interfase GPIB -

La organización de la interfase IEC está basada en un sistema de estructura en línea² en contraste con el sistema de estrella, donde cada instrumento está individualmente conectado a la unidad central por sus propias líneas de control. Los instrumentos conectados formando estructura en línea están todos conectados a una línea común, el *bus*.

La transferencia de datos en el *bus* de IEC o GPIB es en serie para los bytes³ y en paralelo para los bits⁴.

Una ventaja de la interfase GPIB (generalmente llamada *bus GPIB*) es que no hay problemas con cables y la extensión del sistema es muy simple. Todos los instrumentos que cuentan con *bus GPIB* pueden utilizarse juntos en un mismo sistema sin importar en que forma fueron hechos, ni por quién. El usuario puede seleccionar el instrumento que considere más apropiado para cada propósito sin que sea necesario tener en cuenta las facilidades de interfase.

Otras características de la interfase GPIB son:

Pueden ser acoplados sin problemas instrumentos con diferente relación de transferencia de datos. Pueden transferirse datos directamente de un instrumento a otro sin hacer uso de la unidad de control. Es posible la transferencia asincrónica de datos hasta 1 Mbyte por segundo.

El *bus IEC 625* es conocido también como IEEE 488, HP-IB y GPIB.

B.2 Estructura de la interfase

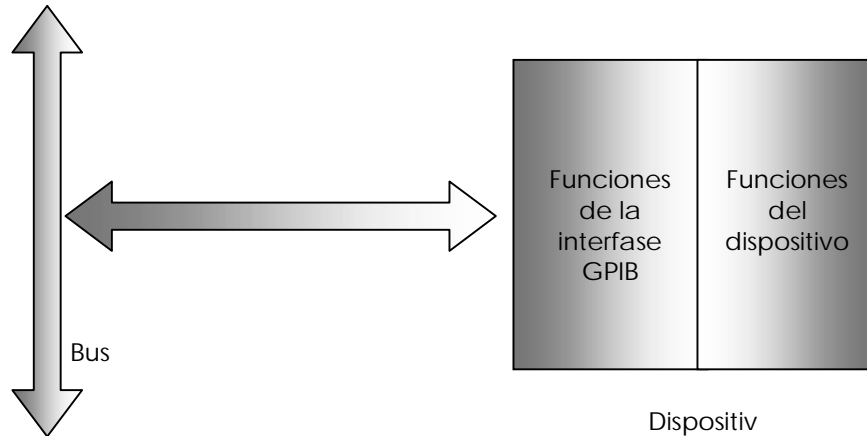
La conexión entre diferentes dispositivos es realizada por un cable (*bus*) que cuenta con 16 líneas paralelas para transferencia de datos, sincronización y mensajes de control. La interfase admite hasta 15 instrumentos conectados a la vez.

El largo del cable entre dos dispositivos no debe exceder los 2 metros para asegurar una buena transferencia de datos (alta velocidad). El largo total máximo del cable debe ser menor a 20 metros. Si es necesario superar dicha longitud, se deberá contar con dispositivos especiales de extensión.

² *Busline system*

³ Contracción del inglés de *Binary digit*: es un octeto de dígitos binarios

⁴ *Bit*: dígito binario, toma los valores 0 ó 1.



- Fig. 2 Disposición de funciones en un dispositivo conectado al Bus GPIB -

El cable cuenta con conectores denominados *piggyback*, cada cable tiene en sus extremos un terminal que cuenta con una unión macho y otra hembra, además de tornillos de fijación, de modo que es posible conectar varios cables apilados a un mismo instrumento y fijarlos.

B.2.1 Instrumentos compatibles con la interfase GPIB

Cada instrumento con interfase GPIB debe estar adaptado a los requerimientos normalizados de este sistema. Además de este sistema, cada instrumento tendrá sus propias funciones (funciones del dispositivo), las cuales serán desarrolladas independientemente de los requerimientos de la interfase de conexión.

La estructura de la interfase GPIB dependerá del instrumento en cuestión. Por ejemplo, la interfase de un voltímetro digital diferirá de la de una impresora.

El DVM (*Digital VoltMeter*) debe enviar los datos medidos en el *bus* y debe recibir las instrucciones de comandos del programa que controla el sistema. La impresora solo debe recibir datos e imprimirlos.

Teniendo en cuenta estos conceptos podríamos subdividir los instrumentos conectados a la interfase GPIB en las siguientes clases: *Listener* (el que escucha), *Talker* (el que habla), *Controller* (el que controla).

Listener: Un dispositivo *Listener* es aquel que puede recibir datos de otros instrumentos. Por ejemplo: impresoras, unidades de displays, fuentes de energía programables y fuentes de señal.

Talker: Un dispositivo *Talker* puede enviar datos a otros instrumentos. Por ejemplo: Voltímetros y contadores, lectores de cinta magnética, etc.

Controllers: Un dispositivo *Controller* puede controlar el tráfico de la información de las líneas comunes (*bus lines*). El controlador determina que dispositivo debe mandar señal y cuál debe recibir. Además, puede enviar comandos especiales y señales de control. Es el encargado de ejecutar los comandos que están definidos en el programa principal del sistema.

La comunicación entre instrumentos está organizada por reglas claramente definidas. Por ejemplo, solo un dispositivo a la vez puede “hablar” (transmitir datos al *bus*), mientras que varios pueden “escuchar” (recibir mensajes simultáneamente).

El flujo de información se transmite adaptando su velocidad a la del instrumento que más lento procesa los datos.

La estructura de la interfase de un dispositivo determina cuando un instrumento puede funcionar como *talker*, *listener* o *controller* del sistema.

Además de las funciones ya mencionadas existen otras funciones, las de *handshake* o coordinación que desempeñan un rol importante en la transferencia y control de datos.

Un *talker* debe contener una función que se llama *source handshake* además de la función *talk*, mientras el resto de los instrumentos GPIB compatibles debe tener una función *acceptor handshake*. A continuación se listan, a modo de ejemplo, las funciones de algunos instrumentos.

Instrumento:	Función:	Explicación:
Lectora de cinta	<i>Talker</i>	Puede hablar
Grabador de posición XY	<i>Listener</i>	Puede escuchar
Teclado	<i>Controller</i>	Puede controlar + hablar
Computadora	<i>Controller</i>	Puede controlar (sistema) + hablar + escuchar
Voltímetro digital	<i>Talker</i> <i>Listener</i>	Puede hablar Puede escuchar La función <i>talk</i> habilita al DVM a enviar datos medidos La función <i>listen</i> a recibir instrucciones del programa principal
Grabadora de cinta digital	<i>Talker+Listener</i> <i>Listener</i>	Puede hablar + escuchar Puede escuchar. La combinación de las funciones <i>talk</i> y <i>listen</i> es para transferencia de datos bidireccional. Una función <i>listen</i> adicional habilita a la grabadora a recibir instrucciones de programa principal.

- Tabla 1, Funciones de algunos instrumentos -

B.2.2 Funciones de interfase

- **System control function:** Habilita a un instrumento a enviar a la interfase los mensajes de *Clear (IFC)* y *Remote Enable (REN)*.
- **Control Function:** Habilita a un instrumento a enviar múltiples mensajes de interfase. Con esta función es posible responder a un requerimiento de servicio (*service request*) o ejecutar un *parallel pool*. Esta función además habilita al dispositivo en cuestión a recibir el control del *bus* y pasarlo, si fuera necesario, a otro dispositivo.
- **Talk Function:** Habilita a un instrumento a enviar datos a otro instrumento a través del *bus*. Con esta función también se envían *Status Bytes*.
- **Listen Function:** Habilita a un instrumento a recibir datos de otro instrumento a través del *bus*.
- **Source Handshake Function:** Se utiliza para sincronizar la transmisión de datos cuando la función *talk* envía datos (datos medidos) o la función *control* envía mensajes a la interfase.

- **Acceptor Handshake Function:** Se utiliza para recibir información, por ej.: datos del display o mensajes de la interfase provenientes del *bus* de datos.
- **Device Clear Function:** Puede ser utilizada para inicializar el dispositivo o instrumento por el controlador.
- **Device Trigger Function:** Puede ser utilizada para sincronizar la lectura de uno o más instrumentos habilitándolos a comenzar simultáneamente, con la ayuda de los comandos del *bus*.
- **Remote/Local Function:** Puede ser utilizada para elegir entre modo control remoto o local (esto es operación manual por panel de control) de uno o varios instrumentos.
- **Service Request Function:** Puede ser utilizada por un instrumento para informar al controlador del sistema que algo ha sucedido y requiere que el controlador del sistema tome determinada acción asincrónica con respecto a otras operaciones del *bus*.
- **Parallel Poll Function:** Puede ser utilizada por el instrumento para transmitir el estado de los datos por vía de las 8 líneas de transmisión, después que el controlador ha requerido información.

Un *bus* GPIB generalmente consiste de un sistema controlador junto con un número de *listeners* y *talkers* o combinaciones de estos. Como se había expresado anteriormente un *talker* puede estar activo por vez, mientras que varios *listeners* pueden estar activos juntos. En un sistema con varios controladores, al menos uno debe contener el *System Control Function*. Solo un controlador puede estar activo por vez.

Es posible pasar el control de un controlador a otro.

En sistemas que contengan al menos un controlador, los *talkers* y *listeners* sólo pueden estar activos después de que hayan sido direccionados por el controlador activo.

Es posible un sistema sin controlador, un sistema puede estar formado por un *talker* y uno o dos *listener*.

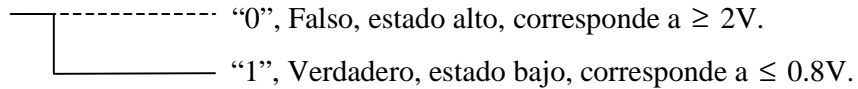
B.2.3 Líneas del Bus

Las 16 líneas del *bus* pueden ser subdivididas en tres grupos, cada uno con diferentes funciones:

- 8 líneas son combinadas en el data *bus*, DIO 1-8.

- 3 líneas son utilizadas para el control de transferencia de datos; DAV, NRFD, NDAC (*Handshake lines*).
- 5 líneas para mensajes de manejo general de la interfase; ATN, IFC, SRQ, REN, EOI. El significado de estas abreviaciones se verá más adelante.

Las señales en la interfase del *bus* son en lógica negativa con potencial positivo.



B.2.3.1 *Bus* de datos

Las 8 líneas de datos de entrada salida (DIO-8) son utilizadas para enviar bytes de datos, direcciones, instrucciones de programación, bytes de estado y comandos especiales del *bus*. Un byte de datos consiste en 8 bits de datos en paralelo. Como se había mencionado anteriormente la transferencia en el *bus* GPIB es en serie para los bytes y en paralelo para los bits, además de que sea asincrónica y bidireccional. Un mensaje completo puede consistir de uno o varios bytes en serie. La máxima velocidad de transporte es de 1 Mbyte.

El *bus* GPIB tiene dos formas de transferir datos: el *data mode* y el *command mode*.

La selección entre estos dos modos de transferencia se realiza mediante la línea de control ATN = *Attention*.

En *data mode* (ATN = 0), se envía por el *bus* GPIB datos o instrucciones de programa desde un dispositivo funcionando en modo *talk* a uno o más dispositivos direccionados con la función *listen*.

- Las instrucciones de programa dependen de cada dispositivo y son seleccionadas de acuerdo al equipo y funcionamiento del general del sistema. El código lo envía el dispositivo controlador actuando como *talker* o como un simple dispositivo seleccionado.

- Los bytes de datos son códigos de estado, datos medidos o mostrados. El contenido de cada byte está enviado a un dispositivo en especial. Se envían desde la dirección de un *talker* a uno o más *listeners*.

En el *command mode* (ATN = 1) el *bus* de datos es utilizado para direccionar un dispositivo como *talker* o *listener*, o bien para enviar comandos especiales para programar uno o más dispositivos. Todos estos mensajes son enviados por el dispositivo controlador.

Generalmente el código utilizado para la transferencia de datos y mensajes es el ASCII. Este código cubre todas los signos requeridos en la práctica.

B.2.3.2 Control de transferencia de datos

La transferencia de mensajes es organizada por el control de transferencia de datos (*data-byte transfer control*, o función *handshake*).

Cada byte enviado en las 8 líneas del *bus* de datos en *data mode* o en *command mode* es controlado con la activación de las 3 líneas de *handshake* DAV, NRFD y NDAC.

Líneas de *handshake* para transferencias de datos son:

- DAV: *Data valid*. Señal proveniente del dispositivo fuente, DAV = 1 indica que el mensaje en el *bus* de datos es correcto y está disponible para ser aceptado.

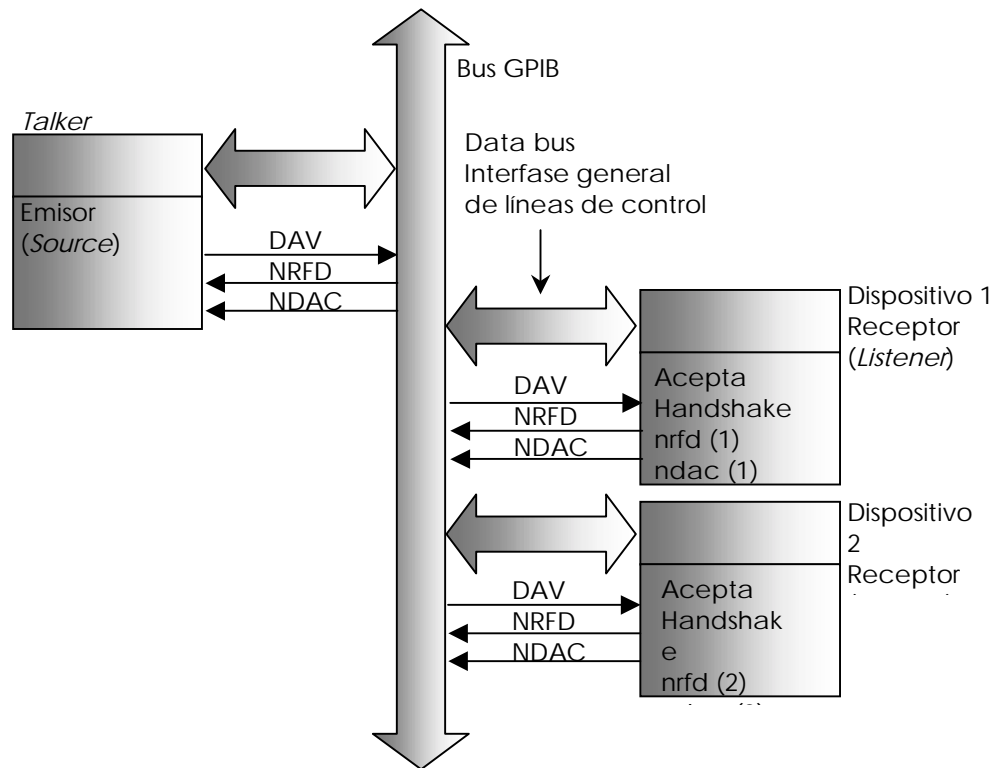
- NRFD: *Not Ready For Data*. Señal enviada por el dispositivo receptor NRFD = 0 indica que el instrumento está listo para recibir un mensaje.

- NDAC: *Not Data Accepted*. Señal enviada por el dispositivo receptor NDAC = 0 indica que todos los dispositivos que estaban en línea confirmaron haber recibido el dato.

NRFD y NDAC son líneas cableadas en configuración OR. El mensaje en el *bus* puede ser "0" (falso) si todos las salidas de los instrumentos dan "0". Esto es NRFD = 0 implica datos válidos listo para recibir datos y NDAC = 0 implica datos aceptados.

B.2.3.3 Procedimiento de *Handshake*

En el siguiente esquema da algunos detalles de las líneas de control involucradas en el *Handshake*. Supongamos que tenemos direccionados 3 dispositivos en un *bus* GPIB, uno como emisor de instrucciones de programa, datos o controles (*talker*) y dos como receptores (*listeners*). Si se envían datos de los emisores a los receptores (*data mode*) la comunicación se efectuará de la forma y sentido que se muestra en la figura.



- Fig.3 Control de transferencia de datos -

B.2.3.4 Control general de la interfase

Cada una de las 5 líneas del manejo general de la interfase tiene una función específica en la comunicación entre el controlador y los instrumentos. Algunas de estas funciones ya han sido mencionadas. En los siguientes párrafos se describen brevemente todas:

- ATN: *Attention*, pedido de atención

El controlador puede indicar por medio de la línea ATN cuando el *bus* GPIB está en data mode (ATN = “falso”) o en modo comando (ATN = “verdadero”).

En el modo datos (*data mode*), las líneas DIO son utilizadas para enviar datos de un emisor (*source*) a uno o más receptores (*acceptors*).

En el modo comando (*command mode*), se envían al *bus* de datos mensajes como direcciones y comandos especiales de control.

- IFC: *Interface Clear*, borrar la interfase

Solamente un dispositivo que contenga funciones de control de sistema puede enviar este mensaje.

Esta línea (IFC = “verdadero”) se utiliza para configurar a todos los instrumentos del *bus* a un estado predefinido.

Todos los emisores (*talkers*) y receptores (*listeners*) quedan en *stand by*. Si hay más de un controlador en el sistema, el control del *bus* GPIB será dado al dispositivo que envió el mensaje IFC. Se provee de esta manera un punto de partida uniforme para cualquier operación.

- REN: *Remote Enable*, habilitación de control remoto.

Esta línea puede ser activada (REN = “verdadero”) solo por un dispositivo que posea funciones de control. Habilita al instrumento a permutar el control entre local (del panel frontal) o remoto.

El instrumento en cuestión debe estar preparado y contar con esta posibilidad en su interfase GPIB. Para activar un instrumento dado en el modo control remoto el controlador activa la línea REN y direcciona el instrumento en cuestión. El instrumento permanecerá en control remoto tanto tiempo como REN esté activa a menos que se ejecute el comando *return to local*, volver a local.

- SRQ: *Service Request*, requerimiento de atención para entrar en servicio.

Un instrumento con esta línea activada en la interfase puede indicar que necesita la atención del controlador, SRQ = “verdadero”.

El controlador puede entonces interrumpir la secuencia de eventos que se esté ejecutando en ese momento y atender al dispositivo que envió el SRQ.

La línea SRQ se utiliza para *serial polling*.

- EOI: *End Or Identify*, final o identificación.

La línea EOI tiene dos funciones en combinación con ATN. En el modo datos (ATN = “falso”), un emisor puede utilizar EOI para indicar el final de una transferencia múltiple de datos (EOI = “verdadero”). En el modo comando (ATN = “verdadero”) el controlador utiliza EOI = “verdadero” para ejecutar en paralelo un conjunto de datos.

B.3 Control del tráfico de mensajes en la interfase GPIB

Todos los items de información transferidos en el *bus* se denominan mensajes remotos. Un mensaje remoto puede ser un mensaje de la interfase o de un dispositivo. Los mensajes a su vez se subdividen entre los de información multilíneas o unilínea.

Los mensajes multilíneas son de información codificada enviadas al *bus* con la ayuda de 3 líneas de handshake. Los mensajes unilínea son los que se transmiten vía una de las 5 líneas generales de control de la interfase. Por ejemplo IFC y SRQ. Los mensajes de interfase son generalmente enviados por el sistema de control para administrar el tráfico. Por ejemplo: comandos del *bus* (multilínea) y REN (unilínea). Datos medidos, estado de datos, instrucciones de programa son ejemplos de mensajes multilínea dependientes de los dispositivos.

El conjunto total de los mensajes de interfase utilizados para controlar el tráfico en el *bus* GPIB puede subdividirse en cuatro grupos:

- Instrumentos Seleccionables: Estos mensajes multilínea se utilizan para seleccionar los instrumentos conectados al *bus* como emisores o receptores.
- Comandos Universales: Incluyen mensajes multilínea y los tres mensajes unilínea, IFC, ATN y REN. Estos comandos se utilizan para que los instrumentos realicen determinadas tareas.
- Comandos Seleccionables: Estos comandos son como los universales multilínea, excepto que solo responden a ellos los instrumentos seleccionados.
- Comandos Secundarios: Estos mensajes multilínea son siempre utilizados en serie con la dirección del instrumento seleccionado.

B.3.1 Instrumentos seleccionables o direccionables

Si un sistema de medición automático trabaja apropiadamente, significa que debe estar previsto para indicar claramente y sin ambigüedades que dispositivo está actuando como fuente emisora de datos (*talker*) y cuál como receptor de datos (*listener*) en un determinado momento.

Para este propósito, cada dispositivo tiene una dirección asignada, la cuál no es más que un nombre codificado exclusivamente para ese dispositivo. Los instrumentos son direccionados por el controlador activo con la ayuda del código de direcciones que envía al DIO líneas en el modo comando.

Una dirección *talk* selecciona un instrumento como fuente de datos y deshabilita los *talkers* previos.

Una dirección *listen* selecciona a un instrumento como receptor de datos, pero no afecta a los otros, permanecen en el estado en que se encontraban previamente.

Una dirección consiste de 7 bits, esto es un byte. El bit más significativo (MSB) (b_8) del código de 8 bits (*byte*) en la línea de datos no tiene importancia para estas direcciones. Esto se indica con una “x”, el bit menos significativo (LSB) (b_1) es en la línea de datos DIO-1

En la Tabla 2 puede verse el grupo de caracteres **ASCII** para *talk* y *listen* y con que código binario se relacionan.

Direcciones que reciben datos (Listen)							Direcciones que envían datos (Talk)										
b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	ASCII	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	ASCII
x	0	1	0	0	0	0	0	SP	x	1	0	0	0	0	0	0	@
x	0	1	0	0	0	0	1		x	1	0	0	0	0	0	1	A
x	0	1	0	0	0	1	0	”	x	1	0	0	0	0	1	0	B
x	0	1	0	0	0	1	1	≠	x	1	0	0	0	0	1	1	C
x	0	1	0	0	1	0	0	S	x	1	0	0	0	1	0	0	D
x	0	1	0	0	1	0	1	%	x	1	0	0	0	1	0	1	E
x	0	1	0	0	1	1	0	&	x	1	0	0	0	1	1	0	F
x	0	1	0	0	1	1	1	,	x	1	0	0	0	1	1	1	G
x	0	1	0	1	0	0	0	(x	1	0	0	1	0	0	0	H
x	0	1	0	1	0	0	1)	x	1	0	0	1	0	0	1	I
x	0	1	0	1	0	1	0	*	x	1	0	0	1	0	1	0	J

x 0 1 0 1 0 1 1	+	x 1 0 0 1 0 1 1	K
x 0 1 0 1 1 0 0	,	x 1 0 0 1 1 0 0	L
x 0 1 0 1 1 0 1	-	x 1 0 0 1 1 0 1	M
x 0 1 0 1 1 1 0	.	x 1 0 0 1 1 1 0	N
x 0 1 0 1 1 1 1	/	x 1 0 0 1 1 1 1	O
x 0 1 1 0 0 0 0	0	x 1 0 1 0 0 0 0	P
x 0 1 1 0 0 0 1	1	x 1 0 1 0 0 0 1	Q
x 0 1 1 0 0 1 0	2	x 1 0 1 0 0 1 0	R
x 0 1 1 0 0 1 1	3	x 1 0 1 0 0 1 1	S
x 0 1 1 0 1 0 0	4	x 1 0 1 0 1 0 0	T
x 0 1 1 0 1 0 1	5	x 1 0 1 0 1 0 1	U
x 0 1 1 0 1 1 0	6	x 1 0 1 0 1 1 0	V
x 0 1 1 0 1 1 1	7	x 1 0 1 0 1 1 1	W
x 0 1 1 1 0 0 0	8	x 1 0 1 1 0 0 0	X
x 0 1 1 1 0 0 1	9	x 1 0 1 1 0 0 1	Y
x 0 1 1 1 0 1 0	:	x 1 0 1 1 0 1 0	Z
x 0 1 1 1 0 1 1	;	x 1 0 1 1 0 1 1	[
x 0 1 1 1 1 0 0	<	x 1 0 1 1 1 0 0	\
x 0 1 1 1 1 0 1	=	x 1 0 1 1 1 0 1]
x 0 1 1 1 1 1 0	>	x 1 0 1 1 1 1 0	^

- Tabla 2, los bits b_1 al b_5 corresponden a códigos binarios de caracteres ASCII -

En la Tabla 3 se destaca que los bits b_6 y b_7 determinan cuando la dirección en cuestión es una dirección que recibirá datos (*listen*) o que los enviará (*talk*).

b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1
x	1	0	•	•	•	•	•
x	0	1	•	•	•	•	•

- Tabla 3, la primera fila es una dirección talk y la segunda fila es listen -

La Tabla 4 muestra los 5 bits remanentes del código que determinan el dispositivo involucrado. La primera fila es la dirección “4” *listen*, la segunda fila es la dirección *talk* “E”.

x	0	1	1	0	1	0	0
x	1	0	0	0	1	0	1

- Tabla 4, primera fila listen "4", segunda talk "E" -

Las direcciones de cada dispositivo no están fijas de manera definitiva. Los instrumentos suelen venir de fábrica con una dirección predeterminada, pero esta puede alterarse por *software* utilizando funciones del panel de control o por *hardware* modificando la posición del *dip-switch*. Si un controlador quiere direccionar un determinado instrumento, coloca el código binario para dicho dispositivo en el *bus* de datos y a su vez hace un ATN = "true".

Todos los dispositivos en el *bus* GPIB comparan el código de dirección presentada con su propia dirección. Para esta función cada dispositivo debe tener una función de aceptación en el *Handshake* que esté activa en el durante el modo comando.

La dirección del dispositivo y la dirección en el *bus* son comparadas por la función decodificadora de mensajes. Solo el dispositivo que tiene igual dirección a la que hay en el *bus* asume la función de *talker* o *listener*.

Cuando se trabaja con varios instrumentos conectados a una interfase GPIB es básico asegurarse de que cada instrumento tenga una única dirección, a pesar de que dos instrumentos *listener* podrían tener la misma dirección si siempre fueran a recibir los mismos datos. Un instrumento que tenga las dos funciones: *talk* y *listen*, tendrá dos direcciones, una para cada función. En este caso, los 5 bits (b₅, b₄, b₃, b₂, b₁) suelen tener el mismo valor en las direcciones de *talk* y *listen*, y solo uno de ellos se utiliza para cambiar la dirección de una vez.

Por ejemplo:

Nanovoltímetro *Talk address* "Nanovolt" x10 10110

Listen address "6" x01 10110

Solo puede ser direccionado un dispositivo por vez como *talker*. En cambio, varios pueden ser direccionados como *listener* simultáneamente, ya que los *listener* pueden operar en paralelo.

Para evitar la posibilidad de que dos *talkers* estén en paralelo al mismo tiempo, cada vez que se direcciona un *talker*, todos los demás *talkers* son automáticamente configurados como *untalk*.

Talkers y *listeners* pueden ser deseleccionados o configurados como inactivos con el comando IFC de control de tráfico de la interfase. Este mensaje debe ser enviado por el dispositivo controlador del sistema.

Para deseleccionar a los dispositivos *listener* solamente, se puede utilizar el comando UNL (*Unlisten*). El código para este comando es puesto en el *bus* de datos por el controlador en el modo comando.

De un modo similar, se puede deseleccionar a un *talker* con el comando UNT (*Untalk*) o también, con menos elegancia se podría seleccionar una dirección inutilizada. De todos modos este comando UNT no se utiliza a menudo por lo mencionado recientemente: cada vez que un dispositivo entra en modo *talk* por haber sido direccionado, los demás *talkers* quedan automáticamente configurados como *untalk*.

B.3.2 Múltiple direcciones para *Talk* y *Listen*

Algunos instrumentos tienen más de una función *talk* y *listen*. Por ejemplo una grabación digital de cinta podría tener una función combinada de *talk* y *listen*, para transmitir y recibir datos y una función *listen* extra para recibir instrucciones del programa controlador.

Hay tres posibles soluciones para manejar este caso:

- 1) Utilizar una segunda dirección primaria.
- 2) Utilizar dos direcciones *listen* extendidas.
- 3) Utilizar secuencia de escape.

B.3.2.1 Segunda dirección primaria

Si se utiliza una segunda dirección primaria, el sistema con la grabadora podría quedar configurado de la siguiente manera: direcciones de *talk* y *listen* (por ejemplo: “R” y “2”) para la transmisión y recepción de datos y otra dirección *listen* (por ejemplo “3”) para recibir las instrucciones del programa. En este caso el controlador ve a esta grabadora como dos dispositivos independientes.

B.3.2.2 Direcciones extendidas

El *bus* GPIB prevé la posibilidad de trabajar con direcciones extendidas sumando un segundo carácter a la dirección principal.

Los bits sexto y el séptimo de este segundo carácter (b_6 y b_7) son iguales a “1”.

Ejemplo de dirección extendida: “Ad”

Donde “A” = x 1000001 (dirección primaria)

“d” = x 1100100 (dirección secundaria)

Las direcciones extendidas pueden ser utilizadas para seleccionar partes de un instrumento dado, o bien identificar un grupo de instrumentos idénticos.

Siguiendo con el ejemplo del grabador de cinta se podría configurar una parte del dispositivo como para grabar y reproducir mientras que otra sección podría programarse para recibir las instrucciones que comanden las funciones como la búsqueda, el avance rápido de cinta o el rebobinado.

Como ejemplo de la selección de un grupo de instrumentos idénticos podemos ver la ilustración de la Figura 4 donde los instrumentos en cuestión son voltímetros.

Cada voltímetro tiene su propia dirección *talk* extendida. Las otras funciones de la interfase GPIB son utilizadas por todos los voltímetros (pero no simultáneamente). Solamente puede ser direccionado un voltímetro por vez como *talker*. Similarmente, un grupo de instrumentos puede ser combinado con dirección extendida *listen*. Además es posible utilizar combinaciones de direcciones extendidas para con *talk* y *listen*. Ambas funciones pueden utilizarse como direcciones secundarias.

Este método aporta las siguientes ventajas:

- El grupo de instrumentos es visto por el controlador como un único dispositivo. Esto hace posible conectar una gran cantidad de instrumentos en el *bus*.
- Los requerimientos de *hardware* disminuyen, ya que una gran parte de la interfase GPIB es común a varios instrumentos de un grupo.

Sin embargo, como contraparte, hay un número de desventajas a considerar si se emplea esta configuración. La estructura del sistema se torna un poco más complicada.

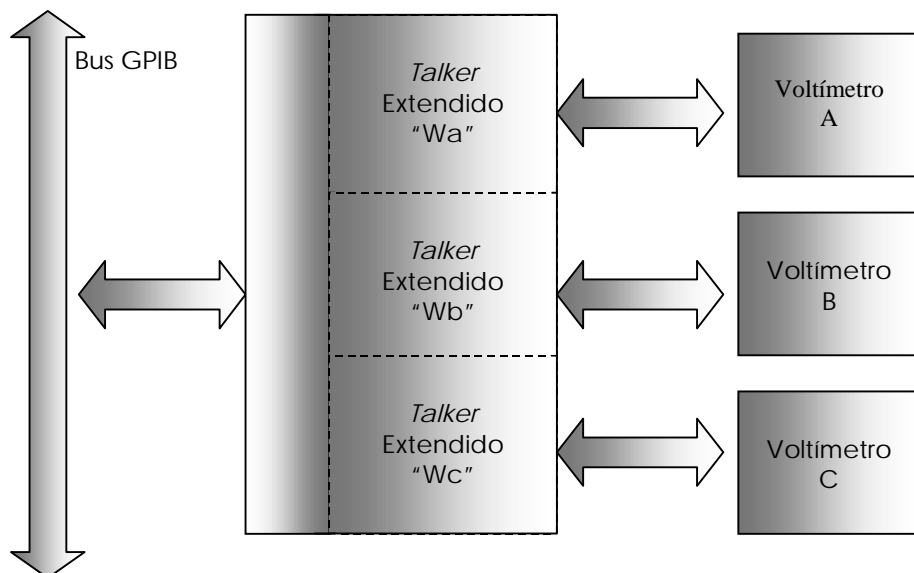
Se necesitan dos códigos distintos de dirección para distinguir y seleccionar las direcciones primarias y secundarias. Si el controlador utilizado en el sistema (esto es la tarjeta GPIB) no tiene posibilidad de generar los códigos para direcciones secundarias, entonces no pueden utilizarse direcciones extendidas. Sin embargo el problema podría resolverse de otra manera.

En el caso del grupo de instrumentos idénticos podría incorporarse un selector a la configuración del sistema. Este selector funcionaría indicando que instrumento del grupo es el elegido para comunicarse con el *bus*. El selector tiene sus propia dirección *listen*. De esta manera una interfase con su propia dirección *talk* y/o *listen* se incluye en el sistema para transferir los *bytes* de datos entre el instrumento seleccionado y el *bus*. La Figura 5 muestra como podría reconfigurarse la implementación de la Figura 4.

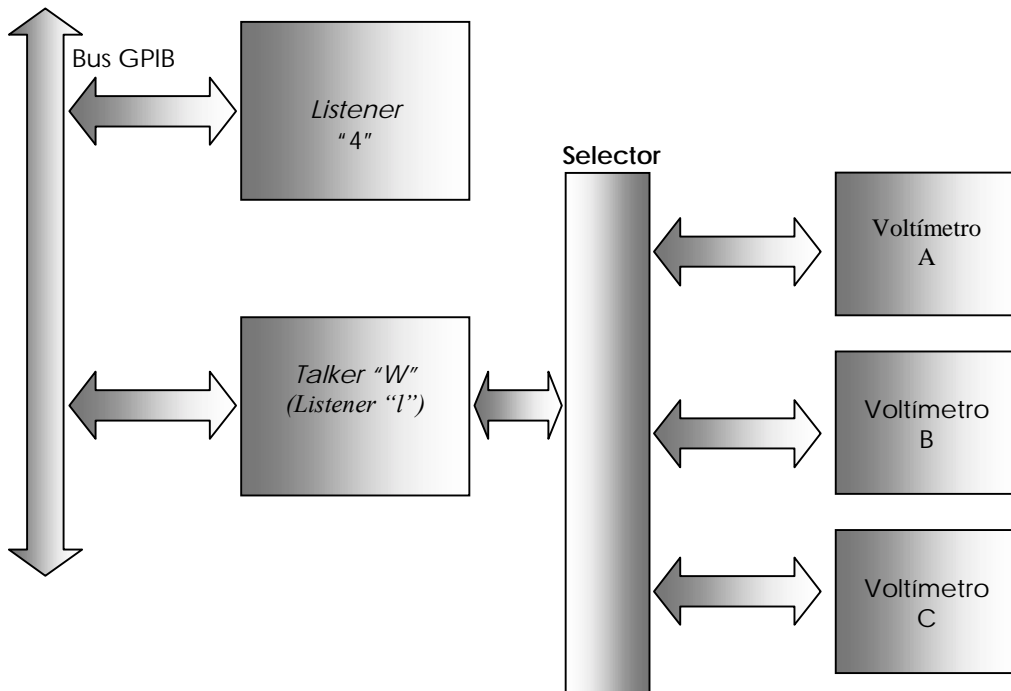
Ahora el controlador GPIB está “viendo” solo dos dispositivos conectados al *bus*.

Para transferir un *byte* de datos desde el voltímetro B a un receptor (*listener*) arbitrario en el *bus*, habría que completar la siguiente secuencia:

- Configurar el controlador de direcciones como *talker* y el selector como *listener* “4”.
- Transmitir un código al selector para seleccionar el dispositivo B
- Configurar el selector como *unlistener*.
- Direccionar el “Grupo de medida” como *talker* “W” y un receptor arbitrario como *listener*.
- De este modo ahora pueden transmitirse los *bytes* de datos del Voltímetro B a receptor en cuestión (*listener*).



- Fig. 4 Grupo de instrumentos idénticos compartiendo la interfase GPIB por direcciones extendidas -



- Fig.5 Configuración alternativa para un grupo de instrumentos idénticos utilizando solo dos interfaces GPIB, una para transferir datos y la otra para programar y controlar.-

B.3.2.3 Secuencia de escape

La tercera solución es utilizar la “secuencia de escape”. En este caso la interfase GPIB no necesita ser extendida de ninguna manera. Siguiendo con el ejemplo del grabador de digital de cinta todo lo que se necesita es una interfase combinada de *talk/listen*.

La distinción de mensajes entre los diferentes tipos de dispositivos ahora se realiza haciendo uso del carácter ASCII **ESC** (=Escape).

Los mensajes que reciben los dispositivos pueden ser instrucciones de programa para tomar una determinada configuración o bien pueden ser instrucciones para enviar datos. Un dispositivo determinado se puede configurar para recibir y enviar datos, si fuera necesario enviarle una instrucción de programa, se le envía al dispositivo un protocolo encabezado con el carácter ASCII **ESC**, seguido de un carácter 0 (ISO 3/0). Así se configura el dispositivo en “modo” programa quedando de esta manera listo para recibir una o más secuencias de caracteres (*string*) que no son otra cosa más que códigos de instrucciones de programa.

B.3.3 Comandos del *Bus*

El sistema del *bus* de la GPIB tiene comandos especiales para programar determinados dispositivos utilizando control activo. Estos códigos de instrucciones están definidos y tienen el mismo significado en todos los sistemas que posean *bus* GPIB.

Para “colocar” los comandos en el *bus*, el controlador envía un código determinado a las líneas DIO mientras están en el modo “comando”. Un dispositivo solo puede reaccionar a un comando dado si la decodificación del mensaje contiene las facilidades del código en cuestión. En la hoja de datos donde se brinda la especificación técnica de la tarjeta GPIB suele encontrarse un detalle de cual es la capacidad de decodificación de dicha tarjeta, allí puede verse que comandos puede decodificar.

Los comandos del *bus* pueden dividirse en tres grupos:

- Comandos no direccionados (*unaddress commands*)
- Comandos universales
- Comandos direccionados

B.3.3.1 Comandos no direccionados

Estos comandos pueden ser utilizados para enviar un *untalk* o *unlisten* a un dispositivo.

- UNL *Unlisten*: Todos los dispositivos que están activos en línea son reseteados y configurados para quedar en estado no-direccionado. Este comando puede utilizarse antes de reasignar nuevos dispositivos *listener*.
- UNT *Untalk*: El dispositivo que en este momento es *talker* debe dejar de serlo, de esta manera ningún *talker* queda en el *bus*. Un *talker* puede además ser configurado para transmitir el comando *talk* a otro dispositivo (no utilizado aún) en el *bus*.

B.3.3.2 Comandos universales

Este grupo de comandos afecta a todos los dispositivos, estén direccionados o no, siempre que tengan la posibilidad de decodificar los comandos en cuestión.

- DCL *Device Clear*: Todos los dispositivos que tengan la capacidad de leer este comando quedan en un estado de configuración inicial predeterminada.

La mayoría de los instrumentos cuentan con la capacidad de poder ser programados para que al encenderse posean una configuración preferencial determinada. Esa configuración podría verse alterada, en el transcurso de una medida; ya sea de manera remota (*bus* GPIB), o bien localmente desde el panel de control. El comando *Clear* coloca al instrumento en su configuración inicial predeterminada.

- LLO *Local Lockout*: Cuando este comando está en el *bus* junto con la activación de la línea REN, deshabilita el Remote/Local switch en el panel de control frontal del instrumento.
- SPE *Serial Poll Enable*: Este comando habilita al modo *serial poll* en el *bus* GPIB. De esta manera queda habilitado el *bus* para recibir datos y agruparlos por orden de llegada. Esta secuencia debería darse siempre que un dispositivo envíe una señal requiriendo ser atendido, esto es activando la línea SRQ.
- SPE *Serial Poll Disable*: Este comando deshabilita al modo *serial poll* en el *bus* GPIB, de esta manera puede ser utilizado para indicar el final de la secuencia *serial poll*.
- PPU *Parallel Poll Unconfigure*: Este comando configura a todos los dispositivos *bus* GPIB en el modo *parallel poll*. De este modo el controlador puede “ver” 8 dispositivos a la vez para determinar, por ejemplo, algunos estados como si se encuentra listo para enviar datos, si está ocupado en otra tarea, etc.

B.3.3.3 Comandos direccionados

Solo los dispositivos que han sido direccionados pueden responder a estos comandos. Algunos comandos direccionados son para dispositivos que han sido direccionados como *talkers* y otros como *listener*.

- SDC *Selective Device Clear*: Los dispositivos que han sido direccionados como *listener* pueden volver a su estado predeterminado con este comando.

- *GTL Go To Local*: Este comando deja configurado al instrumento para accionarlo con control local, esto es por el panel frontal de control.
- *GET Group Execute Trigger*: El controlador puede utilizar este comando para inicializar acciones simultaneas programadas previamente. Para que esto sea posible los instrumentos en cuestión deberían haber sido programados previamente y deberían encontrarse direccionados y en modo *listener*. O sea esperando el comando.
- *TCT Take Control*: Este comando se da cuando el controlador activo transfiere el control a otro dispositivo controlador. El nuevo controlador debería haber sido direccionado previamente como un *talker*.
- *PPC Parellel Poll Configure*: Este comando permite que varias líneas sean asignadas a dispositivos en el *bus* GPIB con el propósito de responder a un *parallel poll*. Debe estar acompañado por un *PPE (Parallel Poll Enable)*, comando secundario conteniendo la clave para la línea DIO asignada en el caso de un mensaje *PPR (Parallel Poll Response)*. El comando *PPC* será recibido solo por los dispositivos que hayan sido programados previamente como *listeners*.

No todos los instrumentos del *bus* GPIB necesitan aplicar este tipo de comandos. Por ejemplo, una impresora que trabaja en modo *listener* no tiene necesidad de decodificar comandos como *TCT*, *PPC* y *PPU*.

