



ESCUELA POLITÉCNICA DEL EJÉRCITO ESPE - LATACUNGA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA E INSTRUMENTACIÓN

**“Diseño e implementación de una red de
comunicación de instrumentos utilizando la interfase
GPIB”**

José G. Bucheli Andrade

Latacunga – Ecuador

2005

CERTIFICACIÓN

Certificamos que el presente documento fue elaborado por el señor:

José G. Bucheli Andrade

Bajo nuestra dirección, como un requisito para la obtención del título de Ingeniero en Electrónica Especialidad Instrumentación

Ing. Eddie Galarza Z.

DIRECTOR DEL PROYECTO

Ing. Marco Singaña

CODIRECTOR DEL PROYECTO

AGRADECIMIENTO

Mi profundo agradecimiento a mis padres, esposa e hijos, y a la Carrera de Electrónica de la Escuela Politécnica del Ejército sede Latacunga

Un agradecimiento especial al director del proyecto Ing. Eddie Galarza, al Ing. Marco Singaña codirector, por la valiosa colaboración en el desarrollo del presente proyecto.

José G. Bucheli Andrade

DEDICATORIA

El presente trabajo está dedicado a mis padres, a mi esposa y a mis hijos.

José G. Bucheli Andrade

ÍNDICE

CAPÍTULO I

INTRODUCCIÓN A LAS COMUNICACIONES INDUSTRIALES

1.1	Generalidades e historia	1
1.1.1	Estándares	3
1.1.2	Protocolos.	7
1.2	Principios de comunicación	8
1.2.1	Bits , Bytes y Caracteres.	8
1.2.2	Modos de comunicación	10
1.2.3	Sistemas asincrónicos	11
1.2.4	Comunicación sincrónica	12
1.2.5	Detección de errores	14
1.2.6	Técnicas de corrección de errores	16
1.3	Características de la transmisión	18
1.3.1	Codificación de datos	19
1.4	Redes de comunicación industrial	23
1.4.1	Las redes	23
1.4.2	Clasificación de las redes:	24
1.4.3	Topología de redes:	25
1.4.4	Protocolos de los buses y redes de campo industriales	28
1.4.5	Cables industriales	41
1.5	Estándares de la comunicación física	45
1.5.1	Normas y Asociaciones	45
1.5.2	Comunicación serie	46
1.5.3	Comunicación paralela	54

CAPÍTULO II

LA INTERFASE GPIB

2.1 Historia del bus GPIB	55
2.1.1 La Norma IEE488.1(GPIB) Nivel Físico: Estructura del bus GPIB	56
2.1.2 Tipos de mensajes que intercambian los equipos	60
2.1.3 Modos de operación de un equipo	60
2.1.4 Protocolos y Normas	63
2.2 La Norma IEEE 488.2 (GPIB)	65
2.3 Comunicación GPIB	67
2.3.1 Principios de comunicación	67
2.3.2 Protocolo básico	68
2.3.3 Protocolos de excepción:	69
2.3.4 Estatus del equipo por el protocolo IEEE 488.2.	70
2.4 Características Eléctricas de la Interfase GPIB	73
2.5 Reglas de Construcción	75
2.6 Otros estándares	76
2.6.1 SCPI (Standard Commands for Programmable Instruments)	76
2.6.2 HS488	76

CAPÍTULO III

HARDWARE Y SOFTWARE GPIB

3.1 Configuración del hardware	77
3.1.1 Tarjeta GPIB 82350 A	78
3.1.2 Interfaces soportadas por Agilent IO Libraries	81
3.1.3 Instalación de IO Libraries	82
3.1.4 Grupos de programas creados	83
3.1.5 Instalación de la Tarjeta 82350 A	84
3.1.6 Configuración de interfase	84

3.2 Comandos de programación GPIB	91
3.2.1 Sintaxis de las órdenes de medida	91
3.2.2 Envío de la información	92
3.2.3 Peticiones de servicio	93
3.2.4 Comandos del bus	96
3.2.5 Polling	98
3.2.6 Órdenes básicas del estándar IEE 488.2.	103
3.3 Lenguajes de Programación para GPIB	104
3.3.1 Software existente para GPIB	105
3.4 Otras Tecnologías para la conexión de instrumentos de medida	112
3.4.1 USB: Universal Serial Bus	112
3.4.2 Ethernet	113
3.4.3 Firewire (IEEE 1394)	113

CAPÍTULO IV

DISEÑO DE UNA RED GPIB

4.1 Osciloscopios Digitales	114
4.2 Comunicación con el osciloscopio AGILENT HP 54621 A	126
4.2.1 Inicialización del Osciloscopio	128
4.2.2 Estructura de los mensajes de orden	128
4.3 Instalación	129
4.4 Configuración	130
4.5 Programación	131
4.6 Pruebas de operación	143
4.7 Resultados Obtenidos	143

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones	145
5.2 Recomendaciones	147

Referencias Bibliográficas

ANEXOS

Anexo A Comandos del osciloscopio AGILENT 54621 A

Anexo B Glosario de términos

Manual del usuario

INTRODUCCIÓN

Para la instrumentación virtual o los procesos de laboratorio, la información puede venir dada no sólo por sensores, sino también por otros sistemas de medida (osciloscopios, multímetros, etc.) con capacidad de comunicación. Partiendo de la información recogida podemos cambiar las condiciones de la prueba, modificando parámetros de los aparatos (generadores de funciones, fuentes de alimentación, osciloscopios) .

El conocimiento, y la implementación de redes de comunicación desde una PC con múltiples dispositivos de medición a través estándares específicos de instrumentación como el IEEE-488 más conocido como GPIB son de vital importancia en el campo de la electrónica y la instrumentación y permitirán estar acorde con el avance tecnológico en los modernos procesos industriales

En un entorno como el descrito, la tendencia actual es que sea un software especializado quien se encargue del control de la red, coordinando el funcionamiento de los distintos instrumentos. Uno de los programas de software es VEE PRO de Agilent, el cual permite recoger, analizar datos, controlar y monitorear dentro de un entorno de programación gráfico en el que se ensamblan objetos para formar el programa de aplicación con el que interactuará el usuario y que se denomina instrumento virtual.

Es por esta razón que se ha desarrollado el diseño e implementación de la red de instrumentos en el laboratorio de electrónica para que docentes y alumnos puedan interactuar con los equipos de medida y que además pueda servir como una referencia práctica para algunas asignaturas de la Carrera.

Para cumplir con este objetivo se ha dividido el presente documento en 5 capítulos que se describen a continuación.

En el primer capítulo se realiza una breve análisis de los estándares y protocolos, se analiza los principios de la comunicación, los parámetros de la transmisión, se revisa los tipos de redes para la comunicación industrial y por último se analiza los estándares mas comunes para la comunicación serie y paralela

En el capítulo II se analiza historia de la interface GPIB, sus especificaciones y protocolos. Las características de la Norma IEEE 488.1 y 488.2, se describe las características eléctricas y las reglas que se deben cumplir para conectar varios instrumentos a la tarjeta.

En el capítulo III se describe la instalación de la librerías VISA, la instalación y configuración de la interface GPIB 82350 A. Se realiza un estudio de los comandos básicos y especiales del estándar del estándar 488.1 y 488.2. Se analiza el software existente que puede interactuar con la tarjeta GPIB sobre todo del lenguaje de programación gráfico HP VEE Hewlett Packard utilizado en este proyecto

En el capítulo IV se describe las características mas importante de operación del osciloscopio HP 54621 A de Agilent así como la estructura de los comandos para su control. Se describe la forma en que se conectó la red y la configuración de los diferentes dispositivos conectados a ella. Se muestra la programación realizada que permite interactuar con el usuario a través del lenguaje VEE. Por último se hace referencia a las pruebas realizadas y a los resultados obtenidos con respecto a la red de osciloscopios.

En el capítulo V se presentan las conclusiones obtenidas luego de todo el proceso de investigación, implementación y pruebas del sistema así también se presentan ciertas recomendaciones para un funcionamiento óptimo de la red. Se muestra al final un manual del usuario que permite operar la red y todo el proceso de comunicación en una forma adecuada

CAPÍTULO I

INTRODUCCIÓN A LAS COMUNICACIONES INDUSTRIALES

1.1 GENERALIDADES E HISTORIA

La comunicación de datos es la transferencia de información desde un punto a otro. Siendo un dato la información que es representada por una secuencia de ceros y unos, la misma clase de datos manejadas por computadora. Muchos sistemas de comunicación manejan datos análogos, ejemplos son los sistemas telefónicos, radio y televisión. La instrumentación moderna esta casi totalmente preocupada con la transferencia de datos digitales

Cualquier sistema de comunicación requiere un transmisor para enviar información, un receptor para aceptarlo y un enlace entre los dos. Los tipos de enlace pueden ser hechos con alambre de cobre, fibra óptica, radio y microonda.

Para distancias cortas algunos enlaces usan la conexión paralela, lo que significa que varios alambres son requeridos para llevar la señal, este tipo de conexión está limitada a equipos tal como las impresoras. Prácticamente toda comunicación moderna de datos utiliza un enlace serial, en el cual el dato es transmitido en secuencia sobre un solo circuito.

Debe haber un acuerdo mutuo como los datos son codificados, es decir, el receptor debe poder entender lo que el transmisor está enviando. La estructura mediante la cual los dispositivos se comunican es conocida como protocolo. En la última década se han establecido muchas normas y protocolos que permiten usar la tecnología de comunicaciones de datos más eficazmente en la industria.

Aunque inicialmente existían sistemas de comunicación de datos, su forma electrónica moderna empezó con la invención del telégrafo. Los primeros sistemas usaron varios alambres paralelos, pero pronto se hizo obvio que para largas distancias un método en serie sobre un solo par de alambres era el más barato.

El primer sistema del telégrafo práctico generalmente se atribuye a Samuel Morse. A cada fin de un enlace había un operador con una tecla enviando y resonando. Un mensaje era enviado como una serie codificada de puntos (pulsos cortos) y raya (pulsos largos). Esto se volvió conocido como el código de Morse y comprendía de aproximadamente 40 caracteres incluyendo el alfabeto completo, números y algunos signos de puntuación.

Inicialmente el trabajo era manual, posteriormente se crearon maquinas para leer códigos, luego para generar y posteriormente para imprimir.

Quizás la limitación más severa del Código de Morse es el uso de un número variable de elementos para representar los diferentes caracteres. Esto podía variar desde un solo punto o raya hasta seis puntos y/o rayas, que lo hizo inadecuado para un sistema automatizado.

Una alternativa del código fue inventada a fines de 1800 por el ingeniero telegráfico Maurice Emile Baudot. El código de Baudot fue el primer código binario de longitud uniforme. Cada carácter fue representado por un tamaño estándar de 5 bits. Se codificó $32 (2^5)$ caracteres que incluyeron todas las letras del alfabeto pero ningún número.

La Unión Internacional de Telecomunicaciones (ITU) después adoptó un código estándar de 64 caracteres utilizando los 32 iniciales de Baudot.

La transmisión por facsímil que usa tecnología computarizada, codificación y sistemas de comunicación más sofisticada, casi han remplazado las comunicaciones telegráficas. La constante evolución de la comunicación de datos ha conducido a la era moderna de sistemas de muy alta velocidad.

1.1.1 Estándares

Los protocolos son la estructura usada dentro de un sistema de comunicaciones para que por ejemplo, una computadora pueda comunicarse con una impresora.

Para desarrollar instrumentación más integrada y sistemas de control, se ha desarrollado estandarización de estos protocolos de comunicaciones por instituciones como ITU, EIA, IEEE, ISO, ANSI , etc.

Los Estándares permiten a los fabricantes desarrollar productos que pueden comunicarse con equipos ya en uso, que para el cliente simplifica la integración de productos de diferentes fuentes

- **El modelo OSI.**- Al principio de los años 80 las empresas proveedoras de Hardware comenzaron a desarrollar diferentes Arquitecturas de Comunicaciones con el fin de lograr Sistemas Distribuidos mediante la interconexión de sus propios equipos.

En 1984, la Organización Internacional de Estandarización (ISO) desarrolló un modelo llamado **OSI(Open Systems Interconnection)**, Interconexión de sistemas abiertos). El cual es usado para describir el uso de datos entre la conexión física de la red y la aplicación del usuario final. Este modelo es el mejor conocido y el más usado para describir los entornos de red.

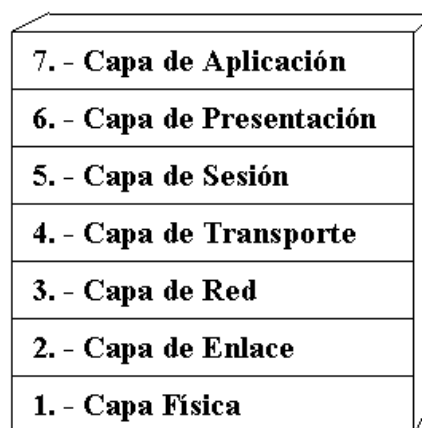


Figura 1.1. Estructura del modelo OSI

Como se muestra en la figura 1.1 , las capas OSI están numeradas de abajo hacia arriba. Las funciones más básicas, como el poner los bits de datos en el cable de la

red están en la parte de abajo, mientras las funciones que atienden los detalles de las aplicaciones del usuario están arriba.

Las capas del modelo OSI, tienen las siguientes funciones:

- **Capa Física:** Encargada de la Transmisión Física de los datos de la estación a la red. La norma RS-232C es una norma de capa física.
- **Capa de Enlace de Datos:** Regula el acceso de los datos al medio, especialmente si el medio es compartido por varias estaciones como en una LAN. La norma IEEE 802.3, más conocida como Ethernet y la IEEE 802.5 conocida como Token Ring, son normas de capas 2 y 1.
- **Capa de Red:** Es la responsable de que una comunicación atraviese distintos medios físicos o redes y llegue a destino. Incluye, entre otras cosas, las direcciones de red, el establecimiento y mantenimiento de las conexiones. Entre los protocolos de capa de red podemos citar el IP de las redes tipo Internet, el IPX de las redes Novell y la norma de redes públicas de datos X.25 del CCITT.
- **Capa de Transporte:** Puede suceder en una red, que dos paquetes de datos (el dato con la información de control proporcionada por cada capa) tomen rutas distintas y lleguen desordenados, o no lleguen. La misión de la capa de transporte es, justamente, ordenarlos, descartar o pedir retransmisión de los datos faltantes. Por otra parte, esta capa es la encargada de establecer la calidad de servicio de la conexión. El TCP es un ejemplo de Protocolo de Control de Transmisión, al igual que el UDP y SPX de las redes Novell.
- **Capa de Sesión:** Es la encargada de la coordinación de las conversaciones, entre uno o más procesos corriendo en las distintas estaciones. Los protocolos SMTP, FTP y Telnet son algunos de los protocolos de las capas 5 y 6.
- **Capa de Presentación:** Maneja la conversión, compresión y formato de los datos. La conversión de caracteres de código ASCII, utilizados en las Pcs, y el código EBCDIC utilizado en los mainframes IBM, se produce en esta capa.

- **Capa de Aplicación:** Provee una interfase utilizada por los procesos del usuario de la estación. Un caso conocido es el protocolo NFS (Network File System), de Sun Microsystems, que permite utilizar los discos de cualquier estación de trabajo por los usuarios de la red.

En una comunicación pueden estar presentes las siete capas, o un subconjunto de ellas, pero si existe una capa, obligatoriamente deben existir las precedentes.

En el modelo OSI el propósito de cada capa es proveer los servicios para la siguiente capa superior, resguardando la capa de los detalles de como los servicios son implementados realmente. Las capas son abstraídas de tal manera que cada capa cree que se está comunicando con la capa asociada en la otra computadora, cuando realmente cada capa se comunica sólo con las capas adyacentes de las misma computadora.

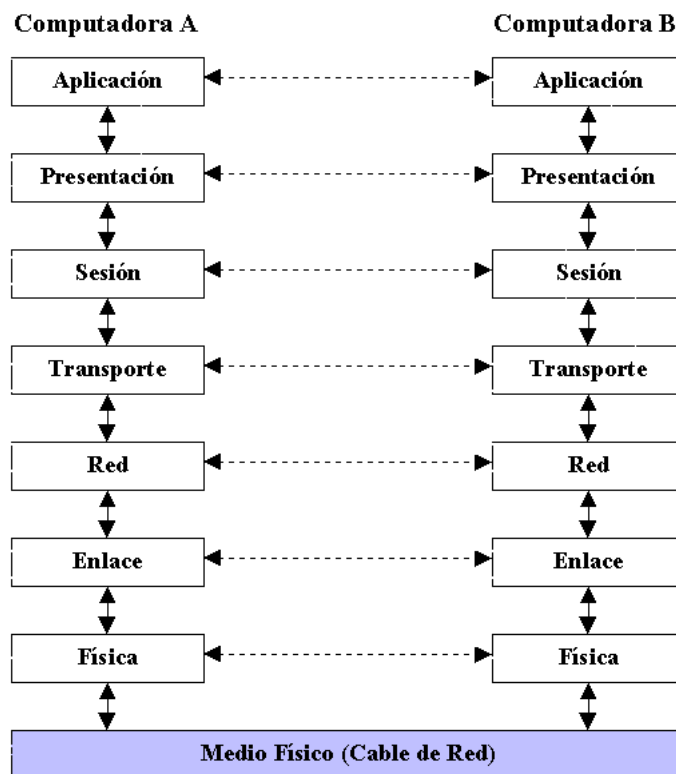


Figura 1.2 Representación del modelo OSI

Con esta última figura se puede apreciar que a excepción de la capa más baja del modelo OSI, ninguna capa puede pasar información directamente a su contraparte en la otra computadora. La información que envía una computadora debe de pasar

por todas las capas inferiores, La información entonces se mueve a través del cable de red hacia la computadora que recibe y hacia arriba a través de las capas de esta misma computadora hasta que llega al mismo nivel de la capa que envió la información. Por ejemplo, si la capa de red envía información desde la computadora A, esta información se mueve hacia abajo a través de las capas de Enlace y Física del lado que envía, el paquete adquiere información de la cabecera de cada capa, esto dice a la próxima capa que hacer con el paquete, pasa por el cable de red, y sube por las capas de Física y Enlace del lado de el receptor hasta llegar a la capa de red de la computadora B con cada parte de la información de cabecera que se va eliminando en el camino.

La interacción entre las diferentes capas adyacentes se llama interface. La interface define que servicios la capa inferior ofrece a su capa superior y como esos servicios son accedados. Además, cada capa en una computadora actúa como si estuviera comunicándose directamente con la misma capa de la otra computadora. La serie de reglas que se usan para la comunicación entre las capas se llama *protocolo*.

Generalmente se envían mensajes o datos en paquetes que simplemente son una sucesión de bytes. El protocolo define la longitud del paquete que es normalmente fijo.

Cada paquete requiere una dirección de la fuente y una dirección del destino para que el sistema sepa dónde enviarlo, y el receptor saber de donde vino. Las flechas entre capas en la figura 2, indican que cada capa lee como el paquete está viniendo o como esta yendo a la correspondiente capa de en frente. Esto es conocido como la comunicación par a par, aunque el paquete real es transportado vía el enlace físico.

El modelo OSI es útil ya que mantiene una estructura universal para todos los sistemas de comunicaciones. Sin embargo, no define el protocolo real a ser usado en cada capa. Algunos protocolos industriales que consideran el modelo OSI son el MAP, TOP, Fieldbus and HART.

En 1980 el IEEE propuso normas (Proyecto IEEE 802) unificando lo existente dividiendo la capa 2 en dos subniveles: LLC y MAC (control de enlace lógico y control de acceso al medio). Para MAC se adoptan 2 protocolos de acceso al medio

- CSMA/CD: Cada interfaz “escucha” al medio de transmisión y lo ocupan si esta libre. Si hay colisión esperan un cierto tiempo que se duplica si se colisiona con el paquete.
- TOKEN: se pasa una “fichas” de estación siguiendo un orden. Cada interfaz solo transmite cuando tiene la “ficha” la topología puede ser en anillo o en línea (BUS)

Ejemplos : 802.3 (CSMA/CD) / 802.4 (Token Bus) / 802.5 (Token Ring)

1.1.2 Protocolos.

Como previamente se mencionó, el modelo de OSI proporciona una estructura dentro del cual un protocolo específico puede ser definido. Un protocolo podría consistir en lo siguiente. El primer byte puede ser una serie de 1s y 0s para sincronizar el receptor o banderas para indicar el inicio de la estructura (para ser usado por el receptor). El segundo byte podría contener la dirección del destino que detalla donde el mensaje va. El tercer byte podría contener la dirección de la fuente que describe donde el mensaje se originó. Los bytes en el medio del mensaje podrían ser los datos reales que tienen que ser enviados del transmisor al receptor. El último byte es indicador del fin de la estructura que pueden ser códigos de detección de error y / o banderas indicadores de fin.

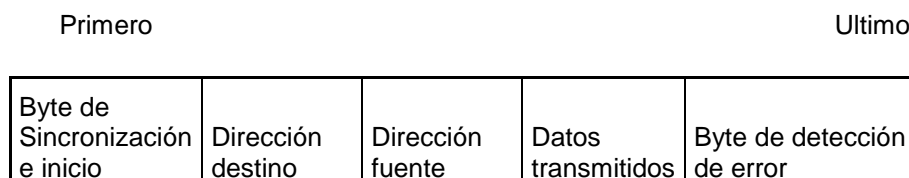


Figura 1.3 Estructura básica de la información de una estructura definida por un protocolo.

Los protocolos varían del muy simple (tal como ASCII basado en protocolos) al muy sofisticado (como el MAPA , Protocolo de Automatización industrial), que opera a velocidades altas transfiriendo megabits de datos por segundo. No hay ningún protocolo correcto o malo, la selección depende de la aplicación particular.

1.2 PRINCIPIOS DE COMUNICACIÓN

1.2.1 Bits , Bytes y Caracteres.

Una computadora usa el sistema de la enumeración binario que tiene sólo dos dígitos, 0 y 1. Cualquier número puede representarse por una serie de estos dígitos, conocido como bits (del digito binario). por ejemplo, el número 5 decimal es igual al número binario 101.

Tabla 1.1 Diferentes juegos de bits.

Bit	1 o 0
Dibit	Dos bits (10)
Nibble	Cuatro bits (1001 o un carácter hexadecimal)
Byte	Ocho bits o dos nibbles (11000001, C1 Hex)
Word	La anchura del bus de la computadora (16 bits)

Como un bit puede tener solo dos valores, puede ser representado por un voltaje en que cualquiera es on (1) u off (0). Esto es también conocido como lógica 1 y lógica 0. Valores típicos usados en una computadora son 0V para lógica 0 y 5V para lógica 1.

Una serie de ocho bits se llama un **byte** (o más normalmente un octeto), y puede tener valores que van de 0 (0000 0000) a 255 (1111 1111).

Las computadoras generalmente manipulan los datos en bytes o múltiplos de bytes.

Los programadores usan la notación hexadecimal porque es una manera más conveniente de definir y tratar con los bytes. En el sistema de numeración hexadecimal hay 16 dígitos (0 – 9 y A – F) cada uno de los cuales está representado por 4 bits. Un byte se representa por consiguiente por dos dígitos hexadecimales.

Un carácter es un símbolo que puede ser impreso. Las mayúsculas, minúsculas, números, signos de puntuación y símbolos especiales tales como ‘ * ’ y ‘ & ’ son todos caracteres.

Para una comunicación un sistema requiere:

- Una fuente de datos (un transmisor) que convierta la información en una forma adecuada para la transmisión sobre el enlace.
- Un receptor que acepte la señal y que la regrese a su forma original.
- Un medio de enlace por el cual se transporte la señal. Este puede ser alambre de cobre, fibra óptica, radio, satélite, etc.

Se debe establecer un acuerdo entre el transmisor y el receptor que defina:

- El tipo de señal utilizado
- Como se representa el ‘1’ y el ‘0’.
- Los códigos que representan los símbolos
- La forma de sincronización entre el emisor y el receptor.
- Como se controla el flujo de datos, de tal manera que no se sobrepongan los datos.
- La forma en que se detectan y corrigen los errores.

Los factores físicos corresponden al estándar de la interfase y los otros factores comprenden los protocolos. El método físico de transferir los datos por un enlace de comunicación varía según el medio usado

El valor binario 0 y 1, puede ser indicado por la presencia o ausencia de voltaje en un alambre de cobre, por un par de tonos de audios generado y decodificado por un módem , o por el uso de luz modulada en el caso de fibra óptica.

1.2.2 Modos de comunicación

En cualquier enlace de comunicación entre dos dispositivos, pueden enviarse los datos en uno de los tres modos de comunicaciones. Estos son: Simplex , Half duplex y Full duplex

Simplex.- Un sistema simplex es aquel que está diseñado para enviar mensajes en una sola dirección. Esto se ilustra en la figura 1.4.

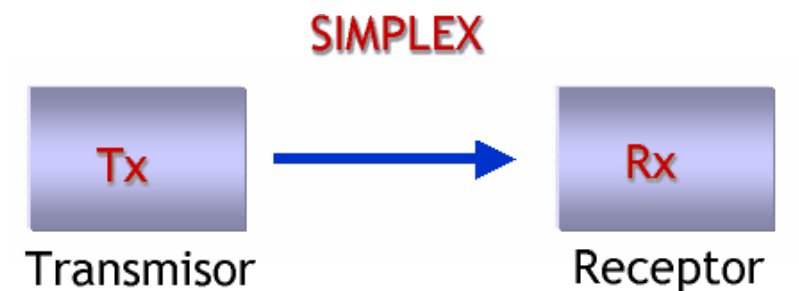


Figura 1.4 Comunicación Simplex

Un sistema duplex esta diseñado para enviar mensajes en ambas direcciones.

Half duplex.- Half duplex ocurre cuando los datos pueden fluir en ambas direcciones, pero en una sola dirección al mismo tiempo, como se ilustra en la figura 1.5.

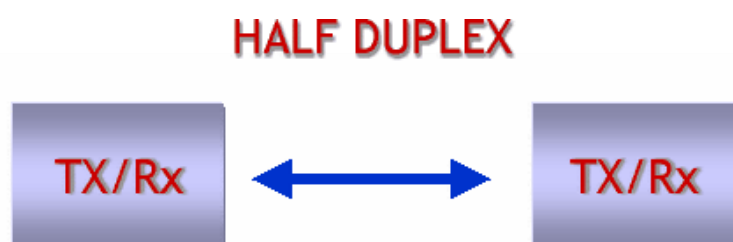


Figura 1.5. Comunicación Half - duplex

Full- duplex.- En un sistema full- duplex los datos pueden fluir simultáneamente en ambas direcciones como se ilustra en la figura 1.6.

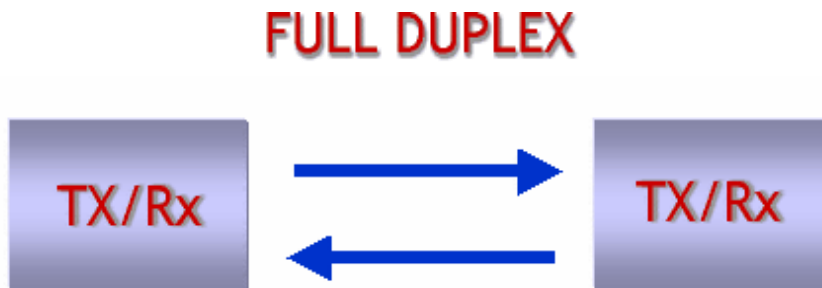


Figura 1.6 Comunicación Full duplex

1.2.3 Sistemas asincrónicos

Un sistema asincrónico es aquel en el cual cada carácter o byte se envía dentro de una estructura. El receptor no empieza la recepción hasta que reciba el primer bit, conocido como bit de arranque. El bit de arranque tiene un voltaje opuesto al estado del voltaje de parada y permite al receptor sincronizarse al transmisor para los siguientes datos en la estructura.

Un sistema asincrónico recibe cada carácter en intervalos de tiempos no específicos. Luego del bit de arranque, se determina un conjunto de bits, que son parte de la información.

El reloj del receptor se sincroniza con el del transmisor en una relación de 1 a 16, los datos se adquieren en la mitad de los mismos. La comunicación asincrónica puede tener problemas de velocidades altas. Las velocidades permitidas están en el orden de los 100kbps.

Un sistema asincrónico puede tener la siguiente estructura.

- Bit de arranque: señal de inicio de la estructura.
- Datos: Generalmente 7 u 8 bits. Pero también pueden ser 5 o 6 bits.

- Bit de parada: generalmente 1, 1.5 o 2 bit. Un valor de 1.5mseg significa que el nivel se mantiene para un tiempo de 1.5mseg.

El transmisor y el receptor deben fijarse exactamente la misma configuración, de tal manera que los datos puedan extraerse correctamente de la estructura. La transmisión de datos útiles es del 70%

Algunas de las características de la transmisión asíncrona son:

- Los equipos terminales que funcionan en modo asíncrono, se denominan también “terminales en modo carácter”.
- La transmisión asíncrona también se le denomina arrítmica o de “start-stop”.
- La transmisión asíncrona es usada en velocidades de modulación de hasta 1,200 baudios.
- El rendimiento de usar un bit de arranque y dos de parada, en una señal que use código de 7 bits más uno de paridad (8 bits sobre 11 transmitidos) es del 72 %.

Ventajas y desventajas del modo asíncrono:

- En caso de errores se pierde siempre una cantidad pequeña de caracteres, pues éstos se sincronizan y se transmiten de uno en uno.
- Bajo rendimiento de transmisión, dada la proporción de bits útiles y de bits de sincronismo, que hay que transmitir por cada carácter.
- Es un procedimiento que permite el uso de equipamiento más económico y de tecnología menos sofisticada.
- Se adecua más fácilmente en aplicaciones, donde el flujo transmitido es más irregular.
- Son especialmente aptos, cuando no se necesitan lograr altas velocidades.

1.2.4 Comunicación Sincrónica

La transmisión síncrona se hace con un ritmo que se genera centralizadamente en la red y es el mismo para el emisor como para el receptor. La información útil es transmitida entre dos grupos, denominados genéricamente delimitadores.

En los sistemas sincrónicos el receptor se sincroniza inicialmente a los pulsos de reloj del transmisor, con información que son incorporados en los datos transmitidos. Esto posibilita al receptor que mantenga su sincronización a lo largo de mensajes grandes de hasta 4500 bytes (36000 bits). Lo que permite por lo tanto transmitir estructuras largas eficientemente. El sistema sincrónico empaqueta muchos caracteres y los envía como un flujo continuo, llamado bloque. Un formato de la trama de un sistema sincrónico típico se muestra en la figura

Primero						último
Sincronización	SFD	Dirección destino	Dirección fuente	Longitud	Datos	FCS

Figura 1.7 Formato de una trama sincrónica

SFD: The Star of frame Delimiter: Señales al inicio de la trama

FCS The Frame Check Sequence: Para la detección del error.

Cada uno de los cuadros es llamado un campo.

Algunas de las características de la transmisión sincrónica son:

- Los bloques a ser transmitidos tienen un tamaño que oscila entre 128 y 1,024 bytes.
- La señal de sincronismo en el extremo fuente, puede ser generada por el equipo terminal de datos o por el módem.
- El rendimiento de la transmisión sincrónica, cuando se transmiten bloques de 1,024 bytes y se usan no más de 10 bytes de cabecera y terminación, supera el 99 por 100.

Ventajas y desventajas de la transmisión sincrónica:

- Posee un alto rendimiento en la transmisión.
- Los equipamientos necesarios son de tecnología más completa y de costos más altos.

- Son especialmente aptos para ser usados en transmisiones de altas velocidades (iguales o mayores a 1,200 baudios de velocidad de modulación).
- El flujo de datos es más regular.

1.2.5 Detección de errores

Todo canal de comunicación de datos práctico está sujeto a ruido, particularmente los alambres de cobre en ambientes industriales debido al alto ruido eléctrico. El ruido que se incluye en la transmisión puede generar errores.

No existe ningún sistema de comunicación de datos que pueda impedir que ocurran errores durante la transmisión, aunque la mayoría de estos pueden detectarse mediante diseños apropiados que permiten saber si la información recibida es la misma que se transmitió originalmente.

Entre las técnicas para la detección de errores podemos describir las siguientes:

- **Técnica del eco**

Es una forma simple de detección de errores usada en situaciones interactivas. Cuando una estación recibe una transmisión, la almacena y retransmite de nuevo a la estación emisora (eco), ésta compara el eco con el mensaje original y de esta forma se puede determinar si se presentó un error y corregirlo. Esta técnica tiene la desventaja de requerir al menos el doble de transmisiones, y además está la posibilidad de una "corrección" espontánea durante la retransmisión.

- **Técnicas de detección automática de errores**

Estas técnicas consisten en la adición al dato por enviar de un marco de verificación de secuencia o FCS (frame check sequence), el cual es obtenido a partir de los datos a transmitir por medio de un algoritmo. Una vez recibido el mensaje, la estación receptora aplica el mismo algoritmo a los datos recibidos y compara el FCS obtenido de esta forma con el que se adicionó a los datos

originales. Si son iguales se toma el mensaje, de lo contrario se supone un error.

Estas técnicas están basadas en dos métodos comunes:


- **Verificación de paridad en dos coordenadas**

Cuando se transmiten datos a un dispositivo que cuente con un buffer, es posible extender la verificación de paridad simple añadiendo un bloque de verificación de carácter (Block Check Character BCC) al final del bloque de datos, el cual realizará la segunda verificación de paridad a todo el bloque.

Tabla 1.2 Verificación de paridad en dos coordenadas

Carácter	Bits del carácter							Bit de paridad
1	1	0	0	1	1	0	0	1
2	0	0	1	1	1	0	1	0
3	0	1	1	0	0	0	0	0
4	1	1	0	1	0	1	1	1
5	1	0	1	0	1	0	1	0
6	0	0	1	1	0	0	0	0
7	1	1	0	0	0	0	1	1
BCC	0	1	0	0	1	1	0	1

Verificación de paridad en dos coordenadas.



- **Verificación por redundancia cíclica (CRC)**

Esta técnica es ampliamente usada debido a que es fácil de implementar en los circuitos integrados a muy gran escala (VLSI) que forman el hardware. Un mensaje puede verse como un simple número binario, el cual puede ser dividido por una cantidad que consideraremos constante, al efectuar la división (a módulo 2) se obtiene un cociente y un residuo, este último es transmitido después del mensaje y es comparado en la estación receptora con el residuo obtenido por la división de los datos recibidos y el mismo valor constante. Si son iguales los residuos se acepta el mensaje, de lo contrario se supone un error de transmisión. En el proceso de datos comercial es ampliamente usada la verificación por redundancia cíclica de 16 bits de longitud, aunque también es posible usar 32 bits lo cual puede ser más efectivo.

1.2.6 Técnicas de corrección de errores

- **Por operador humano**

Si los mensajes transmitidos son únicamente textos, puede resultar más económico y fácil que un operador humano reciba e interprete el mensaje y de ser necesario lo corrija usando su propio criterio. Algunos sistemas que aplican verificación por paridad cambian automáticamente los caracteres con error de paridad por el símbolo “?” para que el operador humano pueda identificarlos y corregirlos.

- **Código Hamming de corrección automática de errores**

Este sistema inventado por Richard W. Hamming (1950) asocia bits de paridad par con combinaciones únicas de bits de datos. Este método permite detectar y corregir con seguridad hasta un bit por cada bloque de información transmitida.

A cada n bits de datos se le añaden k bits de paridad de tal forma que el carácter transmitido tiene $n+k$ bits de longitud. Los bits se numeran de izquierda a derecha (el 1º bit es el más significativo). Todo bit cuyo número sea potencia de 2 es un bit de paridad, los restantes serán bits de datos. Los bits de dato se acomodan en sus posiciones y los bits de paridad se calculan de modo que tengan una paridad par sobre los bits cuyo número de bit formen, por ejemplo: El bit 1 (paridad) es determinado por los bits de datos: 3 ($1+2=3$), 5 ($1+4=5$), 7 ($1+2+4=7$), 9 ($1+8=9$), etc...

De esta forma cada bit está verificado por una combinación única de bits de paridad, de modo que analizando los errores de paridad se puede determinar que bit es el que ha invertido su estado. A continuación se dan algunos ejemplos que muestran cómo se pueden localizar los bits alterados:

Tabla 1.3 Código Hamming de corrección automática de errores

Paridad incorrecta en los bits	El error está en el bit número
4	4
1 y 4	5
1, 2 y 4	7
1 y 8	9

En el caso que exista más de un error en el bloque de información, se llegan a producir varias situaciones que pueden llevar a la "corrección" de un bit no alterado (Ej: si cambian los bits 1 y 2 llevan a la corrección del bit sano 3), entre muchas otras situaciones.

Una variante del código Hamming es adicionarle 1 bit de paridad global. De esta forma es posible tener la seguridad de detección de 2 errores, manteniendo la capacidad de corrección si se produce sólo 1 error.

La cantidad de bits de paridad empleados en la transmisión de la información le restan eficiencia al proceso. Se define la eficiencia de transmisión con la siguiente fórmula:

$$\text{Eficiencia} = \frac{\text{Información} / \text{Tiempo unitario}}{\text{Capacidad} / \text{Tiempo unitario}}$$

Suponiendo que se desea transmitir bloques de 8 bits de información, se necesitan 4 bits de paridad para ello, con lo que se tiene un total de 12 bits. La eficiencia sería:

$$\text{Eficiencia (8+4)} = \frac{8}{12} = 0.6666 \text{ (x 100)} = 66.66 \%$$

La eficiencia de este tipo de transmisión resulta de 66.66% debida solamente al plan de codificación. Además, dependiendo del método de transmisión puede decaer todavía más.

1.3 CARACTERÍSTICAS DE LA TRANSMISIÓN

A continuación se describe los parámetros más importantes tomados en cuenta en la transmisión de la información

- **Rapidez de la señal : Signalling Rate (or Baud Rate) .-** Es una medida de la variación de la señal por segundos. Se expresa en Baudios. Por ejemplo para 1000 baud rate, los pulsos se verían en múltiplos de 1 mseg. $1000 \text{ baudios} = 1 \text{ ms/bit}$
- **Rapidez de los datos (Data rate) .-** Este representa el número real de bits de los datos transferidos por segundo.

La velocidad de los datos o la velocidad de los bits es expresada en bits por segundo , o múltiplos tales como: kbps, Mbps and Gpbs (kilo, Mega and Gigabits por segundos). Por ejemplo un enlace EIA232, se transfiere una estructura de 10 bits, siendo 7 bits de datos mas bits de inicio, parada y paridad. Aquí la velocidad de la señal es 1000 Baud, pero la velocidad o rapidez de los datos es 700 bps.

Aunque hay una tendencia por el término Baud y bps a ser usados intercambiamente, ellos no son lo mismo. En una transmisión asincrónica data rate (bps) es siempre menor que Baud rate. Para sistemas sincrónicos con técnicas de modulación sofisticadas (usadas particularmente en módems) bps excede el Baud rate. El estándar ITU V.22bis Full duplex por ejemplo, define una técnica llamada QAM que efectivamente una transmisión de 600 Baudios corresponde a una información de 2400 bps. Independiente de los métodos usados el máximo data rate está siempre limitado por el ancho de banda de la conexión.

- **Ancho de Banda (Bandwidth)**

El factor más importante que limita las velocidades de las comunicaciones es el ancho de banda del enlace. El ancho de banda esta generalmente expresado en Hertz (HZ). Este representa la máxima frecuencia que puede transportar un medio, sin producirse ninguna degradación de la información.

Como una señal tiende a atenuarse con la distancia, enlaces de comunicaciones pueden requerir repetidores puestos a intervalos a lo largo del enlace, para aumentar el nivel de la señal.

- **Relación señal ruido (Signal to Noise Ratio).-**

La relación señal ruido (S/N) de un enlace de comunicaciones es otro importante factor limitador. Las fuentes de ruido pueden ser internas o externas.

S/N o SNR representa el nivel de la señal con respecto al ruido en un medio. La velocidad de transferencia práctica a través de un enlace está matemáticamente relacionada con el ancho de banda, con el S/N y con el número de niveles codificados en la señal. Si la relación S/N decrece, también decrecerá la velocidad de los bits.

- **Transferencia de datos (Data Throughput).** – Representa la velocidad real o efectiva de transferencia de datos.
- **Relación de error (Error Rate).-** Es la relación de los datos que llegan en forma errónea con respecto a los datos que llegan en forma correcta a un receptor. Error rate está relacionado con factores tales como: relación S/N, ruido e interferencia. Hay generalmente un compromiso entre velocidad de transmisión y el permisible relación de error, dependiendo del tipo de aplicación.. Con el incremento de la velocidad se incrementa los errores.

1.3.1 Codificación de datos

La información, para ser transmitida, necesita ser adaptada al medio de transmisión. Para ello, generalmente, será preciso codificarla de tal forma que pueda asegurarse una recepción adecuada y segura.. Podemos definir codificación como a la realización de dicha transformación, siendo el código la correspondencia existente entre cada símbolo del alfabeto fuente y cada conjunto de símbolos del alfabeto destino. A continuación se describe los códigos más comunes.

ASCII

American Standard Code for Information Interchange; Código Estándar para el Intercambio de Información. Creado aproximadamente en [1963](#) por el Comité Americano de Estándares (ASA) como una refundición o evolución de los conjuntos de códigos utilizados entonces en telegrafía.

Define 128 códigos posibles (7 bits de información por código), aunque utiliza menos de la mitad, para caracteres de control, alfabéticos (no incluye minúsculas), numéricos y signos de puntuación. Su principal ventaja, aparte de constituir un estándar, consiste en la ordenación alfabética de los códigos.

Más tarde, en [1967](#), se incluyen las minúsculas y se redefinen algunos códigos de control para formar el conocido US-ASCII. Actualmente llamamos ASCII a una extensión que utiliza 8 bits para proporcionar códigos adicionales (otros 128) usados en idiomas distintos al inglés (por ejemplo , 'ñ', 'é', 'ö')

Tabla 1.4 Código ASCII

ASCII-1967 (US-ASCII)																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	sp	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Leyenda			
Caracteres de Control	Caracteres Alfabéticos	Caracteres de Puntuación	Dígitos
Control de Flujo	Control de Formato	Separadores	

Por ejemplo para representar la palabra DATA en forma binaria usando el código ASCII de 7 bits cada letra es codificada como sigue:

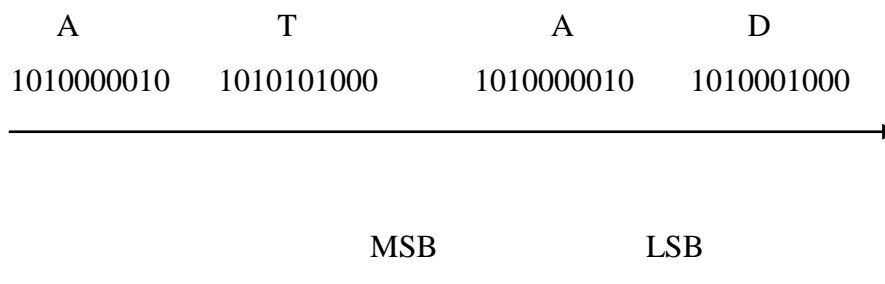
Tabla 1.5 Codificación de la palabra DATA

Palabra	Binary	Hex
D	100 0100	44
A	100 0001	41
T	101 0100	54
A	100 0001	41

El bit mas a la derecha es conocido como Least Significant Bit (LSB). El bit mas a la izquierda es conocido como Most Significant Bit (MSB).

En comunicaciones de dato por convención se transmite el LSB de cada carácter primero y por ultimo los bits MSB. Por ejemplo si la palabra D-A-T-A es transmitida, los caracteres son transmitidos en esa secuencia, pero en el código ASCII de 7 bits para cada carácter es inverso.

A continuación se realiza el ejemplo



El código ASCII de A es 1 0 0 0 0 0 1

Adicionando el bit de parada (1), un bit de paridad (1 o 0) y el bit de inicio (0) al carácter ASCII, el carácter ASCII de A enviado será:

STOP	P	MSB						LSB	START
1	0	1	0	0	0	0	0	0	1 0

EBCDIC

Extended Binary Coded Data Interchange Code (EBCDIC), originalmente desarrollado por IBM., usa 8 bits para representar cada carácter. EBCDIC es similar en concepto al código ASCII, pero los modelos de bit son diferentes y son incompatible con el código ASCII. EBCDIC no tiene mucha relevancia en las comunicaciones de datos del ambiente industrial

Código Baudot

El código Baudot fue inventado por Emile Baudot en 1874 y se utilizó en las primeras transmisiones telegráficas y radioeléctricas. Se trata de un código de 5 bits capaz de representar hasta 32 caracteres distintos, pero tiene además dos de ellos que permiten conmutar entre dos grupos denominados letras y figuras. El grupo de letras contiene el abecedario completo de mayúsculas de la A a la Z, mientras que el grupo de figuras contiene las cifras del 0 al 9, los signos de puntuación y caracteres especiales hasta un total de 26.

Código Gray.

Este código resulta interesante en aplicaciones industriales , ya que reduce las posibilidades de fallos por errores en el código. Se denomina como código progresivo, en los que cada combinación difiere de la anterior y siguiente en uno de sus dígitos. También conocido como códigos continuos, cuando en la primera y última combinación difieren en un solo bit y se les denomina cíclicos.

Código BCD natural.

BCD (Binary Coded Decimal): decimal codificado en binario. Entre estos códigos, el de más interés práctico, es el BCD natural, que se basa en representar

cada dígito decimal a su correspondiente binario natural. Cada dígito corresponde a un grupo de 4 bits. Es un código ponderado; a cada bit le corresponde un valor (peso) , igual que el binario natural., $8-4-2-1$..Para codificar un número decimal de N dígitos se requieren N grupos de 4 bits.

Por ejemplo el valor 15 se representa en BCD natural por **0001 0101**

Código EXCESO 3.

Es un código BCD no ponderado, cada combinación se obtiene sumando el valor 3 a cada combinación binaria BCD natural.

1.4 REDES DE COMUNICACIÓN INDUSTRIAL

1.4.1 Las redes

Las redes de comunicación de datos son utilizadas para que varias computadoras se comuniquen y puedan intercambiar datos y información. Así como compartir recursos de computo, almacenamiento, impresión.

En los primeros años de las redes, las grandes compañías incluyendo IBM, Honeywell y Digital Equipment Corporation, crearon su propio estándar de cómo las computadoras debían conectarse. Estos primeros estándares, sin embargo, no eran eternamente compatibles.

En años posteriores, organizaciones de estándares, incluyendo la **Organización Internacional de Estandarización(ISO)** y el instituto de **Ingenieros Eléctricos y Electrónicos (IEEE)**, desarrollaron modelos que llegaron a ser globalmente reconocidos y aceptados como estándares para el diseño de cualquier red de computadoras. Ambos modelos describen la red en términos de capas funcionales.

1.4.2 Clasificación de las redes:

- **LAN (Local Area Network):** Redes de Área Local

Es un sistema de comunicación entre computadoras que permite compartir información, con la característica de que la distancia entre las computadoras debe ser pequeña. Estas redes son usadas para la interconexión de computadores personales y estaciones de trabajo. Se caracterizan por: tamaño restringido, tecnología de transmisión (por lo general broadcast), alta velocidad y topología.

Son redes con velocidades entre 10 y 100 Mbps, tiene baja latencia y baja tasa de errores. Dentro de este tipo de red podemos nombrar a INTRANET, una red privada que utiliza herramientas tipo internet , pero disponible solamente dentro de la organización. *Ej.: IEEE 802.3 (Ethernet), IEEE 802.4 (Token Bus), IEEE 802.5 (Token Ring)*

- **MAN (Metropolitan Area Network):** Redes de Área Metropolitana

Es una versión de mayor tamaño de la red local. Puede ser pública o privada. Una MAN puede soportar tanto voz como datos. Una MAN tiene uno o dos cables y no tiene elementos de intercambio de paquetes o conmutadores, lo cual simplifica bastante el diseño. La razón principal para distinguirla de otro tipo de redes, es que para las MAN's se ha adoptado un estándar llamado DQDB (Distributed Queue Dual Bus) o IEEE 802.6. Utiliza medios de difusión al igual que las Redes de Área Local.

- **WAN (Wide Area Network):** Redes de Amplia Cobertura

Son redes que cubren una amplia región geográfica, a menudo un país o un continente. Este tipo de redes contiene máquinas que ejecutan programas de usuario llamadas hosts o sistemas finales (end system). Los sistemas finales están conectados a una subred de comunicaciones. La función de la subred es transportar los mensajes de un host a otro.

En la mayoría de las redes de amplia cobertura se pueden distinguir dos componentes: Las líneas de transmisión y los elementos de intercambio

(Conmutación). Las líneas de transmisión se conocen como circuitos, canales o troncales. Los elementos de intercambio son computadores especializados utilizados para conectar dos o más líneas de transmisión. *Ej. : X.25, RTC, ISDN, etc.*

1.4.3 Topología de redes:

La configuración de una red, recoge tres campos: físico, eléctrico y lógico. El nivel físico y eléctrico se entiende como la configuración del cableado entre máquinas o dispositivos de control o conmutación. El nivel lógico trata la información dentro de la red, como se dirige de un sitio a otro o como la recoge cada estación.

- **Topología en Estrella:** Todos los elementos de la red se encuentran conectados directamente mediante un enlace punto a punto al nodo central de la red, quien se encarga de gestionar las transmisiones de información por toda la estrella. La topología de Estrella es una buena elección siempre que se tenga varias unidades dependientes de un procesador, esta es la situación de una típica mainframe, donde el personal requiere estar accediendo frecuentemente esta computadora.

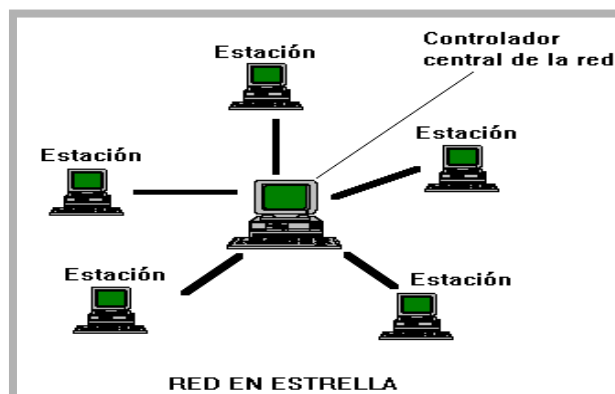


Figura 1.8 Topología en estrella

- **Topología en Bus:** En esta topología, los elementos que constituyen la red se disponen linealmente, es decir, en serie y conectados por medio de un cable; el bus. Las tramas de información emitidas por un nodo (terminal o

servidor) se propagan por todo el bus(en ambas direcciones), alcanzado a todos los demás nodos. Cada nodo de la red se debe encargar de reconocer la información que recorre el bus, para así determinar cual es la que le corresponde, la destinada a él.

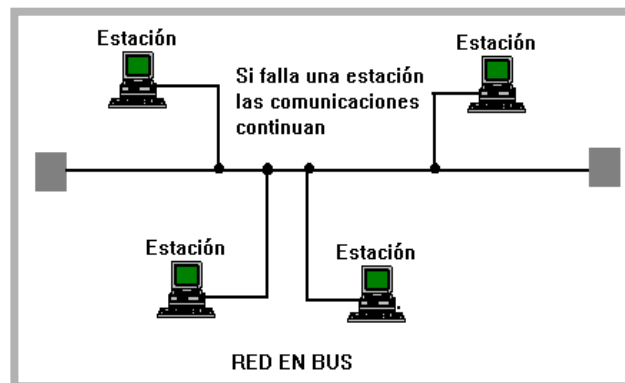


Figura 1.9 Topología en bus

Es el tipo de instalación más sencillo y un fallo en un nodo no provoca la caída del sistema de la red. Como ejemplo más conocido de esta topología: la red Ethernet de Xerox. El método de acceso utilizado es el CSMA/CD, método que gestiona el acceso al bus por parte de los terminales y que por medio de un algoritmo resuelve los conflictos causados en las colisiones de información.

- **Topología en Anillo:** Los nodos de la red se disponen en un anillo cerrado conectados a él mediante enlaces punto a punto. La información describe una trayectoria circular en una única dirección y el nodo principal es quien gestiona conflictos entre nodos al evitar la colisión de tramas de información. En este tipo de topología, un fallo en un nodo afecta a toda la red aunque actualmente hay tecnologías que permiten mediante unos conectores especiales, la desconexión del nodo averiado para que el sistema pueda seguir funcionando.

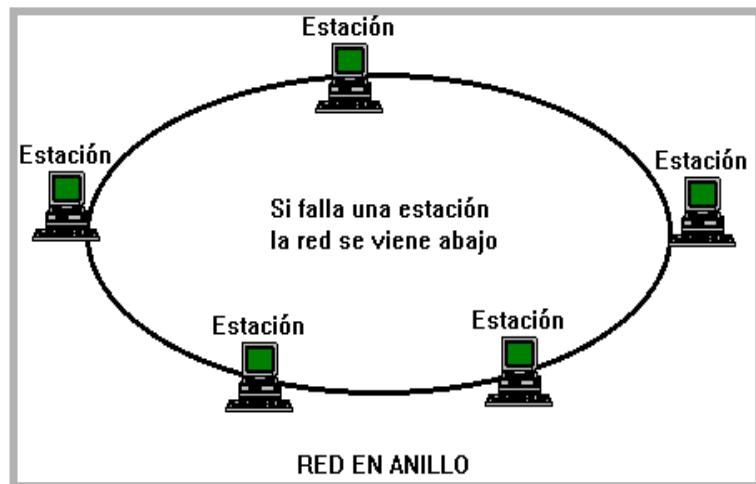


Figura 1.10 Red en anillo

- **La topología de árbol:** Esta topología es un ejemplo generalizado del esquema de bus. El árbol tiene su primer nodo en la raíz, y se expande para afuera utilizando ramas en donde se encuentran conectadas las demás terminales. Ésta topología permite que la red se expanda, y al mismo tiempo asegura que nada más existe una "ruta de datos" (data path) entre 2 terminales cualesquiera.

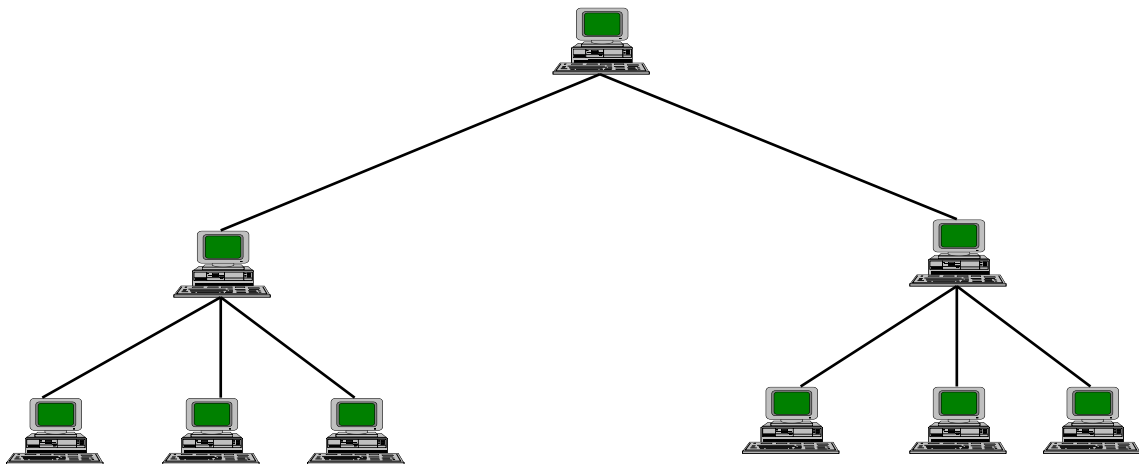


Figura 1.11 Topología en árbol

1.4.4 Protocolos de los buses y redes de campo industriales

En la búsqueda de la integración de las comunicaciones industriales, fueron desarrolladas las Redes de Comunicaciones Industriales (RCI). Las R.C.I, tienen su origen en los estudios efectuados por la fundación FieldBus, que buscaba la creación y desarrollo de esquemas de comunicaciones universales y de arquitectura abierta.

Las redes industriales, en la actualidad han tenido que desarrollarse para poder satisfacer las necesidades de información que ahora se tiene no solo a nivel de proceso sino también a nivel de gerencia. Esto implica que en una red industrial moderna deban coexistir equipos de todo tipo, por lo tanto es necesario agruparlos en una forma jerárquica, de tal forma que se optimice su uso, administración y mantenimiento. Un diagrama representativo de una planta organizada de esta forma se indica en la figura 1.12

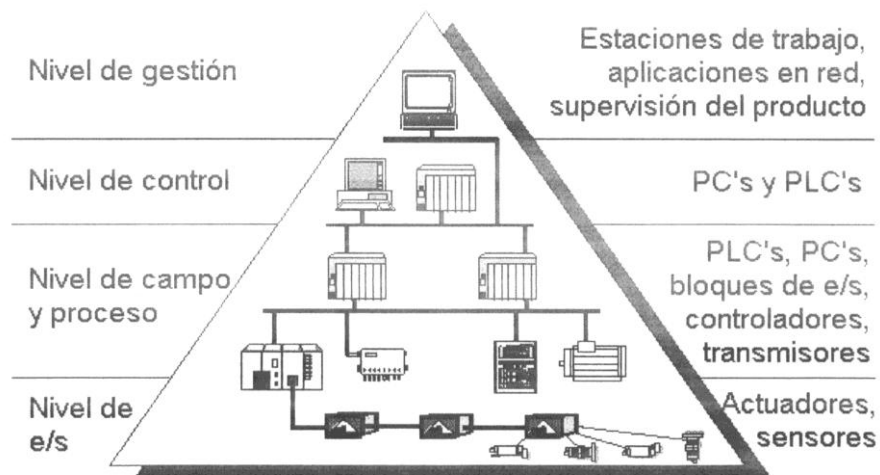


Figura 1.12. Niveles de una red industrial

En el área del bus de proceso dos organizaciones principales, la Fundación Fieldbus (que es el resultado de la unión de la Fundación ISP y FIP) y la Organización de Comercio Profibus (bus de proceso inteligente) están trabajando para producir estándares para protocolos y redes industriales.

Otras organizaciones como la ISA y el IEC están también participando en el desarrollo de estos estándares. Ésta es la razón por la que algunos fabricantes

especifican que sus productos analógicos son compatibles con Profibus, Fieldbus
La Figura 1.13 ilustra un diagrama de bloques de los protocolos disponibles.

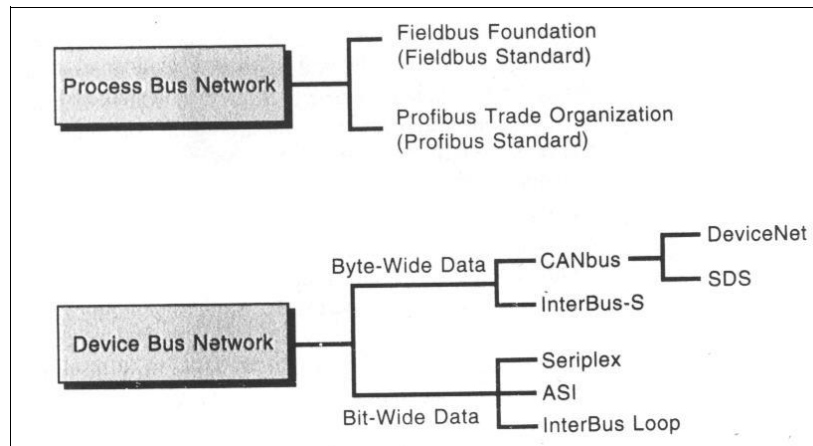


Figura 1.13 Estándares de protocolos

Tanto los buses de dispositivos como los de proceso transmiten su información de la misma manera; esto es, digitalmente. La comunicación digital es la que permite que más de un dispositivo pueda conectarse a un mismo medio y compartirlo debido a la capacidad de direccionamiento que ofrecen y a que los nodos involucrados sean capaces de reconocer los datos. Por otro lado, los datos digitales son menos propensos a sufrir de distorsión producto de la EMI o RFI. También, los PLCs pueden transmitir datos digitales sin tener que convertirlos a analógicos y viceversa. Se reduce el cableado al no tener que conectar los dispositivos de campo con cables de conexión individuales.

Una red con buses de proceso: es una red digital de comunicaciones, abierta y de alto nivel que se emplea para conectar dispositivos de campo analógicos a un sistema de control. Una red de proceso se emplea en aplicaciones donde los sensores o actuadores analógicos de entrada / salida responden más lentamente que aquellos en aplicaciones con dispositivos discretos (redes de buses de dispositivos). El tamaño de los paquetes de información de / desde estos dispositivos de campo analógicos es grande, debido a la naturaleza de la información que se recoge.

Los protocolos que se emplean en estas redes de proceso transmiten a velocidades de 1 a 2 Mbps, En la Figura 1.14 se muestra una red de bus de proceso típica.

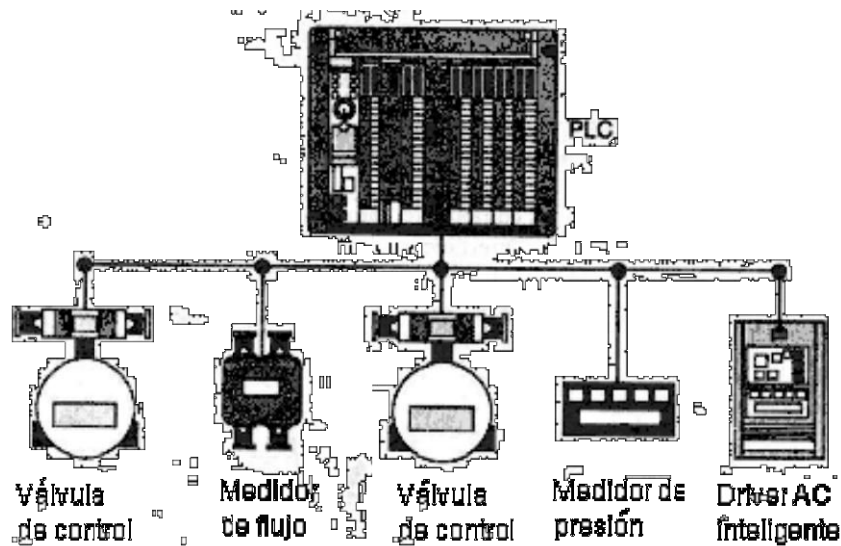


Figura 1.14. Configuración típica de una red de bus de proceso

Un PLC o computadora se comunica con una red de proceso por medio de módulo o tarjeta interfaz controladora que pueden emplear sea el formato del protocolo MODBUS, Fieldbus o Profibus.

- **MODBUS**

Los controladores programables de Modicon pueden comunicarse entre sí y con otros dispositivos sobre una variedad de redes, dentro de estas se incluyen las redes industriales Modicon Modbus y Modbus Plus, y las redes estándar MAP y Ethernet. Las redes son accedidas por medio de puertos incorporados en los controladores o por adaptadores de red, módulos opcionales, o gateways disponibles desde Modicon.

El idioma común empleado por todos los controladores de Modicon es el protocolo Modbus. Este protocolo define una estructura de mensaje que los controladores reconocen y usan, sin importar el tipo de redes sobre el que se

comunican. Establece un formato común para el diseño y contenidos de los campos de un mensaje.

Sobre otras redes, los mensajes del protocolo Modbus se integran en la trama o estructura del paquete utilizada sobre esa red. Por ejemplo, para los controladores de red Modicon para Modbus Plus o MAP. Con software de aplicación asociado – drivers y librerías - se proporciona la conversión entre el mensaje de protocolo Modbus y las tramas específicas de los protocolos que esas redes utilizan para comunicar entre sus dispositivos nodo.

Esta conversión también alcanza a la resolución de direcciones de nodos, caminos de enrutamiento y métodos de comprobación de error específicos para cada tipo de red.

Los pódicos Modbus estándar en los controladores Modicon usa una interfaz RS-232C, que define los pines del conector, cableado, los niveles de la señal, la velocidad de transmisión, y chequeo de paridad.

El usuario debe seleccionar:

- Medio de transmisión
- Rata de datos
- Número de bits de parada
- Bit de paridad (par, impar, ninguno)
- Modo de transmisión (ASCII o RTU)

Los controladores se comunican usando la técnica maestro – esclavo, en la cual sólo un dispositivo (el maestro) puede iniciar transacciones (llamadas ‘peticiones’). Los otros dispositivos (los esclavos) responden suministrando al maestro el dato solicitado, o realizando la acción solicitada en la petición.

La trama genérica de Modbus tiene el formato siguiente:

Tabla 1.6 Formato de la trama Modbus

Dirección del dispositivo	Código de función	Datos	Chequeo de error
---------------------------	-------------------	-------	------------------

Campo de dirección:

- Este campo posee dos caracteres ASCII u 8 bits RTU
- Contiene la dirección del dispositivo esclavo
- La dirección 0 se usa para transmisión

Campo de Código de Función:

- El campo de función utiliza dos caracteres ASCII u 8 bits RTU
- Contiene el código de la función a desempeñar por el esclavo
- El esclavo hará un eco de este campo. Si ocurre un error pondrá el MSB del código en 1 y enviará una respuesta de excepción
- Direcciones válidas de 1 a 255 (decimal). 73-119 son no permitidas

Algunas funciones de ilustración:

Tabla 1.7 Funciones Modbus

Código	Función	Parámetros
01	Leer estado de las salidas	Dir. de inicio, cantidad de salidas a ser leídas
02	Leer estado de las entradas	Dir. de inicio, cantidad de entradas a ser leídas
03	Forzar una salida	Dir. de la salida, dato a ser forzado (On :FF00, Off: 0000)

Los parámetros de las funciones Modbus, van incluidos en el campo de datos

Campo de Datos:

- Contiene la información adicional que el dispositivo esclavo necesita para desempeñar la función que le fue asignada.
- Aquí se envían las respuestas de excepción cuando ocurre un error.
- Este campo utiliza n bytes dependiendo del tipo de función. Aunque también puede ser vacío

Campo de Chequeo de Error:

Puede ser LRC (Longitudinal Redundancy Check) o CRC (Cyclical Redundancy Check) dependiendo del modo de transmisión que se esté usando.

ASCII - LCR

RTU - CRC

Los Controladores pueden configurarse para comunicarse en redes Modbus estándar empleando uno de dos modos de transmisión: ASCII o RTU. Los usuarios deben seleccionar el modo deseado, junto con los parámetros de comunicación serial (velocidad de transmisión, modo de paridad, etc), durante la configuración de cada controlador.

- **MODBUS PLUS**

Modbus Plus es un sistema de red de área local diseñado para aplicaciones de mando industriales. Cada red soporta 64 dispositivos de nodo direccionables, a una tasa de transferencia de datos de 1,000,000 bps. Las aplicaciones incluyen supervisión de mando de un proceso y los mensajes de supervisión.

Modbus Plus mantiene el protocolo de comunicación par-a-par en los diferentes niveles de la red. Modicon proporciona una variedad de controladores programables y adaptadores de red. La red también permite módulos de comunicación I/O distribuidos (DIO) en los que los controladores Modicon se comunican directamente con los subsistemas I/O.

Modbus es un estándar que ha estado mucho tiempo en el mercado y su difusión ha sido muy extendida.

- **FIELD BUS**

Al igual que Modbus y Profibus, Fieldbus busca conectar las redes de campo y las administrativo-financieras de una forma jerárquica, tal como se ilustra en la Figura 1.15.

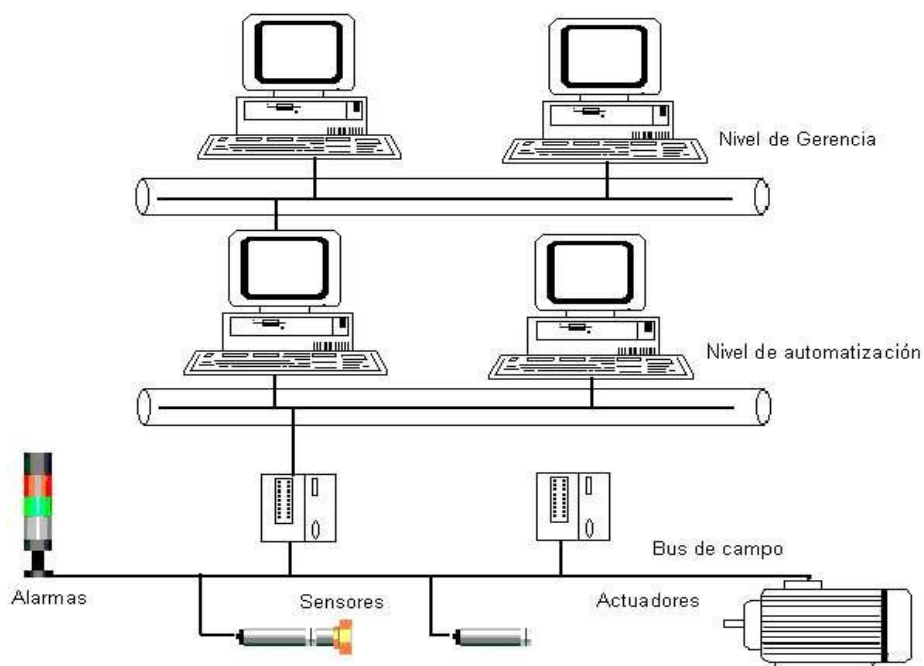


Figura 1.15. Red administrativa y redes de campo.

El sistema FIELD BUS propone sustituir los enlaces punto a punto, de tipo analógicos por la digitalización en el mismo sensor (medición digital distribuida), luego multiplexando, estableciendo una conexión mediante un método físico común (RS 485 de 2 hilos).

El FIELD BUS es una red local de medidores, adquisidores, controladores y PC.

- El acceso al medio de transmisión se realiza como una forma de interrogación distribuída, en las que todas las estaciones intervienen en la circulación de un paquete especial de información: Testigo (Token).
- Con la red con tipología en bus (Token – Bus) las estaciones tienen asignada una posición lógica en una secuencia ordenada y circular. Cada estación conoce la identidad de su antecesora y de su sucesor dentro de un anillo lógico, la ordenación física es independiente de la ordenación lógica.
- Las estaciones que solo reciben pueden excluirse del anillo lógico.
- También puede incluirse un esquema simple de Maestro-esclavo, entre la estación maestra y una o varias estaciones esclavas.

En este esquema el maestro tiene el control del acceso al medio, existiendo dos posibilidades: o es el maestro el que ocupa el medio físico, o el esclavo que fue consultado por el maestro, que está respondiendo.

Los esclavos solo envían mensajes cuando el maestro así se lo solicita. Hay dos tipos de transacciones

- Consulta –respuesta (Query-response): Cada esclavo tiene una dirección (address) única.
- Difusión sin respuesta (broadcast/no response) Todas reciben y ninguna contesta.

Se ha dicho que cada protocolo tiene sus características de funcionamiento propias. Al unir redes con diferentes protocolos, como se muestra en la Figura 1.15, surgen incompatibilidades. En muchos casos no se trata de protocolos que compitan entre sí, sino que funcionan bajo reglas diferentes y hay que procurar que se complementen, particularmente en una arquitectura de comunicación de varios niveles tal como se muestra en la figura anterior.

FOUNDATION FIELDBUS Este es el nombre que se da a la organización que da soporte a este protocolo para lograr su difusión a nivel mundial. La red de bus de proceso Fieldbus es un sistema de comunicación de dos vías, serial, digital y

multipuerto que conecta equipo de campo, como sensores y actuadores inteligentes, con controladores, como PLCs.

El protocolo está desarrollado a partir del modelo de comunicaciones OSI. Es un protocolo para redes industriales pensado específicamente para aplicaciones de control distribuido. La longitud máxima por mensaje es de 256 bytes, lo que permite transferir funciones de control con el concepto de objetos. La figura 1.16 muestra la comparación del protocolo Fieldbus con el modelo OSI

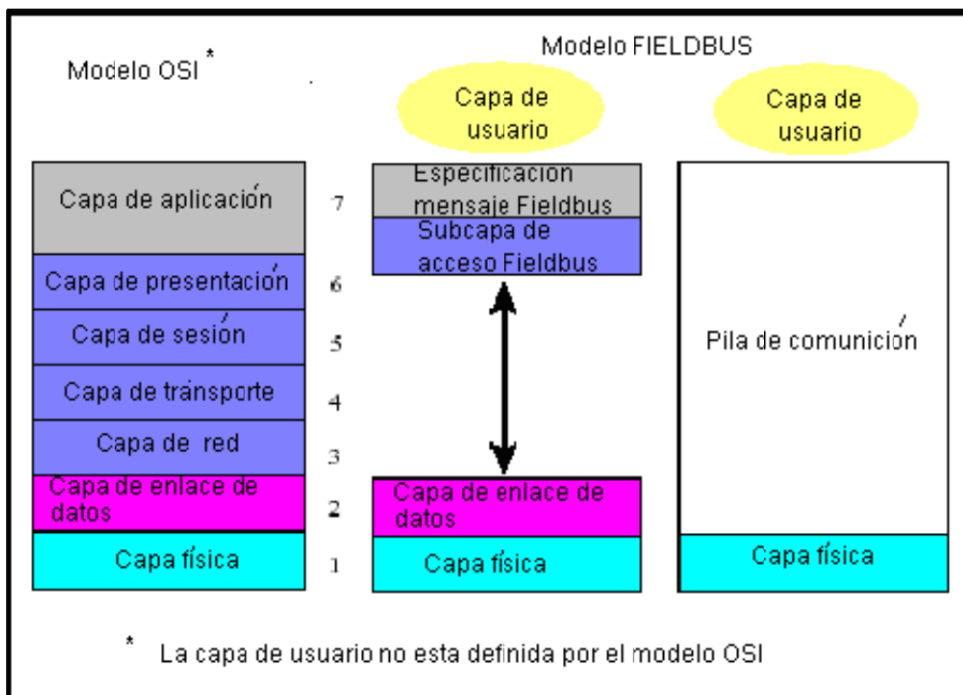


Figura 1.16 Protocolo Fieldbus y su comparación con el modelo OSI

Fundamentalmente consta de tres niveles: Capa física, Capa o pila (Stack) de comunicaciones, y Capa de usuario. En este protocolo no se implementan los niveles 3, 4, 5 y 6 del modelo OSI, pues estos no se requieren en aplicaciones de control de procesos, pero si se tiene en cuenta dentro del stack de comunicaciones a la capa de aplicación que tiene que comunicarse con el Nivel de Usuario.

Poseen velocidades de 31,25 Kbps (H1, baja), 1 Mbps (H2, media) y 2,5 Mbps (H2, alta) que depende del modo que se haya configurado: modo de corriente AC

o modo de voltaje DC. Se emplea codificación Manchester Bifásica – L, llamada serial sincrónica pues la señal del reloj se “inserta” en los datos

Por otro lado, el receptor emplea el limitador de inicio para encontrar el comienzo de una mensaje (trama) Fieldbus. Después de que lo encuentra, el receptor acepta los datos hasta que recibe el limitador final.

RESUMEN CARACTERISTICAS FIELDBUS

Proveedor	Fundación Fieldbus
Bus Abierto	Sí
<i>Grupo de Usuarios</i>	Foundacion Fieldbus, TEXAS
Estándares	IEC 1158-2
Topología	
<i>Tasa de transmisión</i>	2.5 Mbps 500m
	1Mbps 750m
	31.75 Kbps 1900m
Requerimientos del cableado	Par trenzado con blindaje
Estructura del Paquete de datos	Synchronous Manchester Coding
Interfaz eléctrica	Energía y señales sobre la misma línea

- **PROFIBUS**

La base de la especificación del estándar PROFIBUS (PROcess Field BUS) fue un proyecto de investigación (1987-1990) llevado a cabo por los siguientes participantes: ABB, AEG, Bosch, Honeywell, Moeller, Landis & Gyr, Phoenix Contact, Rheinmetall, RMP, Sauter-cumulus, Schleicher, Siemens y cinco institutos alemanes de investigación. Hubo además un pequeño apoyo por parte del gobierno alemán. El resultado de este proyecto fue el primer borrador de la norma DIN 19245, el estándar Profibus, partes 1 y 2. La parte 3, Profibus-DP, se definió en 1993.

Existen tres variantes principales de PROFIBUS de acuerdo a las características de la aplicación:

TABLA 1.8 La familia PROFIBUS

<p>Automatización de fábricas</p> <p>PROFIBUS DP</p> <p>(DIN 19245 T1 + T3)</p> <p>pr EN 50170</p> <p>Alta velocidad de transferencia de data para periféricos descentralizados</p>	<p>Automatización para propósitos generales</p> <p>PROFIBUS FMS</p> <p>(DIN 19245 T1 + T2)</p> <p>pr EN 50170</p> <p>Perfiles de aplicación específicos:</p> <p>Máquinas textiles</p> <p>Automatización de edificios</p> <p>Drivers, sensores y actuadores, PLCs, Switch gear de bajo voltaje</p>	<p>Automatización de procesos</p> <p>PROFIBUS PA</p> <p>(DIN 19245 T4)</p> <p>en preparación</p> <p>Técnicas de transmisión con seguridad intrínseca de acuerdo a IEC 1158-2</p>
--	---	---

En la Figura 1.17 se muestra como se conectan y relacionan estas redes de la familia Profibus, mientras que en la Figura 1.18 se muestra la relación que existe entre los protocolos.

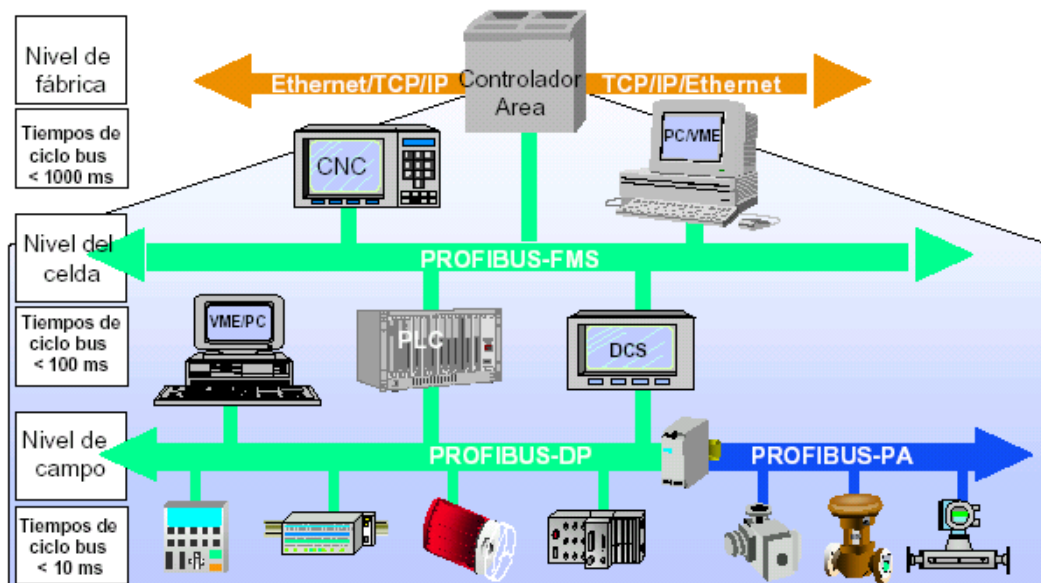


Figura 1.17 Jerarquía de la Red Profibus

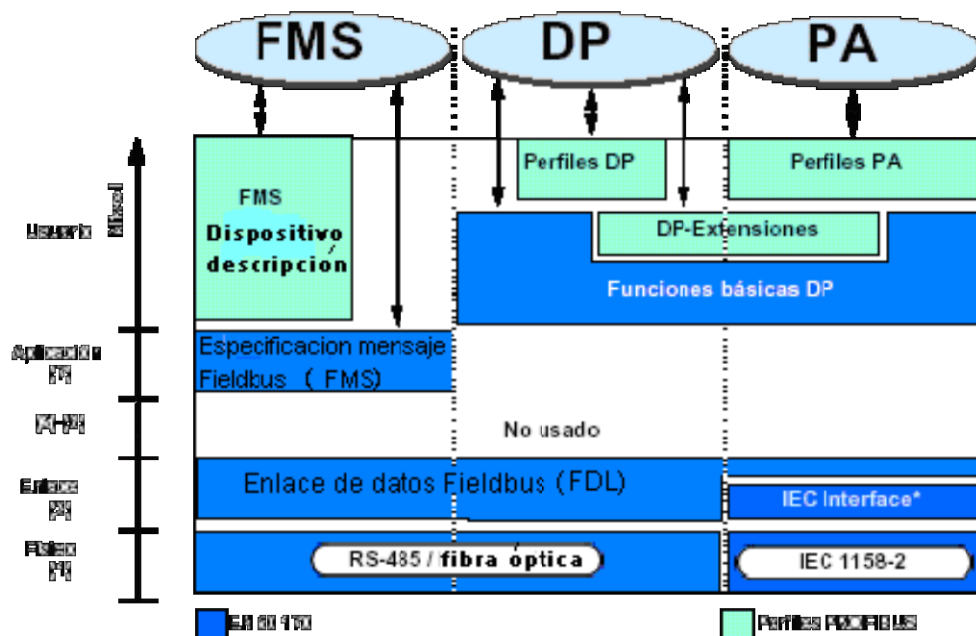


Figura 1.18 Arquitectura de los protocolos PROFIBUS.

- **PROFIBUS CON INDUSTRIAL ETHERNET:**

Industrial Ethernet es un sistema de bus, basado en IEEE 802.3, apto para la industria. Se caracteriza por:

- Conexión de sistemas de automatización entre si y con PC y Workstations para lograr una comunicación homogénea y heterogénea.
- Posibilidad de realizar amplias soluciones mediante redes abiertas.
- Elevado rendimiento de transmisión
- Diferentes soportes de transmisión (cable triaxial, par trenzado industrial, cable de fibra óptica)

La red Industrial Ethernet funciona por el método de acceso normalizado CSMA/CD (carrier sense multiple access with collision detection) en la norma IEEE 802.3. La red Industrial, ofrece una amplia gama de componentes de red para sistema de transmisión eléctricos y ópticos.

Industrial Ethernet, puede realizar en paralelo aplicaciones para Novell Network o TCP/IP sin que se produzcan interferencias mutuas.

Con las interfaces industriales Ethernet, existe la posibilidad de utilizar simultáneamente en la red varios protocolos distintos. De esta manera resulta fácil la transmisión entre las comunicaciones de producción y las de oficina.

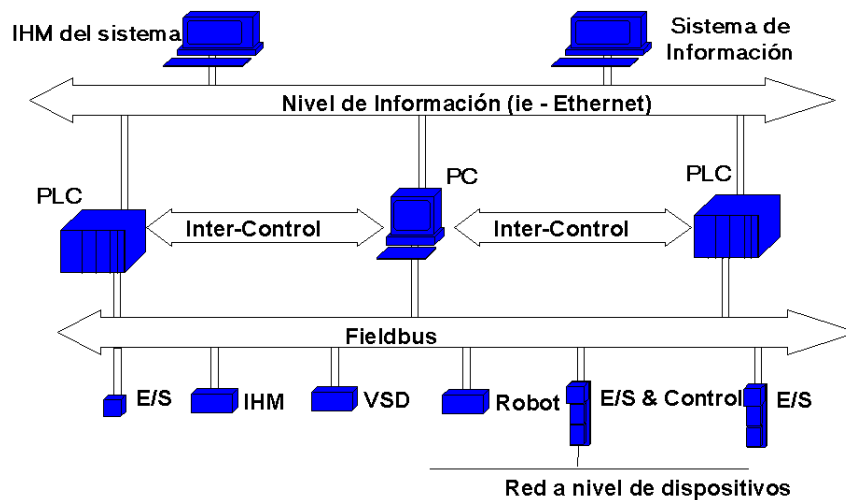


Figura 1.19 Red Industrial Ethernet (jerarquía)

Las redes de comunicación industrial, permiten conocer todo lo referente a un proceso industrial a través de las variables fundamentales medidas por instrumentos instalados en campo, permitiendo a la gerencia saber como está funcionando su empresa.

1.4.5 Cables industriales

Los cables que se emplean en la industria son variantes más resistentes que los cables UTP o Fibra óptica que se usan en las redes LAN tradicionales. Esto es para poder resistir el ambiente industrial que es más agresivo. Una muestra de estos cables industriales se muestra en la Figura 1.20.

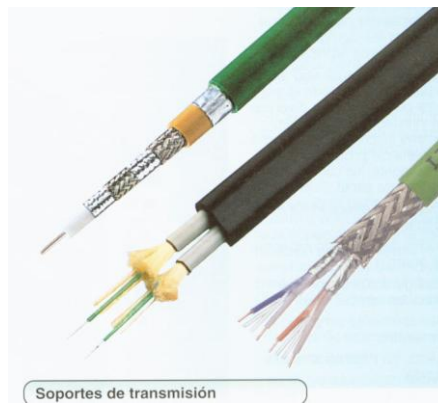


Figura 1.20 Cables Industriales.

- **EL CABLE ITP**

El medio de transmisión más antiguo es el par trenzado cuyo uso se difundió con la telefonía. Este consiste en dos alambres de cobre aislados, en general de 1 mm de espesor, que se entrelazan en forma helicoidal para reducir la interferencia eléctrica que puede venir desde otros cables o pares cercanos, o bien desde una fuente de interferencia EMI/RFI. La interferencia originada al tener varios pares trenzados colocados paralelamente recorriendo distancias considerables se conoce como Cross-Talk o diafonía

Su rendimiento se deteriora con la distancia que recorren; en las redes LAN, por ejemplo, este efecto limita la longitud de las corridas a 100 m

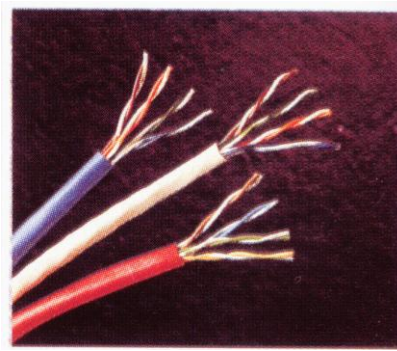


Figura 1.21 Cable UTP

Para transmisión de datos en las redes LAN se emplea una “variante” del par trenzado que se conoce como UTP (Unshielded Twisted Pair) y que tiene cuatro pares de cables (Figura 1.21). Una versión que se emplea en Europa, el (Shielded Twisted Pair) es un cable UTP pero con un blindaje para proteger los pares contra interferencias electromagnéticas y de radio frecuencia (EMI y RFI); por lo mismo, es más inmune al ruido y puede transportar datos a más velocidad que el UTP. La desventaja del STP es que es muy caro y difícil de manejarlo.

En las redes industriales el cable que se emplea es el ITP que se muestra en la Figura 1.22.



Figura 1.22 Cable ITP.

Se caracterizan por tener doble blindaje de muy alta estanqueidad para uso industrial donde se tiene bastante interferencia EMI/RFI y superan la categoría cinco del cable UTP de las LAN; lo que implica que se los puede emplear en

aplicaciones de 100 Mbps. Se los puede utilizar, por ejemplo, para conexión entre armarios eléctricos.

Para ambientes industriales sin mucha interferencia, se puede emplear el cable TP que tiene un calibre apto para montar directamente conectores RJ 45.

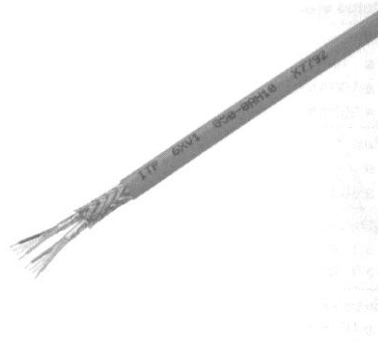


Figura 1.23. Cable TP.

Este cable se lo vende preconfigurado con una longitud máxima de 10 m y cumple con las especificaciones de la categoría 5 de la ISO/IEC 11 801. Es un cable de cuatro hilos (trenzados) agrupados para formar un núcleo de cuatro. Posee dos pantallas globales de lámina compuesta de aluminio y una malla de pantalla de hilos de cobre estañados.

- **CABLE COAXIAL**

El cable coaxial ("coax"), es otro medio típico de transmisión. Hay dos tipos de cable coaxial que se utilizan con frecuencia, uno de ellos es el cable de 50 ohmios, que se utiliza en la transmisión digital y es precisamente el coaxial de banda base; a diferencia del otro tipo, el cable de 75 ohmios, que se emplea en transmisión analógica.

El cable coaxial que se utiliza a nivel industrial (Figura 1.24) posee, a más de las características típicas de un cable de fibra óptica, un blindaje adicional (de allí el nombre de cable triaxial) para darle más solidez.



Figura 1.24 Cable de bus triaxial.

La mejor característica del cable triaxial es su elevada resistencia a la interferencia, gracias a la capa de blindaje extra que al conectarse a tierra lo vuelve “hermético” a altas frecuencias. Cumple con la norma IEEE 802.3 lo que indica que se le puede emplear para construir segmentos de bus industrial Ethernet 10Base 5. Gracias a la pantalla de aluminio exterior este cable es idóneo para tendido subterráneo. Las longitudes de este cable, cuando se usa un segmento completo, puede variar desde 2,5 m hasta 500m.

- **FIBRAS ÓPTICAS**

Los desarrollos recientes en el campo de la tecnología han hecho posible la transmisión de información mediante pulsos de luz. La luz visible tiene un rango de frecuencia de alrededor de 10^8 MHZ, por lo que el ancho de banda de un sistema de transmisión de fibra óptica presenta un potencial enorme.

Un sistema de transmisión óptica tiene tres componentes: el medio de transmisión, la fuente de luz y el detector. El medio de transmisión es una fibra ultra delgada de vidrio o silicio fundido. La fuente de luz puede ser un LED o un diodo láser. El

detector es un fotodiodo que genera un pulso eléctrico en el momento en que recibe un rayo de luz.

En la actualidad los sistemas de fibras ópticas son capaces de hacer transmisiones de datos de 1000 Mbps en 1 kilómetro. Para la industria se emplean diferentes tipos de cable de fibra, todos con fibra de vidrio, como la que se muestra en la figura 1.25.



Figura 1.25 Fibra óptica de vidrio estándar para aplicaciones industriales.

Su construcción es robusta lo que le permite ser empleada en aplicaciones industriales tanto interiores como exteriores.

La fibra óptica, presente ya desde hace mucho en las redes Ethernet o Token Ring, poco a poco está penetrando el ambiente de las redes de campo industriales, precisamente por su aislamiento eléctrico y su resistencia a la interferencia, aún a pesar de su elevado costo y la destreza que se requiere para realizar los empalmes.

1.5 ESTÁNDARES DE LA COMUNICACIÓN FÍSICA

1.5.1 Normas y Asociaciones

Las normas se refieren al acuerdo de los diferentes fabricantes para llevar a cabo una comunicación, tanto en el nivel físico como en el lógico. Para conseguirlo, se

establecen una serie de normas a las que se pueden acoger los fabricantes. Los estándares pueden ser de dos tipos:

- De facto (de hecho) Aceptado por su uso generalizado.
- De iure (de derecho) Propuesto por una asociación.

Algunas de las asociaciones de estándares de comunicaciones más conocidas son:

- ITU- T Unión Internacional de Telecomunicaciones (ITU anteriormente CCITT).
- ISO Organización Internacional de Normalización.
- ANSI Instituto Nacional Americano de Normalización.
- IEEE Instituto de Ingenieros Eléctricos y Electrónicos.
- EIA Asociación de industrias Electrónicas

1.5.2 Comunicación serie:

Se han creado varios estándares para la comunicación serie. Los principales se han definido por la EIA cuyas características principales se presentan el siguiente cuadro.

Tabla 1.9 estándares de la comunicación serie

Norma	RS-232	RS-423	R-422	RS-485
Modo	Simple	Simple	Diferencial	Diferencial
Número de transmisores	1	1	1	32
Número de receptores	1	10	10	32
Longitud máxima (metros)	15	1200	1200	1200
Velocidad máxima (baudios)	20K	100K	10M	10M
Salida del transmisor	±5V mín. ±15V máx	±3,6V mín. ±6V máx.	±2V mín.	±1,5V

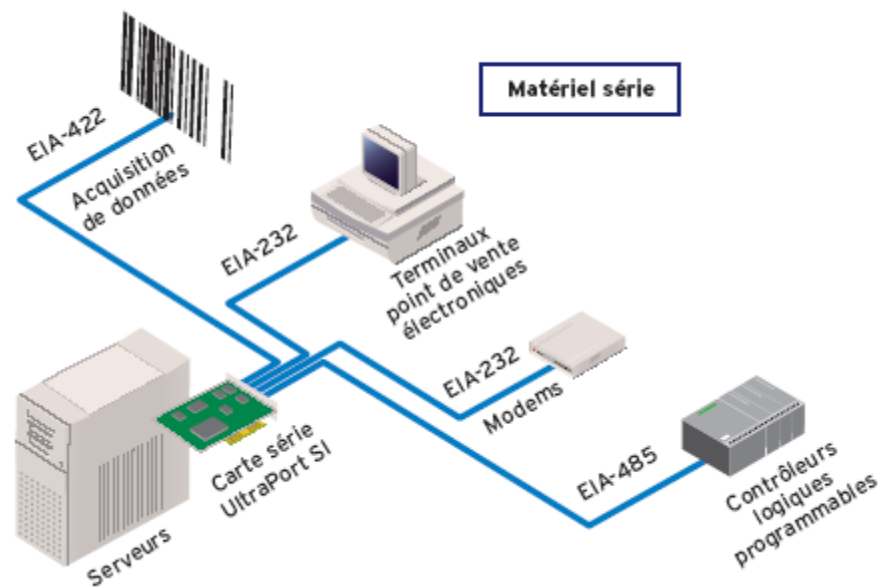


Figura 1.26 Conexión de los estándares serial

- **LA INTERFAZ SERIAL RS-232**

Esta fue una de las primeras técnicas para transmitir datos digitales sobre un medio físico. Hasta ahora sigue vigente sobre todo para comunicar dispositivos de tipo industrial como un PLC con una PC o su consola de configuración.

En la transmisión asíncrona no se emplea una señal de reloj, más bien se utiliza una técnica que recurre a “encapsular” los datos con un bit de inicio y uno o dos bits de parada, y así no es necesaria la línea extra de sincronismo como en la comunicación sincrónica.

Pero encapsular con un bit de inicio y otro de parada los datos no es suficiente, hay otras reglas de transmisión sobre las que se deben poner de acuerdo el transmisor y receptor. Por ejemplo, mientras el estado de la línea está en alto, el receptor deberá interpretar como que no existe transmisión y, por lo mismo, el canal está en modo de espera (idle). Cuando la línea de comunicación cambia de

estado (se recibe el bit de inicio), el receptor debe interpretar ese cambio como el comienzo de la transmisión. Pero, ¿qué sucede si ese cambio de estado fue producido por un transitorio? En situaciones así, el algoritmo de comunicación debe instruir al receptor esperar cierto tiempo para volver a leer la línea y así asegurarse que el cambio de estado corresponde efectivamente al bit de inicio. Pero, volver a leer la línea ¿después de qué tiempo? Igualmente, ¿cada qué tiempo el receptor debe leer la línea para leer un bit de la trama? En ambas situaciones se comprenderá que se necesita saber el tiempo de duración de un bit. Por todo lo indicado, antes de iniciar cualquier comunicación con el puerto RS-232 se debe determinar el protocolo a seguir. Esto debe ser hecho por el usuario quien debe decidir sobre:

- El Protocolo serial: esto es, el número bits de datos, la paridad, el número de bits de parada)
- La velocidad de transmisión.
- El protocolo de control de flujo (RTS/CTS o XON/XOFF).

Con RS232C se puede transmitir los datos en grupos de 5, 6, 7, u 8 bits aunque los más usados son 7 y 8 bits.

La velocidad de transmisión (normalmente 9600 bits por segundo para aplicaciones industriales) debe ser constante durante la transmisión de una trama para garantizar que los bits lleguen uno tras de otro en el momento correcto. Cualquier retardo provocaría una lectura incorrecta.

Un concepto muy importante que debe entenderse es el del **throughput** (rendimiento) de la transmisión de datos.

Si bien el encapsulamiento es esencial para la transmisión asíncrona, por otro lado, tiene el defecto de añadir bits a los datos propiamente dichos (la trama de datos) disminuyendo consecuentemente la velocidad efectiva de la transmisión. Note que si se transmiten 8 bits y se añade un bit de inicio y un bit de parada, independientemente de la velocidad de transmisión, el rendimiento de la transmisión se reduce en un 20%. Esto es, el throughput apenas llega al 80%. Par

protocolos más complejos, el encapsulamiento (que es referido también como **overhead**) es mayor y por lo mismo afecta más al throughput.

El RS-232C es un estándar que constituye la tercera revisión de la antigua norma RS-232, propuesta por la EIA (Asociación de Industrias Electrónicas), realizándose posteriormente una versión internacional por el CCITT, conocida como V.24. Las diferencias entre ambas son mínimas, por lo que a veces se habla indistintamente de V.24 y de RS-232C (incluso sin el sufijo "C"), refiriéndose siempre al mismo estándar.

El estándar RS-232 establece que un 1 lógico se representa con un voltaje entre -3V y -15V, mientras que un 0 lógico se representa con un voltaje entre +3V y +15V.

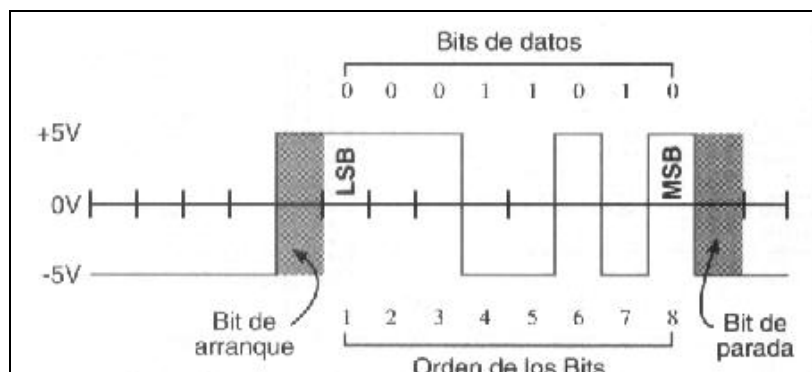


Figura 1.27 Trama RS-232 típica.

Los voltajes más usados son +12V y -12V. El estado de reposo (idle) se representa con un 1 lógico; es decir, -12V. Dependiendo de la velocidad de transmisión empleada, es posible tener cables de hasta 15 metros. En la Figura 1.27 se muestran los voltajes RS-232 para voltajes de +/- 5V.

Para compensar los efectos de la atenuación de voltaje en la línea y los efectos del ruido, se han establecido diferentes niveles de voltaje tanto para el lado del transmisor como del receptor, tal como se muestra en la Figura 1.28

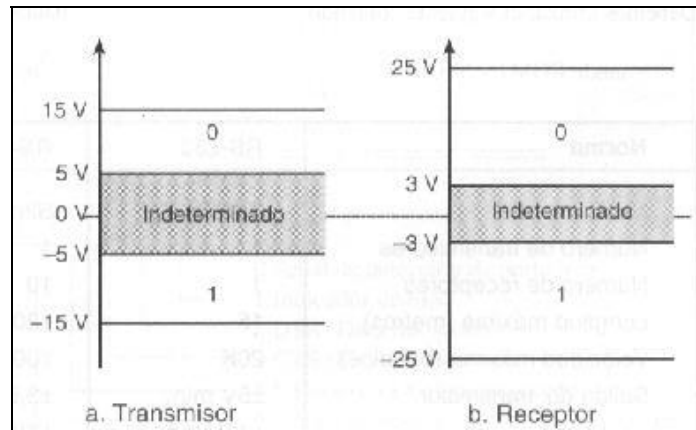


Figura 1.28 Niveles de voltaje en el Tx y en el Rx.

El puerto serial de una PC se rige por el estándar RS-232C. Este estándar fue empleado en los 60s para comunicar un Equipo Terminal de Datos, DTE (Data Terminal Equipment, el PC en este caso) y un equipo de comunicación de datos, DCE (Data Communication Equipment, habitualmente un modem).

Para conseguir los niveles de voltaje correctos hay circuitos que dan soporte para convertir los niveles de voltaje TTL a RS-232 y viceversa. En el ejemplo de la Figura 1.29 se muestran los circuitos que se emplean en una computadora para manejar la interfaz serial RS-232 de la misma.

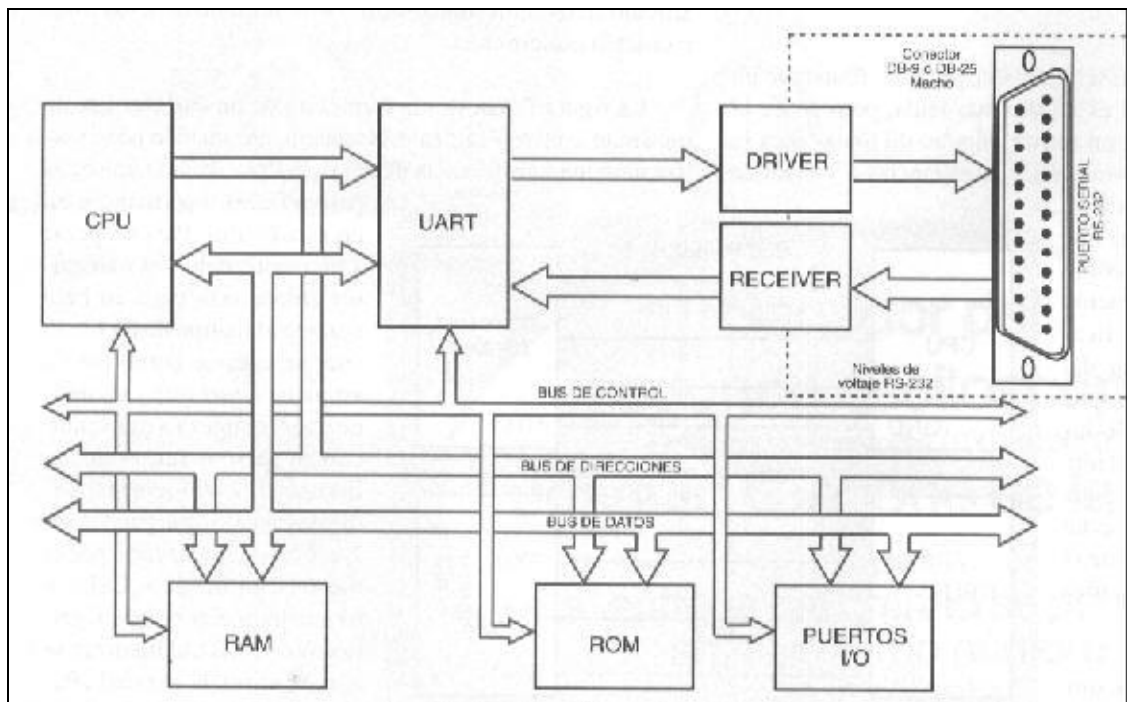


Figura 1.29 Trabajo del UART.

El driver convierte de TTL a RS-232 para la transmisión y el receptor (receiver) de RS-232 a TTL.

El estándar especifica un conector DB-25 de 25 pines y que el conector DTE debe ser macho y el conector DCE hembra.

Por esta razón en muchas PC modernas se utiliza el conector DB-9 macho.

Para controlar el puerto serie, la computadora usa direcciones de E/S e interrupciones (IRQ): para COM1 la dirección 3F8h y el IRQ4 y para COM2 la 2F8h e IRQ3. Luego el estándar añadió nuevos puertos serie siendo las direcciones de E/S, 3E8 para COM3 y 2E8 para COM4, sin especificar las IRQ, debiendo el usuario definir las en función de las que tenga libres o el uso que vaya a hacer de los puertos serie. Se puede tener en diferentes puertos las mismas IRQ siempre que no se utilicen los dos al mismo tiempo.

la RS-422, pero, los drivers RS-422 no deben ser empleados con la RS-485 porque no pueden proveer control del bus.

Tabla 1.10 RS-485 and RS-422 Standards

	RS-422	RS-485
Modo de operación	Diferencial	Diferencial
No. permitido de Tx y Rx	1 Tx, 10 Rx	32 Tx 32 Rx
Máxima longitud del cable	4000ft	4000ft
Máxima taza de datos	10Mbps	10Mbps
Mínimo rango salida driver	±2V	±1.5V
Máximo rango salida driver	±5V	±5V
Máxima corriente corto	150mA	250mA
Impedancia de carga Tx	100	54
Sensibilidad entrada Rx	±200mV	±200mV
Máxima resistencia ent. Rx	4k	12k
Rango volt. Entrada Rx	±7V	-7V to +12V
Uno lógico Rx	>200mV	>200mV
Cero lógico Rx	<200mV	<200mV

Los enlaces diferenciales de las interfaces RS-485 y RS-422 proporcionan una transmisión confiable ante la presencia de ruido, y las entradas diferenciales de los receptores pueden rechazar voltajes en modo-común elevados.

Con velocidades de transmisión altas se comienza a tener problemas con las reflexiones. Para contrarrestar éstas, se instalan resistencias de terminación con un valor igual al de la impedancia del cable. Para cables RS-485 comunes (pares trenzados de 24AWG), esto significa resistencias de 120 en los dos terminales.

Se puede afirmar que la RS-485 fue el primer intento por evitar un enlace físico de datos para cada sensor o transmisor. Es útil cuando no se deben monitorear o controlar muchas variables. Sin embargo, se tiene todavía un problema: no se puede conectar todos los dispositivos a un solo pórtico RS-485, para lograr una mejor flexibilidad y eficiencia que es lo que actualmente demanda la industria moderna.

1.5.3 Comunicación paralela

Los dos más importantes interfaces estándares para la comunicación paralela de datos estos son:

- Bus de propósito general (GPIB) o IEEE-488, usado principalmente para la medición y prueba automática de equipos científicos, el cual será estudiado en detalle como parte de este trabajo.
- The Centronics Standard, usado principalmente para el cableado entre Pcs e impresoras.

CAPÍTULO II

LA INTERFASE GPIB

GPIB es un bus y un protocolo estándar para el control y comunicación con instrumentos de medida, como multímetros digitales, osciloscopios, etc, que permite configurar sistemas automáticos en el laboratorio y en la industria con gran flexibilidad y potencia.

2.1 HISTORIA DEL BUS GPIB

GPIB es un estándar de conexión que permite la comunicación de un ordenador con instrumentos electrónicos de medida, como pueden ser generadores de funciones, osciloscopios, etc. Las siglas corresponden a *General Purpose Interface Bus*, pero a pesar de este nombre, fue diseñado específicamente para la conexión de instrumentos de medida. Fue creado en 1965 por la compañía Hewlett-Packard, que lo denominó originalmente HP-IB, y se popularizó con rapidez, debido a sus altas tasas de transferencia de datos (8 Mbytes/s). Para evitar la dispersión de características, los principales fabricantes acordaron la estandarización del GPIB en 1975 (IEEE 488.1), centrándose en las características eléctricas y mecánicas del bus.

Una segunda estandarización (IEEE 488.2 de 1987) delimitó de forma más concreta la programación del GPIB, definiendo comandos de aparatos, formato de mensajes y estado de los instrumentos. El siguiente paso de importancia fue la adopción del formato de comandos SCPI, que estructura las órdenes a los aparatos de forma coherente, permitiendo (hasta cierto punto), la sustitución de instrumentos de distintos fabricantes con mínimos cambios. En la figura 2.1 se muestra la evaluación del estándar 488.

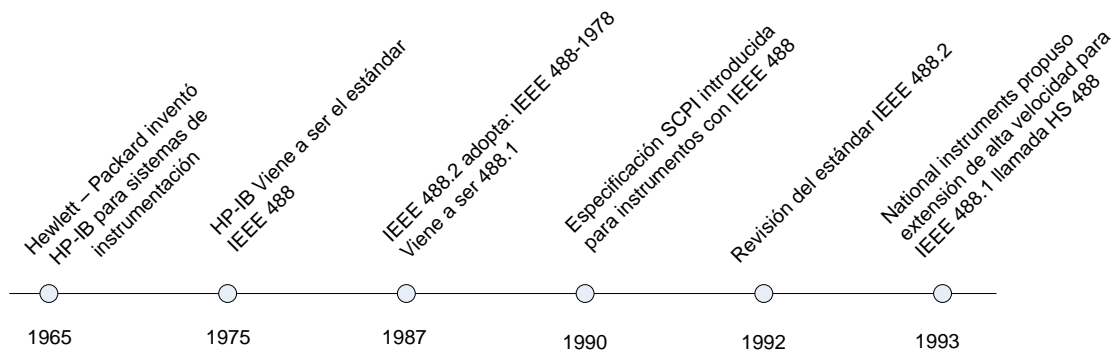


Figura 2.1 Evolución del Estándar 488

2.1.1 La Norma IEE488.1(GPIB) Nivel Físico: Estructura del bus GPIB

El bus GPIB está basado en 16 líneas activas, además de la tierra. Estas 16 líneas se organizan en tres buses como se indica en la figura 2.2.

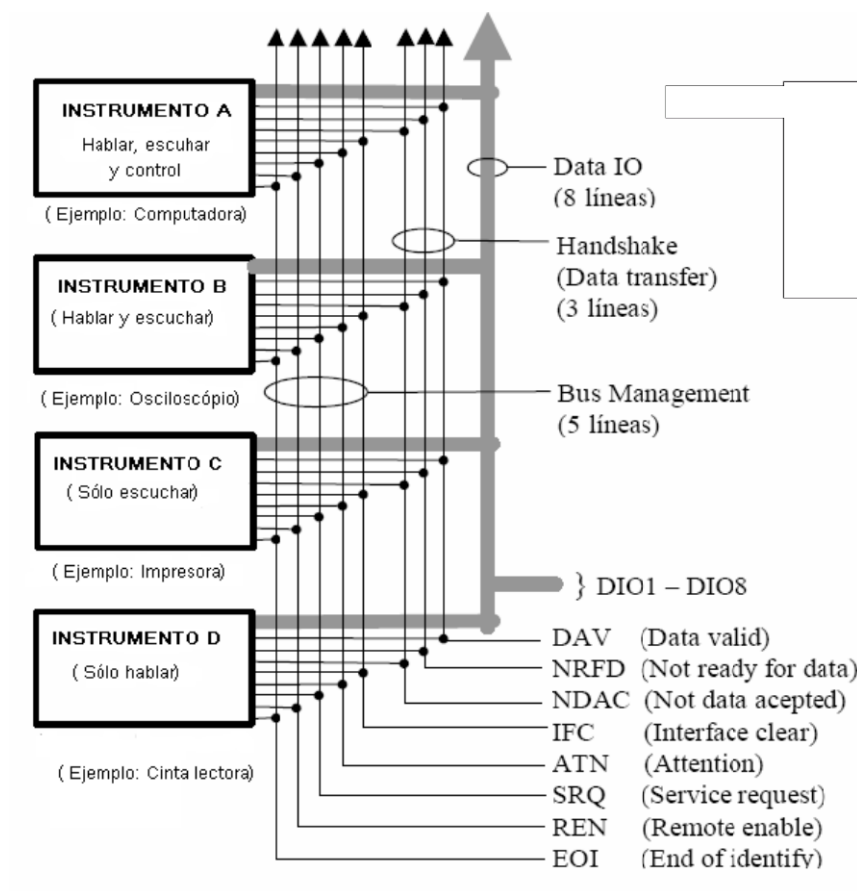


Figura 2.2 Líneas del bus GPIB

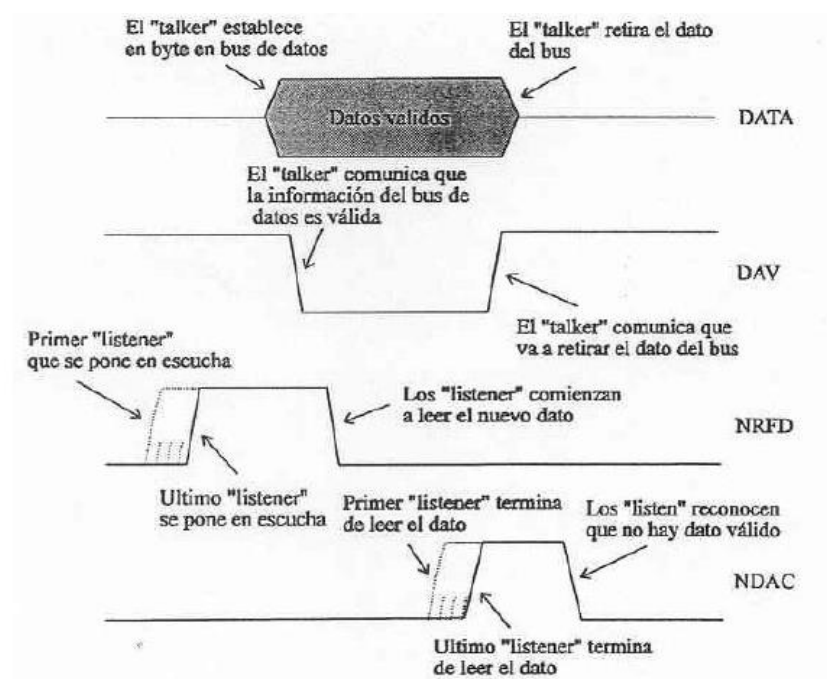
- a) **Bus de Datos (DIO1-DIO8)** (Data input/output): Es un bus bidireccional de 8 líneas orientado a la transferencia de byte o de caracteres ASCII.

Las 8 líneas de datos permiten que el único equipo establecido como "talker" (o en otros casos el "controller") envíe un byte en paralelo hacia todos aquellos equipos que en ese instante estén definidos como "listener".

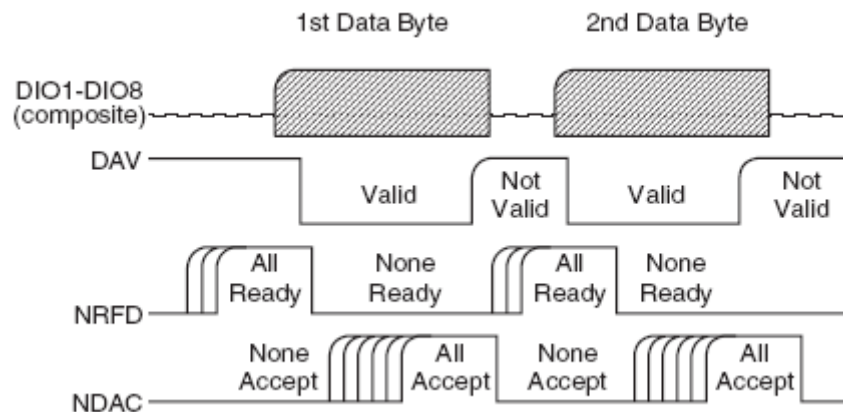
- b) **Bus de sincronización de la transferencia de datos:** Es un conjunto de tres líneas (**DAV**: Data valid, **NRFD**: Not Ready For Data y **NDAC**: Not Data Accepted) que se utilizan de forma coordinada para asegurar la transferencia de datos entre los equipos. Su operación se muestra en la figura 2.3.

- **DATA VALID (DAV):** Es una de las líneas de sincronización que permite la transferencia de datos por el bus de datos. Un TRUE lógico en esta línea significa que el equipo establecido como "talker" activo ha establecido unos datos válidos sobre el bus de datos que deben ser leídos por todos los equipos establecidos como "listener".
- **NOT READY FOR DATA (NRFD):** Es una de las líneas de sincronización, que es gobernada por los equipos establecidos como "listener". Cuando esta línea está en estado lógico TRUE, significa que algún equipo de entre los "listener" no está aún dispuesto para aceptar nuevos datos. El que esta línea se encuentre en estado lógico TRUE, inhibe al equipo "talker" a que inicie el envío de un nuevo dato. El que esta línea esté en estado lógico FALSE, significa que todos los equipos "listener" se encuentran a la espera de un nuevo dato.

- NO DATA ACCEPTED (NDAC):** Es la tercera línea de sincronización de datos, y es gobernada por los equipos que están establecidos como "listener". Cuando se encuentra en estado lógico TRUE, significa que alguno de los equipos establecidos como "listener" aún está pendiente de leer un dato, y en consecuencia, el talker debe esperar aún para retirar los datos. Cuando esta línea se encuentra en estado lógico FALSE significa que ya todos los equipos establecidos como "listener" han leído el dato transferido.



(a)



(b)

Figura 2.3 (a,b)Típico Handshake de una operación con GPIB

c) **Bus de control:** Está constituido por 5 líneas (**ATN:** ATteNtion, **IFC:** InterFace Clear, **SRQ:** Service ReQuest, **REN:** Remote ENable, y **EOI:** End Or Identify) que se utilizan para transferir comandos entre los equipos relativos al modo de interpretar los datos que se transfieren o comandos básicos de gobierno de la interfaz del bus.

- **Attention (ATN):** Es una señal que establece el "controller" para definir con un estado lógico TRUE en ella que el dato que se envía por el bus de datos es un comando enviado por el "controller". Cuando esta línea toma el estado lógico FALSE indica que el byte del bus de datos debe ser considerado como un dato.
- **Interface Clear (IFC):** Está bajo el exclusivo control del "system controller". Cuando es establecido en esta línea un estado lógico TRUE, todos los equipos conectados al bus deben ser reseteados, y todos ellos deben pasar a su estado base.
- **Service request (SRQ):** Es utilizado por los equipos conectados al bus para comunicar al "controller" que requieren ser atendidos por alguna causa (ha concluido una actividad, se ha producido un error, existe algún dato para transferir, etc.). Cuando el "controller" detecta un estado lógico TRUE en esta línea, debe iniciar una encuesta (polling) para determinar que equipo causó el requerimiento, y en el caso de que proceda, satisfacer su demanda.
- **Remote Enable (REN):** Es una línea con la que el "controller" al establecerla a un estado lógico TRUE, habilita a todos los equipos conectados al bus para que reciban datos, o comandos.
- **End or Identify (EOI):** Esta línea tiene dos funciones:

- En primer lugar, cuando un dispositivo actúa como transmisor al bus y quiere finalizar esta transmisión, avisa a sus receptores y al controlador de que ha acabado conectando esta línea.
- En segundo lugar, esta línea es utilizada por el "controller" para iniciar una encuesta paralela. En este caso el "controller" debe poner simultáneamente a estado lógico TRUE las señales ATN y EOI, y como respuesta a ello, los equipos que previamente hayan sido configurados para participar en la encuesta paralela transfieren sus bit de status sobre el bus.

2.1.2 Tipos de mensajes que intercambian los equipos

Entre los equipos conectados al bus GPIB se transfieren mensajes constituidos por secuencias de byte por transacción. De acuerdo con el estado de la señal de control "ATN", existen dos tipos de mensajes:

"Data": Mensaje que contiene información relativo a la funcionalidad de un equipo. Ejemplos son: instrucción de programación, resultado de medida, estatus de un equipo, etc.

"Commnad": Mensaje que tiene como función controlar el modo de operación del bus. Ejemplos son: Inicialización del bus, cambio del modo de operación de un equipo, transferencia del control, etc.

2.1.3 Modos de operación de un equipo

En cada momento, los equipos conectados al bus GPIB pueden estar operando como uno o varios de los siguientes modos de comportamiento:

"Controller": Con capacidad de establecer quien envía o recibe datos y el modo de operación del bus (solo un equipo puede ser "controller").

"Talker": Con capacidad de enviar datos a otros equipos.

"Listener" : Con capacidad de recibir datos de otros equipos.

"Idler" : Sin ninguna capacidad respecto del bus.

- a) **"Controller"**: Normalmente la mayoría de los sistemas GPIB consisten de un ordenador y varios instrumentos. En ese caso el ordenador suele ser el sistema consolador. Si hay varios ordenadores conectados, es decir, varios controladores en un instante de tiempo, sólo puede haber un controlador master, el llamado CIC (Controller –In – Charge) . Este controlador puede pasar el control a otro que esté inactivo.

Las principales responsabilidades del controlador son definir las normas de comunicación, responder a los dispositivos que solicitan servicio, enviar comandos GPIB, pasar y recibir el control y establecer el protocolo necesario para indicar que dispositivos trabajarán como listeners y cual como talker.

Existen dos tipos de controller:

- **"System Controller"**: Tiene capacidad en hardware de tomar el control del bus en todo momento, a través de las líneas "IFC" y "REN".

En un bus sólo puede existir un único System Controller y está caracterizado por tener las capacidades especiales en hardware de poder establecer el estado de las líneas "IFC" y "REN".

- **"Active controller"**: Tiene la capacidad de transferir mensajes de tipo Command para:
 - Establecer los modos de operación "Listener" y "Talker" en los restantes equipos.
 - Enviar los comandos de inicialización y sincronización del bus.
 - Supervisar mediante encuesta el status de los equipos.

Cada bus puede tener conectado uno o más dispositivos capaces de asumir la función de "active controller", aunque en cada momento, solo uno de esos equipos puede operar como tal.

En una situación estándar, hay un computador conectado al bus que actúa a la vez como "system controller" y como único "active controller".

En algún momento un equipo puede requerir del controlador del sistema su interés en convertirse en controlador activo, a fin de llevar a cabo una operación compleja, tal como transferir unos datos al "plotter", o almacenar un fichero en un disco, etc.. Como respuesta a este requerimiento el controlador del sistema, transferirá el control al equipo que lo ha solicitado, el cual pasa a constituirse en controlador activo del bus. Cuando concluye su operación, retorna de nuevo el control del bus, al controlador de sistema. Obsérvese que las capacidades propias de controlador de sistema no pueden ser transferidas.

- b) **Talkers y Listeners:** En general todos los dispositivos que se conectan al bus GPIB son tanto Talkers como Listeners aunque existen algunos que sólo hablan o escuchan.

Talker: Equipo con capacidad de enviar Data. En cada bus pueden existir uno o varios equipos con capacidad de enviar datos a otros equipos por el bus, pero en cada instante sólo uno de ellos puede ser establecido por el controller para que opere como Talker.

Es el único equipo (además del Active Controller) con capacidad de establecer el estado de las líneas DAV.

El equipo Talker sólo puede enviar un dato si todos los equipos que se encuentran en modo Listener esté en disposición de leerlo (Linea NRFD a valor lógico FALSE).

Listener: Equipo que recibe y lee todos los datos que se transfieren por el bus. En cada bus pueden existir uno o varios equipos con capacidad de recibir datos desde el bus, y uno o varios de ellos se puede encontrar simultáneamente en modo “Listen”. El Active Controller es el que establece a través de un comando que un equipo pasa o deja de estar en modo Listen. Todos los equipos que se encuentran en estado Listen reciben simultáneamente todos los datos que son transferidos por el bus.

Hay que destacar el hecho de que ni Talkers ni Listeners pueden realizar sus funciones a no ser que el controller se lo indique.

- c) **Idler:** Estado base sin ninguna respuesta respecto del bus.

2.1.4 Protocolos y Normas

En las especificaciones funcionales del bus GPIB hay que destacar una serie de protocolos y normas que siguen controllers, talkers y listeners para comunicarse entre ellos. De esos protocolos se puede destacar el protocolo de direccionamiento y el de reconocimiento y petición de servicio, y de normas, la que el final de las transferencias de datos.

- a) Protocolo de direccionamiento. Antes de que tenga lugar cualquier transferencia de datos en el bus, es necesario que algún dispositivo sea designado como talker y algún otro como listener. El controlador es el encargado de decidir quien hablará y quien escuchará, colocando un mensaje de comando especificando la dirección Talker y Listener.

El formato del byte de mensaje de direccionamiento es el siguiente:

Tabla 2.1 Formato del protocolo de direccionamiento

7	6	5	4	3	2	1	0
	TA	LA	X	X	X	X	X

Los bits del 0 al 4 contienen la dirección binaria GPIB del dispositivo y los bits 5 y 6 informan de si es una dirección Listener o una dirección Talker.

Respecto a la dirección secundaria, si un dispositivo la tiene, se direcciona enviando primero su dirección primaria y posteriormente la secundaria. La utilidad de la dirección secundaria versa en situar en un solo dispositivo varios instrumentos cada uno en una dirección diferente

Además de estos comandos de direcciones también aparecen dos comandos adicionales: Untalk y Enlisten, que sirven para deshabilitar a los posibles listeners y al posible talker que estén activos en ese momento

- b) Protocolo del Polling Serie: En el bus GPIB existe una línea de control, SRQ, que activada por un dispositivo conectado al bus, informa al controller de que existe algún instrumento que necesita atención.

Para descubrir quién ha activado la línea existen dos métodos: Un polling o muestreo serie y otro paralelo.

Al generar un Serial Poll el controller envía un mensaje de comando especial (mensaje remoto) a un dispositivo en concreto que está en cierta dirección: Es el SPE (Serial Poll Enable). En ese momento el dispositivo envía un byte donde informa sobre si ha sido él o no, el que ha solicitado el servicio. Al recibir el controller ese byte, envía otro mensaje especial, SPD (Serial Poll Disable), con lo cual el instrumento retorna a su estado normal de Talker o Listener.

El modelo del byte que se envía para el Standard IEEE 488.1 es el siguiente:

Tabla 2.2 Formato del protocolo del Polling Serie

7	RQS	5	4	3	2	1	0
---	-----	---	---	---	---	---	---

Donde el bit 6 informa de si ha sido o no ése, el dispositivo que ha solicitado el servicio.

c) Norma sobre finalización de las transferencias de mensajes: Cuando lo que se está transmitiendo son mensajes propios de cada dispositivo, es decir la línea ATN se encuentra inactiva y en el bus hay un talker enviando datos y un listener recibiéndolos, existe formas de indicar de que el byte que se está transmitiendo es el último. Los métodos son varios:

- Método EOS (End of String). Este método utiliza un caracter EOS para marcar el final de la transferencia de datos. Ese caracter puede ser cualquiera; sin embargo son comunes el carácter de retorno de carro o de salto de línea. Cuando el listener lee el carcter EOS sabe que no hay más datos y completa la operación de lectura. Para utilizar este método es necesario configurar tanto Talker como Listener para que utilicen el mismo EOS.
- Método EOI: Se utiliza la línea EOI del propio bus para indicar cuando se está enviando el último caracter. Los listeners monitorizan esta línea y al detectar que está conectada completan el ciclo.
- Método de cuenta: Si se conoce el número de bytes que se va a recibir se puede leer esa cantidad en concreto.

2.2. LA NORMA IEEE 488.2 (GPIB)

En 1987 se revisó la norma en su versión 1 y, a su vez, se introdujo una segunda versión: La ANSI/IEEE Std 488.2-1987. Este hecho supuso un paso más en la compatibilidad de instrumentos, introduciendo soluciones como:

- Definir el conjunto mínimo de capacidades que ha de tener un instrumento.
- Especificó la forma de presentar los datos a través del bus
- Definió un protocolo para enviar mensajes de dispositivo y la forma de enviar varios mensajes en una sola cadena de caracteres

- Dio un conjunto de comandos comunes a todos los instrumentos.
- Definió el modelo estándar de bytes de información sobre el estado del dispositivo.

Los equipos que satisfacen los modos operativos definidos en el estándar IEEE-488.2., necesariamente satisfacen los niveles de comunicación físicos definidos en el estándar IEEE-488.1. La implicación inversa no es requerida.

El modelo de intercambio de un equipo que satisface el estándar IEEE-488.2, incluye a los siguientes componentes:

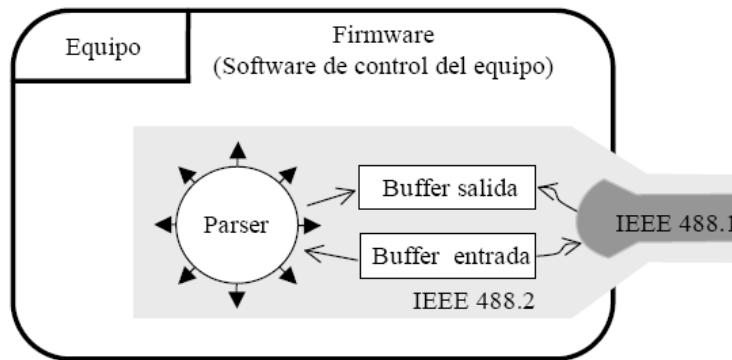


Figura 2.4 Componentes de un modelo de intercambio

- **Buffer de entrada:** Es el área de memoria en la que las ordenes y requerimientos de entrada son almacenadas, antes de que sean interpretadas y ejecutadas por el Parser. El buffer de entrada, permite que el equipo controlador del instrumento, almacene en el buffer un string conteniendo una o varias ordenes que llevan un cierto tiempo ser ejecutada, y mientras esto ocurre pueda realizar otras operaciones con otros equipo.
- **Buffer de salida:** Es el área de memoria en el que los datos de salida (mensajes de salida) son almacenadas en espera de que el controlador decida leerlas.
- **Parser:** Es el controlador interno del equipo que interpreta los mensajes completos almacenados en el buffer de entrada, los ejecuta, y en caso de que sean de requerimiento, deposita en la cola de salida los datos que resulten.

2.3 COMUNICACIÓN GPIB

2.3.1 Principios de comunicación

Un instrumento GPIB recibe su información de otros dispositivos, usualmente este es el sistema controller. El instrumento distingue entre: eventos, comandos, preguntas y datos. La figura 2.5 muestra el flujo de varios tipos de información.



Figura 2.5 Flujo de información con el bus GPIB

- a) **Eventos:** El bus GPIB tiene un número especial de líneas de control que permiten el rápido y simple paso de la información primitiva llamado eventos. Esta información primitiva es también llamada “Interface Events” o “Interface Messages” por que ellas son inicializadas y procesadas por la interfase de hardware. Un ejemplo de este evento es el IFC(Inter Face Clear) evento que obliga al instrumento a escuchar al sistema controlador.
- b) **Comandos:** Los comandos son Strings los cuales hacen que los instrumentos ejecuten una acción, tales como tomar medidas o activar un motor.
- c) **Queries (preguntas).** Son Strings las cuales hacen que el instrumento genere una respuesta. Estas preguntas usualmente no provocan en el instrumento que ejecuten una acción y siempre terminan con un signo de interrogación (?).
- d) **Unidad de Datos:** Son usados para pasar información. Los datos pueden estar en forma de parámetros, los cuales vienen con comandos o preguntas del instrumento. Los datos también pueden ser la respuesta de un instrumento.

2.3.2 Protocolo básico

- El equipo y el controlador se comunican intercambiando mensajes de ordenes y de respuestas completos. Esto significa que el controlador siempre debe terminar un mensaje de órdenes antes de intentar leer una respuesta.
- Los mensajes de ordenes son enviados por el controlador, y pueden ser de dos tipos:
 - **Ordenes de control:** Que requieren un cambio de estado del equipo, pero que no requieren ninguna respuesta.
 - **Ordenes de requerimiento:** Que solicitan información sobre el estado del equipo o sobre la información que posee.
- El equipo habla (envía un mensaje de salida) como respuesta de una orden de requerimiento.
- El controlador solo admite un mensaje de salida (respuesta de una orden de requerimiento), y lo solicita antes de enviar un nuevo mensaje de órdenes. En caso contrario se genera una situación de bloqueo.
- La regla básica del protocolo es:

"El equipo solo habla cuando está dispuesto a ello, y en ese caso, tiene que hablar antes de que se le ordene hacer una cosa nueva"
- Cuando el equipo es encendido, o cuando recibe una orden de inicialización "*CLS" el buffer de entrada y de salida son inicializados, y el parser es inicializado en la raíz de su árbol de ordenes.
- Si el equipo envía un mensaje de respuesta, el controlador debe siempre leer de forma completa el mensaje, antes de que envíe un nuevo mensaje de órdenes al equipo.
- El controlador puede enviar un mensaje conteniendo múltiples órdenes de requerimientos. A esto se le denomina un "requerimiento compuesto". Los diferentes requerimientos dentro del mensaje deben estar separados por el delimitador ";". Los mensajes de respuesta son encolados en la cola de salida, separados entre sí por el delimitador ";" .
- Los comandos son ejecutados en el orden en que han sido recibidos.

2.3.3 Protocolos de excepción:

Cuando un error ocurre en el intercambio de la información, éste no termina en la forma normal, sino que sigue un protocolo de excepción:

a) Equipo direccionado para hablar sin nada en la cola:

- Si es consecuencia de que el equipo no haya recibido una orden de requerimiento, el equipo indicará un error de encolamiento, y no enviará ningún byte por el bus.
- Si es como consecuencia de que la orden de requerimiento previa no se ejecutó como consecuencia de un error, el equipo no indica ningún error de encolamiento, sino que el equipo espera a recibir el siguiente mensaje del controlador.

b) Equipo direccionado para hablar sin que ningún equipo escuche: En este caso, el equipo esperará o bien a que algún equipo escuche, o a que el controlador tome el control.

c) Error de orden: Se genera cuando se detecta un fallo de sintaxis o una orden no reconocible.

d) Error de ejecución: Se genera si un parámetro está fuera de rango, o si el equipo se encuentra en un estado que no permita la ejecución del comando requerido.

e) Error específico del equipo: Se produce cuando el equipo es incapaz de ejecutar una orden, como consecuencia de una razón estrictamente dependiente de él, y no del protocolo seguido por el bus.

f) Error de encolamiento: Se genera si no se sigue el protocolo de lectura de los datos de la cola de salida.

g) Condición inconclusa: Si el controlador intenta leer un mensaje de respuesta antes de que el programa haya concluido de ejecutar la orden que los genera. En este caso el Parser, se inicializa a si mismo, la respuesta ya elaborada es limpiada de la cola de salida, y ningún dato es transferido por el bus.

h) Condición interrumpida: Si el controlador no lee completamente el mensaje

generado por un mensaje de requerimiento y envía otro mensaje de orden, el equipo genera un error de encolamiento, y el segmento de mensaje de salida no leído es eliminado. La orden que interrumpe es inafectada.

- i) **Bloqueo de buffer:** El equipo alcanza un estado de bloqueo si el buffer de entrada, está lleno y también está llena la cola de salida. Esta situación ocurre si un mensaje de orden muy largo que contiene órdenes de requerimiento ha sido enviado, y genera un mensaje de salida superior al que puede contener la cola. El controlador no puede terminar de enviar el mensaje de entrada porque no cabe, y el buffer de entrada no se vacía porque espera que su cola de salida sea vaciada para concluir la orden en ejecución. En este caso el equipo rompe el bloqueo, limpiando la cola de salida y generando un error de encolamiento.

2.3.4 Estatus del equipo por el protocolo IEEE 488.2.

El estandar IEEE 488.2 ofrece un mecanismo estandarizado de presentar y mostrar el estado interno del equipo. A través de este mecanismo, se puede tener información de si el equipo tiene dato dispuesto para transferir, así como si algún tipo de error ha ocurrido.

Aquí la norma obliga a que los instrumentos soporten una estructura de bytes donde guarden y actualicen su estado. El byte de estado de la norma 488.1 no es más que una parte de esta nueva estructura. Los registros de estado del satandard 488.2 se muestra en la figura 2.6

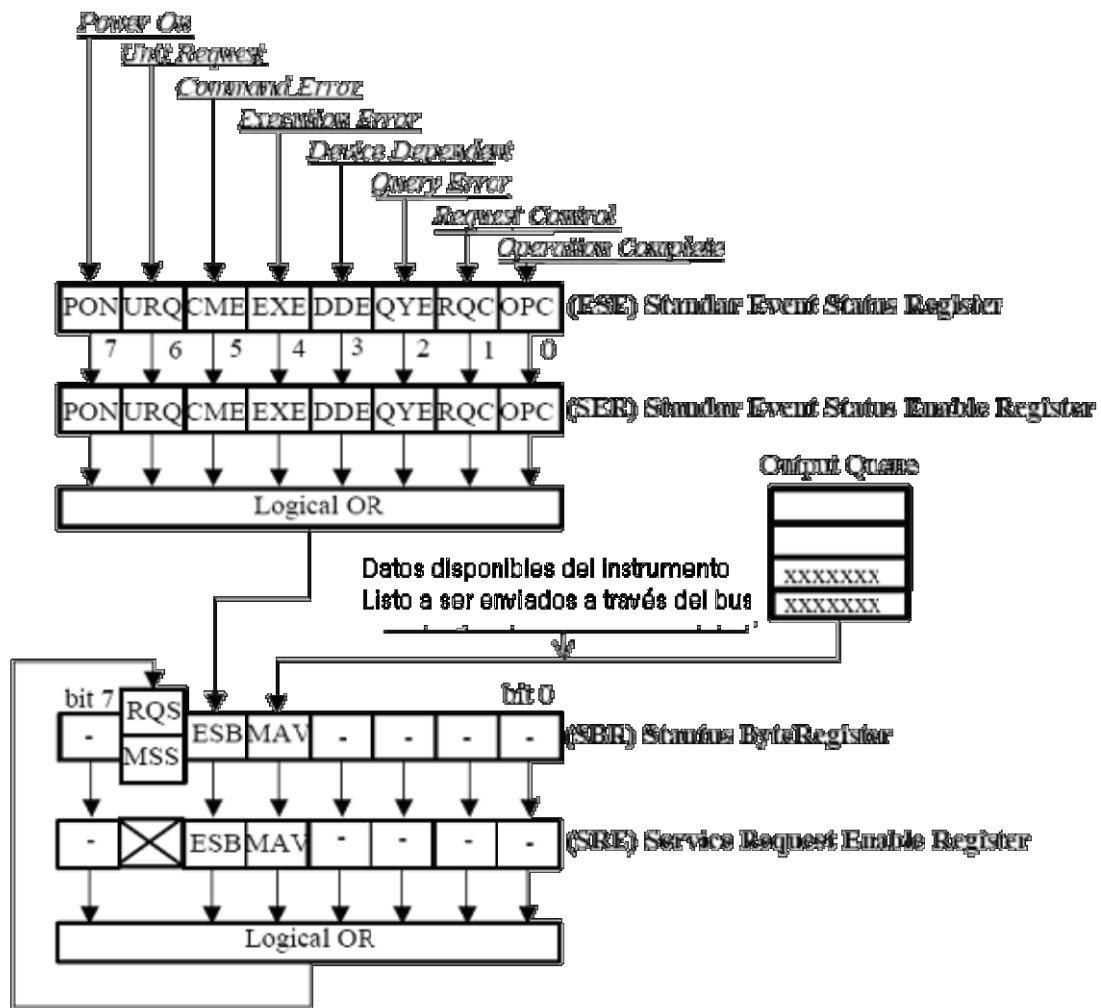


Figura 2,6 Registros de estado

Sobre los diversos registros se puede destacar:

- El “Estándar Event Status Register” (8 bits). Agrupa un conjunto de informaciones que hacen referencia a sucesos y errores muy frecuentes en todos los instrumentos. En el caso de que uno de esos sucesos se verifique, por ejemplo que llegue un comando erróneo, el dispositivo debe encargarse de activar el bit 5 de “ Command Error” y comprobar si su máscara en el Standard Event Enable Register está activada y, si es así, activar el bit 5 del Byte de Estado comprobando a su vez su máscara.

Este registro contiene las siguientes informaciones,

- Operation Complete
- Request Complete
- Query Error
- Device Error
- Execution Error
- Command Error, si llega un comando erróneo se activa
- URQ
- Power On, alimentación encendida o apagada

Para cambiar y conocer las máscaras, se dispone de los comandos *ESE y *ESE?. Para conocer el contenido de todo el Standard Event Status Register, se dispone del comando *ESR?

- b) Questionable Data (16 bits): Es un registro donde se marcan ciertos sucesos propios de ese instrumento. Así por ejemplo, en un multímetro puede ser de interés tener un bit que avise de sobretensiones o sobrecorrientes. Este registro dispone, a su vez, de un registro de máscaras.

Contiene la siguiente información:

- Voltage Overload, caso de sobretensión
- Current Overload, en caso de sobrecorriente
- Ohms Overload
- . Limit Test Fail LO
- Limit Test Fail HI

Para leer el contenido de todo el registro, se puede utilizar este comando combinado:STAT:QUES:EVENT? y para habilitar/deshabilitar así como conocer el contenido de la máscara STAT:QUES:ENAB? y STAT:QUES:ENAB, respectivamente

- c) Status Byte (8 bits): Este registro de estado ya existente en la norma IEEE 488.1, contiene los siguientes bits:

- Questionable Data, si alguno de los bits del registro Questionable Data está activo y habilitado
- Message Available, si hay un mensaje que el instrumento quiere enviar. Tiene una cola de salida (output buffer) donde están los posibles mensajes a enviar
- Standard Event, si alguno de los bits del registro Standard Event Status Register está activo y habilitado (ver gráfico más adelante)
- Request Service (este bit no tiene máscara) el instrumento ha pedido o no una petición al controlador.

Para leer el contenido de este registro, se utiliza el comando: *STB?. Para conocer el registro de máscara se usa *SRE? y para habilitarlo *SRE.

2.4 CARACTERÍSTICAS ELÉCTRICAS DE LA INTERFACE GPIB

El bus de transmisión de datos de GPIB es de 8 bits en paralelo y lógica negativa, con niveles TTL estándar (“cierto” si el voltaje es menor o igual a 0.8 V y “falso” si el voltaje es mayor o igual a 2.0 V).

Las puertas de salida en cada equipo sobre una línea del bus utiliza la tecnología **open collector**. Esto hace que una línea del bus estará en estado "TRUE" lógico (tensión baja) si ese es el valor de salida que corresponde para esa línea en **algún** equipo. Por el contrario, una línea del bus estará en estado "FALSE" lógico solo si en todos los equipos conectados corresponde a esa línea ese valor. Cada línea implementa una función lógica OR (Wire-OR).

Algunas interfases GPIB, utilizan una tecnología tri-state para el control de las líneas del bus, pero siguiendo la misma funcionalidad que con la tecnología "open-collector". Utilizando la tecnología tri-state se puede conseguir mayores velocidades de transferencia de datos.

Dada la tecnología de control de líneas que se utiliza, aunque funcionalmente podrían conectarse a un bus más de 15 equipos (siempre que algunos de ellos sean solo "Listener"), las prestaciones dinámicas del bus pueden deteriorarse con ello.

En la figura 2.7 se muestra una tarjeta GPIB. Los cables y conectores tienen el aspecto típico mostrado en la figura 2.8:



Figura 2.7 Tarjeta GPIB

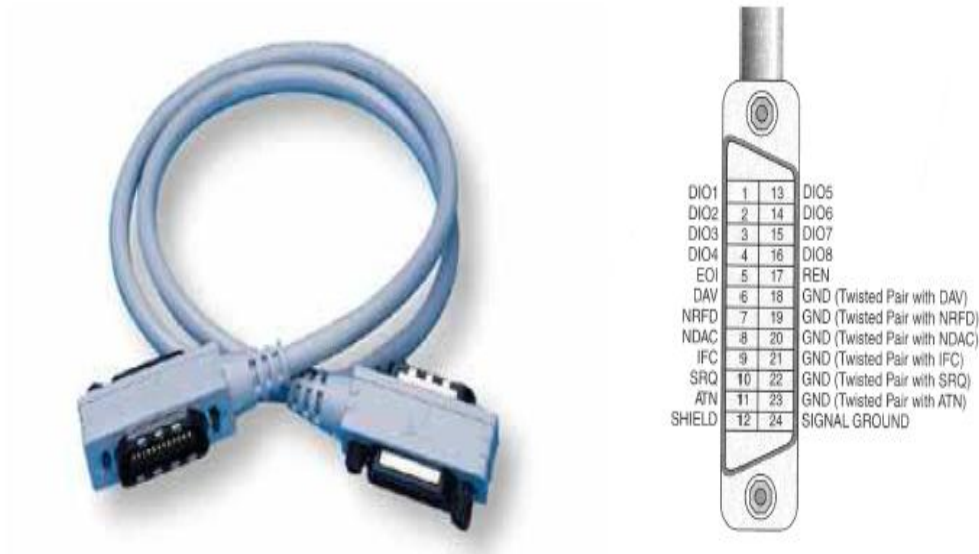


Figura 2.8: Cable de conexión GPIB: aspecto físico y distribución de señales.

El bus consta de 24 pines, repartidos de la siguiente forma:

- 8 líneas de transmisión de datos (DIO1-DIO8)

- 3 líneas para el control asíncrono de la comunicación (NRFD, NDAC y NRDAV). Mediante estas líneas se verifica la correcta transmisión de los datos, que es una de las fortalezas del GPIB.
- 5 líneas que gestionan la transmisión de comandos (ATN, IFC, REN, SRQ y EOI).
- 7 líneas de masa y 1 de apantallamiento.

2.5 REGLAS DE CONSTRUCCIÓN

Para que el bus GPIB alcance la velocidad de transmisión para el que fue diseñado (hasta 8 Mbytes/s), deben cumplirse los siguientes **requisitos**:

- Puede haber un máximo de 15 dispositivos conectados al bus, y al menos dos tercios de ellos deben estar encendidos.
- Los instrumentos están interconectados mediante un cable en estrella o línea y la longitud del cable está limitada a 20m, entre los 15 instrumentos, siendo la distancia máxima entre ellos de 2m.
- Hay 16 líneas de señales, para datos y 8 para control
- La transferencia de mensajes es asíncrona, controlada mediante ciertas líneas de handshaking.
- La velocidad máxima de transferencia es de 1Mbyte/ seg para distancias muy cortas.
- Se pueden direccionar hasta 31 direcciones primarias, aunque solo se puedan conectar 15 instrumentos. Además de las 31 primarias hay 31 secundarias lo que establece un total de 961 posibles direcciones.
- La separación máxima entre dos dispositivos es 4 m, y la separación promedio en toda la red debe ser menor de 2 m.
- La longitud total de la red no debe exceder los 20 m.

2.6 OTROS ESTÁNDARES

2.6.1 SCPI (Standard Commands for Programmable Instruments)

Una de las cosas que no define el estándar IEEE 488.1 es la estructura y sintaxis del lenguaje que se usa para programar los dispositivos, lo que provoca que p. ej. un programa elaborado para un osciloscopio no funcione con un osciloscopio de otra marca. SCPI viene a solventar esa dificultad idiomática entre instrumentos definiendo una estructura y sintaxis común del lenguaje.

2.6.2 HS488

Es una propuesta de la compañía National Instruments para la mejora de las velocidades de transmisión de la norma 488. Entre dos dispositivos compatibles HS488 separados por 2 metros de cable se pueden alcanzar de hasta 8 MB/s. En un sistema con 15 dispositivos y 15 metros de cable la velocidad puede alcanzar 1'5 MB/s

CAPÍTULO III

HARDWARE Y SOFTWARE GPIB

En el siguiente capítulo se realiza un análisis del hardware y software disponible para la comunicación con instrumentos vía GPIB. Se indica los pasos para la instalación de las librerías VISA / SIDL y se realiza una breve descripción de los asistentes de estas librerías que permitirán la configuración de la interfaz GPIB. Se muestra además los comandos básicos y especiales de la interfaz GPIB. Por último se realiza una descripción del ambiente gráfico para programación de instrumentos utilizados en este proyecto.

3.1 CONFIGURACIÓN DEL HARDWARE

Un sistema típico constará de un ordenador con una tarjeta controladora GPIB, más los instrumentos (compatibles con IEEE 488). Existen tarjetas GPIB para prácticamente todos los ordenadores presentes en el mercado (PC, Macintosh, estaciones Sun, Silicon Graphics, DEC Alpha, HP RS/6000, etc). En el caso concreto del PC, las controladoras GPIB pueden conectarse al bus ISA, PCI, PCMCIA (portátiles), USB, Ethernet, Firewire, y los puertos serie y paralelo. Existen así mismo adaptadores para los estándares de comunicación RS-232 y RS-485. La figura 3.1 muestra una tarjeta GPIB:



Figura 3.1 Tarjeta GPIB

En la mayoría de los sistemas de instrumentación o industriales, un PC con el sistema operativo DOS es suficiente para controlar el bus GPIB. La tarjeta GPIB se instala en un slot ISA o PCI libre. Si tanto la tarjeta como el sistema operativo son *plug and play*, en principio será reconocida y configurada automáticamente, en caso contrario (por ejemplo, bajo DOS), se debe elegir de forma manual:

- El puerto de entrada/salida, entre los valores 100h y 3F8h.
- El canal DMA, si se va a utilizar.
- La línea de petición de interrupción (IRQ), entre 2 y 7, también si se quiere utilizarla.

Estos parámetros deben tomarse de forma que no produzcan conflictos con otras tarjetas instaladas en el PC.

La interfaz utilizada en este proyecto es la tarjeta GPIB 82350A, cuyas características, instalación y programación se describe a continuación

3.1.1 Tarjeta GPIB 82350A

La tarjeta 82350 A de Agilent Technologies, es una tarjeta económica, IEEE-488 de alto rendimiento, interfase y software para PCI basadas en PCs. LA tarjeta 82350A permite un fácil acceso y control de instrumentos así como el intercambio de datos. Es una tarjeta PCI de alta velocidad con buffer incorporado, la cual desacopla la transferencia a través del bus PCI, proporciona I/O y el desempeño de sistemas que es superior al acceso directo de memoria (DMA) a 750KB/s

La interfaz GPIB 82350 está definida como un sistema en el cual los instrumentos GPIB están conectados vía cables GPIB a la tarjeta de interfase GPIB 82350 PCI. Esta viene con software SICL y VISA para Windows 95/98/NT/2000/Me/XP. El hardware y software es configurable y compatible con plug-and-play para fácil instalación del hardware. La figura 3.2 muestra un sistema típico.

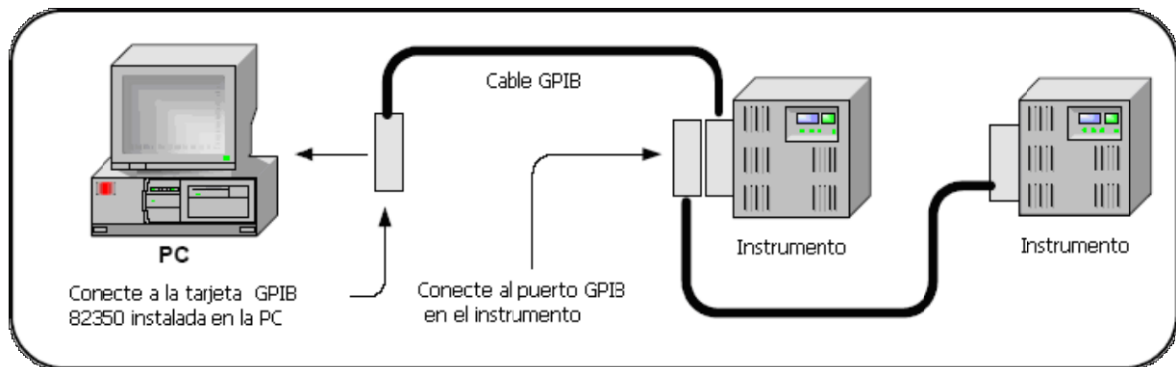


Figura 3.2 Sistema típico con GPIB

Especificaciones de la tarjeta GPIB 82350 A para PCI

- Sistema operativo: Windows 95/98/NT/2000/Me/XP
- Controlador: PC
- Librerías I/O: SICL/ VISA
- Backplane: PCI
- Max. Velocidad I/O: 750KB/s
- Buffering: Interno
- Lenguajes: Agilent VEE, CC++, etc.
- Estándares que soporta: IEEE-488.1 y IEEE-488.2
- Requerimientos mínimos del sistema: 32 MB RAM, 50 MB de espacio libre de disco

La figura 3.3 muestra una secuencia de pasos para instalar y configurar la tarjeta 82350 y para comunicarse entre una PC y los instrumentos vía la 82350.

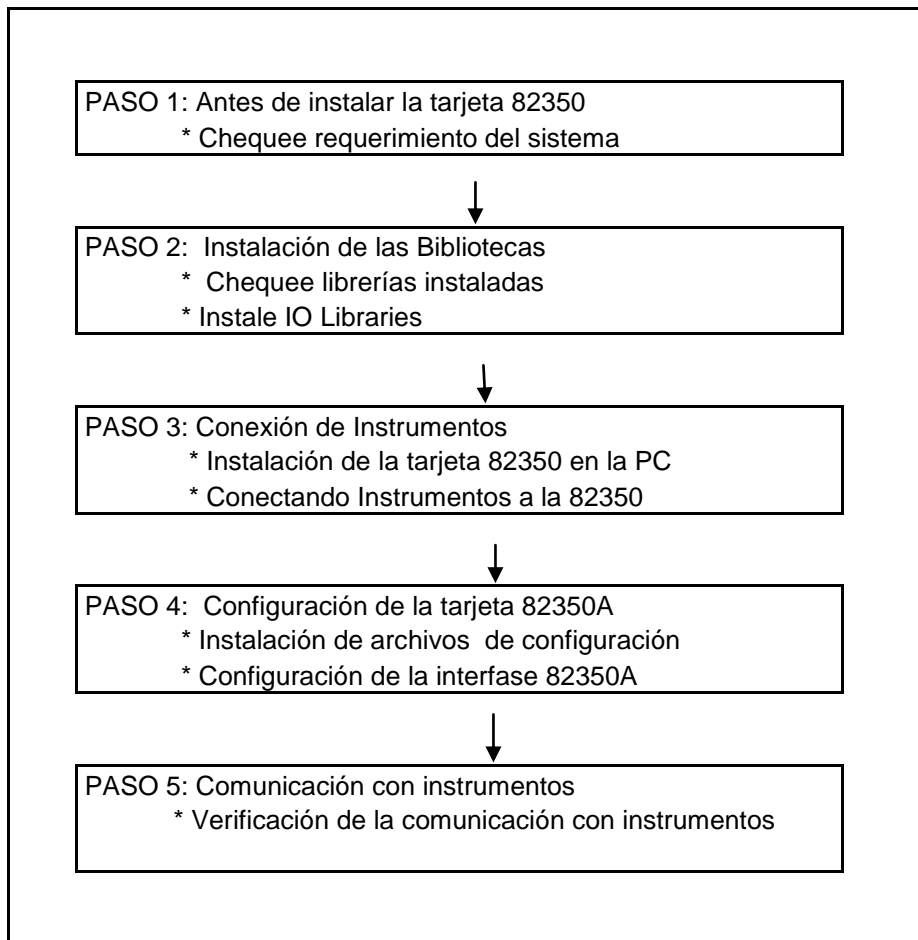


Figura 3.3 Secuencia para instalar y configurar un tarjeta GIPB

El software Agilent IO Libraries consiste de dos librerías y cuatro utilitarios de configuración IO más una librería de control IO, tal como se indica en la tabla 3.1

Tabla 3.1 Agilent IO Libraries

IO Libraries	Descripción
Agilent VISA	Agilent Virtual Instrument Software Architecture: Es una librería que puede ser usada para desarrollar aplicaciones I/O e instrumentos drivers que cumplen con los estándares VXI plug & play.
Agilent SICL	Agilent Standard Instrument Control Library: Es una librería I/O desarrollada por Agilent que es útil para muchas interfaces IO
IO Config	El utilitario I/O Config es usado por Agilent IO Libraries para configurar la interfase de un instrumento IO. Una interfase debe ser configurada con IO Config antes que pueda ser usada con Agilent IO Libraries
VISA Assistant	Es un programa de aplicación que puede ser usado para el control y comunicación con VXI, GPIB, instrumentos serial
Vi Find32	Es un utilitario que lista en una consola de la ventana, todos los recursos VISA encontrados
VXI Resource Manager	Si su sistema incluye E8491 IEEE-1394 PC enlazada a una interfase VXI, se puede examinar la salida del recurso VXI para determinar si el sistema está apropiadamente configurado
Agilent IO Libraries Control	(Icono azul en la barra de tarea de Windows): pulsando sobre este icono se visualiza: <ul style="list-style-type: none"> • Utilitario para ejecutar VISA Assistant IO configuration • Utilitario para ejecutar el ViFind32 debug • Utilitario para ejecutar el IO Config • Documentación y ayuda en línea • Ejecutar Windows Event Viewer • Selección de Agilent VISA Options • Ocultar o salir del icono IO • Ver Version de Agilent IO Libraries

3.1.2 Interfaces soportadas por Agilent IO Libraries

La siguiente tabla muestra las versiones de Agilent IO Libraries recomendadas para cada interfase y el sistema operativo usado para aplicaciones de 32 bits.

Tabla 3.2 Interfaces soportadas por IO Libraries

Interface		Windows Operating Systems							
Type	Product	XP	2000	NT 4.0	NT 3.51	Me	98SE	95	3.1
USB	USB Instruments*	M.01.01 Single or Dual CPU	M.01.01 Single or Dual CPU	No	No	M.01.01	M.01.01	No	No
	82357A GPIB	M.01.01 Single or Dual CPU	M.01.01 Single or Dual CPU	No	No	M.01.01	M.01.01	No	No
PCI	82350B 5V PCI GPIB	M.01.01 Single or Dual CPU	M.01.01 Single or Dual CPU	M.01.01 Single or Dual CPU	No	M.01.01	M.01.01	No	No
	82350A GPIB	M.01.01 Single or Dual CPU	M.01.01 Single or Dual CPU	M.01.01 Single or Dual CPU	No	M.01.01	M.01.01	K.01.00	No
LAN	E5810A GPIB RS-232	M.01.01 Single or Dual CPU	M.01.01 Single or Dual CPU	M.01.01 Single or Dual CPU	No	M.01.01	M.01.01	No	No
	E2050A/B GPIB	M.01.01 Single or Dual CPU	M.01.01 Single or Dual CPU	M.01.01 Single or Dual CPU	G.02.02	M.01.01	M.01.01	K.01.00	No

3.1.3 Instalación de IO Libraries

Existen dos tipos de instalación de Agilent IO libraries: **Runtime Installation** y **Full Installation**.

Una “Full Installation ” permite ejecutar programas que usan Agilent IO Libraries para comunicarse con los instrumentos y también para desarrollar nuevos programas IO. “Runtime Installation” permite correr programas existentes soportados por la interfase, pero no puede desarrollar nuevos programas que corran con Agilent IO Libraries.

“Full Installation” instala todos los componentes de la biblioteca con las opciones recomendadas para el computador y útil a la mayoría de aplicaciones. Para el presente proyecto se seleccionó “Full Installation ” por las características anotadas

Si IO Libraries ya está instalada, se puede pulsar el icono azul IO sobre la barra de tareas (Figura 3.4) y ver información acerca de Agilent IO Libraries Control

para determinar si un paquete full o Runtime esta instalado como se indica en la figura 3.5.



Figura 3.4 Icono de IO sobre la barra de tareas

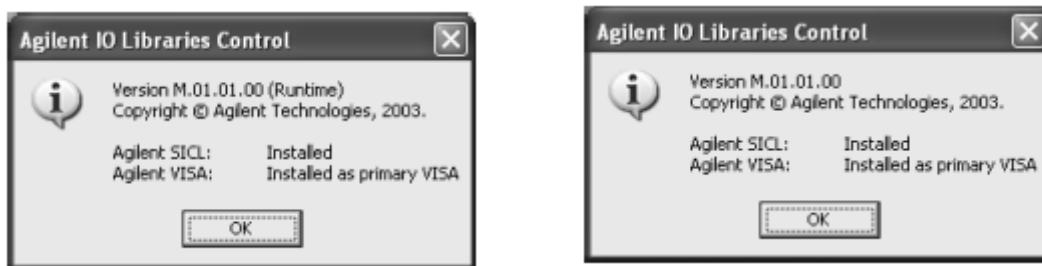


Figura 3.5 Chequeo de librerías instaladas

Instale el CD llamado I/O Libraries for Instrument Control. Al terminar la instalación de este, se detectara la tarjeta GPIB y entonces se podrá comenzar a usar HP VEE.

3.1.4 Grupos de programas Creados

Durante la instalación un grupo de programas de Agilent IO Libraries es creado como se muestra en la figura 3.6

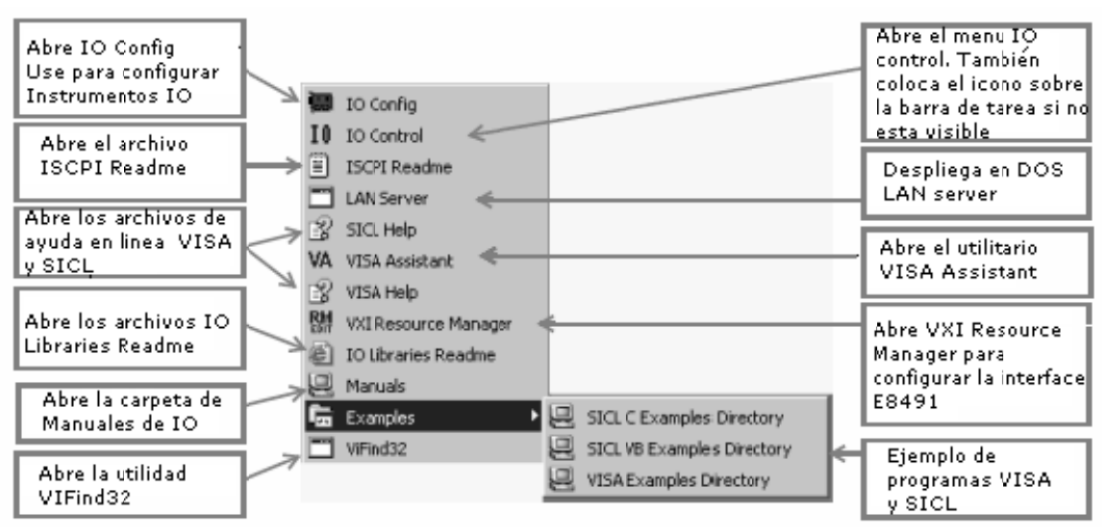


Figura 3.6 Programas de Agilent IO Libraries

3.1.5 Instalación de la Tarjeta 82350 A

Para la instalación de la tarjeta 82350 A, se realiza lo siguiente:

- Se desconecta la fuente de poder y de todos los periféricos y se abre y quita la tapa de la PC para permitir el acceso a los slots I/O.
- Se retira del panel posterior una de las cubiertas. Se escoge un slot PCI que dará un adecuado espacio para el conector de la tarjeta HP-IB
- Se inserta la tarjeta de interfaz 82350 en el slot de expansión de la PC.
- Se conecta un cable GPIB a la interfase usando uno de los siguientes cables:
 - HP 10833 A (1metro)
 - HP 10833B (2 metros)
 - HP 10833C (4 metros)
 - HP 10833D (0.5 metros)
 - HP 8120 – 3448 (6 metros)
 - HP 8120- 3449 (8 metros)

3.1.6 Configuración de interfase

VISA Assistant: VISA Assistant es un programa de aplicación que usa IO Libraries para comunicarse con controladores VXI, GPIB e Instrumentos serial. La pantalla de VISA Assistant y sus interfases se indican en la figura 3.7.

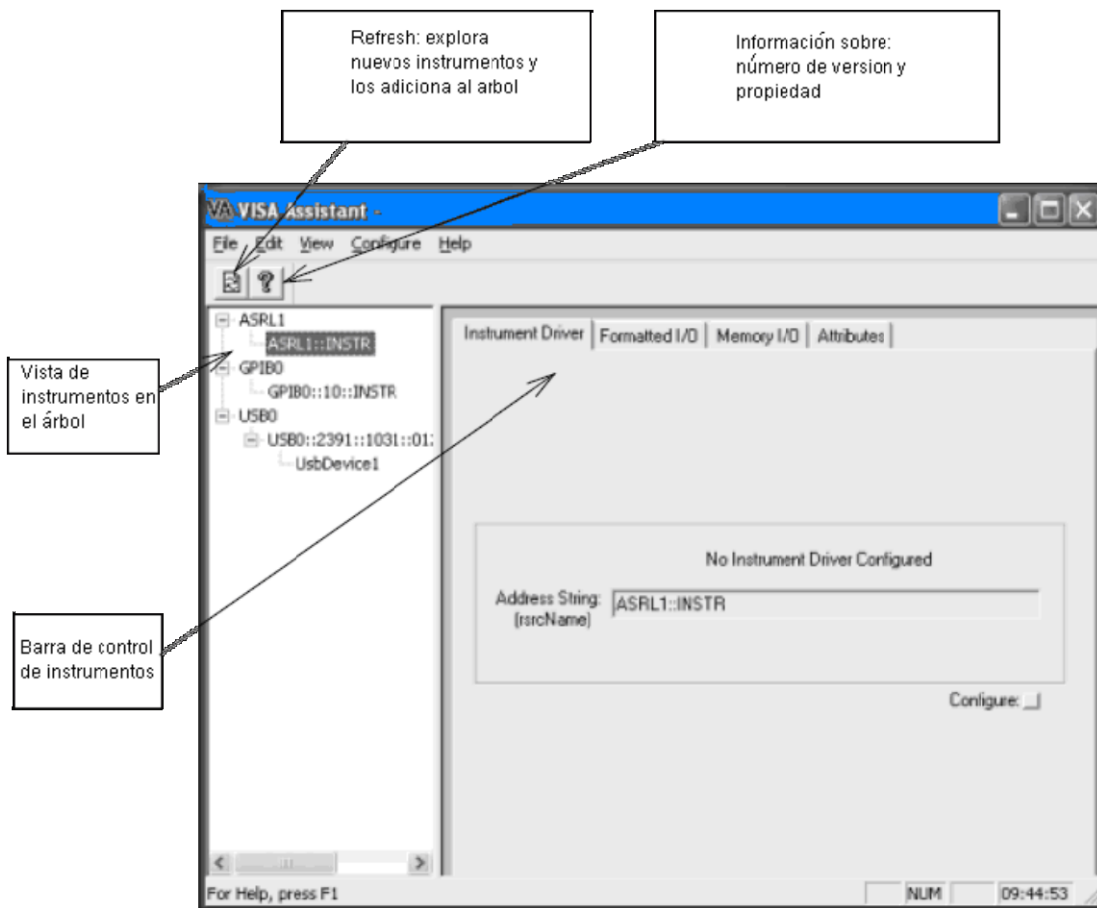


Figura 3.7 VISA Assistant

VISA Assistant puede automáticamente detectar y asignar los drivers de instrumentos VXI plug&play a instrumentos. VISA Assistant también puede usarse para enviar y recibir strings de instrumentos los cuales soportan el formato IO. Para otros instrumentos, VISA Assistant permite leer y escribir áreas de memoria. VISA Assistant también describe atributos que están asociados con un instrumento

IO Config: El propósito de IO Config es configurar una interfase IO para usar con VISA o SICL. IO Config hace que se asocie a un único nombre VISA y o SICL con una interfase de hardware. La figura 3.8 indica la pantalla de IO Config.

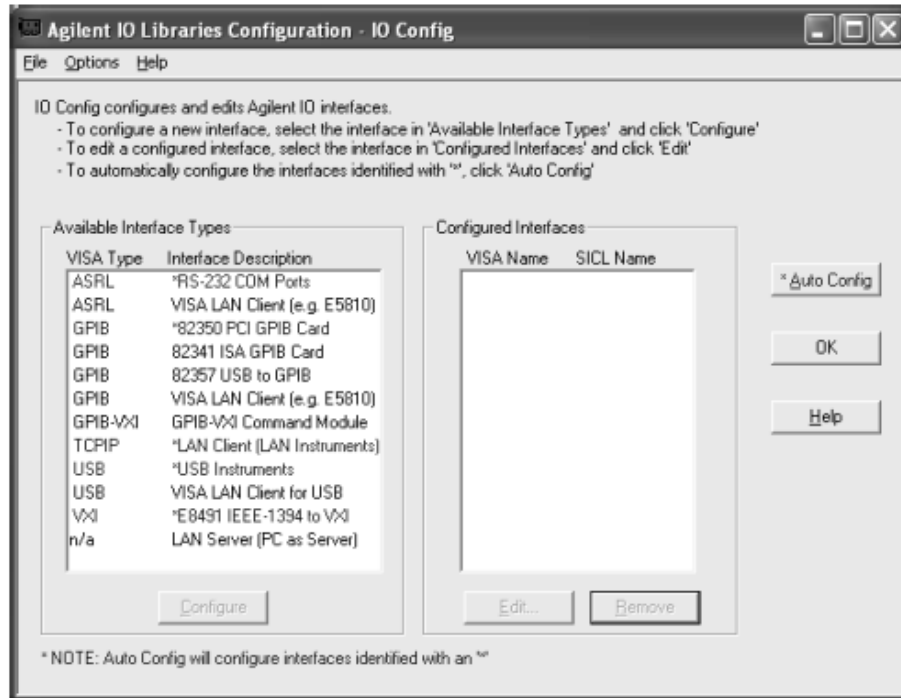


Figura 3.8 Ventana principal de IO Config

Después que la interfase ha sido configurada con IO se puede usar el utilitario VISA Assistant para establecer la comunicación entre la PC y los instrumentos GPIB como se indica en la figura 3.9.

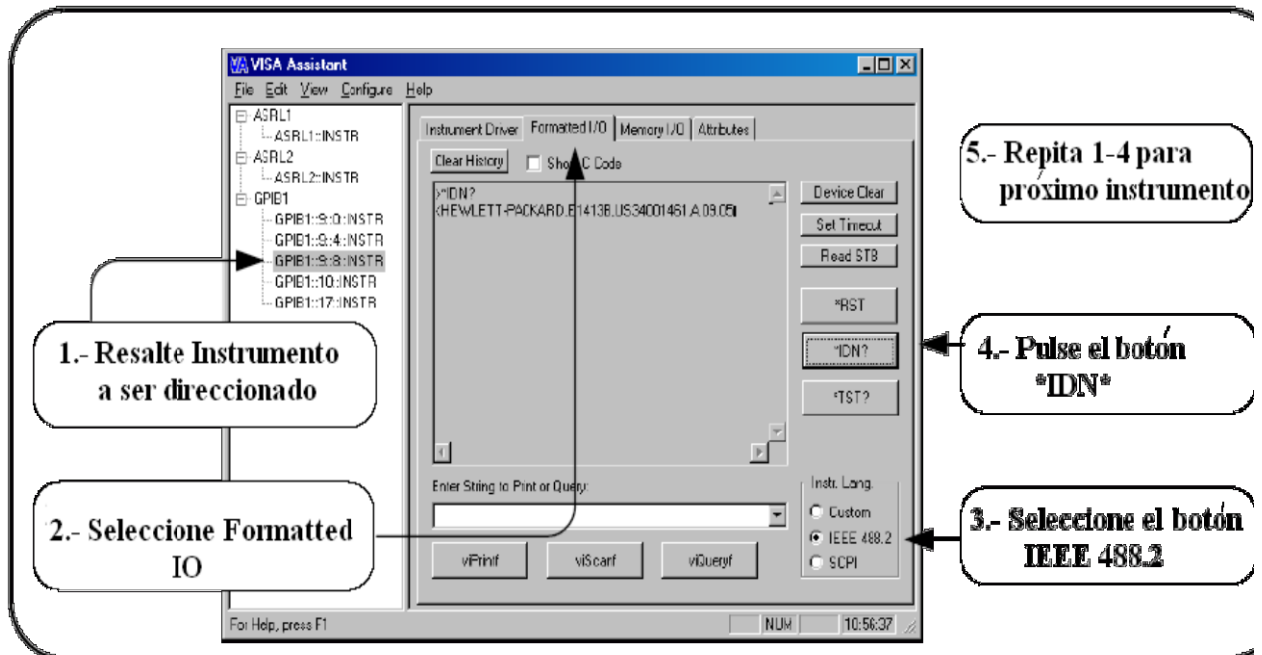


Figura 3.9 Típica ventana de VISA Assistant

Se utiliza IO Config para configurar una nueva interfase. La figura 3.10 indica la caja de diálogo de IO Config.

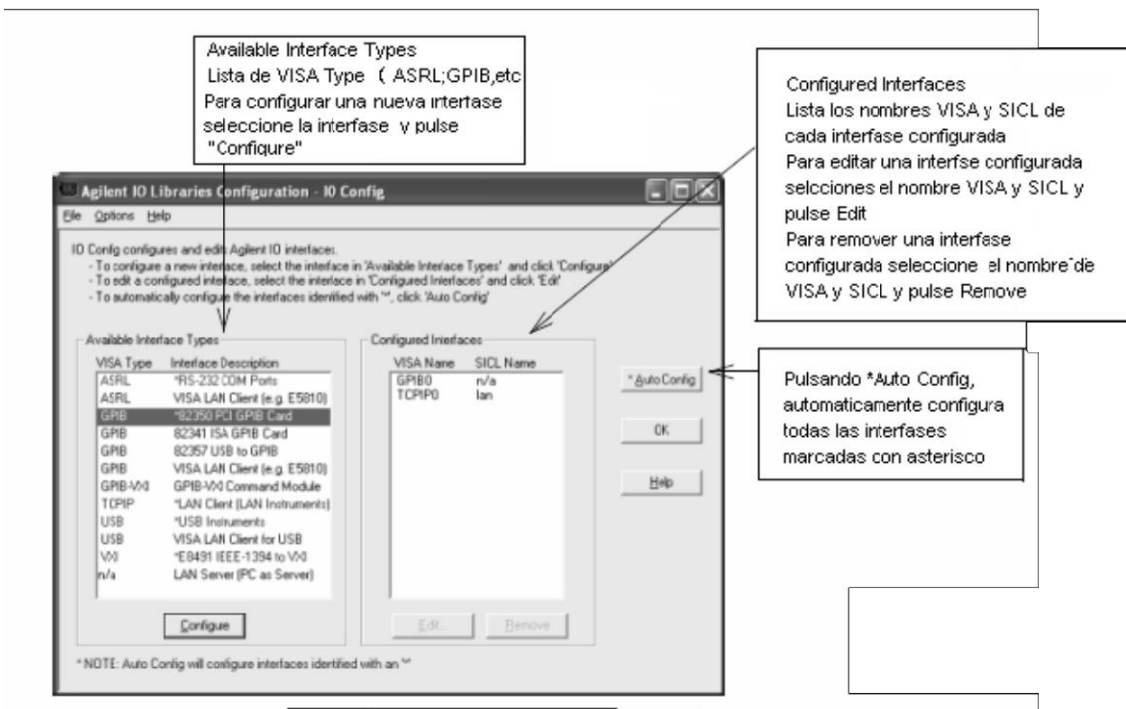


Figura 3.10 Caja de diálogo de IO config para seleccionar la interfase

Por ejemplo si en la figura anterior se escogió la tarjeta GPIB 82350 PCI. En la figura 3.11 se indica la configuración de la tarjeta que aparece después de pulsar el botón "configure".

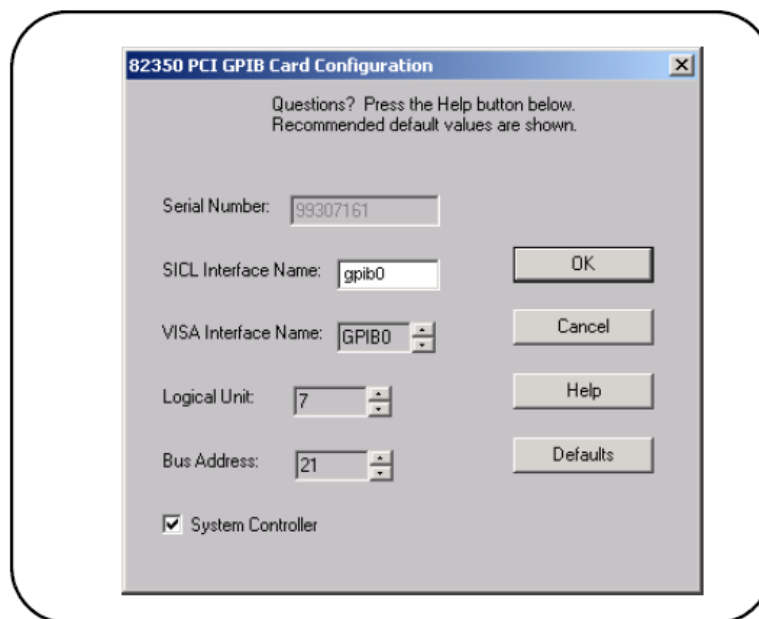


Figura 3.11 Configuración de la tarjeta 82350

Cuando la configuración de la tarjeta GPIB 82350 PCI aparece en la pantalla, se debe fijar el nombre de la interfase SICL, la unidad lógica, el nombre de la interfase VISA y la dirección del bus.

Tabla 3.3 Parámetros para configuración de la tarjeta GPIB 82350 A

Parámetros para configuración de la interfase 82350	
SICL Interfase Name	Nombre simbólico que SICL usa para una única identificación de al interfase GPIB. Por defecto, el nombre de la interfase es “gpib0”. El nombre SICL de la interfaz debe ser una única cadena de caracteres alfanuméricos, empezando con una letra
VISA Interfase Name	Nombre simbólico que VISA usa para una única identificación de la interfase GPIB. Por defecto el nombre de la interfaz es GPIB0. El nombre de la interfase 82350 para VISA deber empezar con una cadena GPIB y tener un entero junto a ella, tal como GPIB0, GPIB1, GPIB2, etc
Logical Unit	Número que SICL usa para una única identificación de la interfaz 82350. El número de la unidad lógica es un número en el rango entre 0 - 10000
Buss Address	Dirección de la interfase controladora sobre el bus GPIB. Usualmente es 21 si la interfase GPIB es un sistema controlador o 20 si la interfase GPIB no es un sistema controlador. Estas direcciones son escogidas por convención, pero cualquier dirección en el rango de 0 a 30 puede ser usada
System Controller	Controla cual dispositivo sobre el bus habla y cual escucha. Si algún dispositivo existe sobre el bus, cada uno tiene una única dirección GPIB y solo un dispositivo en el sistema es el controlador (usualmente el dispositivo instalado en la computadora). Cada interfase GPIB tiene su propio bus, así cada sistema puede ser el controlador del sistema mientras no este encadenado junto a otras interfaces GPIB. Sin embargo dos o más sistemas controladores en el mismo bus causarán que el bus sea inoperante.

Una interfase IO puede ser definida como una interfaz de hardware y una interfaz de software. El propósito del utilitario IO Config es asociar un único nombre de la interfaz con la interfaz hardware. Por ejemplo el sistema de la figura 3.12 consta de dos PC con dos tarjetas GPIB 82350 conectadas a tres instrumentos vía cables GPIB. Para este sistema, IO Config fue usado para asignar a la tarjeta #1 un nombre VISA de GPIB0 y un nombre de SICL de gpib0

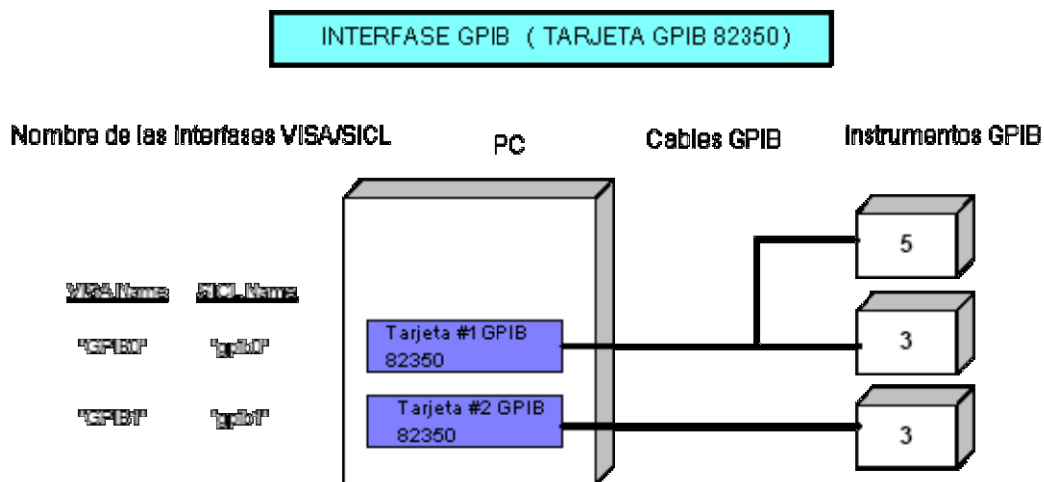


Figura 3.12 Sistema con tarjetas GPIB

IO config también fue usado para asignar a la tarjeta #2 un nombre VISA de GPIB1 y un nombre SICL de gpib1. Con estos nombres asignados a las interfases, las direcciones VISA/SICL son como se muestran en la figura 3.12

Este único nombre ha sido asignado por IO Config, se puede usar VISA vi open command para abrir la ruta IO de los instrumentos GPIB como se muestra en la figura 3.13 , o se puede usar el comando SICL iopen para abrir la ruta IO del instrumento

VISA: viOpen (... "GPIB0::5::INSTR"...))	Open IO path to GPIB instrument at address 5 using 82350 Card #1
viOpen (... "GPIB0::3::INSTR"...))	Open IO path to GPIB instrument at address 3 using 82350 Card #1
viOpen (... "GPIB1::3::INSTR"...))	Open IO path to GPIB instrument at address 3 using 82350 Card #2
SICL: iopen ("gpib0,5")	Open IO path to GPIB instrument at address 5 using 82350 Card #1
iopen ("gpib0,3")	Open IO path to GPIB instrument at address 3 using 82350 Card #1
iopen ("gpib1,3")	Open IO path to GPIB instrument at address 3 using 82350 Card #2

Figura 3.13 direcciones VISA / SICL

Una interfase GPIB tiene uno o más instrumentos GPIB conectados a la PC. Hay tres tipos de interfaz GPIB: USB, ISA y PCI

- Interfaz USB. La comunicación IO es por el puerto USB instalado en la PC, un cable USB y una interfaz USB/GPIB (tal como el pórtico USB/GPIB 82357)
- Interfaz PCI: La comunicación es vía una tarjeta PCI GPIB instalada en la PC y cables GPIB.
- Interfaz ISA: La comunicación es vía una tarjeta ISA GPIB (tal como una 82341) instalada en la PC y cables GPIB.

La figura 3.14 muestra una típica interfase GPIB con instrumentos GPIB que tienen las direcciones 3 y 5

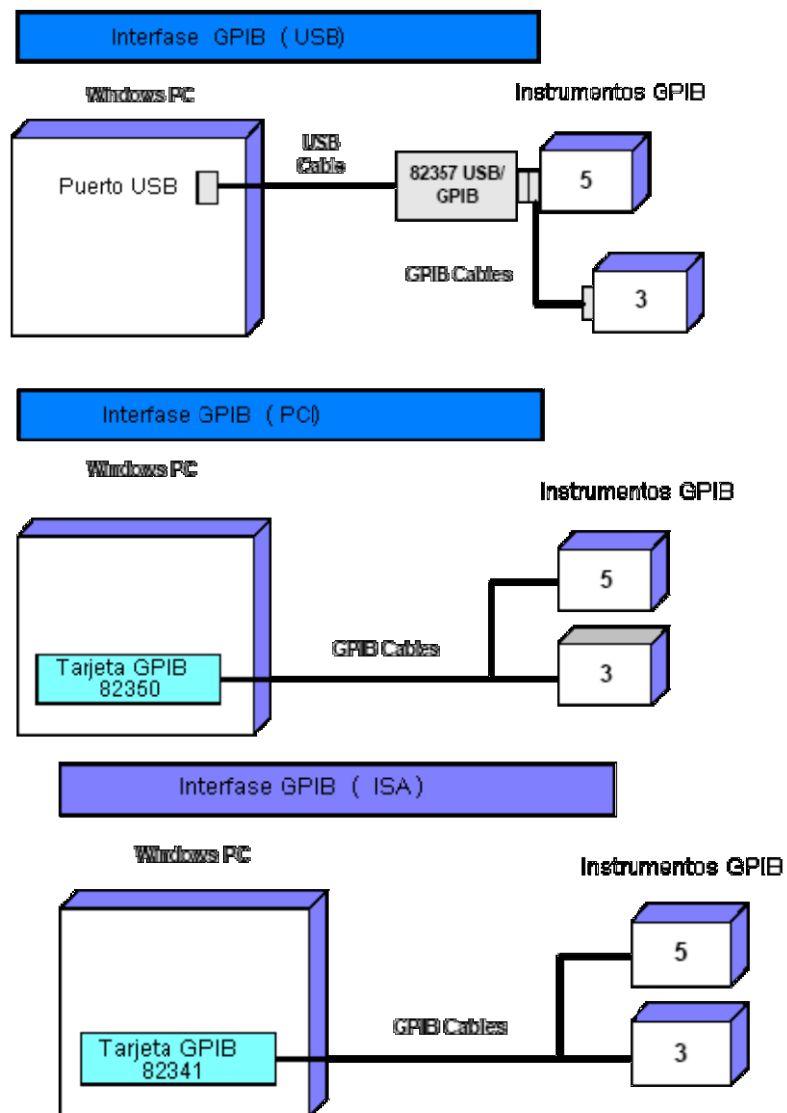


Figura 3.14 Sistema típico con interfases GPIB

3.2 COMANDOS DE PROGRAMACIÓN GPIB

3.2.1 Sintaxis de las órdenes de medida

Para poder programar una función de medida en un instrumento se debe elegir una función que permita enviarle una orden y saber que orden activa cada posible medida que el instrumento es capaz de realizar. Desde los inicios de los instrumentos basados en GPIB, ha habido un cierto desorden en cuanto a la sintaxis y al contenido de las órdenes. Instrumentos de funcionalidad similar de distinto fabricante, o incluso del mismo fabricante, usaban conjuntos de ordenes totalmente distintos, lo cual suponía un gran quebradero de cabeza para el programador. La norma IEEE-488.2-1987 primero y el consorcio SCPI (Standard Commands for Programmable Instrumentation) después racionalizan un poco este panorama.

IEEE-488.2 define la **sintaxis** que deberán usar los comandos de programación, pero sin definir su contenido. SCPI adopta la sintaxis de IEEE-488 y define los contenidos a través de una serie de modelos genéricos de instrumento. De forma muy sucinta, la sintaxis definida en IEEE-488.2 puede explicarse de la siguiente forma (ver figura 3.15)

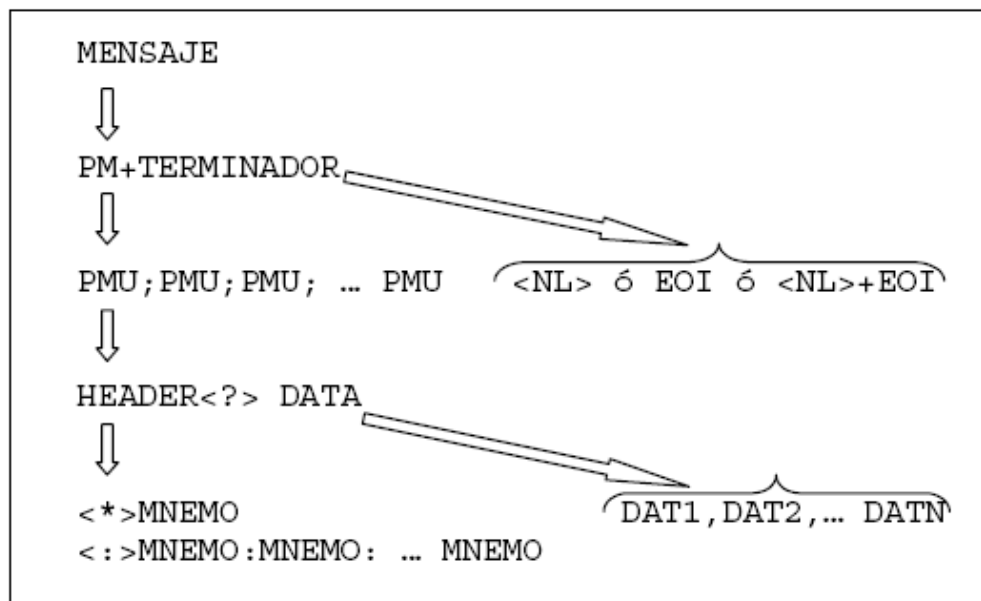


Figura 3.15 Representación esquemática y simplificada de la sintaxis definida en la norma IEEE-488.2-1987

Un mensaje está constituido por una mensaje de programación (Program Message: PM) y un terminador. Un mensaje de programación está constituido por un conjunto de unidades de programación (Program Message Units: PMU) , separadas por un punto y coma. Un terminador puede ser la activación de la línea EOI, un carácter <LF> o ambos simultáneamente.

Una unidad de programación consta de una cabecera (Header) terminada con un carácter de interrogación, si se espera respuesta y, opcionalmente, unos datos, separados de la cabecera por un espacio en blanco. Los datos pueden ser numéricos o alfanuméricos y van separados por comas.

Una cabecera es un conjunto de Mnemotécnicos separados por un carácter ":". Opcionalmente un mnemotécnico puede llevar un carácter "*" al inicio, para indicar que es una orden básica definida en la norma IEEE-488.2.

Así por ejemplo, la orden *IDN? se interpreta dentro de este esquema como: es un PM que contiene una única PMU. No hay datos. El interrogante significa que se espera respuesta del instrumento y el asterisco que se trata de una orden básica obligatoria definida en la norma.

Este es el estado de las cosas después de IEEE-488.2, es decir, los mnemotécnicos que programan una función de medida específica son criterio del fabricante del instrumento.

3.2.2 Envío de la información

Para enviar estos comandos a los instrumentos se puede usar las funciones de la librería GPIB-488.2 o bien de la librería VISA. Por ejemplo, las funciones WRITE de VISA y la función SEND de la librería GPIB 488.2

Aunque parezcan equivalentes, estas funciones no son intercambiables en todos los casos. La función WRITE de VISA asume que se está enviando información a un dispositivo que se comunica por mensajes, esto es, compatible con la sintaxis

de IEEE-488.2. Aunque parezca contradictorio, la función SEND de la librería GPIB 488.2 no hace esta asunción.

La diferencia estriba en que en la función SEND se puede especificar el terminador del mensaje mediante la entrada MODE (0: sin terminador, 1: <NL> + EOI, 2: EOI), mientras que en VISA WRITE no se puede y se enviará uno compatible con IEEE-488.2

Si el instrumento que se tiene es compatible IEEE-488.2 no habrá problema, pero si no lo es se debe averiguar que tipo de terminador requiere y suministrárselo, de lo contrario no entenderá que se ha acabado la comunicación.

3.2.3 Peticiones de servicio

El esquema de funcionamiento secuencial de la relación entre el controlador de un sistema de instrumentación y los instrumentos, en el que el controlador envía una orden de medida y espera la respuesta, es muy simple pero no siempre es el más conveniente.

Algunos instrumentos pueden tener tiempos de medida largos, o puede ser posible programar una secuencia de medidas temporizada de forma que el controlador no deba preocuparse del instrumento hasta que acabe o hasta que la medida este disponible. En estos casos es útil disponer de un mecanismo que permita a los instrumentos notificar al controlador que se ha llegado a un determinado estado.

En un entorno IEEE-488, las interrupciones que pueden generar los instrumentos se llaman peticiones de servicio (Service Request) y hay una línea física en el bus dedicada a este menester (SRQ). El funcionamiento es muy similar al esquema de peticiones de interrupción en un entorno de microprocesador. En primer lugar hay que configurar al instrumento para que pueda activar SRQ cuando un determinado evento ocurra. Esto se hace programando los registros de estado del instrumento.

En segundo lugar hay que definir en el controlador un mecanismo de respuesta en caso de petición de servicio. Genéricamente se puede hablar de dos categorías de mecanismos de respuesta: o bien se guarda la información de que ha llegado una

petición de servicio en una cola que se va consultando periódicamente o bien se activa la ejecución de una rutina al recibir la interrupción. Dentro de VISA, a estos mecanismos se les conoce como Queuing y CallBack.

La primera cosa que se debe hacer al recibir una petición de servicio es identificar al instrumento causante de la misma. La norma IEEE-488.1 establece dos mecanismos llamados Serial Poll y Parallel Poll. En el primero el controlador va preguntando secuencialmente a todos los instrumentos si han generado una interrupción hasta encontrar uno que responda afirmativamente. En un Parallel Poll se hace la pregunta a todos y contestan todos a la vez, usando cada uno una línea de datos. Obviamente no puede haber más de 8 instrumentos en este último caso. Una vez determinado el causante de la petición se puede determinar el motivo de la misma leyendo sus registros de estado.

La norma IEEE-488.2 establece un modelo de registros de estado. El registro de estado (Status Byte Register) indica si el instrumento ha generado la petición (bit 6 a "1") y el motivo. Solo hay dos bits asignados por la norma, el bit 5 asociado al Registro de Estado de Sucesos Normales (Standard Event Status Register) y el bit 4 asociado a la cola de mensajes de salida.

El fabricante puede definir la funcionalidad de los otros bits. Para que la puesta a uno de estos bits genere una petición de servicio hay que habilitar la máscara del registro de estado (Service Request Enable Register).

El registro de estado de sucesos normales tiene 16 bits. 8 de ellos están definidos en la norma. Para que alguno de estos sucesos pueda activar el bit 5 del Status Byte Register también hay que habilitar una máscara.

a) Peticiones de servicio en GPIB 488.2 y LabVIEW

Para poder ejecutar el código deseado al recibir SRQ se debe, en primer lugar, comprobar de forma periódica el valor de esta línea. Hay dos funciones en la librería GPIB 488.2 que permiten realizar esta tarea: WAITSRQ y TESTSRQ. La primera detiene la ejecución del programa hasta que se activa SRQ o bien se produce un TimeOut. La segunda simplemente

comprueba el valor de la línea y devuelve un valor booleano verdadero o falso.

Como indicación general se puede decir: WAITSRQ puede ser útil si se conoce de antemano el tiempo que tardará el equipo en producir la interrupción. No es aconsejable usar “TimeOuts” (tiempo de espera infinitos) porque el procesador puede quedar bloqueado. TESTSRQ puede usarse en un bucle, infinito, hasta que se active SRQ, con lo cual funcionaría de forma similar a WAITSRQ.

Una vez que se ha detectado la activación de SRQ, se debe identificar al causante de ella. La librería GPIB 488.2 ofrece dos funciones para ello: ALLSPOLL y FINDRQS. La primera realiza un Serial Poll a todos los instrumentos de la lista especificada y devuelve un vector con los Registros de Estado de todos ellos.

Con esta información el usuario puede identificar al (o los) culpable(s) de la petición de servicio. FINDRQS realizará un Serial Poll de los dispositivos en la lista que se le proporcione, devolverá la dirección del primero que encuentre con el bit 6 del Registro de Estado activo y enviará un **SDC** (Selected Device Clear) a todos los dispositivos de la lista, borrando la información de todos los Registros de Estado. Esta función solo debería usarse si se está seguro que un único instrumento ha podido realizar una petición de servicio.

Una vez leído el registro de estado, debe leerse también el Registro de Estado de Sucesos Normales (*ESR?) para conocer (y borrar) la condición que ha provocado la petición de servicio.

b) Peticiones de servicio en VISA

Para gestionar las peticiones de interrupción y otras situaciones anormales en el sistema, en VISA se define el concepto de suceso (event). Un suceso puede ser una petición de servicio GPIB, o una activación de una interrupción VXI o incluso una excepción del procesador.

Existen dos mecanismos de tratamiento de sucesos, que en terminología VISA se conocen como Queuing y Callback. Los funciones que permiten descubrir si se ha producido un evento determinado asociado a estos dos mecanismos se conocen como espera síncrona y espera asíncrona, respectivamente. Es un funcionamiento paralelo al comentado para GPIB 488.2.

Hay dos diferencias fundamentales, a nivel conceptual, entre el tratamiento de peticiones de servicio usando funciones GPIB 488.2 y VISA. En primer lugar, para que el gestor de recursos de VISA detecte algún suceso se debe habilitarlo explícitamente con la función VISA ENABLE EVENT. En segundo lugar, cuando se detecta un evento, se abre una sesión para el tratamiento del mismo, que hay que cerrar una vez se han realizado las operaciones pertinentes. Esta sesión es similar a la sesión que se abre para comunicarse con un recurso del sistema.

Cuando la función finaliza, se pasa a leer los datos obtenidos por el instrumento con VISA READ. También se debe borrar la condición de petición de servicio en el registro de sucesos normales, ya sea leyéndolo o borrando explícitamente su contenido.

3.2.4 Comandos del bus

Existen 5 tipos de información que se pueden transmitir cuando el bus opera en modo de comando (ATN=0).

- *Direcciones de habladores*
- *Direcciones de escuchas*
- *Comandos Universales*
- *Comandos direccionados*
- *Comandos no direccionados*

a) Direcciones de habladores(TAD)

Puede haber en el bus, solamente un dispositivo direccionado como hablador a la vez. Las direcciones de dispositivo están en el rango entre 0 y 30.

Las direcciones de hablador están formadas por la dirección primaria del dispositivo, + 64. Este valor es transmitido activando la señal ATN.

b) Direcciones de escuchas (LAD)

En este sistema pueden haber varios escuchas conectados, y direccionados simultáneamente, que recibirán los datos en paralelo.

Las direcciones de escucha están formadas por la dirección primaria del dispositivo, + 32. Este valor es transmitido activando la señal ATN.

c) Comandos universales

Son recibidos por todos los dispositivos conectados al bus

Tabla 3.4 Comandos universales de la interfaz GPIB

Mnemonic	Comando	Descripción
LLO	Local lockout	Deshabilita el panel frontal del instrumento (REN tiene que ser habilitado antes)
DCL	Universal device clear	Lleva a los instrumentos a un estado conocido
PPU	Parallel poll unconfigure	Deshabilita el polling paralelo
SPE	Serial poll enable	Habilita el modo serial polling
SPD	Serial poll disable	Deshabilita el modo serial polling

d) Comandos direccionados

Son ejecutados solo para los instrumentos que están direccionados como escuchas.

Tabla 3.5 Comandos direccionados de la interfaz GPIB

Mnemonic	Comando	Descripción
SDC	Selected device clear	Lleva a los dispositivos direccionados como escuchas a un estado conocido
GTL	Go to local	Devuelve al instrumento a modo local
GET	Group execute trigger	Indica el comienzo de una actividad para un grupo de dispositivos.
PPC	Parallel poll configure	Habilita el polling paralelo.
TCT	Take control	Transfiere el estado de controlador activo a otro dispositivo

e) Comandos no direccionados

Comandos que pueden ser considerados como extensiones de direcciones de dispositivo escuchas o habladores, debe ser ejecutado por cualquier hablador o escucha aunque no haya sido previamente direccionado. Se incluyen:

UNL (unlisten): Indica a todos los escuchas que deben dejar de escuchar

UNT (untalk): Indica a todos los habladores que deben dejar de ser habladores

3.2.5 Polling

Cuando un dispositivo requiere servicio activa la señal SRQ. El controlador tiene 2 posibles maneras de realizar un polling para ver cual es el que pidió servicio.

Serial polling: Realiza las siguientes operaciones

- Direcciona como habladores a los dispositivos (uno a la vez)
- Controlador envía SPE para activar el modo de polling serial.
- Espera respuesta del dispositivo direccionado
- UNT para deshabilitar a todos como habladores.
- Si el bit 6 del byte de respuesta es 1 entonces: controlador envía SPD para deshabilitar modo de polling serial y atiende al dispositivo; en caso contrario el controlador elige otro dispositivo y realiza nuevamente en mismo procedimiento.

El manual de cada dispositivo tiene información adicional sobre el significado de los otros bytes de la respuesta al polling serial.

Parallel polling El polling paralelo permite chequear simultáneamente el estado de los dispositivos. A cada dispositivo se le asigna un bit de datos, el cual será seteado dentro de la rutina de polling paralelo si se requiere servicio.

Más de un dispositivo puede ser asignado a una línea de datos; luego se realiza un polling serial entre ellos para determinar cual pidió atención.

El polling paralelo comienza cuando se activan ATN y EOI juntas, luego el controlador lee el byte de respuesta y comienza su interpretación.

Algunos dispositivos pueden ser configurados a través del comando PPC para responder en líneas específicas de datos; en otros dispositivos las líneas de respuesta se seleccionan a través de switches o jumpers. Algunos no tienen capacidad para polling paralelo.

Los diagramas de flujo 3.1, 3.2 y 3.3 muestran los ciclos de bus de lectura, escritura y el *Serial polling*

Diagrama de flujo 3.1 : Ciclo de Bus de lectura

CICLOS DE BUS

LECTURA

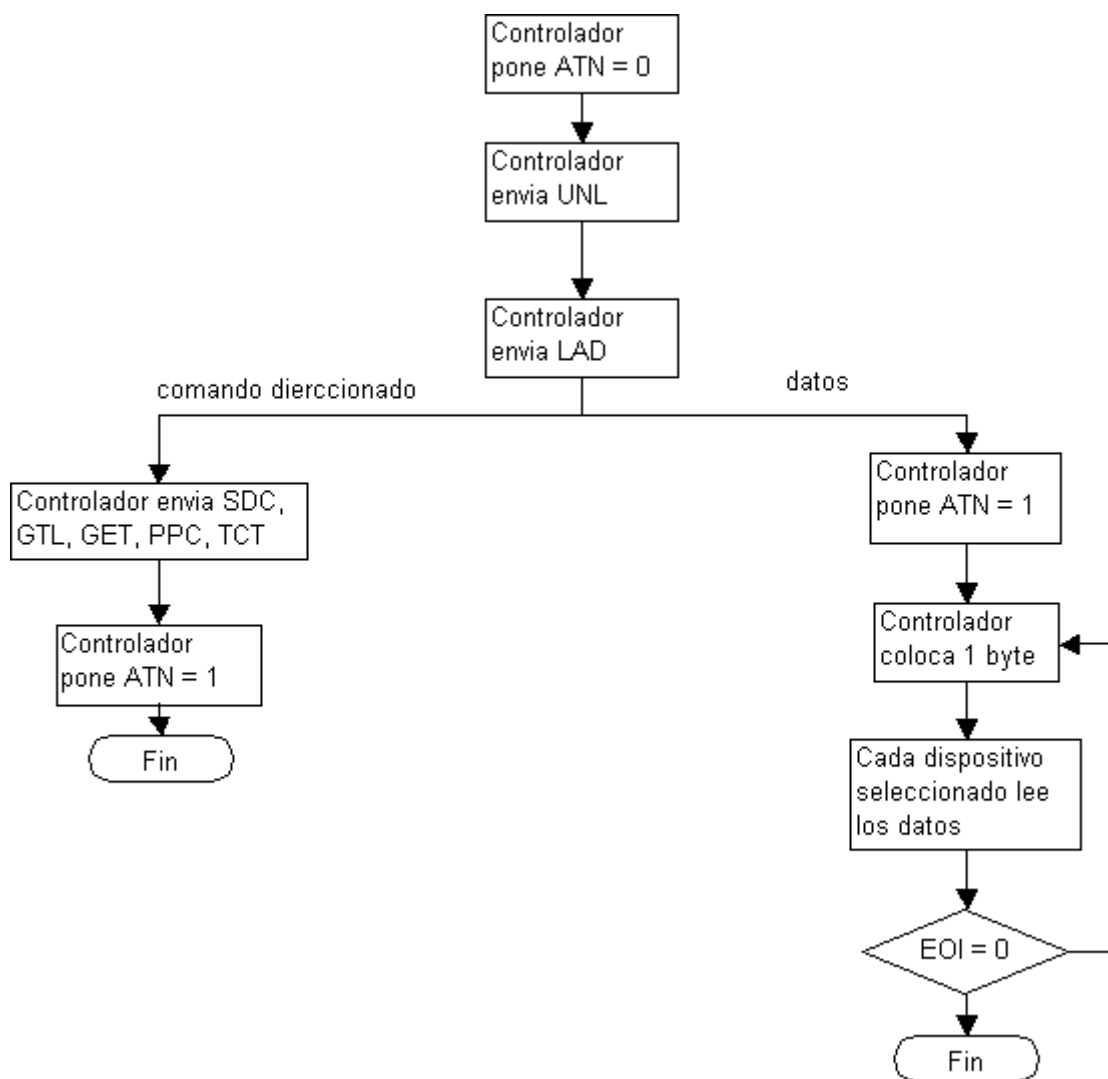


Diagrama de flujo 3.2 : Ciclo de Bus de escritura

ESCRITURA

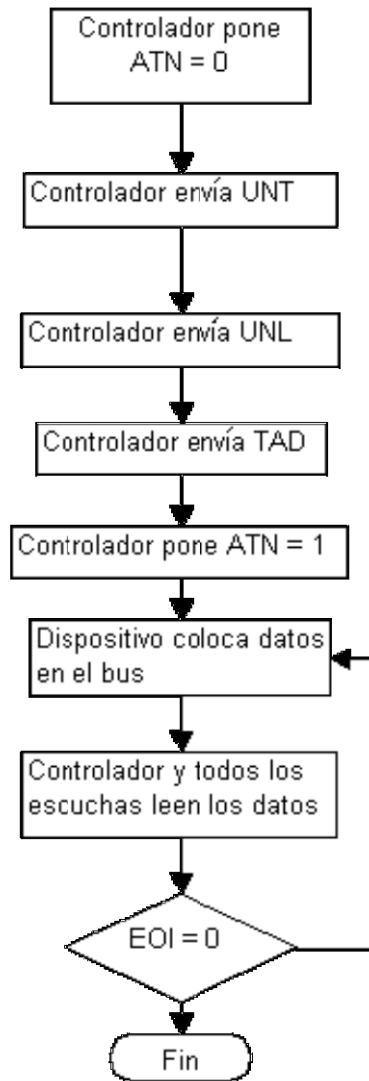
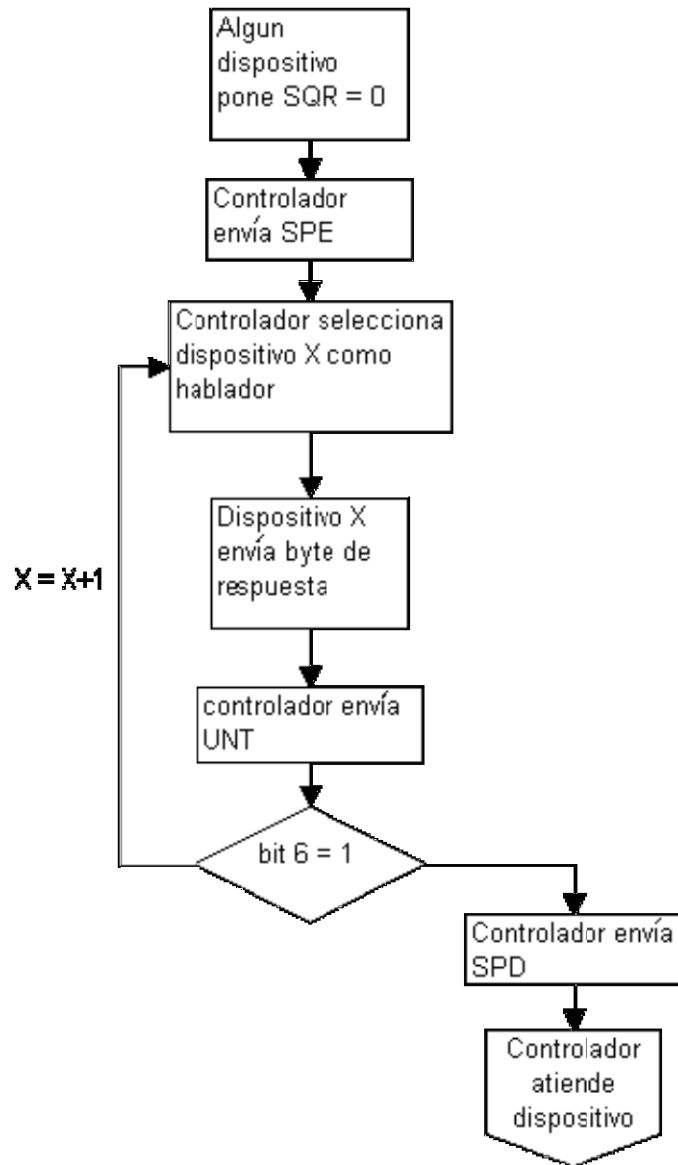


Diagrama de flujo 3.3 : polling serial

POLLING SERIAL



3.2.6 Órdenes básicas del estándar IEE 488.2.

El estándar IEEE-488.2 es un conjunto de órdenes básicas que realizan funciones comunes a todos los equipos, con independencia de su naturaleza.

Tabla 3.6 Comandos Básicos de IEEE 488.2

Orden	Nombre de la orden	Función
*CLS	Clear Status Command	Borra el registro de estado y los registros de incidencia
*ESE	Event Status Enable Command	Habilita bits del registro de habilitación de incidencias
*ESE?	Event Status Enable Query	Interroga el registro de habilitación de incidencias estándar
*ESR	Events Status Register Query	Interroga el registro de incidencias estándar
*IDN	Identification Query	Identifica tipo de instrumento y versión software
*LRN	Learn Device Setup Query	Requiere el estado actual del equipo
*OPC	Operation Complete Command	Fija el bit de “ Operación Completa” del registro estándar
*OPC?	Operation Complete Query	Responde con “1” si se ha ejecutado órdenes previas
*OPT?	Option Identification Query	Requiere la opción instalada en el equipo
*RCL	Recall Command	Restaura el estado del equipo del registro save/recall
*RST	Reset Command	Sitúa al equipo en el estado básico de referencia
*SAV	Save Command	Almacena el estado actual en un registro save/recall
*SRE	Service Request Enable Command	Habilita los bits del registro de habilitación de Byte de estado
*SRE?	Service Request Enable Query	Requiere el contenido del registro SER de habilitación del byte de estado
*STB?	Read Status Byte Query	Requiere el estado del registro resumido del Byte de estado
*TRG	Trigger Command	Arranca o dispara la operación del equipo de forma remota
*TST?	Selft – Test Query	Requiere el resultado del autotest del equipo
*WAI	Wait-to-Continue Command	Espera a que se realicen todas las operaciones pendientes

3.3 Lenguajes de Programación para GPIB

La estructura genérica de programación de un sistema GPIB se muestra en la figura 3.16.

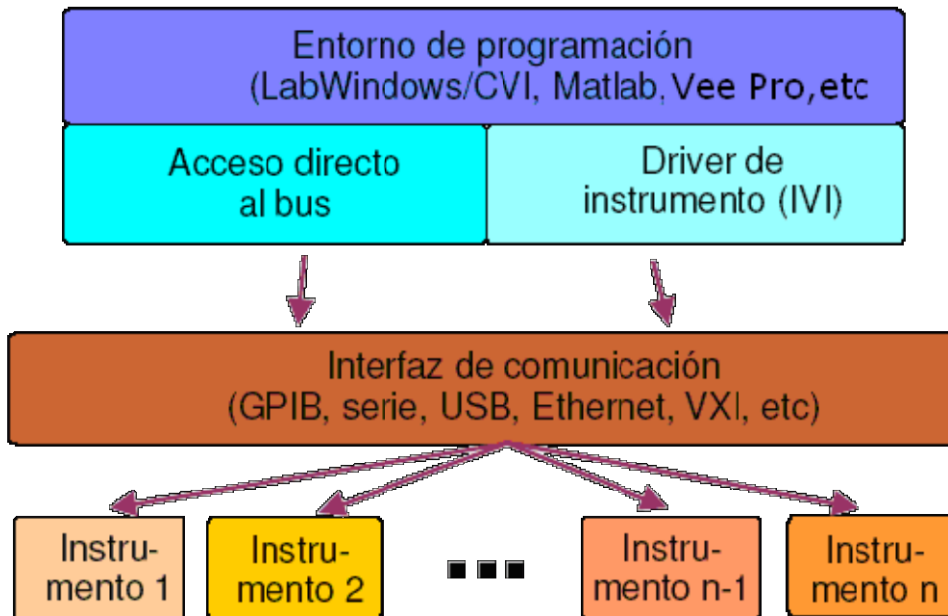


Figura 3.16 Estructura de programación de un sistema GPIB

La programación de los sistemas GPIB se realiza por intercambio de mensajes entre los dispositivos, mensajes que pueden ser de dos tipos:

- De **interfaz**, para la gestión del bus: inicialización y direccionamiento de los aparatos.
- De **datos**, dirigidos a un dispositivo específico, para decirle que tome una medida, cambiar su configuración, etc.

Los dispositivos se identifican en la red GPIB por su dirección numérica (asignada en el instrumento y leída por el *driver* GPIB instalado en el sistema), aunque mediante el lenguaje de programación se le puede dar un *handler* alfanumérico más descriptivo.

3.3.1 Software existente para GPIB

Los lenguajes de programación para aplicaciones de control y prueba se han vuelto muy amigables, también más poderosos. LabWindows/CVI ofrece un completo sistema de programación ANSI C y HPVVEE, ofrece las mismas funciones en un lenguaje de programación gráfico.

La primera alternativa (y, en ocasiones, la más práctica), es el uso de software propietario desarrollado por los mismos fabricantes del instrumento. Por ejemplo, el programa gratuito Intuilink de Agilent Technologies, el programa de pago Wavestar para los osciloscopios de Tektronix Inc, o el programa Vee Pro de Agilent utilizado en este proyecto para el manejo del osciloscopio Agilent 54621A. La ventaja evidente de estos programas es que pueden ser empleados nada más que al conectar los instrumentos, y proporcionan ya hechas las funciones más comunes que uno desea realizar, sin necesidad de programar.

Las desventajas son también claras: por tratarse de software cerrado, sólo puede ser usado para la tarea para la que fue diseñado, y además son imposibles de integrar con otros programas.

a) Labwindows/CVI y LabVIEW de National Instruments

LabWindows/CVI es un entorno de desarrollo completo basado en ANSI C (el nombre significa *C* para *Instrumentación Virtual*). Los aspectos más destacables de este producto son:

- Completas librerías para la comunicación entre dispositivos (puerto serie, paralelo, GPIB, TCP/IP, etc).
- Especial facilidad para el desarrollo de interfases gráficas adaptadas a los instrumentos de medida (dispone de elementos para mostrar formas de onda, conmutadores, potenciómetros, etc)
- Soporte para los *drivers* IVI de numerosos instrumentos (extensión .fp).

- Posibilidad de ofrecer el programa final desarrollado mediante una aplicación de instalación.

National Instruments, creadores de LabWindows/CVI, ofrece también LabVIEW, que, con la misma funcionalidad, está orientado a programación gráfica en vez de al desarrollo de código C. NI mantiene la compatibilidad con otros compiladores de propósito general, como las distintas suites de Microsoft Visual Studio (Visual Basic, Visual C++, .NET, etc), a través de su producto Measurement Studio.

b) Matlab con Instrument Control Toolbox

Matlab, que fue en origen un conjunto de rutinas para manipulación de matrices, ha evolucionado con el tiempo para convertirse en un entorno de programación de propósito general con gran potencia matemática y aplicabilidad en muchos ámbitos de la ciencia y la ingeniería, gracias a sus módulos de extensión (*toolboxes*) de procesamiento de señales, control, ecuaciones diferenciales, y un largo etcétera. En muchos casos las *toolboxes* representan el estado del arte en programación numérica en sus respectivos campos.

Recientemente, los creadores de Matlab, Mathworks, emprendieron una línea clara de ampliación de su mercado hacia la conexión de hardware con el PC, distribuyendo *toolboxes* para el control de tarjetas de adquisición, generación de código y emulación de DSPs, xPC (control remoto de PCs para operación en tiempo real), adquisición de imágenes, etc. La primera versión de la *toolbox* específica de **control de instrumentos** de medida apareció en 2001, y la versión 2.1 disponible en la actualidad (2004) está suficientemente depurada para ser competitiva con productos más maduros como CVI.

La *toolbox* ha sido diseñada de forma que el acceso a los instrumentos de medida se realiza mediante objetos, que pueden ser de dos clases: interfaz o de dispositivo. Mientras que el primer método equivale a la programación con comandos SCPI, el segundo enfoque permite explotar toda la potencia de IVI. Para ello es preciso disponer del *driver* de Matlab correspondiente (extensión

.mdd), obtenible en la página de Mathworks o bien usar una utilidad llamada makemid para convertir (o “envolver”) el *driver* IVI ya instalado en el sistema.

La utilidad midedit permite ver y modificar todas los objetos definidos en los aparatos, sus propiedades, y las funciones ejecutables.

c) Agilent VEE PRO



HP VEE, de Hewlett Packard, ofrece un completo ambiente gráfico como lenguaje de programación. Adicionalmente, FineTest ha escrito una Librería de Prueba, Controladores de Instrumento, un Program Shell para hacer muy fácil el desarrollo de programas de prueba. Se puede escribir un programa de prueba con un mínimo de conocimientos de 'HP VEE'. HP VEE también ha elaborado funciones interconstruídas para crear interfaces de usuario, controlar los puertos seriales, etc. lo cual hace mucho mas fácil la escritura de programas de prueba de las fuentes de poder de hoy.

HP VEE es un lenguaje gráfico y trabaja exactamente como un diagrama de flujo. El flujo del programa se ejecuta de arriba hacia abajo mientras que los datos pueden viajar de derecha a izquierda o de izquierda a derecha. Ver el siguiente diagrama de flujo:

Diagrama de flujo 3.4 Estructura de programación de VEE



Agilent VEE es un idioma de la programación gráfica perfeccionado para aplicaciones de prueba y medidas, programas con interfaz del operador.

Las ventajas de usar Agilent VEE son las siguientes:

- Incrementa la productividad dramáticamente.
- Permite el uso en una amplia gama de aplicaciones que incluyen pruebas funcionales, verificación de diseños, calibración, control y adquisición de datos
- Flexibilidad y control de instrumentos con GPIB, VXI, GPIO serial, PC Plug, instrumentos LAN. Se puede usar “ panel drivers, VXI plug&

play, “ direct I/O sobre la interfase estándar, o importar librerías de múltiples vendedores.

- Usando Active X Automation and Control on PCs permite controlar otras aplicaciones tales como MS Word, Excel, Acces, para el análisis de datos o poner el resultado de la prueba sobre una base de datos para un futuro uso
- Elaborar grandes programas con facilidad, viene a ser más flexible en el manejo de instrumentos. VEE tiene compilador, un adecuado ambiente de trabajo para programas largos y complejos y capacidad de manejo de instrumentos avanzados.
- VEE soporta Visual Studio NET

Los programas de VEE son creados seleccionando los objetos de los menús y conectándolos juntos. El resultado en VEE es un diagrama de flujo de datos, que es más fácil usar y entender que las tradicionales líneas de código. Usando VEE no es laborioso editar- compilar- enlazar- ejecutar ciclos.

En VEE, el programa se construye con elementos del programa llamados “objetos”. Los objetos son estructuras en bloques de un programa en VEE. Ellos realizan varias funciones tales como: operaciones I/O, análisis, y visualización. Cuando se ve los objetos con todas las conexiones se llama “vista de detalle”. La vista de detalle es análoga al código fuente de un lenguaje textual.

En VEE los datos se mueven de un objeto al próximo objeto de un modo que consiste en: Los datos de entrada a los pines de la izquierda, los datos de salida de la derecha, la secuencia de operación de los pines de arriba y de abajo como se muestra en la figura 3.17

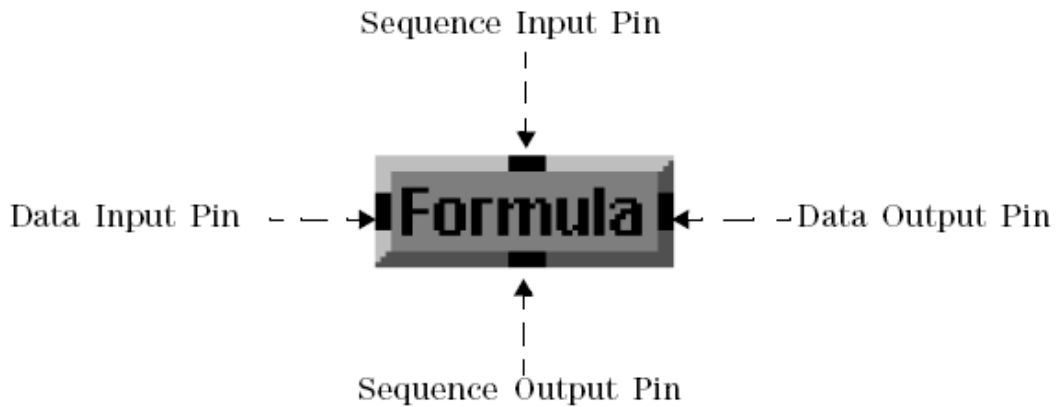


Figura 3.17 Terminales de un objeto de VEE

VEE provee muchas opciones para el control y comunicación con instrumentos, algunos de ellos se indican a continuación

- Panel drivers (Instrument Drivers) para más de 450 instrumentos de diferentes vendedores. La mayoría de los drivers están disponibles para varios vendedores Plug&play compatibles con Windows 98, Windows 2000, Windows NT o Windows XP.

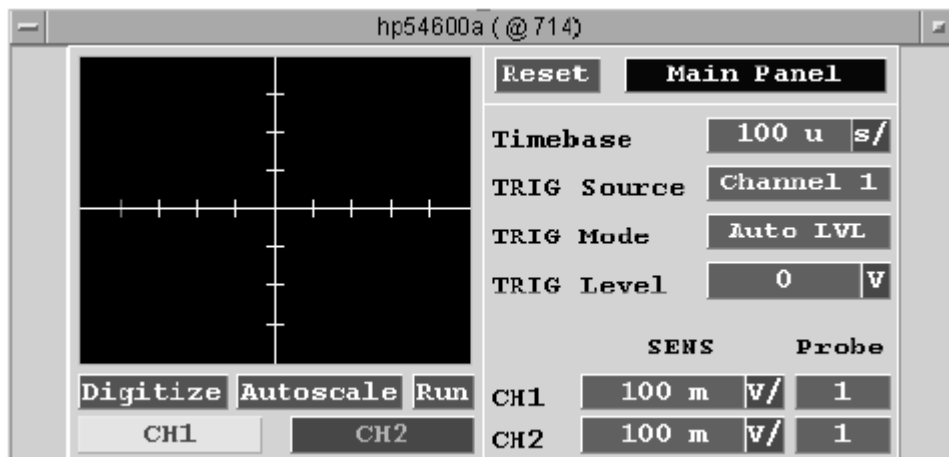


Figura 3.18 Panel Driver del osciloscopio HP 54600A

- VEE direct I/O se usa para enviar a los instrumentos, comandos strings sobre interfases estándar tales como GPIB (IEEE-488), OPIO, RS 232, VXI o LAN
- Control de una gran variedad de tipos de instrumentos con facilidad, organización y capacidad de dirección de los instrumentos.

Después de que haya instalado e iniciado VEE, se muestra la siguiente ventana de trabajo como se indica en la figura 3.19

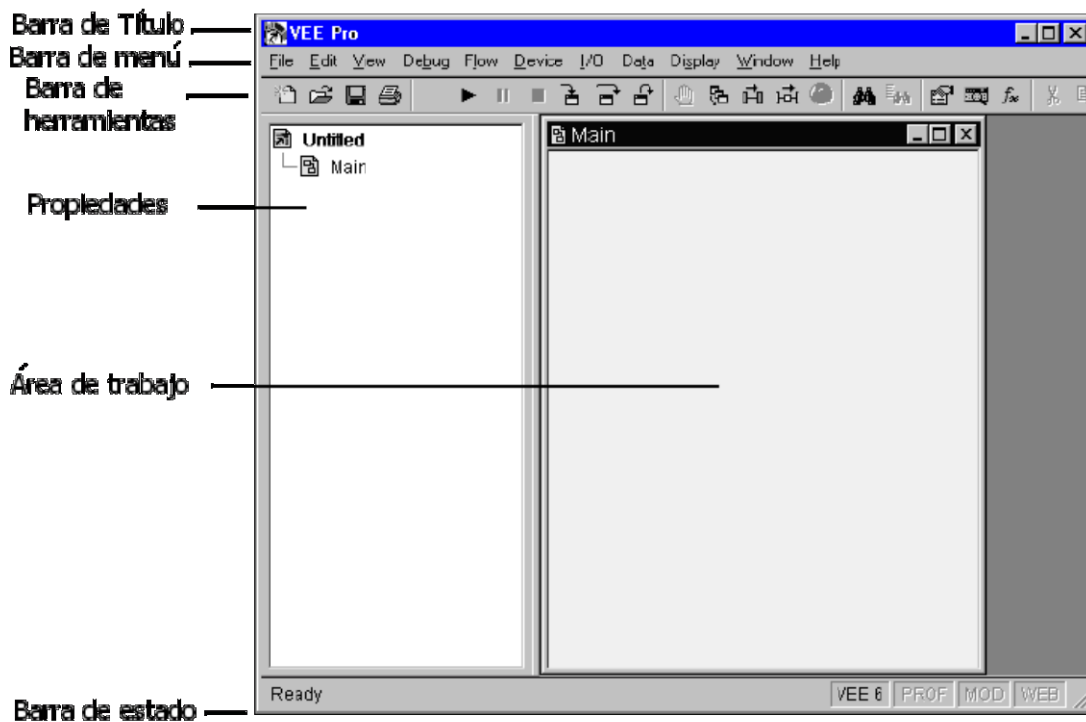


Figura 3.19 Ambiente de trabajo de VEE

Se incluye VEE Direct I/O object que permite con cualquier instrumento de cualquier vendedor comunicación sobre una interfase estándar (si no hay un driver disponible para el instrumento). The Direct I/O object trabaja transmitiendo comandos al instrumento y recibiendo datos del instrumento. Usando Direct I/O, generalmente rinde velocidades de ejecución mas rápidas. La figura 3.20 muestra un ejemplo del uso de Direct I/O para controlar un generador de funciones:

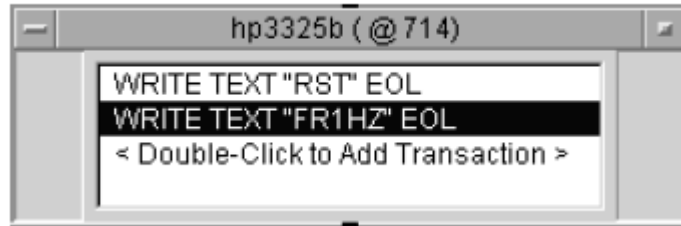


Figura 3.20 Un generador de funciones con Direct I/O

En el capítulo 4 se mostrará con más detalle la aplicación de este lenguaje de programación gráfica para la implementación de la red de comunicación de instrumentos utilizando al interfaz GPIB 82350 A y el osciloscopio HP 54621A

3.4 OTRAS TECNOLOGÍAS PARA LA CONEXIÓN DE INSTRUMENTOS DE MEDIDA

Las dos tecnologías de comunicación con instrumentos más comunes en la industria, GPIB y el puerto serie, provienen de los años 70; es claro que desde entonces la industria informática ha experimentado cambios radicales.

3.4.1 USB: Universal Serial Bus

El USB fue creado como puerto para la conexión de periféricos (impresoras, cámaras digitales, unidades de disco, escáner, etc) a los PCs. En el diseño de USB se optó por *plug and play*, de forma que el PC reconozca y configure los dispositivos en el momento de su conexión. Permite la conexión simultánea de hasta 127 dispositivos en un puerto, con una velocidad de transferencia de datos de hasta 60 Mbytes/s (estándar USB 2.0). USB es un sistema de comunicación barato y está implementado en cualquier PC moderno. Sin embargo, los cables USB no están preparados para entornos industriales (con posible pérdida de datos ante el ruido electromagnético), carecen de un mecanismo de enganche al PC, y la distancia de conexión está limitada a 30 m. Además, no existe un protocolo estándar sobre USB: cada fabricante debería desarrollar el suyo propio.

3.4.2 Ethernet

Ethernet es una tecnología de conexión de ordenadores madura y de amplia implantación. Las ventajas son su ubicuidad, la posibilidad de control remoto, y la facilidad para compartir los instrumentos entre varios usuarios. La velocidad de transmisión de datos tiene un máximo teórico entre 10, 100 o 1000 Mbits/s (estándar Ethernet Gigabit), aunque hay que considerar que raramente es alcanzable en condiciones normales, siendo muy dependiente del tráfico de la red. Una desventaja importante inherente al diseño de Ethernet es que el orden de llegada de los datos no es determinista. También se puede plantear la seguridad global del sistema, y, como factor psicológico en contra el que los ingenieros tuvieran que recurrir a los administradores de red para montar su sistema. Otra alternativa interesante es la Ethernet inalámbrica (IEEE 802.11).

3.4.3 Firewire (IEEE 1394)

Diseñado por Apple, y estandarizado por el IEEE 1394 en 1995, Firewire fue concebido para la transmisión masiva de datos en entornos multimedia (por ejemplo en cámaras de vídeo digitales), alcanzando velocidades de 400 Mbits/s (el siguiente estándar establecerá velocidades de hasta 3.2 Gbits/s). Firewire soporta hasta 16 dispositivos en cadena, pudiendo estar separados mutuamente un máximo de 4.5 m de distancia. Al contrario que USB, existe un protocolo definido para el control de instrumentos sobre IEEE 1394; sin embargo, pocos instrumentos de medida lo reconocen. Dos inconvenientes para la implantación de este estándar son la ausencia de los chipsets correspondientes en las placas base de los PCs, por lo que el usuario debe adquirir una tarjeta PCI por separado (Macintosh la incorporan de fábrica) y que los cables Firewire no están inmunizados al ruido, siendo por tanto no aptos para entornos industriales.

CAPÍTULO IV

DISEÑO DE UNA RED GPIB

En el presente capítulo se realiza una descripción de las características del osciloscopio digital Agilent 54621 A. Se analiza la estructura de los comandos para la comunicación con el osciloscopio. Se describe los elementos requeridos tanto de software y hardware para la instalación de la red. Se indica la forma direccionar instrumentos tanto en el equipo como en la PC a través del programa Agilent VEE . Luego se describe la programación de las diferentes ventanas creadas para establecer la comunicación entre los osciloscopios y la PC vía GPIB. Por último se indica las pruebas realizadas de la red en funcionamiento así como una comparación entre los valores observados en el osciloscopio y los valores presentados en la pantalla de la PC

4.1 OSCILOSCOPIOS DIGITALES

Estos osciloscopios están basados en un conversor A/D, su principio de funcionamiento difiere totalmente del de los analógicos. Toman muestras de la señal a una frecuencia de muestreo determinada y lo manejan generalmente mediante Procesamiento Digital de señales (DSP).

Una característica nueva que aparece entonces en la especificación es la frecuencia de Muestreo (Sampling Rate).

Cuando se conecta la punta de un osciloscopio digital al circuito, la sección vertical ajusta la amplitud de la señal de la misma forma que lo hacía el osciloscopio analógico.

El conversor A/D del sistema de adquisición de datos hace un muestreo de la señal a intervalos de tiempo determinados y convierte la señal de tensión continua en una

serie de valores digitales llamados muestras. En la sección horizontal, una señal de reloj determina cuando el conversor A/D toma una muestra. La velocidad de este reloj se denomina velocidad de muestreo (Sample Rate) y se mide en muestras (samples) por segundo.

Los valores digitales muestreados se almacenan en una memoria como puntos de señal. El número de los puntos de señal utilizados para reconstruir la señal en pantalla se denomina registro. La sección de disparo determina el comienzo y el final de los puntos de señal en el registro. La sección de visualización recibe estos puntos del registro, una vez almacenados en la memoria, para presentar en pantalla la señal.

Dependiendo de las capacidades del osciloscopio se pueden tener procesos adicionales sobre los puntos muestreados, incluso se puede disponer de un predisparo, para observar procesos que tengan lugar antes del disparo.

Los osciloscopios digitales hoy han avanzado notablemente en prestaciones, facilidad de uso, velocidad y han reducido su costo. Por ésta y otras razones, la tendencia hacia el futuro es que vayan desplazando progresivamente a los analógicos.

Las principales ventajas de los osciloscopios digitales son:

- Almacenamiento digital
- Mayor precisión
- Idéntica focalización en cualquier base de tiempo
- Procesamiento digital de las señales
- Mayor cantidad de funciones
- Autocalibración
- Menor tamaño
- Conectividad con PCs e impresoras

La única desventaja - además de su mayor costo - es que quizá pueda ser más difícil de operar dada la mayor cantidad de funciones que poseen, especialmente para operadores habituados a los analógicos.

En cuanto a las especificaciones: los **anchos de banda** van de los 100 a 200 MHz hasta los 600 MHz en los convencionales. En cuanto a los de alta frecuencia, la oferta llega hasta varias decenas de GHz.

No se debe confundir esta especificación con la de **Sampling Rate**, que como se dijo antes, es la cantidad de muestras por segundo. Este valor es mucho mayor, sus valores oscilan en el orden de los GS/s (Giga Samples por segundo).

Los osciloscopios digitales se pueden clasificar de la siguiente manera:

- De tiempo real :
 - De almacenamiento digital (DSO)
 - De fósforo digital (DPO)
 - De señal combinada (MSO)

- De muestreo

Los de **Tiempo Real** tipo de "**Almacenamiento Digital** " u "**Digital Storage Oscilloscopes** " (**DSO**) cubren una amplia gama de aplicaciones generales como son la depuración de señales digitales y de señales repetitivas, etc.

Sus características principales son:

- Anchos de banda de 100 MHz a 4 GHz
- Sample rates de 1.25 GS/s a 20 GS/s
- Capturas de formas de onda a razón de 400000 formas de onda por segundo
- 32 MB de almacenamiento

A continuación se muestra con mayor detalle la especificación y las características de operación del modelo **Agilent 54621A**, que son osciloscopios **DSO** de la gama de 60 a 500 MHz, utilizado en el presente proyecto.

En la figura 4.1 se muestra la estructura del osciloscopio modelo Agilent 54621 A con las señales adquiridas a través de las puntas de prueba

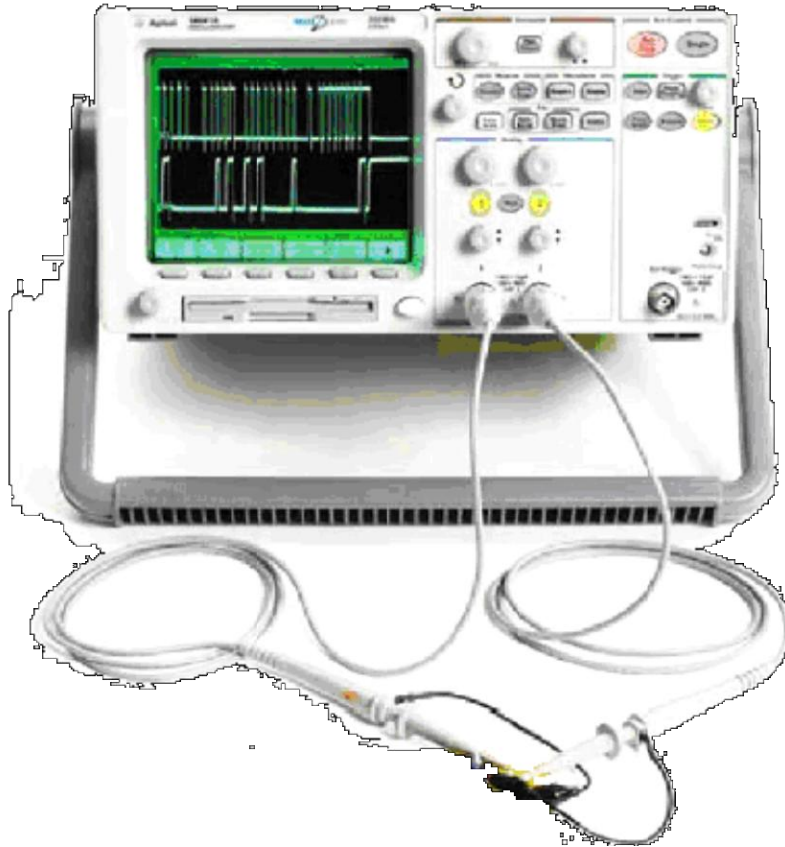


Figura 4.1 Adquisición de ondas con el osciloscopio 54621 A

A continuación se indican las características principales del osciloscopio 54621 A de Agilent

- Máx. frecuencia muestreo: 200 MHz
- Profundidad de memoria: 2 MB/canal
- Resolución vertical: 8 bits
- Resolución temporal: 12 bits
- Detección de picos: 5 ns
- Promediado capturas: 1,2,4,8,...,16383
- N° Canales: 2
- Anchura de banda: dc a 60 MHz
- Acoplo AC: 3.5 a 60 MHz

- Rango Amplificación: 1 mV/div a 5V/div
- Max rango de entrada: 300 Vrms a 400 Vp
- Impedancia entrada: 1M Ω y 14 pF
- Modos de disparo: Ch1,Ch2,Ext, Line

Medidas automáticas:

Tensión: Peak-to-Peak, Maximum, Minimum, Average, Amplitude, Top, Base, Overshoot, Preshoot, RMS (DC).

Tiempo: Frequency, Period, + Width, - Width and Duty Cycle, Rise time, Fall time, Max, Min, Delay y Phase.

Cursores: Tiempo (X, ΔX y $1/\Delta X$), tensión (Y y ΔY)

Procesado forma de onda: Suma, diferencia, producto, FFT, derivación e integración.

Este **osciloscopio digital**, presenta muchas ventajas sobre los osciloscopios analógicos entre ellas se puede destacar:

- Pueden visualizar simultáneamente muchas señales.
- Permite visualizar señales adquiridas con un disparo único (“single shot”)
- Permite disparos basados en criterios muy sofisticados.
- Permite visualizar la forma de onda tanto anterior como posterior al instante de disparo.
- Se puede obtener información muy elaborada de las señales que se adquieren.
- Se puede transferir posteriormente las formas de ondas adquiridas a sistemas externos (procesador plotter, etc.).

La figura 4.2 muestra el panel frontal del osciloscopio 54621 A con las funciones de los principales botones

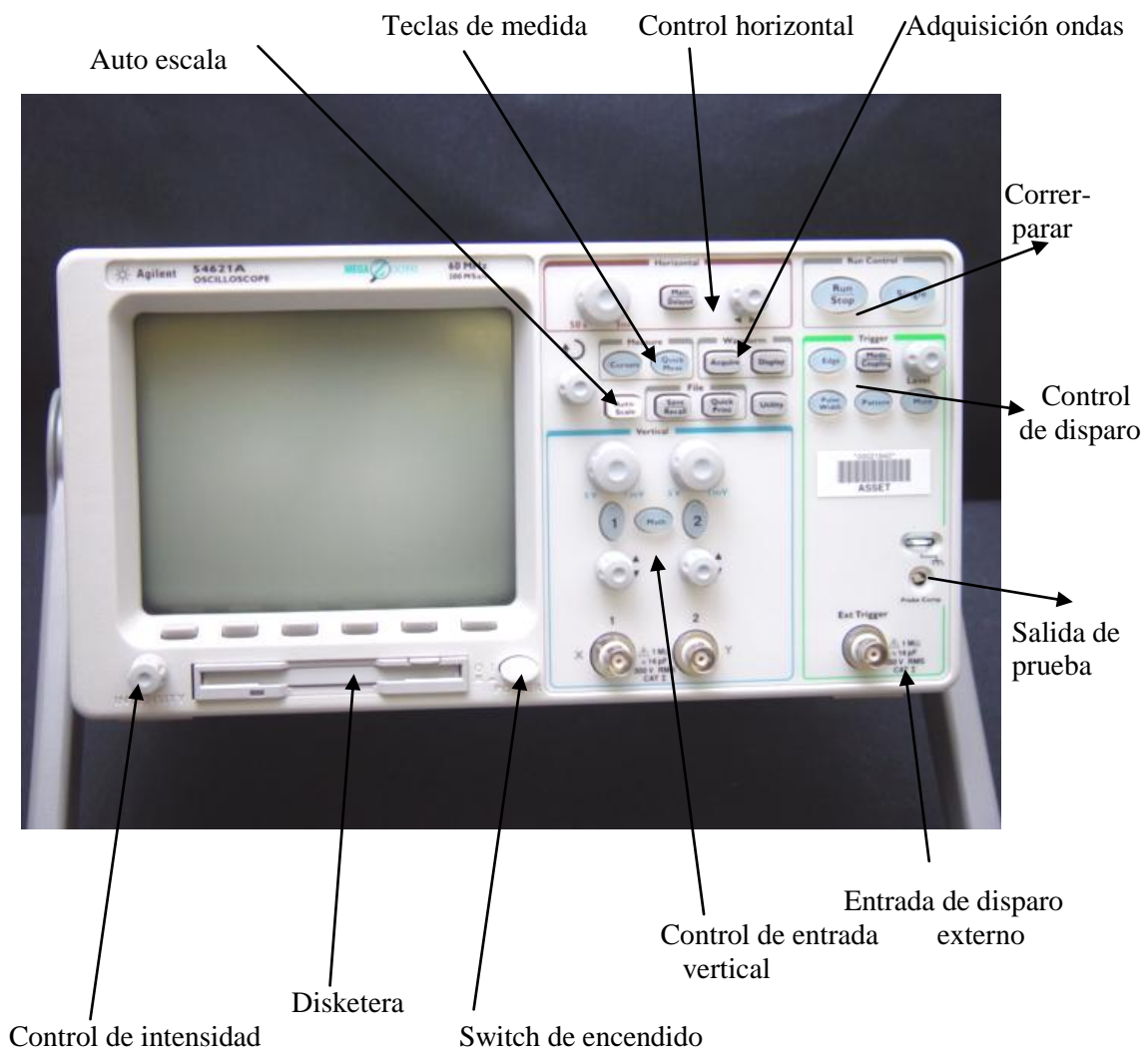
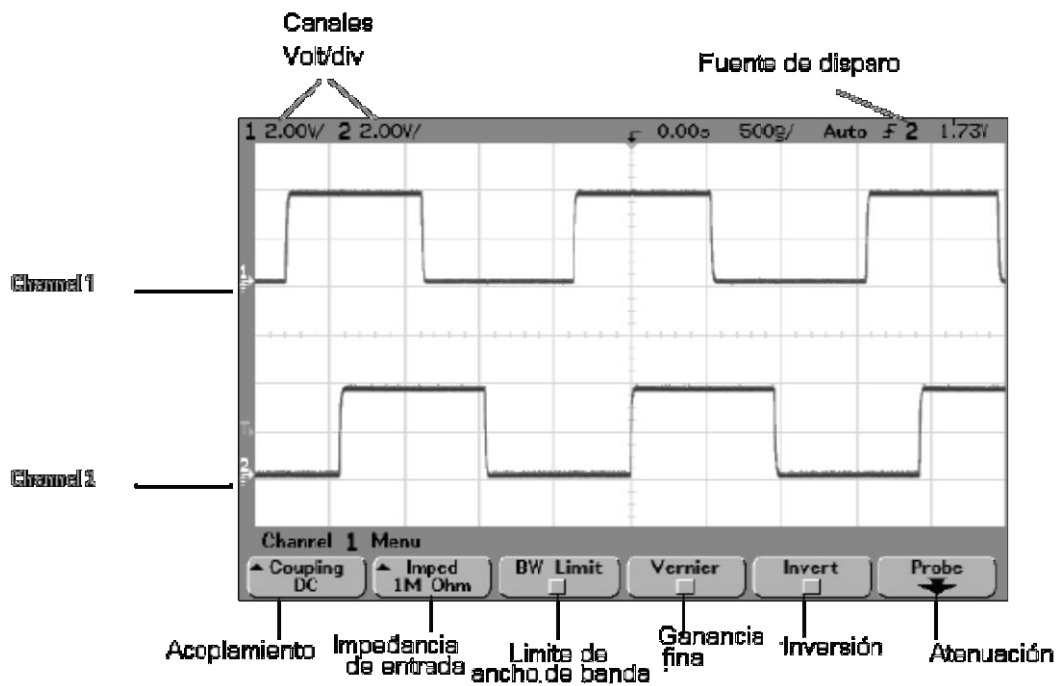


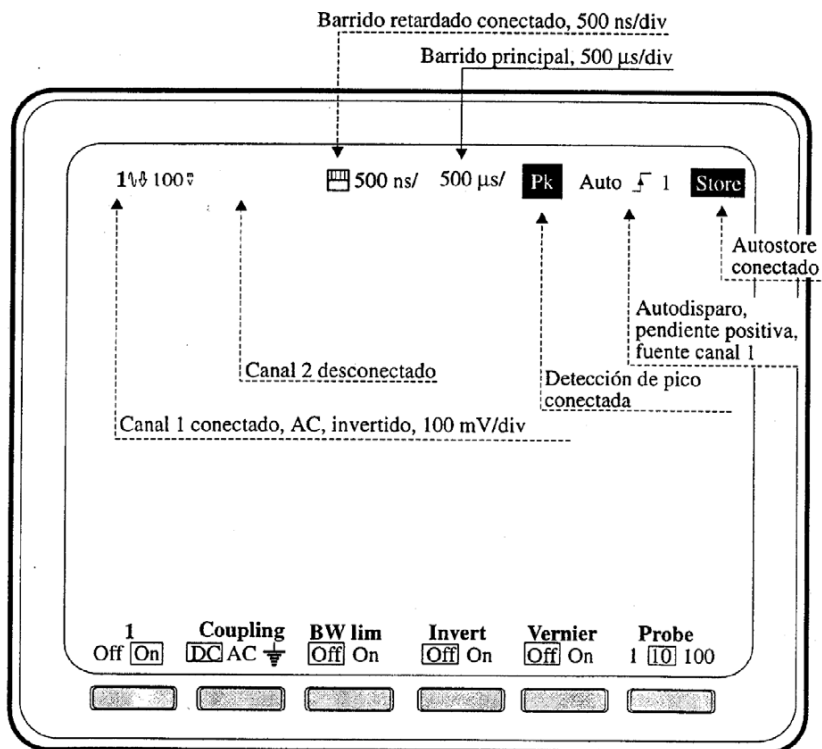
Figura 4.2 Panel frontal del osciloscopio 54621 A

Los diferentes elementos del panel de control no tienen escalas cuantitativas y así mismo su función puede variar según el estado de operación que se encuentre el osciloscopio.

Las informaciones correspondientes se visualizan directamente sobre los bordes del display, y se van modificando según se vaya accionando los elementos de control. La forma habitual de operar es mover los dispositivos de control y observar sobre el display los efectos que producen los cambios. La figura 4.3a y 4.3b muestra las señales adquiridas, las escalas utilizadas, y las opciones de menú que se pueden escoger



(a)



(b)

Figura 4.3 a y b (Selección de la opción del menú) Opciones del menú

Control de los canales verticales: Al pulsar sobre la tecla del canal 1 o 2 se tiene:

Off: Se deja de visualizar el canal

On: Se visualiza el canal.

Coupling: DC : Se visualiza la señal con todo el espectro DC –100 MHz

AC : Se introduce un filtro paso alto de 3.5 Hz

Gnd : Se visualiza la tierra de la señal.

BWLim: Off : Se visualiza la señal con todo su espectro DC-100 MHz.

On : Se introduce un paso bajo de 20 MHz para reducir el nivel de ruido.

Invert: Off : Se visualiza la señal sin cambio de polaridad.

On : Se visualiza la señal con la polaridad cambiada.

Vernier: Off: La ganancia del canal se modifica a los valores normalizados 1-2-5-10.....

On : La ganancia del canal se modifica en pasos más finos.

Probe: 1 : No se introduce ganancia adicional.

10 : Se introduce una ganancia x10 para compensar la atenuación de la sonda.

100: Se introduce una ganancia x100 para compensar la atenuación de la sonda.

Las sondas introducen una atenuación para compensar la capacidad parásita en la entrada del osciloscopio:

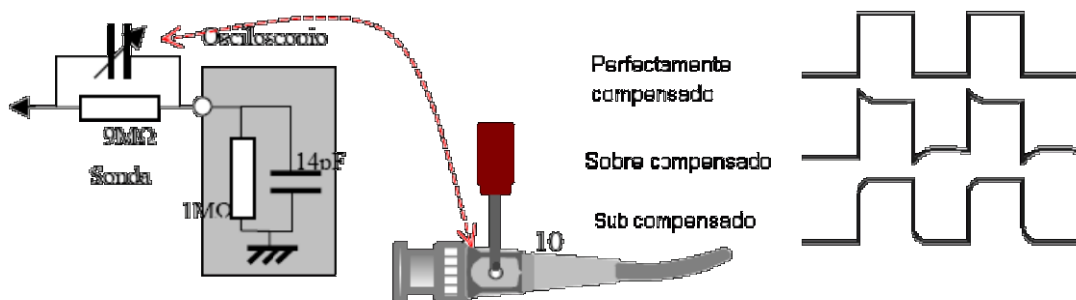


Figura 4.4. Estructura interna de la punta de prueba

Control de la base de tiempo.

Main : Selecciona el control de la base de tiempo principal.

El botón **Time/Div** controla el factor de conversión de tiempo.

El botón **Delay** controla el retraso de la señal que se visualiza respecto del disparo.

Delayed: Representa la señal con una doble base de tiempo.

Roll: Causa que la forma de onda se mueva lentamente a través del display.

Sólo opera con la base de tiempo 500 ms/div o más lenta. En este modo no hay trigger.

Vernier: Off: La base de tiempo se modifica en los valores estándar 1-2-5-10.

On: La base de tiempo se modifica en valores muy finos.

Time Ref: Center: El tiempo de referencia que coincide con el disparo y que se marca con el símbolo (\blacktriangledown), se centra en el punto medio de la pantalla en el margen superior del display.

Left: El tiempo de referencia se sitúa próximo al margen izquierdo.

Estrategias de disparo.

Trigger Source: Permite seleccionar la señal que se utiliza como referencia para establecer el instante de disparo de la base de tiempo:

Channel 1: Se utiliza la señal del canal 1.

Channel 2: Se utiliza la señal del canal 2.

Line: Se utiliza la señal de red

External: Se utiliza la señal externa que se introduce por una entrada independiente.

Mode: Establece el criterio que se utiliza para producir el disparo de la base de tiempo.

AutoLvl: Se utiliza como nivel de disparo el valor medio de la señal que se utiliza como referencia para el trigger. Si pasado un tiempo de *time out*

(25 ms) sin que se encuentre la condición de disparo, el osciloscopio se autodispara.

Auto: Se utiliza como nivel de disparo el valor medio de la señal que se utiliza como referencia para el trigger. No se utiliza autodisparo.

Normal: El nivel de disparo se fija manualmente mediante el botón "Level".

Single: El disparo se realiza una sola vez. Responde al disparo manual ejecutado pulsando el control Run.

TV: Se utilizan estrategias de disparo relativas a señales de video.

Slope/Coupling: Establece el flanco y el modo de filtrado de la señal que se utiliza como referencia.

Flanco +: El disparo se produce cuando la señal de referencia supere el nivel de disparo.

- : El disparo se produce cuando la señal de referencia caiga por debajo del nivel.

Coupling DC/AC: La señal de referencia se introduce directamente o a través de un filtro pasa - altos de 10 Hz que elimina el nivel de continua.

Reject: Off/LF/HF: La señal de referencia se introduce directamente, a través de un filtro paso bajo de 50 Hz o a través de un filtro paso alto de 50 Hz.

Noise Reject: ON/OFF La señal de referencia se filtra o no a través de un filtro de 1 Mhz para eliminar el ruido.

(TV Mode) Ofrece patrones de disparo específicos de señales de vídeo.

Medidas automáticas

Generales: Comunes a los diversos menús:

Source: 1 o 2 : Selecciona el canal sobre el que se hacen las medidas.

Clear Meas: Elimina las opciones de medida seleccionadas previamente.

Next Menú y Previous menú: Opciones para navegar por los sucesivos menús.

Voltaje: Permiten calcular características sobre las tensiones de las formas de onda que se está visualizado en la pantalla.

Vpp : Tensión pico a pico de la señal representada en la pantalla.

Vavg : Tensión media de la señal representada en la pantalla (nivel de continua).

Vrms : Valor rms de la señal representada en la pantalla.

Vmax: Valor de pico positivo de la señal representada en la pantalla.

Nmin: Valor de pico negativo de la señal representada en la pantalla.

Vtop: Valor estacionario positivo al que tiende la señal representada en la pantalla.

Vbase: Valor estacionario negativo al que tiende la señal representada en la pantalla.

Tiempo: Permiten calcular características de la temporización de las formas de onda que se están visualizando sobre la pantalla:

Freq: Frecuencia de la señal seleccionada.

Period: Período de la señal seleccionada.

Duty cycle: Relación pulso/no pulso de una señal cuadrada.

-Width, +Width, Rise time y Fall time: Cálculo de tiempos de una forma de onda tipo pulso.

Modos de visualización

Display: Controla el origen de la traza que se visualiza en la pantalla. Existen tres modos de visualización como se indica en la barra de la figura 4.5



Figura 4.5 Modo de visualización de la traza en la pantalla

Normal: Se visualiza en la pantalla la señal que se ha adquirido en el último barrido.

Peak Det: Monitoriza en la traza cualquier impulso de anchura superior a 5 μ s aún cuando sea inferior al período de muestreo que corresponda a la base de tiempo establecida. (Sólo es necesaria en bases de tiempo más lentas de 1ms/div)

Average: Representa en la pantalla una traza que corresponde al promedio de las señales obtenidas en los 8,64, 256 últimos barridos. Con ello se reduce en un factor 2.8, 8 y 16 la amplitud de los ruidos que sean asíncronos con el instante de disparo.

En la figura 4.6 se indica las señales adquiridas del modo normal y del modo Average

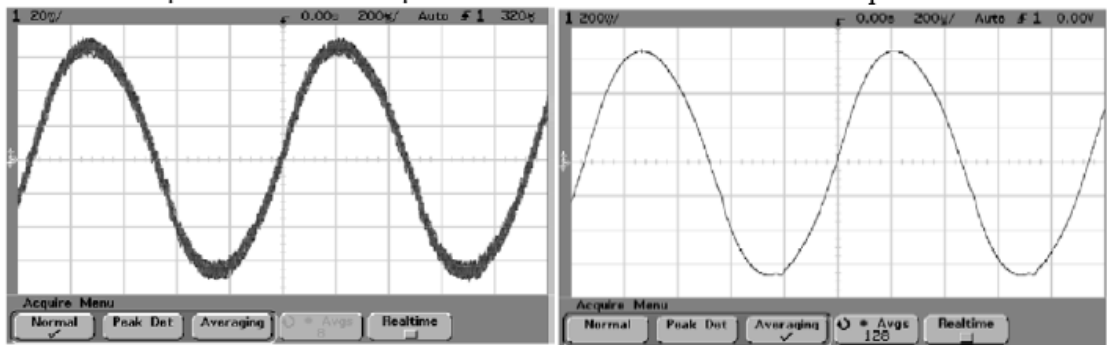


Figura 4.6 Adquisición en forma normal y average

Vector (Off/On): Elimina o introduce los segmentos que unen las sucesivas muestras de la traza.

Grids (Off/On): Elimina o visualiza la regilla superpuesta a la pantalla.

Configuración del Osciloscopio

Autoscale: Configura automáticamente el osciloscopio de acuerdo con las señales que están presentes en sus canales.

Setup: Permite salvar en memoria la configuración actual del equipo y recuperar una previamente salvada.

1/2/3/4/5: Código del registro de memoria para configuración con el que se trabaja.

Save: Almacena en el registro de memoria la configuración actual del osciloscopio.

Recall: Establece en el osciloscopio la configuración que previamente se almacenó en el registro de memoria.

UndoAutoscale: Recupera la configuración que había antes de pulsar autoscale.

DefaultSetup: Establece la configuración del osciloscopio de encendido.

Trace: Permite almacenar en memoria las trazas que están visualizadas en la pantalla y recuperarlas posteriormente.

Mem1/Mem2: Selección del registro de memoria para traza con la que se va a operar.

Trace (Off/On): Visualiza o deja de visualizar la traza en la pantalla.

Save to: Salva las trazas actuales de la pantalla en el registro de memoria.

Clear: Borrar el registro de trazas de memoria.

Recall setup: Recupera la configuración que había cuando se almacenó la traza en el registro seleccionado.

Print/Utility: Conjunto de comandos que permiten configurar el osciloscopio para comunicarse con otros equipos o para procesos de calibración y verificación.

4.2 COMUNICACIÓN CON EL OSCILOSCOPIO AGILENT HP 54621A

El osciloscopio HP54621 puede ser gestionado de forma remota desde un computador utilizando la interfaz para el bus GPIB. A través de la interfaz se pueden realizar operaciones básicas:

- Controlar el modo de operación del osciloscopio.
- Ejecutar operaciones de medida con el osciloscopio.
- Adquirir la información generada por el osciloscopio (formas de ondas, medidas, configuraciones, etc)
- Enviar información para ser representada en un osciloscopio (Imagen de pixels, configuración, etc).

La comunicación entre el computador que actúa como controlador, y el osciloscopio se realiza mediante intercambios de mensajes, siguiendo el estándar IEEE-488.1

Para que el osciloscopio pueda operar de forma remota, necesita que esté dotado de la interfaz HP-IB (que es optativa), como se observa en la figura 4.7

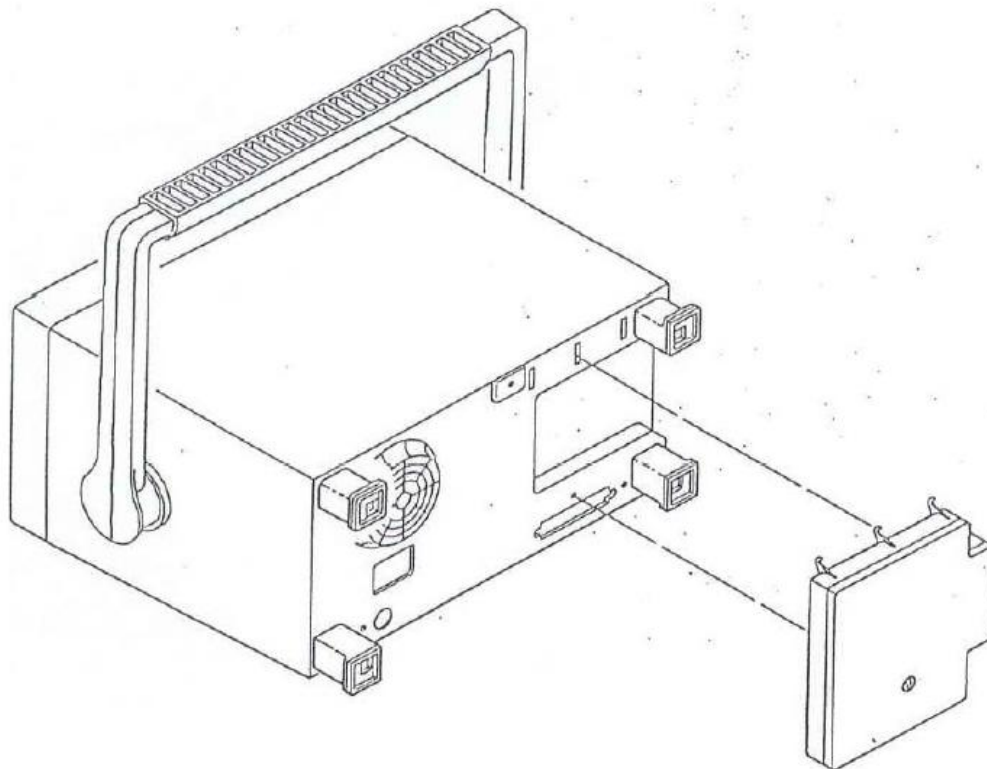


Figura 4.7 Conexión de la interfaz N2757A para conectar cable GPIB

Para que el osciloscopio responda a los comandos enviados a través del bus GPIB, se requiere que previamente haya sido programada su interfaz. Esto se lleva a cabo desde el menú correspondiente al botón **Print/Utility**, como se indica en la figura 4.8. A través del menú se debe establecer:

- 1) Modo **Computer**: que indica que el controlador va a ser un computador externo.
- 2) Dirección GPIB: que será un entero en el rango 0-30

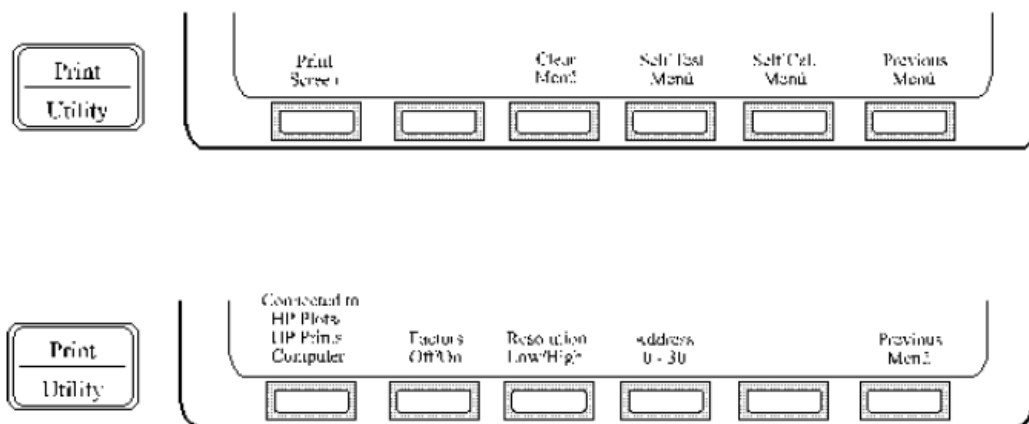


Figura 4.8 Menús para la programación en el osciloscopio de la PC como controlador

4.2.1 Inicialización del Osciloscopio

Antes de comenzar a intercambiar mensajes con el osciloscopio, se debe inicializar tanto su interfaz, como el programa Parser. Esto se hace mediante las dos operaciones:

(*CLR'); *Inicializa la interfaz del osciloscopio, e inicializa el programa de control Parser a su estado base.*

(*RST'); *Establece el Osciloscopio a su estado de inicialización (preset)*

4.2.2 Estructura de los mensajes de orden

Los mensajes que deben ser enviados para ejecutar una orden en el osciloscopio, son strings alfanuméricos, con la siguiente estructura:

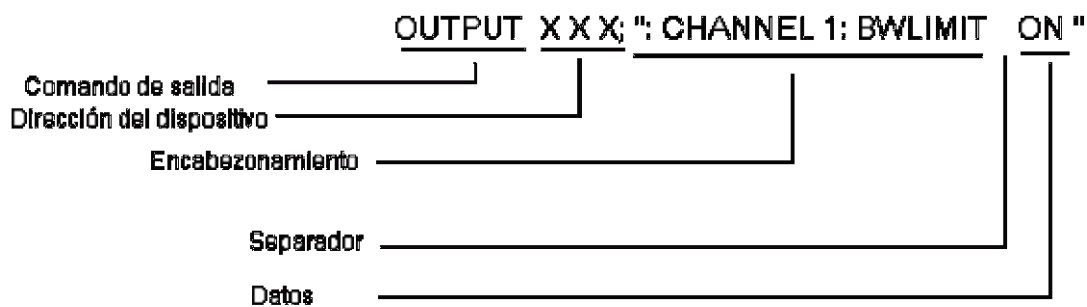


Figura 4.9 Estructura de mensajes de orden

Encabezamiento: Es el identificador de la orden que se ejecuta. Está compuesta de una cadena de campos separados por delimitadores ":". Cada campo identifica un nivel dentro del árbol de órdenes, y en conjunto, identifican unívocamente una orden.

El identificador de un nivel se compone de unos caracteres obligatorios, y otros que pueden incluirse optativamente, para dar mayor legibilidad al programa.

CHANnel ⇒ CHAN ⇔ CHANN ⇔ CHANNE ⇔ CHANNEL

Los identificadores pueden expresarse indiferentemente con mayúsculas o minúsculas

CHANNEL \Leftrightarrow CHANnel \Leftrightarrow Channel \Leftrightarrow channel

Los encabezamientos de las órdenes de requerimiento finalizan con el carácter “?”.

‘:MEASURE:FREQUENCY?’

Los encabezamientos de las órdenes básicas del protocolo IEEE-488.2 no están incluidas en el árbol y siempre empiezan por el carácter “*”.

‘*RST’

Delimitador: Es un espacio o conjunto de espacios.

Datos de programas: Son valores numéricos o enumerados que cualifican la orden. Pueden ser uno o varios, según la orden de que se trate. En este caso los datos deben estar separados por comas o combinaciones de comas y espacios.

Los valores numéricos pueden expresarse en notación decimal o científica: $28 = 0.28E2 = 280e-1 = 28000m = 0.028K = 28e-3K$

4.3 INSTALACIÓN

Para la implementación de la red se utilizó los siguientes equipos y accesorios:

- Computadora Pentium IV, 2 Ghz con Slot PCI
- Plataforma de programación gráfica VEE PRO 7
- Tarjeta GPIB HP 82350 A y librerías VISA y SICL para XP
- Osciloscopios digitales Agilent 54621 A
- Cables GPIB (uno de 1m y 4 de 2mts), el promedio entre los 5 dispositivos debe ser menor o igual a 2mts , que se cumple con el tamaño del cable elegido

La red de osciloscopios puede ser conectada en estrella o tipo bus. Para el presente proyecto se utilizó la conexión tipo bus, que se acondiciona a la disposición de los osciloscopios en el laboratorio como se muestra en la figura 4.10

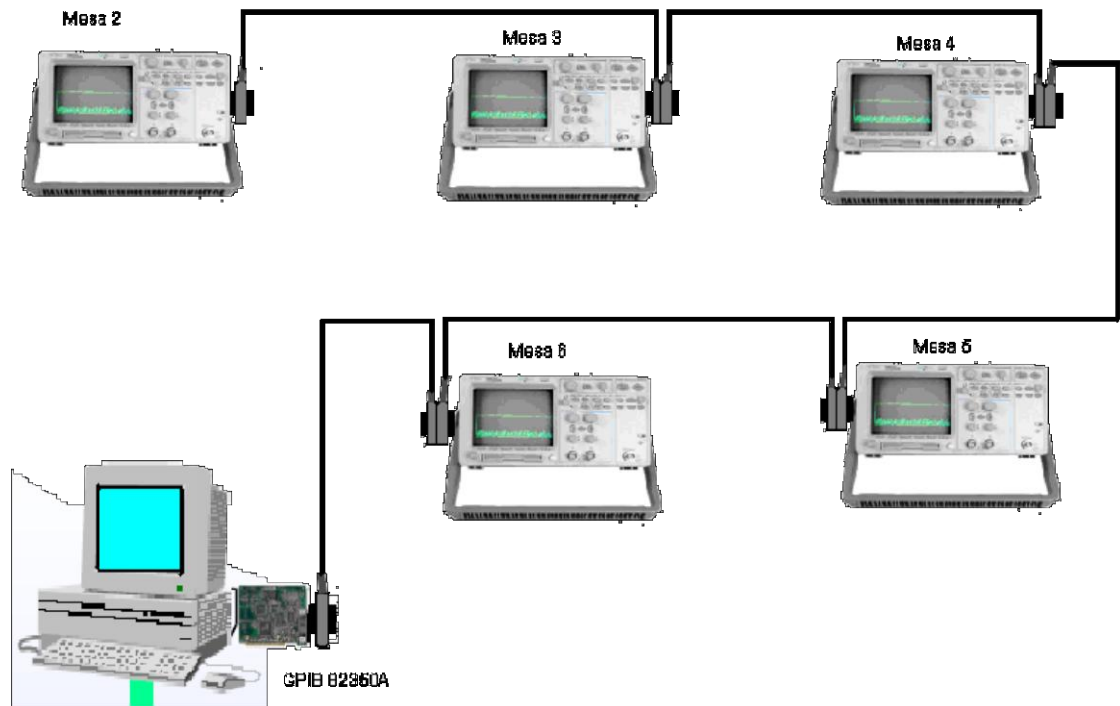


Figura 4.10 Red de osciloscopios con tarjeta GPIB, conexión tipo bus

4.4 CONFIGURACIÓN

Una vez que ha sido instalada la tarjeta GPIB 82350 A e instalado las librerías VISA e SICL, se procede a conectar todos los osciloscopios en la red tipo bus como se indicó en la figura anterior.

En el osciloscopio se programa la interfaz, esto se lleva a cabo desde el menú correspondiente al botón **Print/Utility**.

Se selecciona tipo de interfaz GPIB y se coloca la dirección para cada dispositivo en el rango 0-30.

Para el presente proyecto las direcciones seleccionadas para cada uno de los osciloscopios es la que se indica en la tabla 4.1

Tabla 4.1 Direcciones de cada uno de los osciloscopios

Mesa (osciloscopio)	2	3	4	5	6
Dirección	9	8	10	11	5

Las direcciones que se deben colocar están entre 0 y 30, los valores indicados arriba no tienen ningún orden específico.

En la caja de diálogo para seleccionar la interfase (en este caso GPIB), se puede cambiar el nombre del Instrumento y se coloca la dirección ($7 \times 100 + 9=709$). El 7 es el código de la tarjeta GPIB, el 9 es la dirección que se fijo en el osciloscopio

4.5 PROGRAMACIÓN

Como ambiente gráfico para la programación de instrumentos se utilizó el programa de Agilent VEE versión 7. VEE es un programa que utiliza menos iconos de programación y que aprovecha las líneas de código de MatLab con análisis numérico, gráficos y procesamiento de señales.

El programa cuenta con 5 pantallas para establecer comunicación entre el operador, la PC y los osciloscopios:

- Pantalla de acceso (ingreso de clave)
- Pantalla de selección (ver toda la red, información , salir)
- Pantalla para visualizar toda la red
- Pantalla para comunicación con cada osciloscopio
- Pantalla de información
- Pantalla de grabación

Pantalla de ingreso: Es la ventana principal que permite ingresar al programa, antes de colocar la clave correspondiente. La figura 4.11 muestra el panel en detalle de la ventana

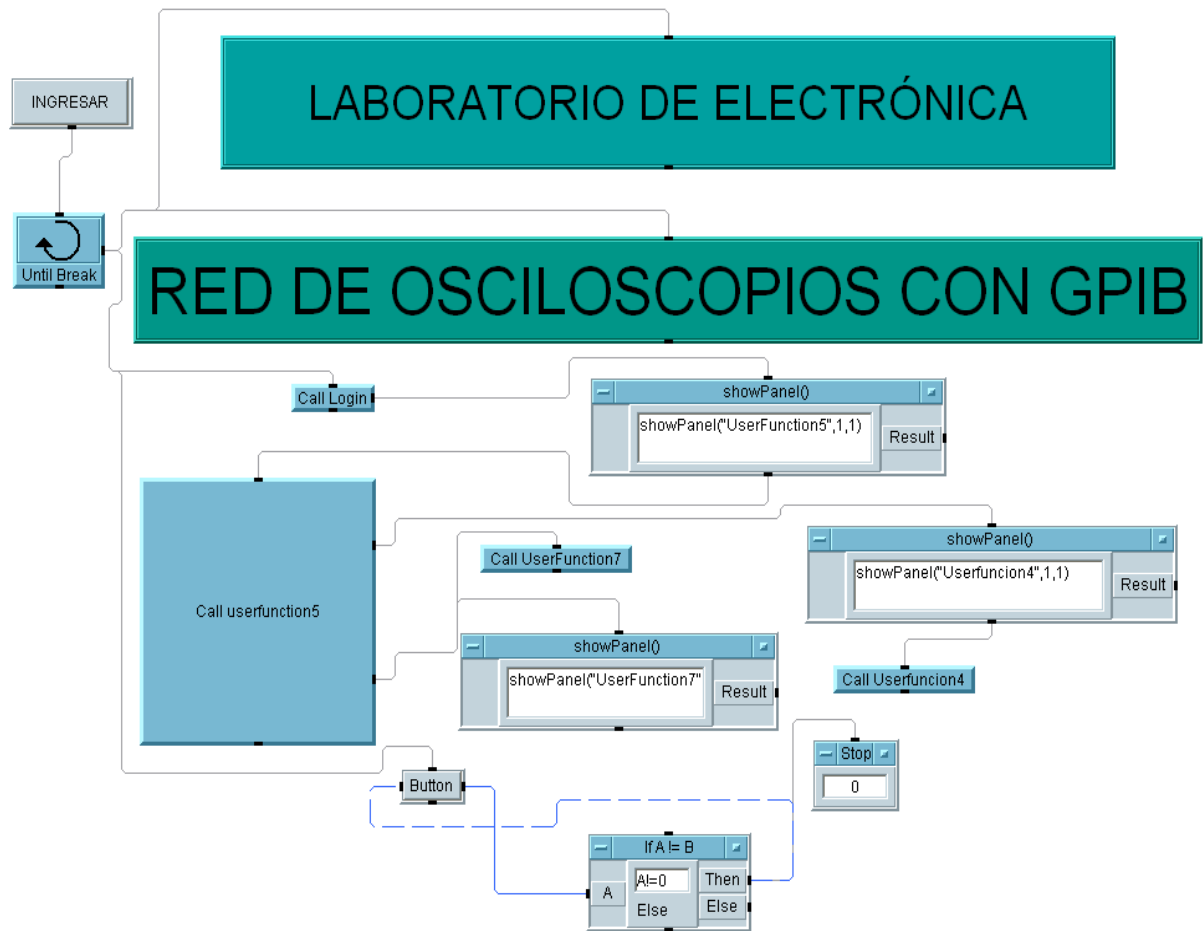


Figura 4.11 Panel de detalles de ventana principal

En esta pantalla se muestran cuatro “UserFunction” . Una “UserFunction” son funciones creadas que realizan una función específica, las cuales pueden ser llamadas en cualquier parte del programa. A continuación se describe la operación de cada una de las funciones creadas:

- Userfunction Login
- Userfunction 4
- Userfunction5
- Userfunction 7

Userfunction Login: A través de esta función, al momento de ejecutar el programa se pide se seleccione una de los tres nombres almacenados. Una vez seleccionado el nombre se pide ingrese la clave correspondiente. En caso de ingresar la clave incorrecta o caracteres no especificados aparece un mensaje

de error para que ingrese nuevamente la clave. La figura 4.12 muestra el programa principal de UserfunctionLogin

La figura 4.13 es un UserObject (subrutina) llamada Login que se ejecuta dentro de UserfunctionLogin y tiene por objeto seleccionar el nombre de la persona, ingresar el código numérico y realizar las comparaciones con los códigos almacenados previamente

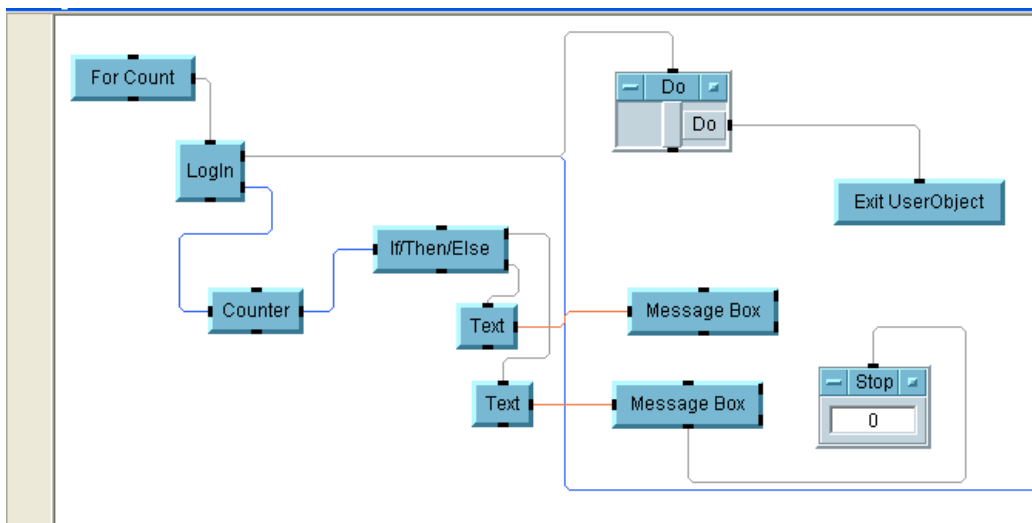


Figura 4.12 Programa principal de UserfunctionLogin

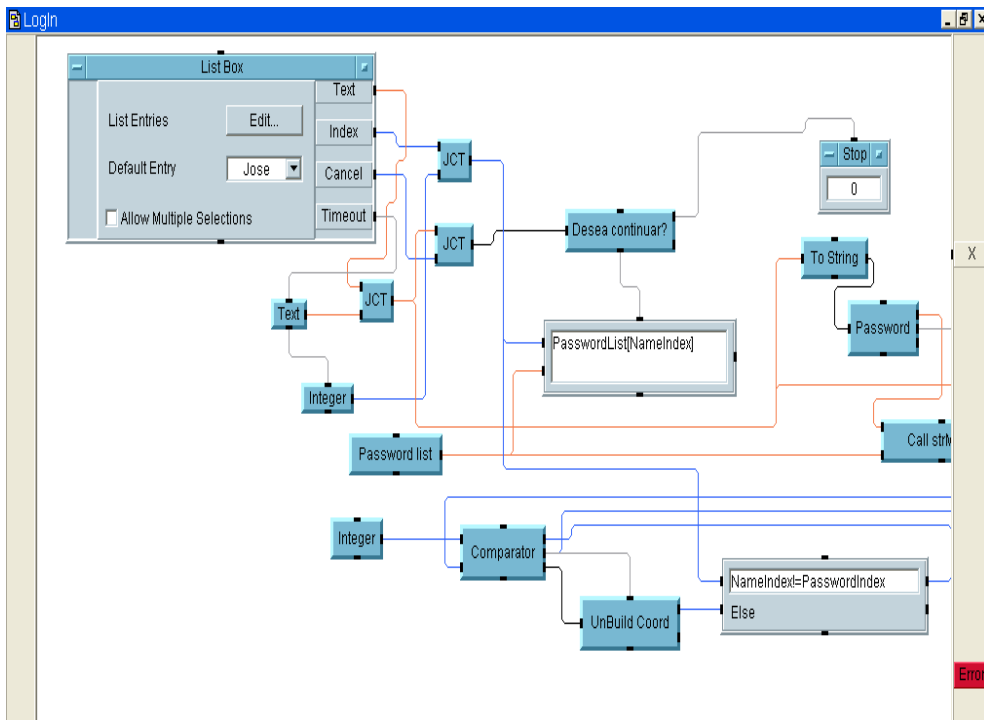


Figura 4.13 Subrutina a ejecutarse dentro de UserfunctionLogin para el ingreso de clave

Pantalla de selección : Creada con UserFunction5 (Figura 4.14): Permite escoger entre las siguientes opciones:

- Seleccionar toda la red: Ingresa a UserFunction4 que muestra la red de los 5 osciloscopios
- Información: Da información sobre las opciones en el manejo de la red y de cada una de las pantallas del osciloscopio
- Salir: Sale de esta función y pide ingrese nuevamente la clave

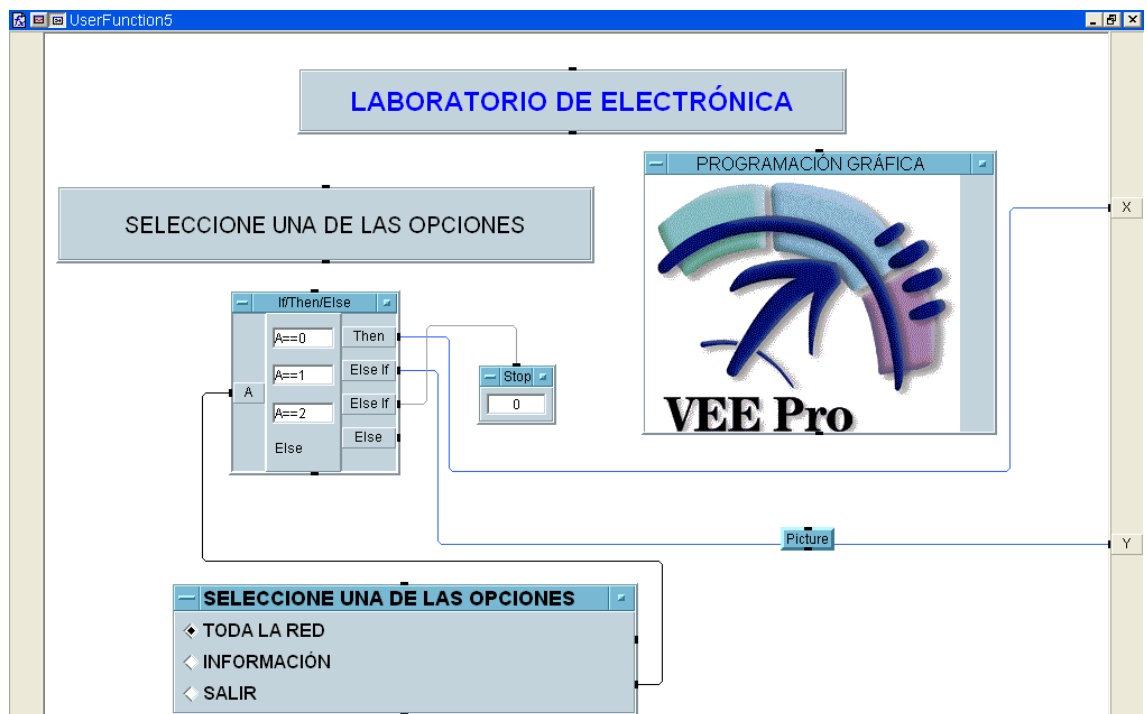


Figura 4.14 Descripción de Userfunction5

Pantalla para visualizar toda la red: Creada a partir de Userfuncion4: Permite observar la red de los 5 osciloscopios, así como ingresar a una cada de las pantallas en donde se podrá:

- Visualizar formas de onda
- Adquirir medidas (Frecuencia, período, Vrms, Vpp, Vmax, Vmin, Vcd, ciclo de trabajo)
- Observar el valor actual de las escalas de amplitud y tiempo.

- Visualizar el canal habilitado
- Enviar comandos hacia el osciloscopio para cambio de escalas, habilitar canal, RUN/STOP, invertir señales, atenuación.
- Grabar formas de onda y valores característicos de las señal

La figura 4.15 muestra los displays de los cinco osciloscopios conectados en la red y la figura 4.16 muestra el panel de detalle de esta Userfuncion.

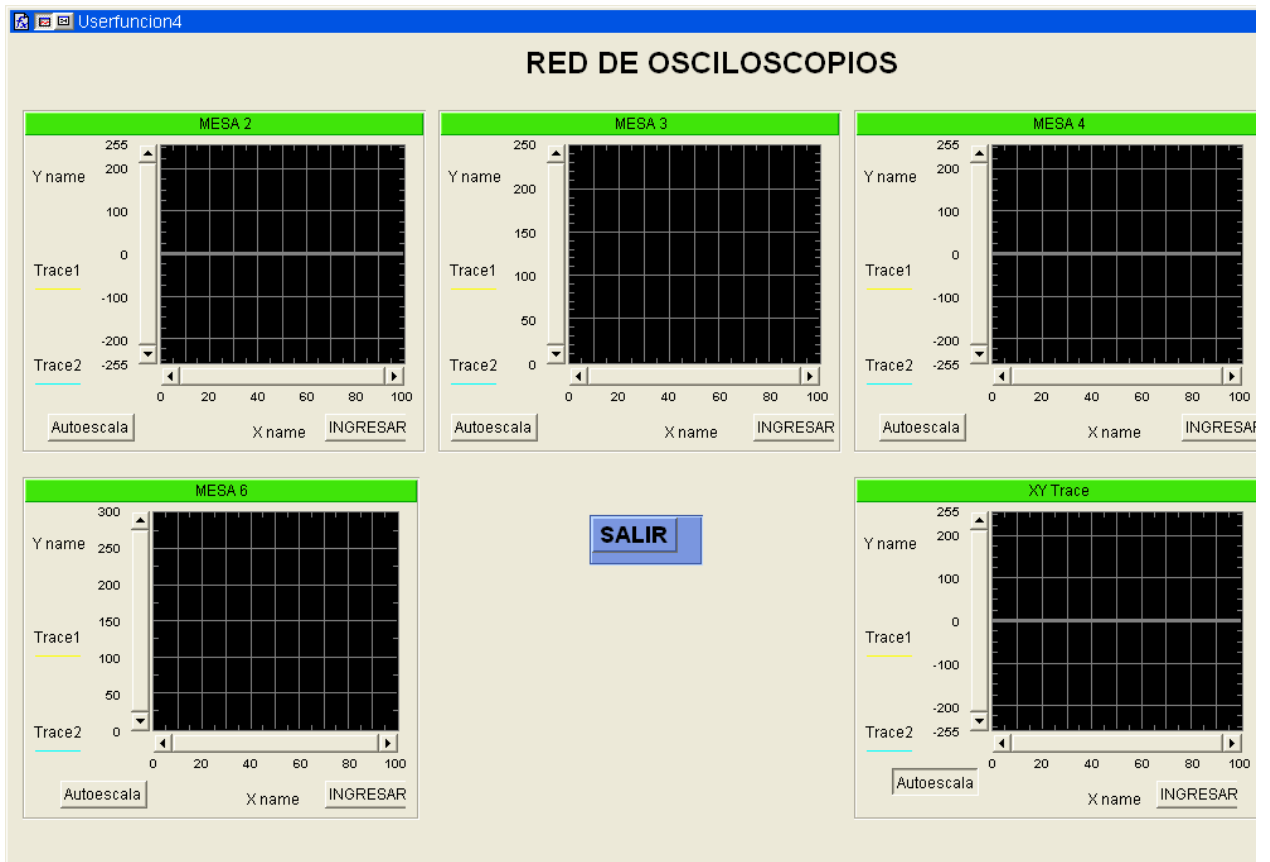


Figura 4.15 Panel frontal de la red de osciloscopios (Userfuncion4)

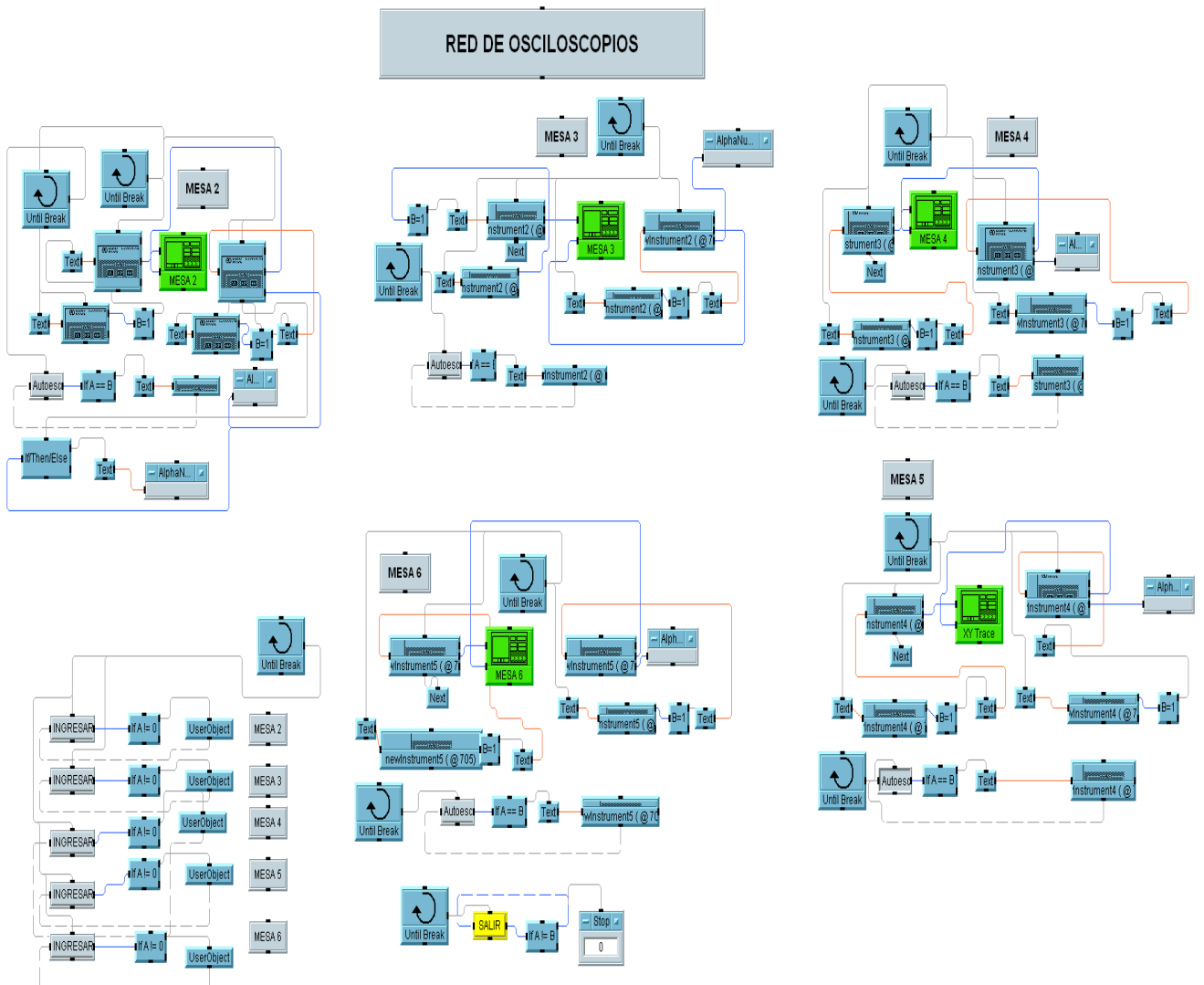


Figura 4.16 Diagrama de conexiones de objetos de Userfuncion4

Pantalla para comunicación con toda la red: Pulsando sobre el botón “ingresar” de cada una de las pantallas de la red mostrada en la figura 4.15 se tiene acceso a la pantalla de la figura 4.17. Esta pantalla permite el manejo de datos desde y hacia el osciloscopio. La figura 4.18 muestra su diagrama de conexiones.

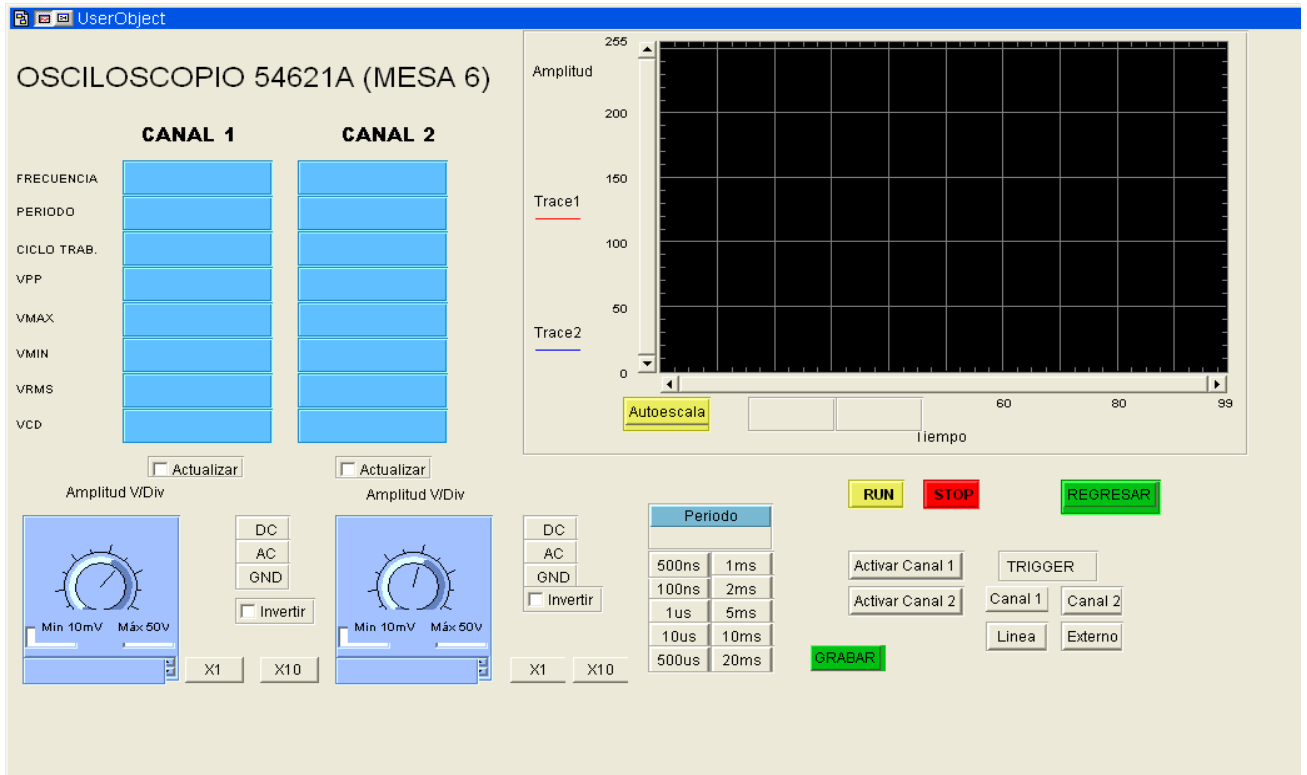


Figura 4.17 Pantalla para enviar y recibir datos desde y hacia el osciloscopio

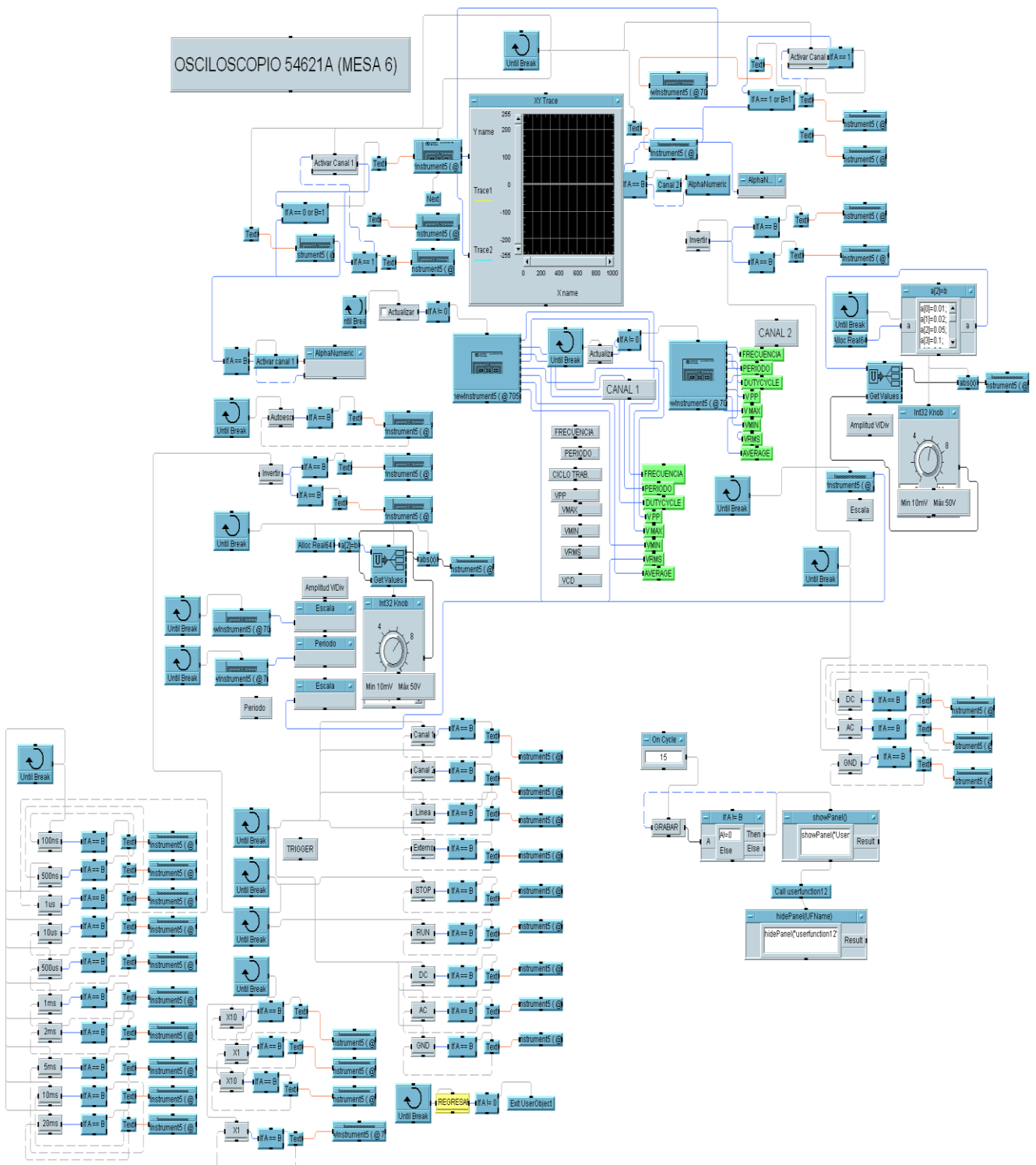


Figura 4.18 Diagrama de conexiones de la pantalla para enviar y recibir datos con el osciloscopio

Pantalla para grabación: Al pulsar sobre el botón grabar que aparece en la pantalla de la figura 4.17 se tiene acceso a la pantalla indicada en la figura 4.19, donde aparecerá al momento de ejecutar el programa una caja de dialogo para ingresar el nombre del archivo con posibilidades de extensión JPG o BMP. El panel frontal y el diagrama de conexiones se indica en las figuras 4.19 y 4.20

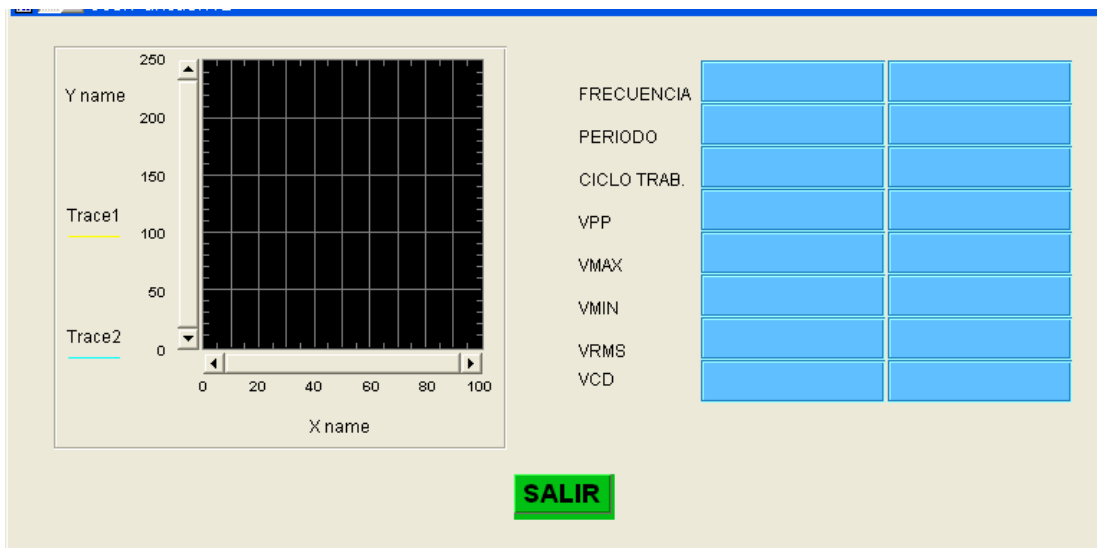


Figura 4.19 Panel frontal para grabación en JPG o BMP

OSCILOSCOPIO 54621A (MESA 6)

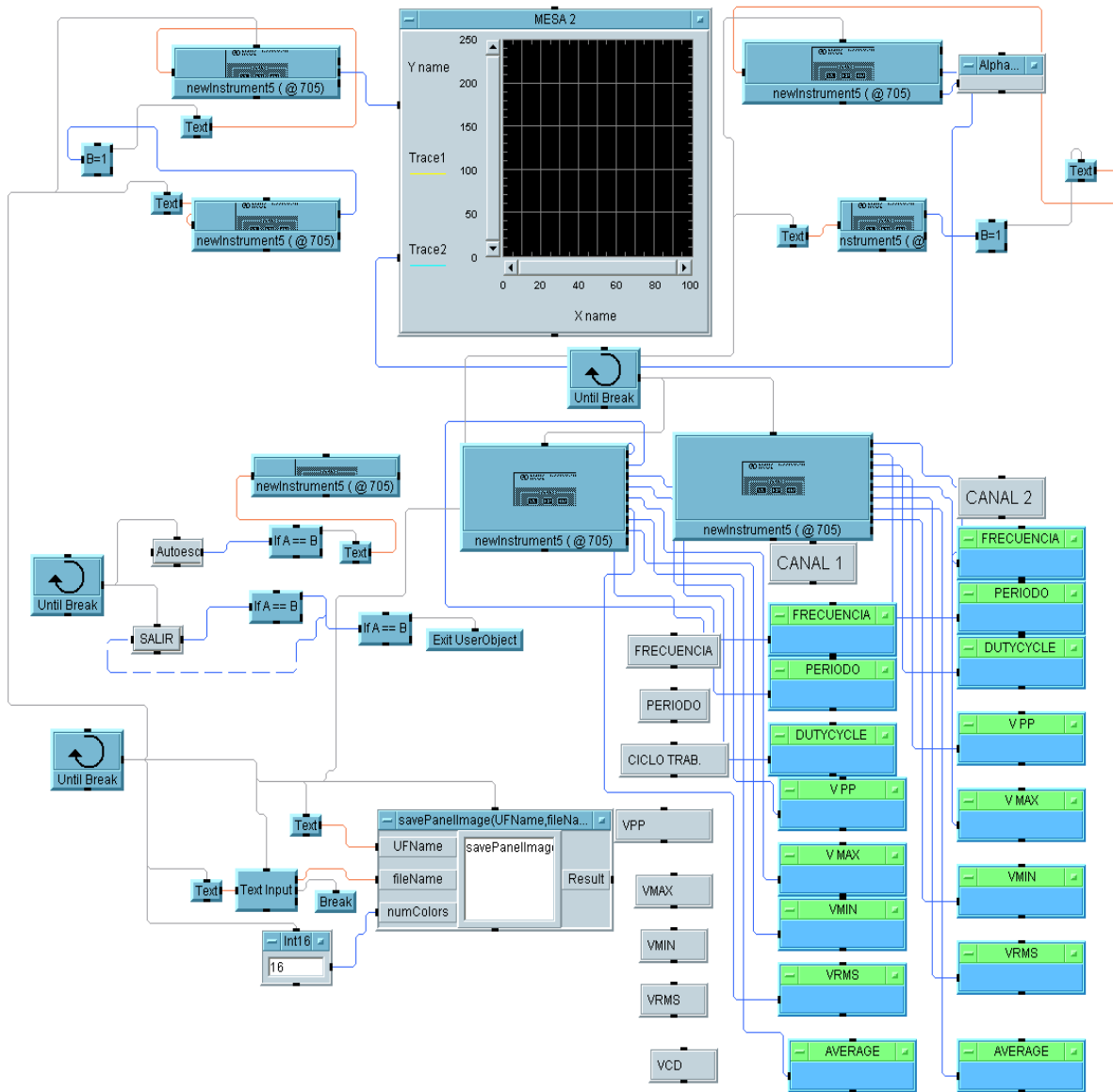


Figura 4.20 Panel de detalle para grabación en JPG o BMP

Pantalla de Información: Creada a partir de Userfunction7: Permite acceder a la información de la red

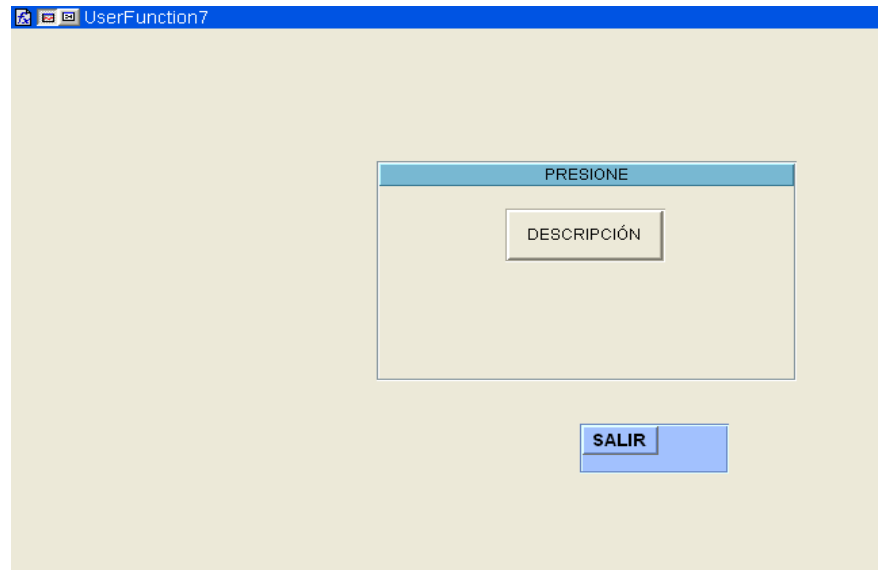


Figura 4.21 Panel para Userfunction7

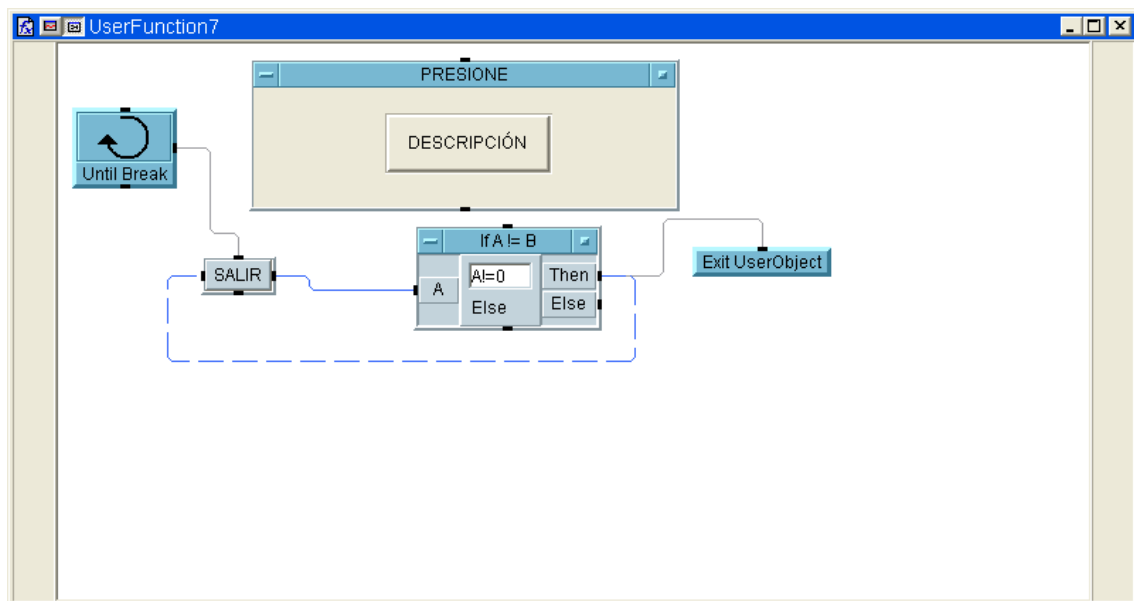


Figura 4.22 Panel de detalles de UserFunction7

Para tomar las medidas y formas de onda tanto del canal 1 y 2, se utilizan los comandos que se escriben en el objeto I/O creado a partir de:

I/O de la barra de herramientas de VEE , luego Instrument Manager y por último Create I/O objeto. Los comandos escritos para pedir datos corresponden a los especificados para el osciloscopio Agilent 54621 A . En el Anexo A se indican los principales comandos del osciloscopio.

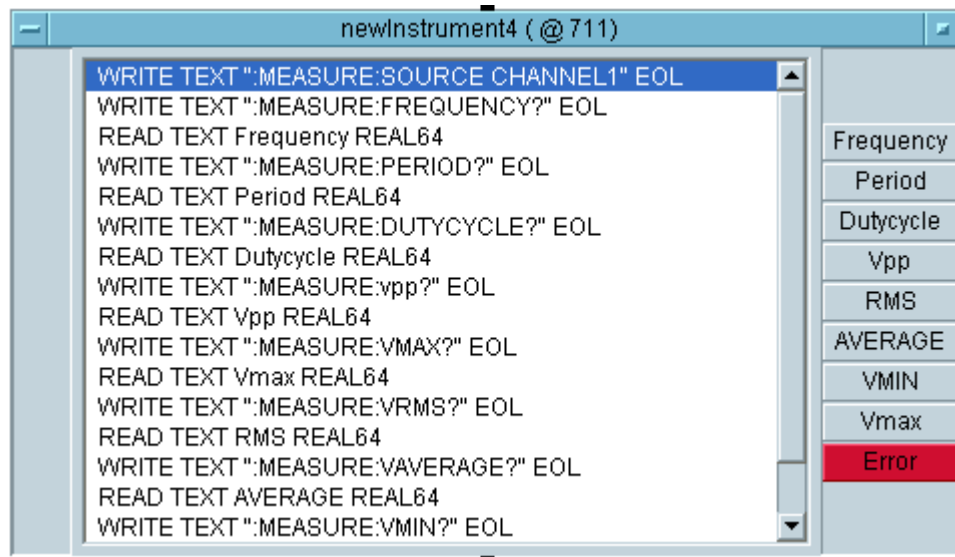


Figura 4.23 Caja de diálogo de direct I/O para adquirir datos de la señal del canal 1

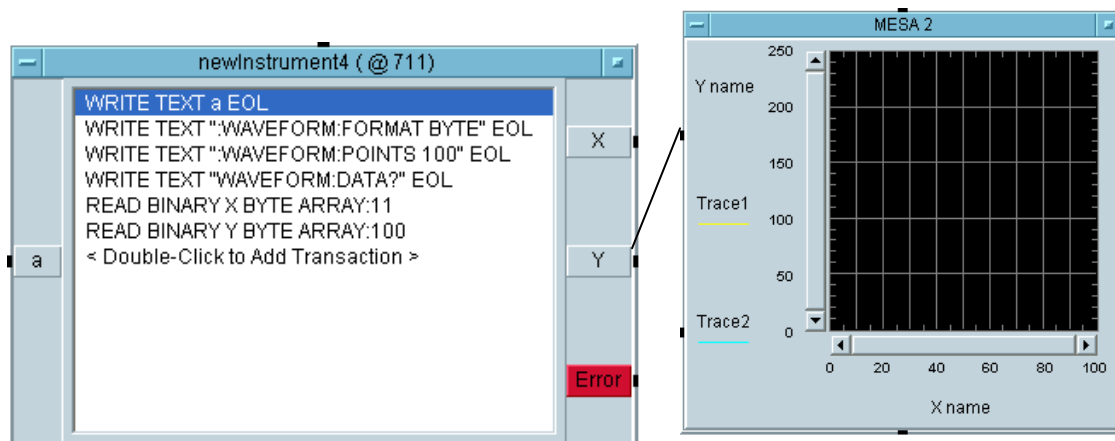


Figura 4.24 Caja de diálogo de direct I/O para adquirir la forma de onda del canal 1

4.6 PRUEBAS DE OPERACIÓN

- En varias ocasiones el sistema fue sometido a pruebas experimentales de laboratorio comprobándose la coherencia de los resultados y la comunicación entre la PC y los instrumentos conectados a la red utilizando la interfase GPIB
- Se creó una versión de seguridad del programa a través de Create Run Time Versión
- Se probó el funcionamiento de la red conectando los 5 osciloscopios que dispone el laboratorio, se aplicaron señales a ambos canales y se observó la forma de onda y los valores de los parámetros de ambas señales tanto en la PC como en el osciloscopio.
- Se probó la respuesta del osciloscopio a los comandos enviados desde la PC vía GPIB, así como el monitoreo en las ventanas creadas de la operación del osciloscopio
- Se realizaron pruebas de grabación de las señales y valores de las formas de onda para cada uno de los osciloscopios tanto el formato JPG o BMP.

4.7 RESULTADOS OBTENIDOS

Luego de varias pruebas de funcionamiento de la red y para confirmar su desempeño se realizaron comparaciones entre señales observadas en el osciloscopio y las obtenidas a través del lenguaje de programación gráfico VEE en las ventanas creadas, así como el monitoreo y control del osciloscopio a través de la PC, obteniéndose los siguientes resultados

Las formas de ondas adquiridas de los 5 osciloscopios se muestran en las ventanas creadas en forma similar y responden a variaciones de las escalas del osciloscopio y variaciones de la señal de entrada con un mínimo tiempo de retardo.

Se pudo observar que al variar varios parámetros manualmente en el osciloscopio, estas variaciones también se muestran en la pantalla de la PC, así mismo se pudo comprobar que al modificar los controles desde la PC el osciloscopio responde en forma rápida a estos cambios.

Al seleccionar la opción de grabación se muestra primeramente la forma de onda adquirida y los valores de los parámetros más importantes de la señal de los dos canales. A través de una caja de diálogo se pide ingresar la dirección el nombre y la extensión JPG o BMP del archivo a grabar. Luego se procedió a abrir el archivo guardado y se comprobó su almacenamiento correcto.

Se comprobó que al realizar las mediciones de los parámetros de las señales aplicada a ambos canales, la comunicación se vuelve mas lenta, esto se debe a que en la programación se ejecutan subrutinas, llamada a funciones para envío de los comandos y recepción de información desde y hacia el osciloscopio, lo que provocan retardos.

En el caso de que algún osciloscopio se apague, el sistema sigue mostrando las formas de onda de los instrumentos restantes, volviendo a adquirir señal automáticamente el momento que se vuelva a encender el osciloscopio

En forma general se cumplen en forma satisfactoria el funcionamiento de la red así como la adquisición de datos y control del osciloscopio.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Al finalizar este proyecto me siento conforme de haber aportado con algo útil para el laboratorio de Electrónica de nuestra Carrera, y que puede ser utilizado por los señores docentes y alumnos, así como la disponibilidad de utilizar el programa, la tarjeta y la información presentada para posteriores aplicaciones en el campo de la comunicación industrial.

Presento las siguientes conclusiones:

- Se cumplió con el objetivo planteado en el perfil del proyecto que consistía en “Analizar, Diseñar e implementar una red de comunicación de instrumentos utilizando la interfase GPIB”.
- Se ha analizado y comprendido el principio de funcionamiento y operación de la tarjeta de interfaz GPIB. Las tarjetas GPIB son controladores que gobiernan el flujo de información a través de la red mediante el envío de comandos a los dispositivos para que hablen, escuchen y atiendan pedidos de servicio de los dispositivos
- El bus de transmisión paralela GPIB ofrece un velocidad de respuesta mucho mayor que otras interfaces, sin embargo su limitante es el número pequeño de equipos que pueden conectarse en red y la corta distancia tanto entre equipos como total.
- Es necesario instalar y configurar el software relacionado con la operación del sistema. En este proyecto se ha instalado y configurado la tarjeta de interfaz HP 82350A a través de los soportes de software como son las librerías VISA y SICL en el sistema operativo XP .
- Mediante este sistema se puede enviar datos binarios o cadena de caracteres desde el instrumento hacia la PC vía GPIB, y estos serán mostrados en las

pantallas creadas a través del lenguaje de programación gráfico HP VEE PRO. Así mismo atender peticiones de instrumentos es decir en el momento que se opere algún cambio en el osciloscopio este debe ser enviado a través de la red para mostrarlo en la pantalla.

- La operación se debe a comandos que son enviados desde el programa HP VEE a través de la tarjeta y cable GPIB hacia el osciloscopio. Estos comandos se seleccionan del manual de programación del propio osciloscopio
- Desde las pantallas creadas se puede operar en forma remota varios controles que tiene el osciloscopio para su funcionamiento y adquisición de señales, con tiempos de retardo despreciables
- La red consta de una computadora utilizada como controlador que monitorea y comanda 5 osciloscopios digitales que dispone en la actualidad el laboratorio de Electrónica utilizando el estándar GPIB (488) y como lenguaje de programación gráfico HP VEE, de Hewlett Packard versión 7
- HP VEE es un lenguaje de programación que ofrece un completo ambiente gráfico, además permite crear interfaces de usuario, controlar la comunicación con los diferentes puertos, flexibilidad y control de instrumentos con GPIB, VXI, serial
- Es necesario que la red disponga de una máster, en este caso una computadora donde se instala la tarjeta y que será el controlador, el cual se encarga de gestionar el bus, enviar órdenes, solicitar el estado de los dispositivos así como controlar el flujo de datos. Para ordenarle una acción a un dispositivo, el controlador debe colocar primeramente en el bus la dirección del mismo.
- Se debe armar en red los osciloscopios antes de ejecutar el programa, ya que una de las características de la tarjeta GPIB es que deben estar conectados mínimo los 2/3 del total de instrumentos direccionados.
- Existen otras posibilidades de implementar la red de instrumentos esto es utilizando la interfase RS-232, VXI, LAN y como lenguajes de programación gráfico, Lab View, VEE, Matlab.

- A la red no solo se pueden conectar osciloscopios sino también cualquier instrumento que tengan entrada GPIB tal como generadores, fuentes de voltaje etc.
- La transacción para adquirir las formas de onda se la debe hacer en el formato de Byte esto permite que la señal sea mostrada tal y como se la ve en la pantalla del osciloscopio.
- El presente proyecto permitirá al docente que realiza prácticas monitorear la adquisición de la señales en cada una de las mesas de trabajo, así como a controlar la mayoría de opciones del osciloscopio de una manera fácil por medio de una interfase de usuario creada en el lenguaje de programación grafica VEE.

5.2 RECOMENDACIONES

- Al seleccionar una tarjeta GPIB observe que tipo de bus dispone su computadora, ya que algunas casas comerciales elaboran varias tarjetas de interfaz GPIB (IEEE 488) insertable para los buses de computadoras tradicionales como: PCI, PCMCIA , USB
- El fabricante de estas tarjetas exige que mínimo los 2/3 de los equipos que están en la red estén encendidos. Se recomienda para tener una buena velocidad de respuesta encender todos los equipos.
- Si desea observar el valor de las medidas de las señales ingresadas a través de los dos canales, seleccionar actualizar en la ventana que muestra un osciloscopio, y luego deshabilitar esta opción esto permitirá una mejor respuesta al adquirir y visualizar la forma de onda
- Se recomienda cumplir con las especificaciones en cuanto al promedio de la longitud del cable que debe ser de 2 metros y la longitud total de no mayor a 20 metros.
- La transmisión de información especialmente para la adquisición de formas de onda se lo debe hacer en el formato Byte y no en ASCII. En byte se transmite el valor de la señal en forma mucho más rápida y permite visualizar en el graficador de la ventana creada en forma exacta como se la

observa en el osciloscopio. El formato ASCII requiere de conversiones de caracteres que provocan retardo en la visualización.

- Disponiendo de computadoras en cada una de las mesas de trabajo del laboratorio e implementado una red LAN con ello se puede lograr que varios alumnos puedan acceder a los instrumentos y compartir los recursos del sistema, pueden implementarse además prácticas de desarrollo enfocadas a evaluar y analizar el bus GPIB.
- Disponiendo de generadores con entrada GPIB, pueden conectarse a este transductores como emisores y al osciloscopio transductores receptores de tal forma que por medio de la tarjeta GPIB y de un lenguaje de programación gráfico se puedan realizar cálculos sobre la onda capturada tal como densidad espectral, distancia del objeto, energía, etc
- Se recomienda a la carrera preocuparse por la adquisición de nuevas interfases de comunicación y lenguajes de programación gráfico para que puedan realizarse prácticas de algunas asignaturas de especialidad, que son de mucha importancia en el ambiente industrial.
- Se recomienda a la ESPEL que se trate de hacer convenios con empresas para que alumnos y profesores puedan realizar pasantías, lo que conllevará a un mejor desempeño en el campo profesional

REFERENCIAS BIBLIOGRÁFICAS

- Mackay, Park, Wright: Data Communication for instrumentation and Control. Techbooks. Sydney Australia. 2000.
- Horak Ray: Communication Systems and networks: MT Books, New York, 1999.
- Hallsall Fred: Data Communications, Computer Networks and Open Systems.
- Manuales del a Tarjeta 82350 PCI
- Manuales del Osciloscopio Agilent 54621 A

ANEXO A

COMANDOS DEL OSCILOSCOPIO AGILENT 54621 A

Comandos	Preguntas	Opciones y respuesta
:ACQUIRE:COMPLETE <complete>	:ACQUIRE:COMPLETE?	<completion_criteria> ::= 100; an integer in NR1 format
:ACQUIRE:COUNT <count>	:ACQUIRE:COUNT?	<count_argument> ::= an integer from 1 to 16384 in NR1 format
n/a	:ACQUIRE:POINTS?	2,000; an integer in NR1 format
n/a	:ACQUIRE:SRATE?	<point_argument> ::= simple rate (simples/s) in NR3 format
:ACQUIRE:TYPE <type>	:ACQUIRE:TYPE?	<type> ::= {NORMAL AVERAGE PEAK}
n/a	:AER?	{0 1}; an integer in NR1 format.
n/a	:CALIBRATE:DATE?	<date> ::= day, month, year in NR1 format
n/a	:CALIBRATE:DATE?	<return value> ::= <day>,<month>,<year>; all in format NR1
:CALIBRATE:LABEL <string>	:CALIBRATE:LABEL?	<string> ::= quoted ASCII string up to 32 characters
n/a	:CALIBRATE:SWITCH?	{PROTECTED UNPROTECTED}
n/a	:CALIBRATE:TIME?	<return value> ::= <hours>,<minutes>,<seconds>;all in NR1 format
:CDISPLAY	n/a	n/a
:CHANNEL<n>:BWLIMIT <bwlimit> { {1 ON} {0 OFF} }	:CHANNEL<n>:BWLIMIT?	{0 1};<n> ::= 1-2 or 1-4; an integer in NR1 format
:CHANNEL<n>:COUPLING<coupling>	:CHANNEL<n>:COUPLING?	<coupling value> ::= {AC DC GND} <n> ::= 1-2 or 1-4; an integer in NR1 format
:CHANNEL<n>:DISPLAY <display value>{ {1 ON} {0 OFF} }	:CHANNEL<n>:DISPLAY?	{0 1};<n> ::= 1-2 or 1-4; an integer in NR1 format
:CHANNEL<n>:IMPEDENCE <impedence>	:CHANNEL<n>:IMPEDENCE?	<impedence> ::= {ONEMEG};<n> ::= 1-2 or 1-4; an integer in NR1 format
:CHANNEL<n>:INVERT { {0 OFF} {1 ON} }	:CHANNEL<n>:INVERT?	{0 1};<n> ::= 1-2 or 1-4; an integer in NR1 format
:CHANNEL<n>:LABEL <string>	:CHANNEL<n>:LABEL?	<string> ::= any series of 6 or less ASCII characters enclosed in quotation marks <n> ::= 1-2 or 1-4; an integer in NR1 format
:CHANNEL<n>:OFFSET <offset> [suffix]	:CHANNEL<n>:OFFSET?	<offset> ::= Vertical offset value in NR3 format. [suffix] ::= {V mV} <n> ::= 1-2 or 1-4; an integer in NR1 format

Comandos	Preguntas	Opciones y respuesta
:CHANNEL<n>:PROBE <attenuation>	:CHANNEL<n>:PROBE?	<attenuation> ::= Probe attenuation ratio in NR3 format ::= X1,X10,X20,X100 (obsolete form) <n> ::= 1-2 or 1-4; an integer in NR1 format
:CHANNEL<n>:RANGE<range> [suffix]	:CHANNEL<n>:RANGE?	<range> ::= Vertical full-scale range value in NR3 format. [suffix] ::= {V mV} <n> ::= 1-2 or 1-4; an integer in NR1 format
:CHANNEL<n>:SCALE<scale>[suffix]	:CHANNEL<n>:SCALE?	<scale> ::= Vertical units per division value in NR3 format. [suffix] ::= {V mV} <n> ::= 1-2 or 1-4; an integer in NR1 format
*CLS	n/a	n/a
:DIGITAL<n>:DISPLAY <display> { {1 ON} {0 OFF} }	:DIGITAL<n>:DISPLAY?	{0 1};<n> ::= 0-15; an integer in NR1 format
:DIGITAL<n>:LABEL <label string>	:DIGITAL<n>:LABEL?	<string> ::= any series of 6 or less ASCII characters enclosed in Quotation marks <n> ::= 0-15; an integer in NR1 format
:DIGITAL<n>:POSITION <position>	:DIGITAL<n>:POSITION	<n> ::= 0-15; an integer in NR1 format <position> ::= 1-8 if display size = large, 1-6 if size = medium, 1-32 if size = small
:DIGITAL<n>:THRESHOLD <threshold value>	:DIGITAL<n>:THRESHOLD?	<n> ::= 0-15; an integer in NR1 format <value> ::= {CMOS ECL TTL <user defined value>} <user defined value > ::= value in NR3 format from -8.00 to + 8.00 [suffix] ::= {V mV uV}

:DIGitize [<source>[,...,<source>]]	n/a	<source> ::= {CHANnel<n> POD1 POD2 FUNCTION}for 5462xA <source> can be repeated up to 5 times. <n> ::= 1-2 or 1-4; an integer in NR1 format
:DISPlay:CLEar	n/a:	n/a
:DISPlay:DATA[format][,][area] <binary block data>	:DISPlay:DATA?[format][,][area]	<format> ::= {TIFF}(command only) <area> ::= {GRATicule}(command only) <format> ::= {TIFF BMP}(query only) <area> ::= {GRATicule SCReen}(query only) <binary block_data> ::= data in IEEE 488.2# format

Comandos	Preguntas	Opciones y respuesta
:DISPlay:LABel {{0 OFF} 1 ON}}	:DISPlay:LABel?	{0 1}
:DISPlay:LABList#80000524 <binary block>	:DISPlay:LABList?	<binary block> ::= a time-ordered list of 75 labels. Each label can maximum of 6 characters followed by a comma.
:DISPlay:PERsistence <value>	:DISPlay:PERsistence?	<value> ::= {MINimum INFinite}
:DISPlay:SOURce <value>	:DISPlay:SOURce?	<value> ::= {PMEMory{0 1 2}}
:DISPlay:VECTors{{1 ON} {0 OFF}}	:DISPlay:VECTors?	{1 0}
*DMC<macro label>,<macro Definition>	n/a	<macro label> ::=quoted ASCII string <macro definition > ::= block data in IEEE488.2 #format
*EMC {{ 0 OFF} {1 ON}}	*EMC?	{0 1}
*ESE < mask>	*ESE?	< mask> ::= 0 a 255; an integer in NR1 format Bit Weight Enables 7128 PON – Power ON 664 URQ – User Request 532 CME – Command Error 416 EXE – Execution Error 38 DDE – Device Dependent Error 24 QYE – Query Error 12 TRG – Trigger Query 01 OPC – Operation complete
n/a	*ESR?	< status > ::= 0 a 255; un entero en formato NR1
:FUNction: CENTer <frequency>	:FUNction: CENTer?	<frequency> ::=the current center frequency in NR3 format. El rango de valores legal es desde 0 a 25GHz
:FUNction:DISPlay{{0 OFF} 1 ON}}	:FUNction:DISPlay?	{0 1}
:FUNction: OFFSet < offset>	:FUNction: OFFSet?	<offset> ::=the value at center screen in NR3 format The range of legal values is generally +/-10 times the current scale selected function.
:FUNction: OPERation < operation>	:FUNction:OPERation?	<operation> ::= {SUBTract MULTiply INTegrate DIFFerentiate FFT}

Comandos	Preguntas	Opciones y respuesta
:FUNction:RANGe<range>	:FUNction: RANGe?	<range> ::=eje vertical en full-scale evaluada en formato NR3 El rango para ADD,SUBT,MULT es 8E-6 a 800E+3 El rango para la función INTgrate es 8E-9 a 400+3 El rango para la función DIFFerentiate es 80E-3 a 8.0 E12 (depends on current sweep speed) The range for the FFT function is 8 a 800 dBV.
:FUNction:REFerence <level>	:FUNction:REFerence?	<level> ::= the current reference level in NR3 format. The range of legal values is from -400.0 dBV to +400.0 dBV (depending on the current range value)
:FUNction:SCALe <scale	:FUNction:SCALe	<scale value> ::= integer in NR1 format

value>[<suffix>]		<suffix> ::= {V dB}
:FUNction:SOURce <source>	:FUNction:SOURce?	<source> ::= {CHANnel<n> ADD SUBT MULT};<n> ::= 1-2 1-4 in NR1 format
:FUNction:SPAN 	:FUNction:SPAN?	 ::= the current frequency span in NR3 format. Legal values are 1 Hz to 100 GHz.
n/a	*GMC? <macro label>	<macro label>::=quoted ASCII string, block data in IEEE 488.2#
:HARDcopy:DESTination <destination>	:HARDcopy:DESTination?	<destination> ::= {CENTronics FLOppy}
:HARDcopy:FACTors { {0 OFF} {1 ON} }	:HARDcopy:FACTors?	{0 1}
:HARDcopy:FFEed { {0 OFF} {1 ON} }	:HARDcopy:FFEed?	{0 1}
:HARDcopy:FORMat <device>	:HARDcopy:FORMat?	<format> ::= {TIFF BMP CSV LASerjet DESKjet EPSON S
:HARDcopy:GRAYscale { {0 OFF} {1 ON} }	:HARDcopy:GRAYscale?	{0 1}
n/a	*IDN	AGILENT TECHNOLOGIES,<model>, serial number>,X.XX.XX <model>:: the model number of the instrument <serial number>::=the serial number of the instrument <X.XX.XX>::= the software revision of the instrument
n/a	*LMC?	<ASCII string>::=string list separated by commas

Comandos	Preguntas	Opciones y respuesta
n/a	*LRN?	<learn_string>::=current instrument setup as a block of data in IEEE 488.2 # format
:MARKer:MODE <mode>	:MARKer:MODE?	<mode> ::= {OFF MEASure MANual}
:MARKer:X1Position <position> [suffix]	:MARKer:X1Position?	<position> ::= X1 cursor position in NR3 format [suffix]::= {s ms us ns ps Hz kHz MHz} <return_value> ::= X1 cursor position in NR3 format
:MARKer:X1Y1source <source>	:MARKer:X1Y1 <source>	<source> ::= {CHANnel<n> FUNction MATH} <n> ::= 1-2 0 1-4 en formato NR1 <return_value>::=<source>
:MARKer:X2Position <position> [suffix]	:MARKer:X2Position?	<position> ::= X2 cursor position value in NR3 format <suffix> ::= {s ms us ns ps Hz kHz MHz} <return_value> ::= X2 cursor position value in NR3 format
:MARKer:X2Y2source <source>	:MARKer:X2Y2<source>	<source> ::= {CHANnel<n> FUNction MATH} <n> ::= 1-2 o 1-4 in NR1 format <return_value>::=<source>
n/a	:MARKer:XDELta?	<return_value>::= X cursors delta value in NR3 format
:MARKer:Y1Position <position> [suffix]	:MARKer:Y1Position?	<position> ::= Y1 cursor position value in NR3 format <suffix> ::= {mV V dB} <return_value>::=Y1 cursor position value in NR3 format
:MARKer:Y2Position <position> [suffix]	:MARKer:Y2Position?	<position> ::= Y2 cursor position value in NR3 format <suffix> ::= {mV V dB} <return_value>::=Y2 cursor position value in NR3 format
n/a	:MARKer:YDELta?	<return_value>::= Y cursors delta value in NR3 format
MEASure:CLEar	n/a	n/a
:MEASure:DELay[<source>]	:MEASure:DELay? [<source>]	<source> ::= {CHANnel 1 CHANnel 2 } <return_value>::=floating- point number delay time in seconds in
:MEASure:DUTYcycle [<source>]	:MEASure:DUTYcycle? [<source>]	<source> ::= {CHANnel<n> FUNction MATH} for 5462xA <n> ::= 1-2 o 1-4 in NR1 format <return_value>::= ratio of positive pulse width to period in NR3

Comandos	Preguntas	Opciones y respuesta
:MEASure:FALLtime [<source>]	:MEASure:FALLtime? [<source>]	<source> ::= {CHANnel<n> FUNcTion MATH} <value> ::= time in seconds between the lower threshold and upper threshold in NR3 format
:MEASure:FREQuency [<source>]	:MEASure:FREQuency? [<source>]	<source> ::= {<digital channels> CHANnel<n> FUNcTion MATH} <source> ::= frequency in Hertz in NR3 format
:MEASure:PERiod [<source>]	:MEASure:PERiod? [<source>]	<source> ::= {<digital channels> CHANnel<n> FUNcTion MATH} <value> ::= waveform period in seconds in NR3 format
:MEASure:RISetime [<source>]	:MEASure:RISetime? [<source>]	<source> ::= {CHANnel<n> FUNcTion MATH} <value> ::= rise time in seconds in NR3 format
:MEASure:SOURce <source>	:MEASure:SOURce?	<source> ::= {<digital channels> CHANnel<n> FUNcTion MATH} <source> ::= {<digital channels> CHAN<n> FUNC NONE}
:MEASure:TEDGe	:MEASure:TEDGe? <slope><occurrence>[,<source>]	<slope> ::= direction of the waveform. <value> ::= time in seconds of the specified transition
MEASure:VAMPlitude [<source>]	:MEASure:VAMPlitude? [<source>]	<source> ::= {CHANnel<n> FUNcTion MATH} <value> ::= the amplitude of the selected waveform in NR3 format
:MEASure:VMAX [<source>]	:MEASure:VMAX? [<source>]	<source> ::= {CHANnel<n> FUNcTion MATH} <value> ::= maximum vertical value of the selected waveform in NR3 format
:MEASure:VMIN [<source>]	:MEASure:VMIN? [<source>]	<source> ::= {CHANnel<n> FUNcTion MATH} <value> ::= minimum vertical value of the selected waveform in NR3 format
:MEASure:VPP [<source>]	:MEASure:VPP? [<source>]	<source> ::= {CHANnel<n> FUNcTion MATH} <value> ::= vertical peak to peak value in NR3 format
:MEASure:VRMS [<source>]	:MEASure:VRMS? [<source>]	<source> ::= {CHANnel<n> FUNcTion MATH} <value> ::= calculated dc RMS value in NR3 format
:MEASure:XMAX [<source>]	:MEASure:XMAX? [<source>]	<source> ::= {CHANnel<n> FUNcTion1 FUNcTion2} <value> ::= horizontal value of the maximum in NR3 format
*OPC	*OPC?	ASCII "1" in the output queue when all pending device operations are completed

Comandos	Preguntas	Opciones y respuesta
*RST	n/a	The *RST command places the instrument in a known state
:RUN	n/a	n/a
*SRE <mask>	*SRE?	Sets the bits in the Service Request Enable Register. T <mask> ::= sum of all bits that are set, 0,...,255; an integer in NR1 format
n/a	*STB?	<value> ::= 0,...,255; an integer in NR1 format
:SYSTem:DATE <date>	:SYSTem:DATE?	<date> ::= <year>, <month>, <day>
:TIMebase:MODE <value>	:TIMebase:MODE?	<value> ::= {MAIN WINDow XY ROLL}
:TIMebase:RANGe <range_value>	:TIMebase:RANGe?	range_value ::= 50 ns through 500 s
:TIMebase:SCALE <scale_value>	:TIMebase:SCALE?	<scale_value> ::= scale value in seconds
:TRIGger:MODE<mode>	:TRIGger:MODE?	<mode> ::= {EDGE GLITch PATTern TV SEQuence DURation} <mode> ::= { NONE EDGE GLIT PATT TV SEQ DUR}
:TRIGger:[EDGE:]COUPling <coupling>	:TRIGger:[EDGE:]COUPling?	<coupling> ::= {AC DC LFRject}
:TRIGger:[EDGE:]SOURce [<source>]	:TRIGger:[EDGE:]SOURce?	<source> ::= {CHANnel<n> EXTernal LINE} for the 54621A
:WAVeform:BYTeorder <value>	:WAVeform:BYTeorder?	<value> ::= {LSBFirst MSBFirst}

na	:WAVeform:COUNT?	<count_argument> ::= an integer from 1 to 16384 in <NR1> for
n/a	:WAVeform:DATA?	<binary block length bytes>, <binary data> <NL>
:WAVeform:FORMat <value>	:WAVeform:FORMat?	<value> ::= {WORD BYTE ASC}
:WAVeform:POINts <# points>	:WAVeform:POINts? [MAXimum]	<# points> ::= {100 250 500 1000 2000 MAXimum}
:WAVeform:SOURce <source>	:WAVeform:SOURce?	<source> ::= {CHANnel<n> POD1 POD2 FUNCtion MAT
n/a	:WAVeform:TYPE?	<mode> ::= {NORM PEAK AVER}
:WAVeform:VIEW <view>	:WAVeform:VIEW?	<view> ::= {NORMal}

ANEXO B

GLOSARIO DE TÉRMINOS

ANSI: Siglas del Instituto de Estandarización Nacional Americano. Fundado en 1918, ANSI es una organización voluntaria compuesta de más de 1,300 miembros (incluyendo todas las compañías de las computadora grandes) que crean normas para la industria de la computadora. Por ejemplo, ANSI C es una versión del idioma de C que ha sido aceptado por el comité de ANSI.

BUS: Una colección de alambres a través de los cuales los datos se transmiten de una parte de una computadora a otra. Se puede pensar en un bus como una carretera en la que el dato viaja dentro de una computadora. Cuando se usa en referencia a las computadoras personales, el término bus normalmente se refiere al bus interior. Éste es un bus que conecta todos los componentes interiores de la computadora al CPU y la memoria principal.

CCITT: Abreviación de Comité Consultivo Internacional para telefonía y Telegrafía (Comité Consultive Interacional for Telephone and Telegraph), una organización que establece normas de comunicaciones internacionales. CCITT, ahora conocido como ITU ha definido muchas normas importantes para las comunicaciones de los datos.

DCE : Nombre que se da al Ambiente Distribuido computarizado, una colección de servicios de tecnología desarrollada por El Grupo Abierto para crear aplicaciones distribuidas que corren en plataformas diferentes.

DTE: Nombre que se da al Equipo Terminal de datos, es un dispositivo que controla datos que fluyen a o de una computadora. Se acostumbra a utilizar el término en referencia a comunicaciones serie definidas por los RS-232C estándar. Esta norma define los dos extremos de las comunicaciones siendo el canal un DTE y el equipo de Comunicaciones de Datos (DCE) el dispositivo. En términos prácticos, el DCE es normalmente un módem y el DTE es la propia

computadora, o más precisamente, el chip UART de la computadora. Para los módems interiores, el DCE y DTE son parte del mismo dispositivo.

(EIA): Asociación de Industrias electrónica: Una asociación de comercio que representa la comunidad de alta tecnología americana. Empezó en 1924 como la Asociación de Fabricantes de Radio. El EIA patrocina varias actividades en nombre de sus miembros, incluso las conferencias y muestras de comercio. Además, ha sido responsable para desarrollar algunas normas importantes, como los RS-232, RS-422 y RS-423 normas para los dispositivos de conexión serial

EMI : Interferencia electromagnética:

GPIB : General Purpose Interface Bus. Bus de interface de Propósito General sinónimo con Hp-IB. Es un Bus estándar usado para el control electrónico de instrumentos en un computador. Conocido también como como bus IEEE 488 definido en este patrón en el año de 1975 y posteriormente ANSI/IEEE 488 en 1978, luego 488.1 en 1975 y 488.2 en 1987

HANDSHAKE : Término utilizado en el proceso de comunicación que significa enlace o protocolo de comunicación.

HP-IB : Hewlett-Packard Interface Bus. Bus de interconexión de instrumentos electrónicos, desarrollado por Hewlett- Packard.

IEEE : Abreviación de Instituto de Ingeniería Eléctrica y Electrónica, pronunciado I-triple-E. Fundado en 1884, el IEEE es una organización compuesta por ingenieros, científicos, y estudiantes. El IEEE es mejor conocido por las normas en vías de desarrollo para la computadora e industria de la electrónica. En particular, el IEEE establece 802 normas que se siguen para las redes del área local .

I/O: Input/ Output. Entrada / Salida

ISO : Así es llamada la Organización Internacional para Estandarización. Note que ISO no son siglas; en cambio, el nombre deriva del iso de palabra griega que

significa igual. Fundado en 1946, ISO es una organización internacional compuesta de las normas nacionales de más de 75 países. Por ejemplo, ANSI (El Instituto de las Normas Nacionales Americanas) es un miembro de ISO. ISO ha definido varias normas para la computadora las más significativas son quizás las OSI (Interconexión de los Sistemas Abiertos), una arquitectura estandarizada para diseñadores de redes.

LISTENER: Escucha u oyente. Función que capacita a un dispositivo recibir mensajes a través del bus GPIB

RFI : Interferencia de radio frecuencia

RS-232C : Así es llamada la recomendación standard-232C, una interface standard aprobada por la Asociación de Industrias Electrónicas (EIA) para los dispositivos de conexión serial. En 1987, el EIA dio a conocer una nueva versión de la norma y cambió el nombre a EIA-232-D. Y en 1991, el EIA se asoció con asociación de Industria de Telecomunicaciones (TIA) y emitió una nueva versión de la norma llamó EIA/TIA-232-E.

Casi todos módems conforman a los EIA-232 estándar y la mayoría de las computadoras personales tienen un EIA-232 puerto por conectar un módem o otro dispositivo. Además de los módems, muchos monitores, mouses, e impresoras seriales son diseñadas para conectar a un puerto EIA-232 .

RS-422 y RS-423 : Interfaces normales aprobadas por la Asociación de Industrias Electrónicas (EIA) para los dispositivos de conexión serial . Las RS-422 y RS-423 normas se diseñan para reemplazar el más viejo RS-232 estándar porque se busca la mayor transmisión de datos y mayor inmunidad a la interferencia eléctrica. Todas las computadoras Macintosh de Apple contienen un puerto RS-422 que también puede usarse para comunicación RS-232C .

RS-422 apoyan conexiones del multipoint considerando que RS-423 realiza sólo conexiones del punto-a-punto.

RS-485 : Una Asociación de Industria de Electrónica (EIA) normal para comunicaciones de multipunto. Utiliza varios tipos de conectores e incluye DB-9 y DB-37. RS-485 son similares a RS-422 pero puede soportar más nodos por la línea porque usa manejadores y receptores de baja impedancia

SCPI : Standard Commands for Programmable Instruments. Comandos normalizados para instrumentos programables. Viene a ser una extensión de la norma 488.2 que define un set de comandos normalizados para la programación de instrumentos

SICL: Agilent Standard Instrument Control Library: Es una librería I/O desarrollada por Agilent que es útil para muchas interfaces IO

SRQ: Service Request. Pedido de servicio

Talker: Locutor o hablante. Es la función que permite a un dispositivo transmitir información a otros dispositivos conectados al bus GPIB.

UTP: (Unshielded Twisted Pair) es una variante del par trenzado que se emplea para transmisión de datos en las redes LAN

VISA: Agilent Virtual Instrument Software Architecture: Es una librería que puede ser usada para desarrollar aplicaciones I/O

MANUAL DEL USUARIO

Para un correcto funcionamiento de la red se debe seguir los siguientes pasos:

- Asegúrese que todos los osciloscopios estén conectados a la red a través del cable GPIB
- Encienda la computadora y abra el programa RED DE COMUNICACIÓN CON GPIB creado en una versión de seguridad a través del lenguaje de programación gráfico HP VEE .
- Encienda mínimo 4 osciloscopio (Mínimo 2/3 del total configurados)
- Pulse sobre la tecla ingresar en la primera pantalla que aparece al ejecutar el programa
- Seleccione uno de los tres nombres de la lista de la caja de diálogo.
- Ingrese la clave correcta
- Si usted ingresó su clave correcta tendrá acceso a una pantalla para seleccionar una de las siguientes opciones:

- Ver toda la red
- Información
- Salir

- Al seleccionar “información” ingresará a una pantalla donde se describe en forma breve sobre la red implementada
- Si seleccionó “ver toda la red” usted ingresará a una pantalla que muestra los 5 displays de los osciloscopios con las formas de onda que se están adquiriendo. En esta pantalla, pulsando sobre la tecla autoescala, el osciloscopio ejecutará este comando y mostrará las formas de onda con una amplitud y tiempo que se ajustan a las divisiones de la pantalla.

En esta misma ventana si se pulsa sobre el botón ingresar, se accede a una ventana donde aparece el display de un solo osciloscopio, sobre la que se muestran las formas de onda que se están adquiriendo, y a través de varios switch y botones presentes se tiene control sobre el osciloscopio.

Para observar los valores característicos de una señal se debe seleccionar el botón “actualizar” y luego deshabilitar esta opción. Si se no se la deshabilita la adquisición de las formas de onda es mas lenta

Para seleccionar “grabar”, asegúrese de que previamente se está adquiriendo señales a través de los canales. Aparecerá una caja de diálogo que indica donde grabar, escriba un nombre con la extensión JPG o BMP.

Para Salir pulse sobre el botón “regresar” de la última ventana presente y luego salir.

Apague la computadora, los osciloscopios pueden seguir funcionando en forma independiente.

Latacunga, Julio del 2005

Elaborado por:

José G. Bucheli Andrade

LA DIRECTORA DE LA CARRERA DE INGENIERÍA EN
ELECTRÓNICA ESPECIALIDAD INSTRUMENTACIÓN

Ing. Nancy Guerrón

EL SECRETARIO DE LA ESPE – LATACUNGA

Ab. Eduardo Vásquez Alcázar