



Sistema de Medición de Campo Lejano

Cecilia Andrea Morras
Maria Cecilia De Vito

TES
05/26
DIF-02966
SALA



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA
Biblioteca
50 y 120 La Plata
catalogo.info.unlp.edu.ar
biblioteca@info.unlp.edu.ar



DIF-02966

Tabla de contenidos

| | |
|---|----|
| 1. Introducción | 5 |
| 1.1 Misiones SAOCOM | 6 |
| 2. Antenas..... | 8 |
| 2.1 Longitud de onda | 8 |
| 2.2 Antena de media onda | 8 |
| 2.3 Impedancia | 9 |
| 2.4 Patrón de radiación | 10 |
| 2.5 Ganancia Directiva y Ganancia de Potencia..... | 10 |
| 2.6 Polarización de la Antena | 11 |
| 2.7 Ancho del haz de la antena | 12 |
| 2.8 Ancho de banda de la antena..... | 12 |
| 3. Mediciones sobre Antenas..... | 13 |
| 3.1 Campos de medida..... | 13 |
| 3.1.1 Campos de medida en reflexión..... | 13 |
| 3.1.2 Campos de medida elevados | 13 |
| 3.1.3 Campos de medida inclinados | 14 |
| 3.1.4 Cámaras anecoicas | 14 |
| 3.1.5 Campos de medida compactos | 14 |
| 3.2 Instrumentación de medida..... | 15 |
| 3.2.1 Sistemas de posicionamiento | 15 |
| 3.3 Medidas de ganancia | 16 |
| 3.4 Medida de la Directividad | 16 |
| 3.5 Medida de impedancia | 16 |
| 3.6 Medida de la distribución de corriente..... | 17 |
| 3.7 Medida de la polarización | 17 |
| 3.7.1 Medida del diagrama de polarización | 17 |
| 3.7.2 Medida de la amplitud y de la fase..... | 18 |
| 3.7.3 Método con tres antenas | 18 |
| 3.8 Medida de la temperatura de la antena | 18 |
| 3.9 Medidas de campo próximo | 19 |
| 4. Teledetección..... | 20 |
| 4.1 Espectro electromagnético | 20 |
| 4.2 Interacción Atmósfera y radiación | 21 |
| 4.3 <i>Interacción radiación y superficie</i> | 22 |
| 4.4 <i>Sensores Activos</i> | 22 |
| 4.5 <i>Sensores Pasivos</i> | 23 |
| 4.6 <i>Radar</i> | 23 |
| 4.7 <i>RAR</i> | 24 |
| 4.8 <i>SAR</i> | 24 |
| 5. Sistemas de Adquisición de Datos..... | 27 |
| 5.1 Introducción..... | 27 |
| 5.2 Computadora..... | 27 |
| 5.3 Sensores..... | 28 |
| 5.4 Señales..... | 29 |
| 5.5 Acondicionadores | 31 |
| 5.6 Hardware de Adquisición y Control | 32 |
| 5.7 Manejadores (“drivers”) y programas de aplicación. | 32 |
| 5.8 El elemento más importante..... | 32 |
| 5.9 Instrumentación Virtual | 35 |
| 5.10 El estándar SCPI | 36 |
| 5.10.1 Uso del Estándar SCPI..... | 37 |
| 5.11 Manejadores de Instrumentos..... | 37 |

| | |
|---|----|
| 6. Sensores de Posición..... | 40 |
| 6.1 Potenciómetros..... | 40 |
| 6.2 LVDTs..... | 40 |
| 6.3 Encoders..... | 41 |
| 6.3.1 Encoders Incrementales y Absolutos: | 41 |
| 6.3.2 El código Gray..... | 42 |
| 6.3.2.1 Ventajas del código Gray: | 45 |
| 6.3.2.2 Un ejemplo de aplicación | 45 |
| 7. Entrada/Salida Digital | 46 |
| 7.1 Conceptos Generales..... | 46 |
| 7.2 Comunicación entre la CPU y los Periféricos..... | 48 |
| 7.3 Operaciones de entrada/salida..... | 49 |
| 7.4 Direccionamiento de entrada/salida..... | 51 |
| 7.4.1 Direccionamiento dedicado | 52 |
| 7.4.2 Direccionamiento mapeado en memoria | 52 |
| 8. Comunicación Serie | 54 |
| 8.1 Características de la Comunicación Serie: | 54 |
| 8.2 RS-232..... | 57 |
| 8.3 Otras Normas:..... | 62 |
| 8.3.1 RS-449: | 62 |
| 8.3.2 CCITT V.24:..... | 62 |
| 8.3.3 Interfase digital X.21. | 63 |
| 8.4 Protocolos de Comunicación (Handshaking Modes)..... | 63 |
| 8.4.1 Software Handhaking – XON-XOFF | 64 |
| 8.4.2 Hardware Handhaking – XON-XOFF | 65 |
| 9. GPIB (General Purpose Interface Bus)..... | 66 |
| 9.1 Introducción..... | 66 |
| 9.2 Conceptos Básicos GPIB..... | 66 |
| 9.2.1 Características del bus GPIB | 66 |
| 9.2.2 Estructura del Bus GPIB | 68 |
| 9.2.3 Direccionamiento GPIB..... | 71 |
| 9.2.4 Reporte de estado | 72 |
| 9.2.4.1 Registros de estado..... | 73 |
| 9.2.5 Comandos Comunes | 75 |
| 10. Proceso Paralelo | 77 |
| 10.1 Exclusión mutua: | 77 |
| 10.2 Sincronización: | 79 |
| 10.3 Espera Activa: | 79 |
| 10.4 Espera no activa: | 79 |
| 10.5 Mecanismos Hardware: | 81 |
| 10.6 Semáforos: | 82 |
| 10.7 Semáforos en Sincronización Barrier: | 82 |
| 10.8 Especificación de la Ejecución Concurrente: | 83 |
| 10.9 Especificación de la Sincronización: | 83 |
| 10.10 Sincronización Barrier: | 84 |
| 11. Descripción del Sistema de medida de Campo Lejano..... | 88 |
| 11.1 Introducción: | 88 |
| 11.2 Restricciones del Sistema y del Proyecto: | 89 |
| 11.3 Ambiente de Desarrollo/ Operación/Mantenimiento: | 89 |
| 11.4 Funcionalidad del Sistema: | 90 |
| 12. Control de Posición | 91 |
| 12.1 Introducción | 91 |
| 12.2 Interfase PCI-DIO-96..... | 91 |

| | |
|---|-----|
| 12.2.1 Mapa de los registros | 92 |
| 12.2.1.1 Programación de la interfase | 93 |
| 12.2.1.2 Conexión con el Encoder | 96 |
| 12.2.1.3 Lectura del encoder | 100 |
| 12.2.1.4 Devolución de los recursos | 102 |
| 12.2.1.5 Control de movimiento del pedestal..... | 102 |
| 13. Control remoto del Generador de Señales en la antena de referencia | 104 |
| 13.1 Descripción del Instrumental relacionado: | 104 |
| 13.2 Implementación: | 105 |
| 14. Control de instrumentos de medida..... | 106 |
| 14.1 Introducción a las funciones básicas GPIB | 106 |
| 14.2 Comunicación con los instrumentos de medida del sistema de Campo Lejano a través de la interfase GPIB: | 108 |
| 14.3 Conclusiones: | 110 |

1. Introducción

El **Programa Espacial Argentino** de la **Comisión Nacional de Actividades Espaciales (CONAE)** ha obtenido buenos resultados con métodos asociativos, que permiten a todos los países participantes de una misión argentina reducir sus costos de acceso y operación en el espacio.

Esto, más una ingeniería de muy buena calidad y costos de construcción muy moderados han hecho que las agencias espaciales de los **Estados Unidos, Francia, Italia, Dinamarca y Brasil** colocarán sus aparatos o servicios en satélites argentinos. Y hay más interesados.

Por su disposición para las alianzas estratégicas y el funcionamiento impecable de sus misiones (que van desde el diseño de la plataforma a la fabricación de sus partes, y desde la integración final de éstas hasta el control del satélite ya instalado en órbita), la **Argentina** empieza a ser vista como un socio espacial confiable –o posible– por muchos países desarrollados.

Presentación de los servicios espaciales de INVAP en su portal www.invap.com.ar



Figura 1: Infografía de Vuelo Satélite SAC-C (Cortesía INVAP-Argentina)

Misiones SAOCOM

Los satélites SAOCOM, de los cuales se fabricarán al menos dos ejemplares, tendrán 1600 kilogramos cada uno. Serán los satélites de mayor tamaño y capacidad construidos en el Hemisferio Sur hasta ahora. Su utilidad fundamental será el monitoreo y la prevención de catástrofes, y su principal instrumento será el radar de apertura sintética.

En la actualidad La Comisión Nacional de Actividades Espaciales (CONAE), está desarrollando un nuevo satélite de manufactura argentina. Se trata del SAOCOM 1-A que constituye una nueva generación de satélites para la observación de la Tierra, con instrumentos que operan en el rango de microondas y dispone de sensores activos (radar).

El radar del SAOCOM emitirá microondas a alta potencia, con una antena gigante de 2,5 por 10 metros. Como vigia de catástrofes, este sensor podrá inspeccionar situaciones antes de que ocurran o mientras suceden, en forma independiente de la hora o la meteorología.

A diferencia de las cámaras comunes, el radar "puede ver" a través de las nubes o de noche, y así detectar agua bajo cobertura de árboles, medir los cambios de altura de un domo volcánico a punto de reventar, e incluso penetrar profundamente en el suelo y discernir ríos o estructuras subterráneas, o estimar si la profundidad de las napas freáticas supone riesgo de inundación, o de sequía.

El radar del SAOCOM será del tipo SAR, o de "apertura sintética", que permite apuntar o hacer oscilar el haz de una antena fija como si ésta fuera móvil. Esto permite barrer o enfocar lo que se desee en la superficie terrestre, pero sin el riesgo y las complicaciones mecánicas de tener que mover una gran antena a centenares de kilómetros de altura, o gastar energía moviendo todo el satélite para apuntarlo lateralmente. Como emitirá microondas en la banda "L", esta antena SAR podrá

detectar objetos mayores de ocho metros.

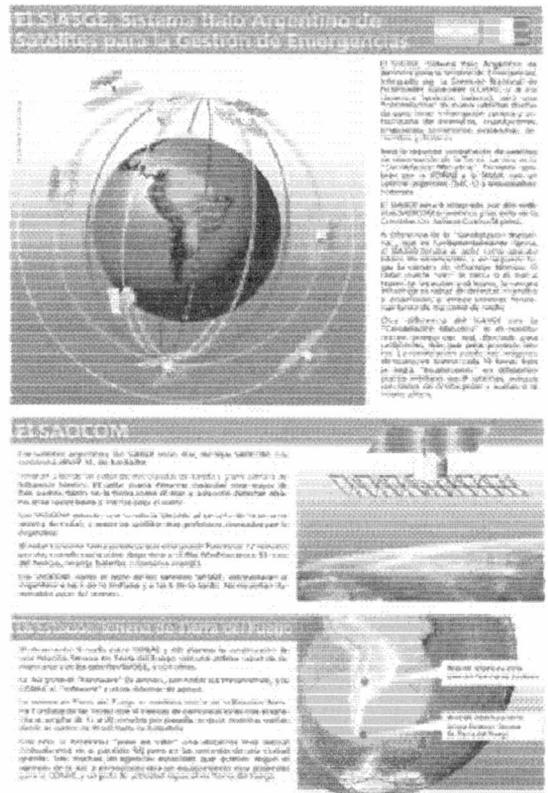


Figura 2: Infografía SIASGE (Cortesía Inva-Argentina)

Es probable que esta capacidad SAOCOM de generar imágenes de radar finas y en tiempo real se combine con los resultados de al menos dos cámaras ópticas a bordo. Habrá una "infrarroja" capaz de detectar fuentes de calor y prevenir incendios en zonas despobladas, y otra nocturna, capaz de funcionar de noche con luz residual (la de la luna o las estrellas). Con tantos y tan diversos sensores, habrá muy pocos eventos naturales dañinos que puedan escapar a la detección o evaluación del SAOCOM.

En el marco de la política de CONAE, que favorece las alianzas estratégicas con instituciones nacionales e internacionales, se selló un convenio con el Instituto Argentino de Radioastronomía. El convenio establecía el desarrollo de los prototipos de los paneles del radar que se instalará en la misión SACOCOM.

Como todo dispositivo que debe cumplir sus funciones en el espacio esta sujeto a especificaciones muy exigentes. Fundamentalmente esta exigencia se encuentra asociada al amplio rango de trabajo en que deben mantenerse sus especificaciones dentro de las bandas de tolerancia. En el caso particular de la antena interesa conocer con mucha exactitud y precisión su impedancia, diagrama de irradiación para ambas polarizaciones, ganancia, ancho del haz, ancho de banda, aislamiento de polarización y estabilidad con la temperatura, entre otros.

Paralelamente al esfuerzo profesional, que implica el desarrollo de un componente que se encuentra en el "estado del arte", surge la necesidad de establecer un sistema de medición que permita verificar los parámetros característicos antes mencionados. En el caso especial del diagrama de radiación se deberá implementar un Sistema de medición de Campo Lejano que dará como resultado el patrón de campo lejos de la antena.

2. Antenas

En esencia, una antena es un sistema conductor metálico capaz de radiar y recibir energía en forma de ondas electromagnéticas, y una guía de onda es un tubo metálico conductor por medio del cual se propaga esa energía electromagnética. Una antena se utiliza como la interfase entre un transmisor y el espacio libre o el espacio libre y el receptor. Una guía de onda, así como una línea de transmisión, se utiliza sólo para interconectar eficientemente una antena con el receptor o el transmisor. Una antena acopla energía de la salida de un transmisor a la atmósfera de la Tierra o de la atmósfera de la Tierra a un receptor. Una antena es un dispositivo recíproco pasivo; pasivo en cuanto a que en realidad no puede amplificar una señal, por lo menos no en el sentido real de la palabra (sin embargo, una antena puede tener ganancia), y recíproco en cuanto a que las características de transmisión y recepción son idénticas, excepto donde las corrientes de alimentación al elemento de la antena se limitan a la modificación del patrón de transmisión.

Sin profundizar en cuestiones físicas, si una corriente circula por un conductor, creará un campo eléctrico y magnético en sus alrededores. Luego la corriente creará un campo eléctrico y magnético, pero como la distancia entre los dos conductores, que forman nuestra línea es pequeña, no se creará una onda que se propaga. En esta situación la contribución que presenta el conductor superior se anulará con la que presenta el conductor inferior.

Pero si separa, en un punto los dos conductores, los campos que crean las corrientes ya no se anularán entre sí, si no que se creará un campo eléctrico y magnético que formará una onda que se podrá propagar por el espacio.

Según esto, dependiendo del punto desde el que se separe el conductor, se tendrá una longitud en los elementos radiantes variable. Al variar esta longitud, la distribución la corriente variará, y lógicamente la onda que se creará y se propagará.

Longitud de onda

Se entiende por longitud de onda la longitud en metros que correspondería a un ciclo de la corriente considerada, sabiendo que las ondas de radio viajan en espacio a la velocidad de la luz, es decir, 300.000 Km/seg. Suponiendo una señal cuya frecuencia fuese 1 ciclo/seg o 1 Hz, esta recorrería en un segundo 300.000 Km y esta sería su Longitud de Onda. Así pues en el caso de la radioastronomía la frecuencia de emisión del hidrógeno neutro es de 1420 MHz, es decir, 1,42GHz y la longitud de onda es:

$$\lambda = \frac{300.000 \text{ Km/seg}}{1.420.000.0001/\text{seg}} \cong 2,1 \cdot 10^{-4} \text{ Km} \cong 21 \text{ cm}$$

Antena de media onda

La Figura 3 representa un dipolo de media onda al que se le suministra alimentación de radiofrecuencia a través de una línea de transmisión. Al aplicar una señal alterna se generará una corriente y un voltaje. Estas están desfasadas 90°, de tal forma que, si se pudiera visualizar la radiofrecuencia aplicada en la antena el voltaje ,en el centro de la misma, sería cero y en los extremos alcanzaría su valor máximo (menos las

pérdidas). En cambio la intensidad de corriente es cero en los extremos y alcanza su valor máximo en el centro de la antena.

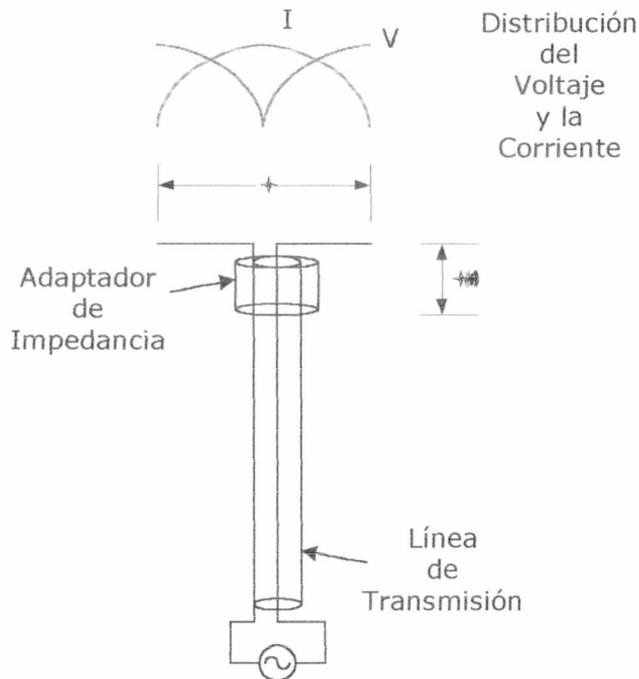


Figura 3: Dipolo de $\lambda/2$

Impedancia

Una antena se tendrá que conectar a un transmisor y deberá radiar el máximo de potencia posible con un mínimo de pérdidas. Se deberá adaptar la antena al transmisor para una máxima transferencia de potencia, que se suele hacer a través de una línea de transmisión. Esta línea también influirá en la adaptación, debiéndose considerar su impedancia característica, atenuación y longitud.

Como el transmisor producirá corrientes y campos, a la entrada de la antena se puede definir la impedancia de entrada mediante la relación tensión-corriente en ese punto. Esta impedancia poseerá una parte real $Re(w)$ y una parte imaginaria $Ri(w)$, dependientes de la frecuencia. Si a una frecuencia una antena no presenta parte imaginaria en su impedancia $Ri(w)=0$, entonces se dirá que esa antena está resonando a esa frecuencia. Normalmente una antena será usada a su frecuencia de resonancia, que es cuando mejor se comporta. Lógicamente esta resistencia también dependerá de la frecuencia.

Esta resistencia de entrada se puede descomponer en dos resistencias, la resistencia de radiación (R_r) y la resistencia de pérdidas (R_L). Se define la resistencia de radiación como una resistencia que disiparía en forma de calor la misma potencia que radiaría la antena. La antena por estar compuesta por conductores presentará pérdidas. Estas pérdidas son las que definen la resistencia de pérdidas en la antena. Cuando la antena está resonando evita tener que aplicar corrientes excesivas, que lo único que hacen es producir grandes pérdidas.

Patrón de radiación

El campo de radiación que se encuentra cerca de una antena no es igual que el campo de radiación que se encuentra a gran distancia. El término campo cercano se refiere al patrón de campo que está cerca de la antena, y el término campo lejano se refiere al patrón de campo que está a gran distancia.

Durante la mitad del ciclo, la potencia se irradia desde una antena, en donde parte de la potencia se guarda temporalmente en el campo cercano. Durante la segunda mitad del ciclo, la potencia que está en el campo cercano regresa a la antena. Esta acción es similar a la forma en que un inductor guarda y suelta energía. Por tanto, el campo cercano se llama a veces campo de inducción. La potencia que alcanza el campo lejano continúa irradiando lejos y nunca regresa a la antena. Por tanto, el campo lejano se llama campo de radiación.

La potencia de radiación, por lo general, es la más importante de las dos; por consiguiente, los patrones de radiación de la antena, por lo regular se dan para el campo lejano. El campo cercano se define como el área dentro de una distancia " D^2/l " de la antena, en donde l es la longitud de onda y D el diámetro de la antena en las mismas unidades. La Figura 4 muestra el patrón de campo sobre los planos Y-Z y X-Y para un dipolo de media onda.

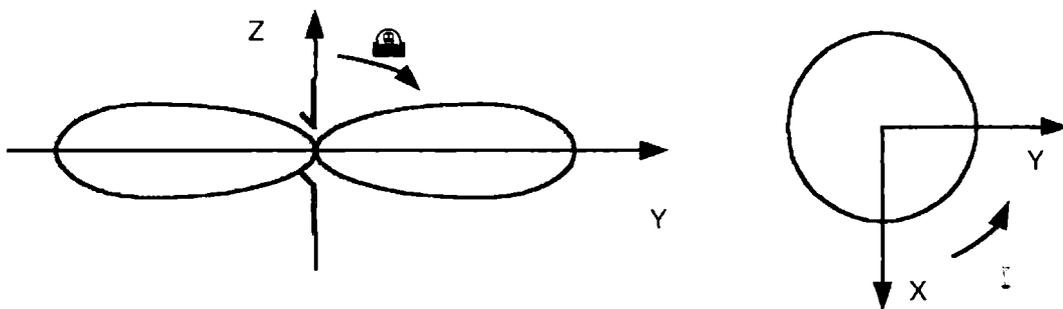


Figura 4: Diagramas de radiación para un dipolo de media onda

Ganancia Directiva y Ganancia de Potencia

Los términos ganancia directiva y ganancia de potencia con frecuencia no se comprenden y, por tanto, se utilizan incorrectamente. La ganancia directiva es la relación de la densidad de potencia radiada en una dirección en particular con la densidad de potencia radiada al mismo punto por una antena de referencia, suponiendo que ambas antenas irradian la misma cantidad de potencia. El patrón de radiación para la densidad de potencia relativa de una antena es realmente un patrón de ganancia directiva si la referencia de la densidad de potencia se toma de una antena de referencia estándar, que por lo general es una antena isotrópica. La máxima ganancia directiva se llama **Directividad**.

La ganancia de potencia es igual a la ganancia directiva excepto que se utiliza el total de potencia que alimenta a la antena (o sea, que se toma en cuenta la eficiencia de la antena). Se supone que la antena indicada y la antena de referencia tienen la misma potencia y que la antena de referencia no tiene pérdidas.

Polarización de la Antena

La Figura 5 representa el esquema más sencillo de onda electromagnética, que es aquella en el cual los frentes de ondas son planos, y en cualquier punto por el que pasa el tren de ondas de intensidad del campo eléctrico y la excitación magnética oscilan a lo largo de líneas rectas perpendiculares entre sí y a la dirección de propagación.

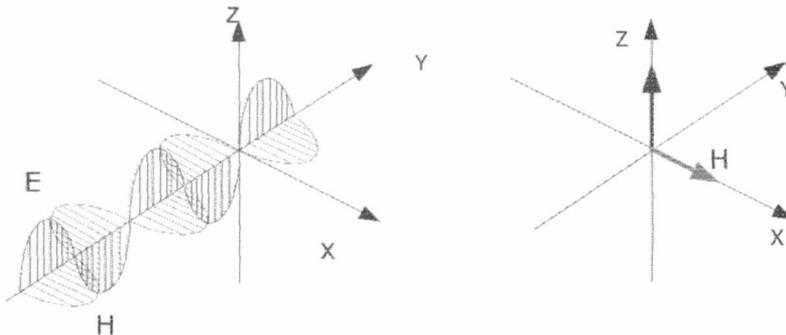


Figura 5: Onda electromagnética

Se dice que una onda está polarizada linealmente cuando, en cualquier punto fijo, la punta del vector E (o H) oscila a lo largo de una recta. La teoría y la experiencia demuestran que las ondas procedentes de un dipolo oscilante están polarizadas linealmente, encontrándose el vector eléctrico en un plano que contiene al dipolo.

La polarización de una antena se refiere sólo a la orientación del campo eléctrico radiado desde ésta. Una antena puede polarizarse en forma lineal (por lo regular, polarizada horizontalmente o verticalmente, suponiendo que los elementos de la antena se encuentran dentro de un plano horizontal o vertical), en forma elíptica, o circular.

Si una antena irradia una onda electromagnética polarizada verticalmente, la antena se define como polarizada verticalmente; si la antena irradia una onda electromagnética polarizada horizontalmente, se dice que la antena está polarizada horizontalmente; si el campo eléctrico gira en un patrón elíptico, está polarizada elípticamente; y si el campo eléctrico gira en un patrón circular, está polarizada circularmente.

Si se considera un plano perpendicular al rayo, en un cierto punto A, si el campo eléctrico en este plano, en lugar de variar su magnitud cambia su dirección se está ante la presencia de una onda electromagnética polarizada circularmente, Figura 6.

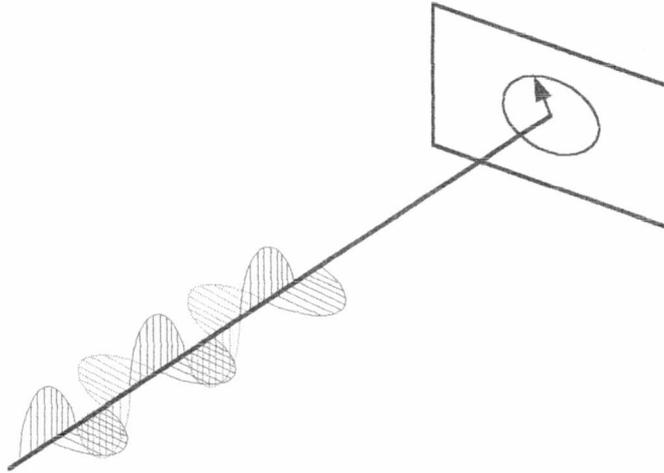


Figura 6: Onda electromagnética circularmente polarizada

Cuando el campo eléctrico varía en dirección y magnitud sobre un plano perpendicular al rayo de manera, que la proyección del extremo de su extremo sobre el mencionado plano describe una elipse se dice que la onda electromagnética se encuentra polarizada elípticamente.

Ancho del haz de la antena

El ancho del haz de la antena es solo la separación angular entre los dos puntos de media potencia (-3 dB) en el lóbulo principal del patrón de radiación del plano de la antena, por lo general tomado de uno de los planos "principales".

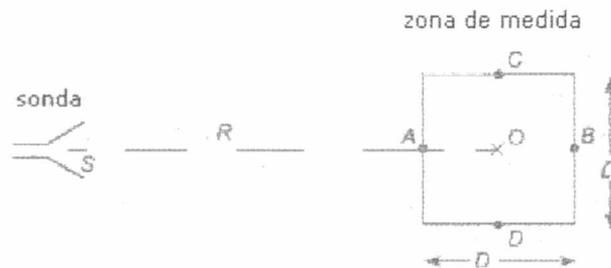
Ancho de banda de la antena

El ancho de banda de la antena se define como el rango de frecuencias sobre las cuales la operación de la antena es "satisfactoria". Esto, por lo general se toma entre los puntos de media potencia, pero a veces se refiere a las variaciones en la impedancia de entrada de la antena.

3. Mediciones sobre Antenas

Campos de medida

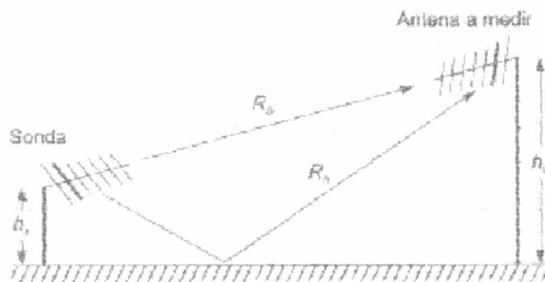
Para medir el diagrama de radiación de la antena se desplaza otra antena, denominada sonda, a una distancia R constante y se van registrando las señales recibidas por la sonda en función de la posición angular. La representación de estas variaciones constituye el diagrama de la antena.



Campo de medida de antenas

Campos de medida en reflexión

A bajas frecuencias es muy difícil eliminar las reflexiones, por lo que, en vez de evitarlas, se intenta controlarlas de manera que sus efectos sean menos desfavorables. Estos campos se diseñan de manera que la suma de la señal directa y las reflejadas produzcan dentro de la zona de medida una onda aproximadamente plana. Se utilizan en VHF y UHF, por lo que al ser las longitudes de onda grandes, las variaciones de amplitud en función de la posición serán pequeñas.



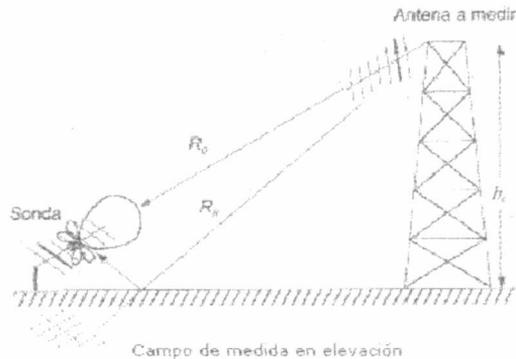
Campo de medida en reflexión

Campos de medida elevados

En algunas ocasiones, para evitar las reflexiones en el suelo, se sitúan la sonda y la antena que se quiere medir en lugares elevados, de manera que por la directividad de la sonda y la rugosidad del suelo el efecto de las reflexiones sea prácticamente despreciable.

Campos de medida inclinados

En ocasiones, cuando existen problemas de espacio y para evitar las reflexiones, se utilizan los **campos de medida inclinados**. En éstos se sitúa la antena que se quiere medir sobre una torre o mástil, usualmente telescópico para facilitar el montaje de la antena. En el suelo y a cierta distancia de la torre se sitúa la sonda de medida cuyo diagrama de radiación se diseña de tal manera que presente un nulo en la dirección del suelo donde se produce la reflexión. Este tipo de campo de medida es usual en la medida de antenas de VHF (Very High Frequency) y UHF (Ultra High Frequency).

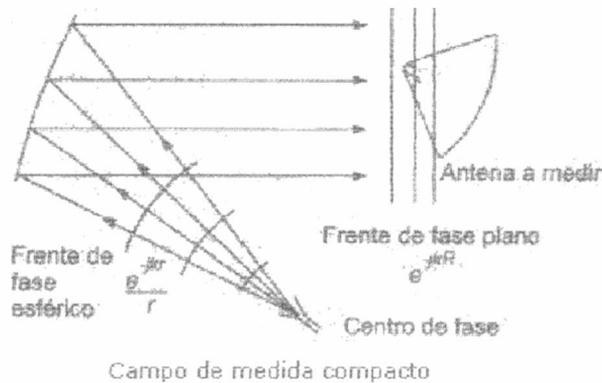


Cámaras anecoicas

La mejor manera de medir antenas es evitar las reflexiones o por lo menos que estas tengan un valor muy bajo. Esto es lo que se intenta lograr en las cámaras anecoicas, que son recintos cerrados recubiertos, en sus paredes, con paneles de materiales absorbentes, de manera que las reflexiones en ellos sean muy pequeñas.

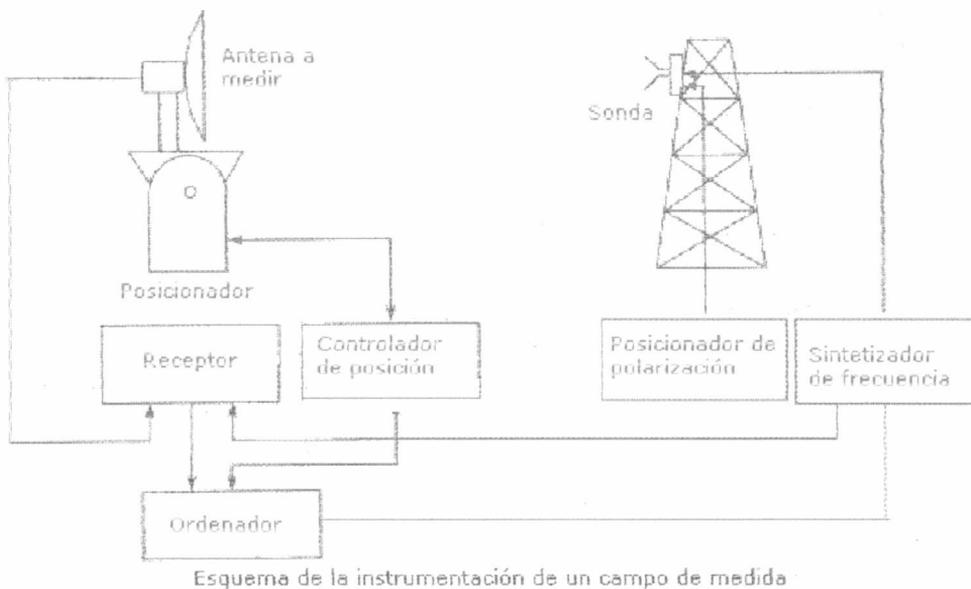
Campos de medida compactos

Para evitar los problemas del bloqueo del alimentador las geometrías del reflector suelen ser asimétricas, u otras aún más complejas (dobles reflectores, cassegrain o gregorianas). La antena que se quiere medir se coloca en frente del reflector, cuyo tamaño deberá ser lo suficientemente grande como para garantizar una onda plana sobre toda la antena. Aún así, las dimensiones totales del conjunto son mucho menores que las exigidas en uno de medida en campo lejano directo. Debido a su reducido tamaño se llaman campos de medida compactos.



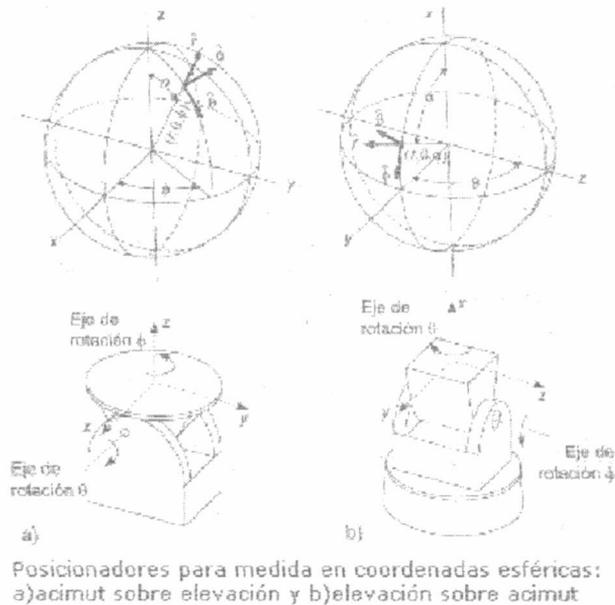
Instrumentación de medida

La **figura** muestra el esquema general de un sistema para la medida de antenas. Los elementos básicos que lo configuran son: un **generador de radiofrecuencia**, una **sonda de medida**, unos **posicionadores** para mover la antena, un **sistema de detección y medida** de la señal recibida y un **sistema de control y almacenamiento de los datos recibidos**.



Sistemas de posicionamiento

Si solo se requiere la medida de un corte del diagrama será suficiente con un motor que mueva la antena respecto a un eje. Pero lo habitual es querer hacer medidas completas del diagrama, con lo cual son necesarios dos motores o sistemas de posicionamiento que se muevan sobre ejes ortogonales.



Medidas de ganancia

Uno de los parámetros más importantes que caracteriza a una antena es su ganancia. Existen dos clasificaciones básicas de los métodos de medida de ganancia: absolutos y por comparación. La medida absoluta se suele hacer con antenas que luego se van a emplear como referencia en otras medidas. Las dos antenas más comúnmente empleadas como patrones en la medida de otras son el dipolo resonante en $l/2$ (con una ganancia de 2.1 db) y las bocinas piramidales (con ganancias entre 10 y 25 dB).

Medida de la Directividad

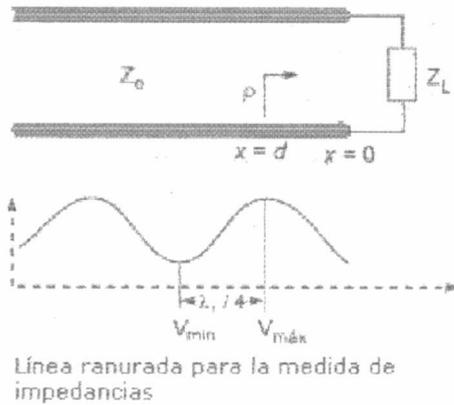
A diferencia de la ganancia, que es un parámetro que también depende de la eficiencia de la antena, la directividad depende únicamente del diagrama de radiación, por lo que para hallarla es preciso conocer éste.

Un método es hallar la directividad a partir de sus expresiones aproximadas en función del ángulo sólido equivalente. Este último se puede estimar como el producto de los ancho de haz a -3 dB de la antena en sus planos principales.

Medida de impedancia

La medida de la impedancia es igual a la de cualquier dispositivo de un acceso, con la particularidad de que hay que tener en cuenta que los elementos que rodean a la antena afectan a su impedancia y por tanto, al hacer la medida, se deberán reproducir lo más fielmente posible las condiciones reales de funcionamiento de la antena. A continuación se mencionan algunos métodos de medida:

- Puente de impedancia.
- Línea ranurada.
- Analizador de redes.



Medida de la distribución de corriente

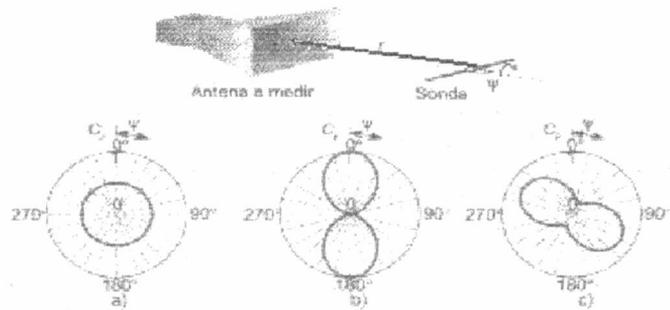
Para la medida de corrientes el método más simple consiste en desplazar una pequeña antena, usualmente una espira, es decir, un dipolo magnético elemental, paralelamente y muy próxima al hilo de corriente. Si el plano de la espira y el del dipolo coinciden, la tensión inducida en ella será proporcional al flujo magnético que la atraviesa. Si el tamaño de la espira es suficientemente pequeño se puede decir que el campo magnético será constante y proporcional a la corriente del hilo en aquel punto.

Medida de la polarización

En la práctica, es difícil construir una antena que mantenga sus características de polarización para cualquier punto de su diagrama, por lo que para una caracterización completa se deberán medir las características de polarización en múltiples direcciones. Hay diferentes técnicas para determinar la polarización de una onda, a continuación se Presentan algunas de ellas.

Medida del diagrama de polarización

Esta medida da la información de la relación axial y del ángulo de inclinación de la onda, pero no de su sentido de giro. La antena que se quiere medir se puede emplear en transmisión o en recepción y se precisa de una antena linealmente polarizada, por el dipolo, montado según se indica en la figura.



Medida del diagrama de polarización de de una antena para tres casos: a)circular, b)lineal y c)eliptica

Medida de la amplitud y de la fase

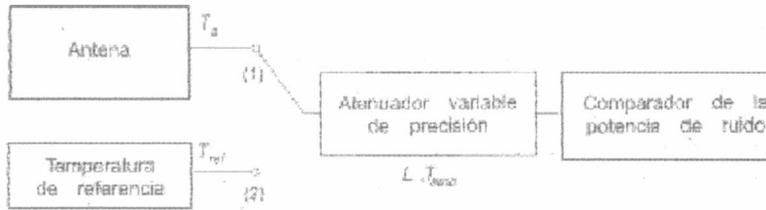
Los instrumentos que se emplean en los campos de medida permiten conocer no sólo la amplitud, sino también la fase de la señal. Midiendo la tensión recibida por una antena linealmente polarizada cuando se orienta según dos direcciones ortogonales del espacio la polarización de la onda queda ya completamente definida. La medida se puede realizar con dos antenas cualesquiera que no tengan la misma polarización pero que estén caracterizadas, tanto en polarización como en ganancia, realizando algunos cálculos para expresar, finalmente, el resultado en función de dos componentes ortogonales.

Método con tres antenas

Si se requiere una elevada precisión se deberá emplear un método de medida absoluto. Para ello se emplean tres antenas cuyas características de polarización son desconocidas. La única limitación es que al menos dos de ellas no tengan polarización circular. Emitiendo por una antena se mide, en módulo y fase, la tensión recibida por otra de ellas. La medida se repite rotando 90° las antenas entre sí. Realizando el mismo proceso con todas las combinaciones de antenas resulta finalmente un conjunto de seis medidas en donde quedan como incógnitas la polarización de las antenas y la ganancia de las mismas.

Medida de la temperatura de la antena

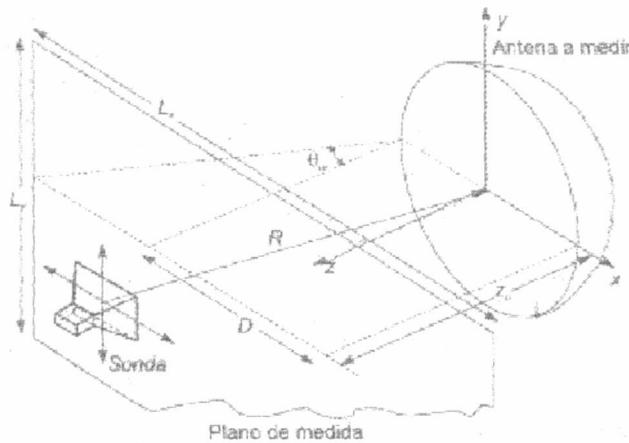
La medida de la temperatura de antena, T_a , se realiza comparando la potencia de ruido recibida por la antena con la de una fuente de temperatura de ruido conocida, T_{ref} . Para mayor exactitud en las medidas es preferible que la temperatura de la fuente de referencia sea lo más parecida posible a T_a .



Medida de la temperatura de ruido de la antena

Medidas de campo próximo

En las **medidas en campo próximo** las componentes tangenciales del campo eléctrico producido por la antena bajo prueba se miden, en amplitud y fase, por la sonda sobre una superficie que puede encontrarse a unas pocas longitudes de onda de la estructura de la antena. Las superficies de medidas más utilizadas son: la **plana**, la cilíndrica y la esférica.

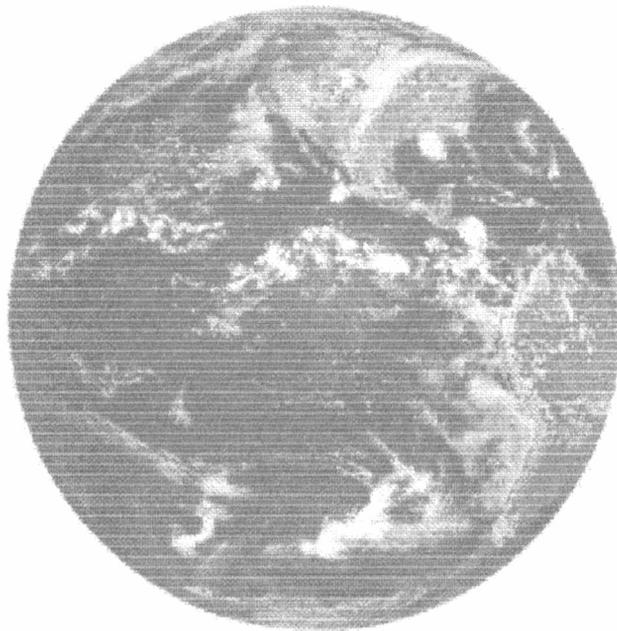


Esquema de una exploración en campo próximo en coordenadas rectangulares

4. Teledetección

La teledetección o percepción remota es una técnica que ha ido evolucionando aceleradamente durante las últimas década con el advenimiento de la tecnología satelital, sus orígenes se ubican con el inicio de las actividades aeronáuticas a principio de siglo XX para ir evolucionando con el desarrollo de las actividades espaciales. Hoy en día, las definiciones de teledetección se utilizan para describir las actividades que realizan los satélites, trasbordadores espaciales o determinado tipo de aviones. Las siguientes definiciones establecen algunos parámetros de lo que la comunidad científica considera a la palabra teledetección o percepción remota:

- La teledetección o lo que es lo mismo la percepción remota es la ciencia de adquirir y procesar información de la superficie terrestre desde sensores instalados en plataformas espaciales, gracias a la interacción de la energía electromagnética que existe entre el sensor y la tierra (Chuvienco, 1996).
- Obtención de imágenes u otro tipo de información acerca de un objeto, utilizando técnicas de mediciones hechas a distancia, además de procesar y analizar los datos (CCRS).
- La percepción remota es la colección de información de un objeto sin estar en contacto físico con el objeto. Aeroplanos, aviones y satélites son las plataformas más comunes desde donde las observaciones remotas son realizadas. El término percepción remota es restrictivo a métodos que emplean energía electromagnética como medio de detección y medición de las características de un objeto (Sabins, 1978).



Espectro electromagnético

El espectro electromagnético es la organización de bandas de longitudes de onda o frecuencia desde las más cortas a las más largas. Para la teledetección solo algunas de

estas bandas son utilizadas hoy en día para obtener información de la superficie de la tierra o de la atmósfera, entre las bandas más utilizadas están:

- Visible Infrarrojo.
- Próximo Infrarrojo.
- Medio Infrarrojo.
- Infrarrojo Lejano o térmico.
- Micro ondas.

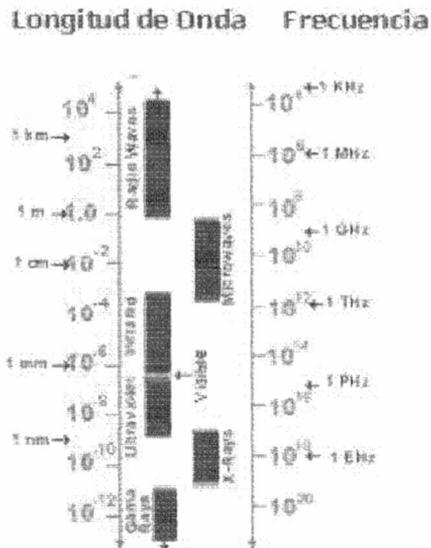


Figura 7: Espectro electromagnético.

Para cualquier proceso de percepción remota es necesario que existan los siguientes aspectos: una fuente de iluminación o energía (el sol en el caso de los sensores activos y un recurso propio a bordo del satélite en el caso de los sensores pasivos), influencia de la atmósfera (que modifica la respuesta que recibe el sensor del satélite), interacción con la superficie (que depende en mucho de sus propiedades físico químicas, su forma y textura), que se registren y procesen los datos, aplicación a un estudio determinado.

La energía que recibe el sensor, depende de las propiedades de la superficie (reflectividad de la cubierta, condiciones atmosféricas, geometría de la observación, rugosidad de la superficie), con base en ello, se dan varias denominaciones de acuerdo a su comportamiento, por ejemplo, los cuerpos Negros absorben y emiten toda la energía, los cuerpos blancos reflejan la energía por completo y los cuerpos grises absorben y emiten en forma constante la energía incidente en diferentes longitudes de onda.

Interacción Atmósfera y radiación

La atmósfera no es un cuerpo transparente en ella se encuentran pequeñas partículas en suspensión que interactúan con la radiación incidente. La energía al chocar con estas partículas puede ser dispersada, absorbida o emitida dependiendo de la longitud de onda. Por lo general a mayor longitud de onda la dispersión disminuye, en tanto absorción presenta comportamientos variables. Las porciones del espectro

electromagnético donde la radiación no es absorbida y llega a la superficie de la Tierra se denominan ventanas atmosféricas y son de gran importancia para diseñar los sensores de los satélites para captar dentro de estas longitudes de onda.

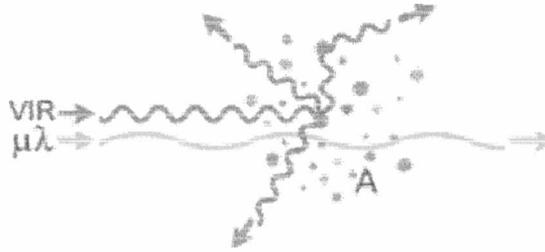


Figura 8: Energía emitida por la atmósfera.

También la atmósfera emite parte de la energía, Figura 8, absorbida en forma de calor y este tipo de energía es registrada principalmente por los sensores diseñados para generar información de las energías electromagnéticas del infrarrojo térmico.

Interacción radiación y superficie

La energía que llega a los cuerpos, incide en la superficie y puede sufrir tres procesos: ser reflejada (R), absorbida (A) o transmitida (T), por lo tanto la energía incidente (I) Será la sumatoria de todos los anteriores procesos.

$$I = R + A + T$$

La energía reflejada y el calor que los cuerpos emiten, es la energía que los sensores de los satélites reciben, después de interactuar nuevamente con la atmósfera.

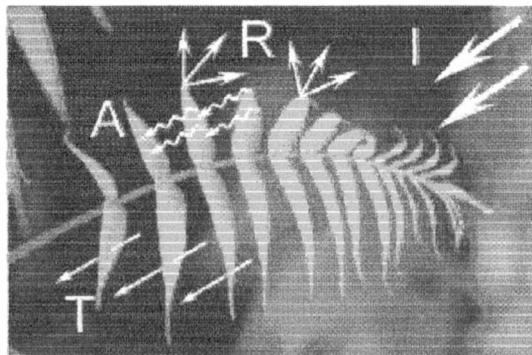


Figura 9: Descomposición de la energía incidente en los cuerpos.

Sensores Activos

Los sensores activos, se basan en que proveen su propia fuente de energía que emiten hacia los cuerpos y reciben la señal de retorno. Entre los sensores activos más comunes están los radares, que pueden trabajar bajo cualquier condición atmosférica, tanto de día como de noche. Estos sensores trabajan principalmente en la región de las micro ondas.

Sensores Pasivos

Los sensores pasivos de teledetección reciben la señal de una fuente de luz o energía externa reflejada por la superficie de los cuerpos. Estos sensores están incluidos sobre la mayoría de los satélites que manejan las bandas del espectro visible.

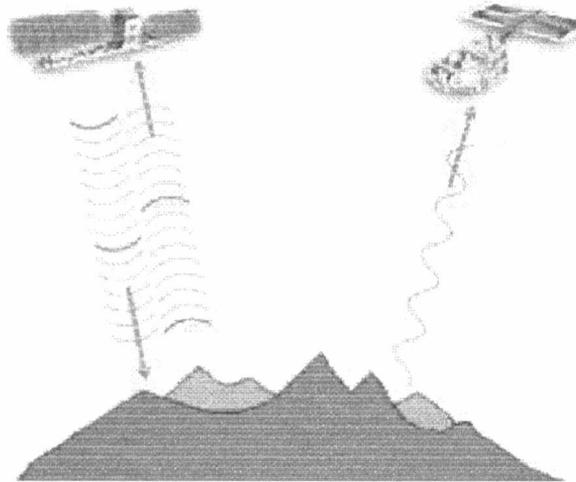


Figura 10: Tipos de sensores.

Radar

El radar (RAdio Detection And Ranging) es un sistema activo que emite un haz energético de microondas y registra la energía reflejada luego de interactuar con la superficie u objetos. Los radares también son denominados radiómetro activo de microondas y trabajan en banda comprendida entre 0.1cm y 1m del espectro electromagnético.

Gracias a que las longitudes de onda de los radares son mayores al tamaño a la mayoría de las partículas en la atmósfera, éstos puede trabajar en cualquier condición atmosférica, ganando la atención e interés de los científicos para realizar importantes aplicaciones sobre áreas con alta proporción de nubes, como en los países tropicales. También los radares a diferencia de los satélites ópticos registran datos en cualquier momento, tanto en el día como en la noche, debido a que emiten su propia fuente de energía y no tiene que requerir de la energía solar.

Los radares pueden ser instalados sobre aviones (aerotransportados) o sobre plataformas espaciales (satélites), éstos poseen una antena que transmite y/o recibe señales generando imágenes a alta resolución, donde pueden observarse características físicas de la superficie de la tierra.

Los radares primeramente pueden dividirse en dos grandes grupos, radares activos y radares pasivos:

- Los radares activos emiten pequeños pulsos de microondas en la dirección de interés y reciben y almacenan la energía dispersada por los objetos dentro de un campo de captura de la imagen.
- Los radares pasivos reciben niveles de radiación de microondas emitidas por los objetos en su ambiente natural.

De acuerdo con el tamaño de la antena, los radares también pueden dividirse en dos grandes grupos:

- Real Aperture Radar (RAR).
- Synthetic Aperture Radar (SAR).

RAR

Los RAR son equipos donde el tamaño de la antena es controlado por la longitud física de la antena. También son conocidos como radares no coherentes.

La ventaja de los equipos RAR esta en su diseño simple y en el procesamiento de los datos. Sin embargo su resolución es pobre para el rango cercano, misiones de baja altitud y longitudes de onda baja. El uso de estos datos estaría limitado para longitudes de onda mas corta y sería difícil aplicarlos a estudios atmosféricos o de dispersión, debido a que las misiones vuelan a baja altitud y su cobertura es pequeña. La resolución de la antena está limitada por la longitud de la antena.

La antena necesita tener varias veces el tamaño de la longitud de onda para reducir el ancho de banda de la señal emitida. Sin embargo es impráctico diseñar una antena suficientemente grande para producir datos de alta resolución.

SAR

Los SAR (Synthetic Aperture Radar) son sistemas de radares coherentes que generan imágenes de alta resolución. Una apertura sintética o antena virtual, consiste en un extenso arreglo de sucesivas y coherentes señales de radar que son transmitidas y recibidas por una pequeña antena que se mueve a lo largo de un determinado recorrido de vuelo u órbita. El procesamiento de la señal usa las magnitudes y fases de la señal recibida sobre sucesivos pulsos para crear una imagen.

Los puntos en los cuales sucesivos pulsos son transmitidos son considerados como largos arreglos sintéticos usados para generar la imagen SAR. Se envían pulsos de señal a los mismos puntos de la superficie terrestre en dos o más momentos distintos de la trayectoria del radar y la resolución que se obtiene es equivalente a la que si se utilizara una antena de similar longitud que la distancia entre los pulsos. Este concepto de larga antena virtual es la base de los radares de apertura sintética.

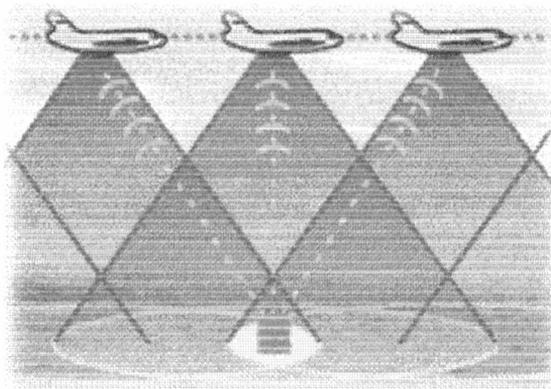


Figura 11: Obtención de una imagen con SAR.

Generación de apertura sintética: el objeto rojo es registrado desde distintas posiciones del sensor. Los SAR son instalados sobre aviones o plataformas espaciales y han servido para el mapa de la superficie de la tierra aun en condiciones atmosféricas adversas, también son herramientas útiles para obtener mapas de la superficie del mar.

Algunas aplicaciones de los sensores SAR son incluidos en diferentes modos de aplicación como son los ScanSAR, SpotlightSAR, polarimetricSAR, interferometricSAR o InSAR. A los RAR y SAR transportados en aviones se los denomina SLAR (Side Looking Airborne Radar) y difieren en el poder de resolución.

El equivalente óptico en un sistema de radar es una antena rectangular que transmite y recibe energía de microondas. La resolución es dependiente de la longitud focal en sensores ópticos y de la longitud de la antena en la dirección del vuelo del sistema.

Las antenas son análogas a los sistemas de lentes en los que una antena larga se puede comparar a la lente de un telescopio (longitud focal larga), mientras que una antena más corta es similar a una lente angulosa ancha (longitud focal corta). Para continuar la analogía, una antena larga proporciona a una imagen detallada o de alta resolución de un área pequeña, mientras que una antena corta provee una imagen de un área grande con menos detalle.

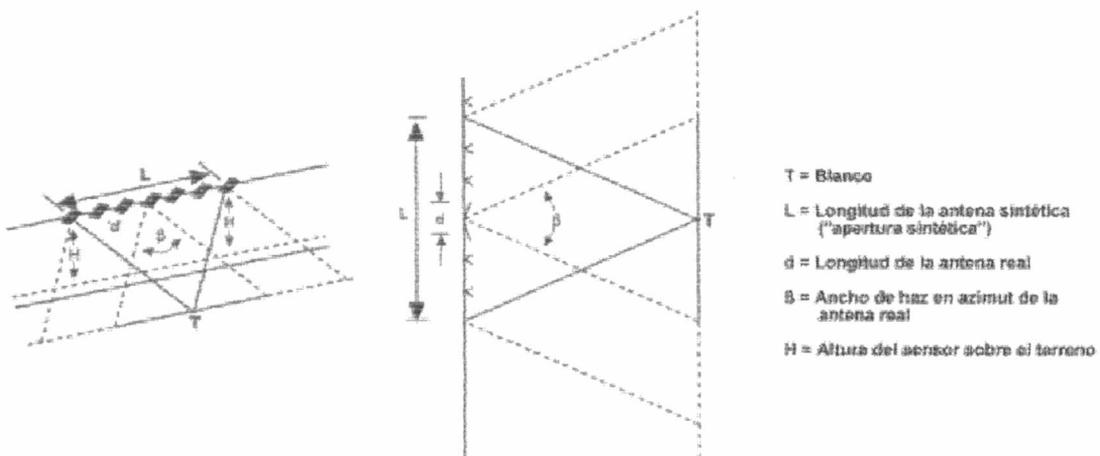


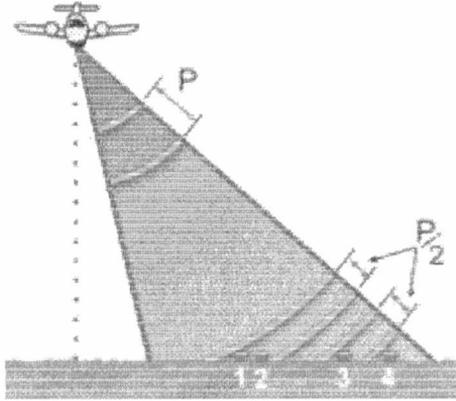
Figura 12: Resolución de una imagen con SAR.

Generación de una antena de apertura sintética de longitud L a partir de una antena real de longitud d. La resolución en un sistema de radar es controlada por la longitud del pulso de la señal y el ancho del rayo proveniente de la antena.

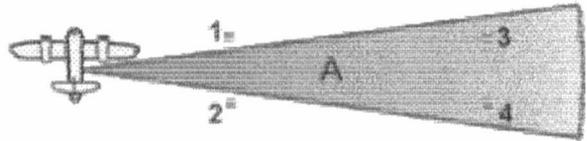
La longitud del pulso determina la resolución en la dirección de propagación de la energía (dirección del alcance). Pulsos más cortos dan lugar a una alta resolución en el alcance.

El ancho del rayo proveniente de la antena determina la resolución en la dirección del vuelo o del azimut. La amplitud de la señal es directamente proporcional a la longitud de onda del radar e inversamente proporcional a la longitud de la antena que la transmite.

Esto significa que la resolución se deteriora con la distancia a la antena. Para tener una alta resolución en la dirección de azimut, la antena de radar debe ser muy larga.



Resolución en el rango



Resolución en azimut.

5. Sistemas de Adquisición de Datos

Introducción

Como humanos, frecuentemente tomamos por hecho nuestro asombroso sistema de percepción. Vemos una copa sobre una mesa y automáticamente realizamos los movimientos para alcanzarla y beber su contenido. Al llevar a cabo esta simple tarea se requiere una compleja interacción de sensores, conocimiento, interpretación y coordinación. Se podría decir que los seres humanos constituimos un complejo sistema que toma información del mundo que lo rodea a través de los sentidos, la acomoda para que pueda ser interpretada, procesa de acuerdo al conocimiento que tenga acumulado y toma acciones. Todo esto bajo la supervisión de una computadora paralela ubicada en nuestro cerebro.

Un Sistema de Adquisición de Datos, basado en una Computadora, tiene una analogía directa con el sofisticado sistema de percepción y manipulación que presenta un ser humano. Efectivamente un Sistema de Adquisición de Datos consiste en la recolección de información, proveniente de diversas fuentes de información, y su digitalización para su posterior almacenamiento, análisis y presentación. Existen seis componentes básicos en la constitución de un Sistema de Adquisición:

- Computadora.
- Transductores y Sensores.
- Señales.
- Acondicionamiento de la señal.
- Hardware de Adquisición y Control.
- Manejadores ("*drivers*") y programas de aplicación.

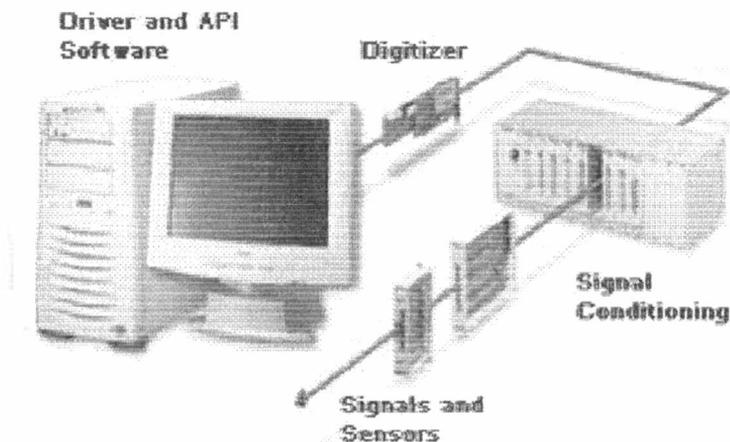


Figura 13: Data Acquisition System (Cortesía de National Instruments)

Computadora

En la actualidad, la computadora personal (Personal Computer - PC) forma parte de nuestra vida cotidiana. El crecimiento de la industria informática ha dado un gran salto en los últimos años acrecentando las capacidades de memoria, velocidad del

procesador y resolución para presentar la información. El bajo costo y un ambiente amigable para el usuario ha facilitado el acceso y manejo de las PCs que se encuentran en el hogar, la oficina, tiendas, museos, centros comerciales, etc. Generalmente se utiliza la PC para el proceso de la información, organización de los datos y comunicación en red.

Por otro lado, el creciente avance en el hardware y el software ha extendido sus funcionalidades en Sistemas de Control y Monitoreo bajo distintos ambientes: oficinas, electromedicina, establecimientos industriales, control de acceso a peajes, viviendas inteligentes, etc.; mientras les ha permitido a los científicos e ingenieros, agilizar su trabajo. Ahora ellos pueden automatizar sus tareas de investigación, diseño, desarrollo y medición y, por lo tanto, pueden aumentar su productividad, precisión y rendimiento [1].

Al presente, las computadoras personales vienen equipadas con procesadores Pentium, AMD o PowerPC con frecuencias de reloj que superan largamente el GHz y buses de alto rendimiento como el PCI, PXI, CompactPCI, IEEE1394 y los tradicionales USB y el cada vez menos frecuente bus ISA. En el caso de requerirse conexiones remotas se dispone de conectividad a redes Ethernet y conexiones seriales RS232 y RS485.

En los casos en que al Sistema de Adquisición de Datos se integra control instrumental, a través de algunos de los buses GPIB/VXI o serial, se tiene un sistema completo que incluye Medición y Automatización.

Sensores

Un Sistema de Medición, Adquisición y Control comienza con el fenómeno físico que se quiere caracterizar o medir. Este fenómeno físico puede ser la temperatura de un horno, la intensidad de luz de una fuente lumínica, la presión dentro de un depósito, la velocidad de un objeto, etc.

Un transductor es un dispositivo que transforma una magnitud física que se quiere medir en un voltaje o corriente eléctricos. La habilidad de un Sistema de Adquisición depende de la habilidad de los transductores de convertir las magnitudes físicas en señales que puedan medir el hardware de adquisición. Los transductores son sinónimos de Sensores en la terminología de los Sistemas de Adquisición y existen distintos tipos de ellos dependiendo de la aplicación. La Tabla 1 enumera algunos de los tipos de sensores o transductores mas conocidos y el tipo de fenómeno físico que pueden medir.

| Fenómeno | Transductor o Sensor |
|---------------------------|--|
| Temperatura | Termocuplas Resistores Variables con la Temperatura (RTDs) Termistores |
| Luz | Tubos de vacío Fotosensores |
| Sonido | Micrófonos |
| Fuerza y Presión | Medidor de deformación Transductores piezoeléctricos |
| Fluidos | Sensores de Efecto Hall |
| Posición y Desplazamiento | Potenciómetros Codificadores angulares |
| PH | Electrodos de pH |

Tabla 1

Diferentes tipos de sensores poseen distintos requerimientos para convertir la magnitud física en una señal eléctrica, más aún, pueden necesitar de circuitos electrónicos adicionales y muy complejos.

Señales

Las señales que producen los sensores pueden ser de dos, tipos a saber:

- Analógicas
- Digitales

Las señales analógicas pueden tomar cualquier valor respecto al tiempo. Algunos ejemplos de señales analógicas lo constituyen la temperatura, la presión, el sonido, etc. Las características principales de una señal analógica son la amplitud, la forma y la frecuencia [1].

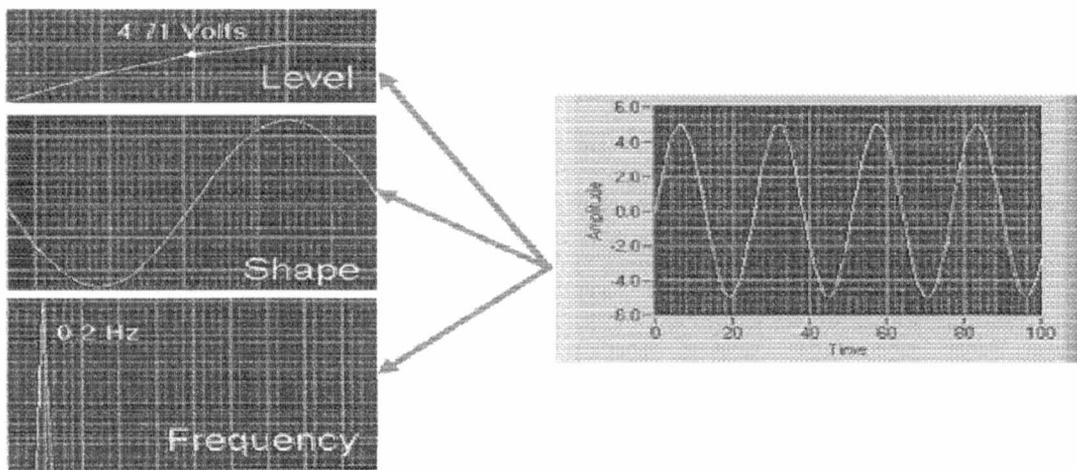


Figura 14: Señales Analógicas (Cortesía de National Instruments).

La amplitud de la señal es importante por cuanto da información de su intensidad. La intensidad de luz de una fuente lumínica, la temperatura de un horno o la presión dentro de un recinto dan muestra acerca de la importancia de la amplitud. Cuando se requiere medir la amplitud la variación temporal de la señal no es muy grande y lo más importante es la exactitud con que se la mide.

Algunas señales reciben un nombre propio como producto de su forma: seno, cuadrada, diente de sierra, triangular. La forma de una señal es tan importante como su amplitud por cuanto su conocimiento puede agregar información como ser valor pico, valor medio y pendiente. En general, en aquellos sistemas, en que la forma de la señal resulta de interés, la señal varía rápidamente respecto al tiempo. Ejemplos de este tipo de señales son los latidos del corazón, señales de video, sonidos, vibraciones y respuestas de circuitos electrónicos. En estos casos la exactitud sigue siendo un aspecto importante en el proceso de adquisición.

La frecuencia de la señal es otra de las características que requiere ser medida. En general, su valor no se mide directamente sino que se obtiene lo que se denomina el espectro de frecuencias a partir de una herramienta matemática conocida como Análisis de Fourier.

Cuando la frecuencia es la característica más importante se deben considerar la exactitud y la velocidad de adquisición. Si bien ésta última es menor que en el caso de requerirse la forma de la señal, debe respetarse el Teorema de Nyquist con el fin de no perder información.

El teorema de Nyquist establece que una señal debe ser muestreada por lo menos al doble de la máxima frecuencia presente en el ancho de banda de la misma para que pueda ser reconstruida. De otro modo las frecuencias mayores se manifestarán como frecuencias menores dentro del ancho de banda de la señal. La Figura 15 muestra una señal de 5MHz muestreada a una velocidad de 6 MS/seg (6 Mega muestras por segundo). Como se puede apreciar la señal de 5 MHz aparece como una señal de 1 MHz.

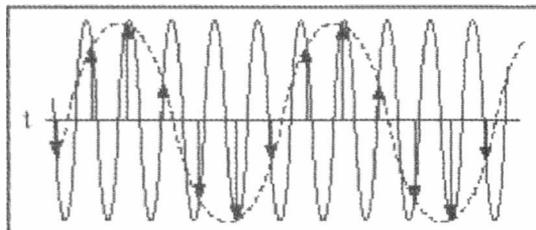


Figura 15

Las señales digitales no pueden tomar cualquier valor respecto al tiempo; es decir, las señales digitales sólo pueden tomar dos estados: bajo y alto. En general, este tipo de señales se encuentran referidas por familias de circuitos integrados que responden a distintas tecnologías. Por ejemplo las señales digitales TTL (Transistor Transistor Logic) responden a integrados contruidos con transistores bipolares y los valores de voltaje para un estado bajo se encuentran comprendidos entre 0V-0.8V, mientras que, para un estado alto van de 2.4V-5V. Las características que reúnen iteres en este tipo de señales son el estado y la velocidad de repetición.

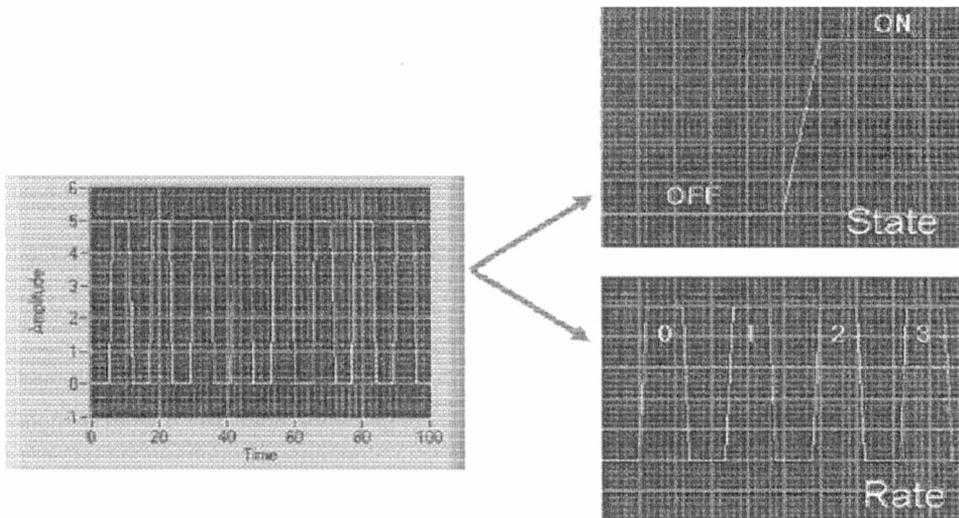


Figura 16: Señales Digitales (Cortesía de National Instruments)

El conocimiento del estado de una señal es importante cuando se requiere conocer el valor de una magnitud que ha sido digitalizada. Por ejemplo el conocimiento del estado de cada uno de los bits en un codificador angular para obtener la posición del objeto.

La velocidad de repetición se encuentra ligada a la velocidad de cambio de estado por unidad de tiempo. Por ejemplo la detección del estado, ON-OFF, en un sensor que detecta el pasaje de una leva, ubicada en el eje de un motor, para obtener la velocidad de rotación.

En cuanto las señales eléctricas, que generan los sensores, deben ser transmitidas para su posterior análisis y almacenamiento, las de tipo digital poseen ciertas ventajas que las hacen más atractivas:

- Inmunidad al ruido, las señales analógicas son más susceptibles que los pulsos digitales a la amplitud no deseada y variaciones de frecuencia y fase.
- Las señales digitales son más fáciles de procesar, almacenar y multiplexar que su contrapartida las analógicas.
- Los sistemas digitales utilizan la regeneración de las señales, en lugar de la amplificación por lo que son más resistentes al ruido electromagnético.
- Las señales digitales son más sencillas de medir y evaluar.
- Los sistemas digitales están mejor equipados para mejorar la tasa de error que los sistemas analógicos ya que disponen de códigos para la detección y corrección de errores.

Acondicionadores

En muchas oportunidades los sensores generan señales muy dificultosas o peligrosas de medir con el hardware de adquisición. En los casos que deben manejarse altos voltajes, señales muy débiles o muy grandes, ambientes muy ruidosos desde el punto de vista electromagnético, etc. Entre los acondicionadores de señal podemos mencionar:

- Amplificación.
- Atenuación.

- Aislamiento.
- Funciones analógicas.
- Multiplexado analógico y digital.
- Filtrado.
- Comparación.
- Muestreo y almacenamiento analógico.
- Conversión analógica-digital.
- Conversión digital-analógica.

Hardware de Adquisición y Control

Estos dispositivos actúan entre la computadora personal y el mundo externo. Su función primaria es la de digitalizar (convertir de analógicas en digitales) las señales provenientes del exterior para que puedan ser interpretadas por la computadora. Entre estos dispositivos se pueden enumerar los siguientes:

- Entradas y Salidas Analógicas.
- Entradas y Salidas Digitales.
- Temporizadores y Contadores.
- Interfases Multifunción.

Este hardware se encuentra disponible para distintos buses. Para aplicaciones distribuidas se dispone de una amplia gama de dispositivos que pueden conectarse a distintos buses de campo, sistemas embutidos y redes ethernet.

Manejadores ("drivers") y programas de aplicación.

El Software transforma a una Computadora Personal y el Hardware de Adquisición (DAQ) en un sistema completo de Adquisición de Datos, análisis y presentación. Sin el software que controla al Hardware de Adquisición (DAQ) este último no funciona apropiadamente. Los manejadores ("drivers") del Hardware de Adquisición constituyen una capa intermedia entre el Programa de Aplicación y el Hardware. En general, los manejadores colaboran con los programadores evitando la necesidad de programar a nivel de registros o comandos complicados para acceder a las funcionalidades del hardware.

La capa correspondiente al programa de aplicación puede estar constituida por un ambiente de desarrollo dentro del cual se desarrolla una aplicación para resolver un problema específico; otra posibilidad es un programa genérico que posee una configuración por defecto que puede ser ajustada a distintos tipos de problema. Normalmente el Software de Aplicación le agrega capacidades relacionadas con el análisis y la presentación.

El elemento más importante

Existe un elemento adicional que está siempre presente en los Sistemas de Adquisición pero que rara vez se lo muestra en los diagramas en bloques: el "***homo sapiens***". Los sistemas son inspirados, diseñados, construidos, programados, verificados, perfeccionados y usados por mujeres y hombres para servir a propósitos humanos.

Es por esta razón que un Sistema de Adquisición debe incluir facilidades que le permitan a los humanos comunicarse, leer sus datos y proveer ajustes e instrucciones: visualizadores, sonidos, teclados, pantallas sensibles al tacto, ratones, joysticks, llaves, sintetizadores de voz, etc.

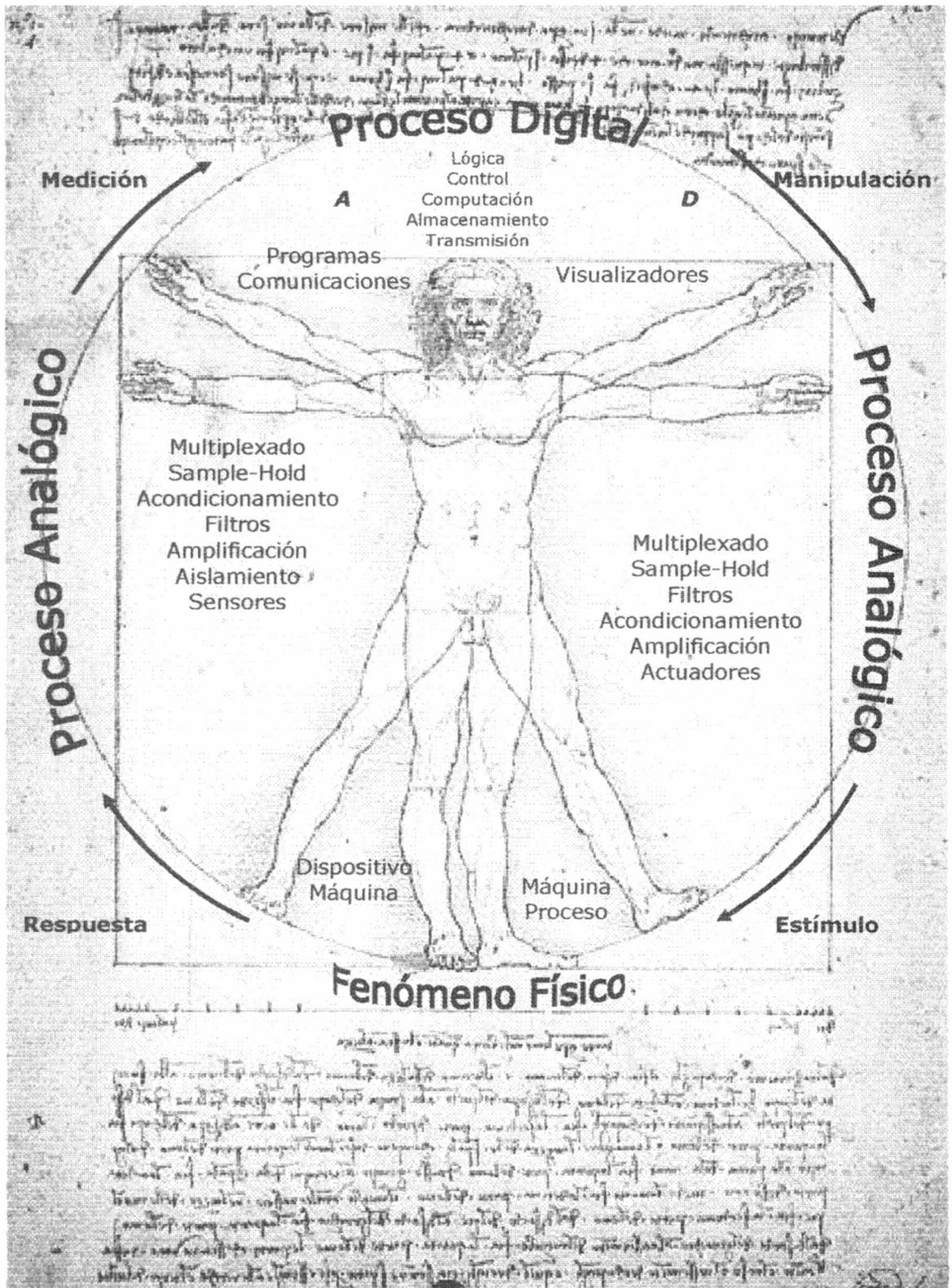


Figura 17: Study of proportions, from Vitruvius's De Architectura Leonardo Da Vinci

Instrumentación Virtual

De acuerdo a lo expresado en los párrafos previos se puede afirmar que, dentro de un Sistema de Medición, Adquisición y Control, la Computadora Personal (PC) se ha convertido en el componente de procesamiento por excelencia. Su adopción en la instrumentación de ensayos, mediciones y automatización ha dado lugar a que aparezca un nuevo tipo de instrumentación, denominado, ***Instrumentación Virtual***.

Cuando se habla de Instrumentación Virtual se hace referencia a *un nivel de software y hardware añadido a un computador de propósitos generales de modo tal que los usuarios pueden interactuar con el computador como si fuese un instrumento electrónico tradicional*.

Un instrumento virtual consiste de una computadora estándar o estación de trabajo equipada con una poderosa aplicación, una óptima plataforma de hardware, basada en tarjetas que se enchufan ("plug-in boards"), y manejadores ("drivers" e "instrument drivers") que, en conjunto, realizan las funciones correspondientes a los dispositivos y/o instrumentos tradicionales. Los instrumentos virtuales constituyen el desplazamiento de la instrumentación tradicional, centrada en el hardware, hacia la instrumentación centrada en el software que explota el poder de cálculo, productividad, visualización y conectividad de las computadoras de escritorio y las estaciones de trabajo. Esto último significa que, a pesar del desarrollo que ha tenido el hardware en las últimas décadas, a partir de circuitos integrados más complejos y rápidos, realmente es el software el que ha sido la palanca que ha permitido construir esta fundación de hardware sobre la que se ha creado la instrumentación virtual. Esto le permite a los ingenieros y científicos crear su propio sistema de instrumentación, definido por el usuario ("user defined"), en lugar de quedar limitados con los sistemas tradicionales, definidos por el fabricante ("vendor defined").

A excepción de los componentes y circuitos especializados que se pueden encontrar en un instrumento tradicional, la arquitectura general de un instrumento no difiere mucho de un instrumento virtual basado en una computadora personal. Ambos requieren de uno o varios procesadores, puertos de comunicación (serial, GPIB), capacidad de visualización y módulos de adquisición. La diferencia entre ambos radica en la flexibilidad y en la posibilidad que tienen los instrumentos virtuales de adaptarse a las necesidades de solución de un problema en particular. Un instrumento particular, en general, debe disponer del hardware apropiado para realizar cierta clase de procesamiento; en cambio, un instrumento virtual puede obtener dicha clase de procesamiento por software. Por supuesto, esto último indica un compromiso que deberá atenderse. En casi todos los casos las funciones implementadas por software son más lentas y esto significa que la reducción de velocidad deberá ser compatible con la velocidad de actualización de los datos que se requieran procesar.

El software es el elemento más importante en un instrumento virtual. Mediante la disponibilidad de las herramientas adecuadas se pueden crear poderosas aplicaciones que constituyen un sistema de instrumentación virtual. Estas aplicaciones definen cuándo y cómo se adquieren los datos, cómo se los procesa, manipula y almacena y cómo los resultados son presentados al usuario a través de una interfase gráfica apropiada. Un software poderoso puede incluir inteligencia y capacidad de decisión de modo que el instrumento se adapta cuando se producen cambios inadvertidos o cuando los requerimientos de procesamiento disminuyen.

Un elemento importante en el software de un instrumento virtual es la modularidad. Cuando se debe tratar con un gran proyecto se lo divide en módulos que solucionan aspectos parciales del problema. Esta división en tareas lo hacen más manejable y fácil de depurar, mientras se disminuyen las dependencias que pueden causar comportamientos no esperados. Más aún se puede diseñar un instrumento virtual para solucionar cada una de estas tareas que luego juntos constituyen un sistema que permite darle solución a una tarea mayor.

Con el avance en la tecnología de redes y la conectividad con la WEB se puede agregar que un instrumento virtual no está confinado a una computadora personal. En realidad con los recientes desarrollos en redes de área local e Internet es muy común que los instrumentos virtuales integren sistemas de monitoreo, control remoto de procesos y de visualización de datos desde múltiples ubicaciones.

Por último se puede afirmar que la instrumentación Virtual facilita la integración de hardware de adquisición, instrumentos tradicionales y otros dispositivos, a través del bus GPIB/IEEE 488 o conexión serie RS 232/RS485, PLCs (Programmable Logic Controllers). El resultado de tal integración puede dar lugar a sofisticados **Sistemas de Medición y Automatización**.

El estándar SCPI

La integración de instrumentos tradicionales en un Sistema de Adquisición de Datos ha dado lugar al desafío que representa la estandarización de un protocolo de comunicación. Efectivamente, desde el punto de vista del software esta estandarización no fue inmediata y, aún en la actualidad, se encuentra en pleno desarrollo e implementación.

Desde el punto de vista de los instrumentos de laboratorio los fabricantes de los mismos vienen trabajando, desde hace varios años, en un proceso de estandarización de las interfases eléctricas y mecánicas. Sin embargo, poco o casi nada se hacía desde el punto de vista de los mensajes enviados a estas interfases. Como consecuencia de ello cada fabricante desarrollaba su propio conjunto de mensajes requiriendo a los usuarios el aprendizaje de un nuevo vocabulario por cada instrumento. Además, la proliferación de comandos causaba que los usuarios perdieran una gran cantidad de tiempo aprendiendo a programar cada instrumento, el mantenimiento se hacía difícil y la actualización de un sistema era complicada.

El estándar conocido como SCPI –“Standard Commands for Programmable Instrumentation”- define un conjunto de comandos para controlar dispositivos de verificación y medida.

Desde el punto de vista del estándar SCPI un sistema de instrumentación es una colección de dispositivos de verificación y medición conectados mediante un bus a una computadora de control llamada Controlador del Sistema (“System Controller”).

Dentro del sistema el controlador del mismo, en este caso la Computadora Personal, puede enviar uno o más comandos al bus. Estos comandos se denominan mensajes. Los instrumentos, a su vez, pueden devolver mensajes. Los mensajes pueden contener el resultado de una medición, la configuración de un instrumento, una condición de error, etc. En el caso particular en que un mensaje genera una respuesta se lo identifica como una pregunta (“query”)[2].

En el pasado, los comandos que controlaban una función variaban entre instrumentos diferentes. El estándar SCPI provee un lenguaje uniforme y consistente que permite controlar los instrumentos de medición y verificación. Esta consistencia se manifiesta en forma vertical y horizontal. La consistencia vertical se da en la programación de una clase de instrumentos, por ejemplo, el comando para realizar la lectura del voltaje de DC es el mismo para todos los multímetros. Por otro lado la consistencia horizontal se da cuando se requiere implementar comandos comunes a instrumentos pertenecientes a diferentes clases de distintos fabricantes, por ejemplo el comando de disparo utilizado en osciloscopios, contadores, multímetros, analizadores de imagen, etc.

El principio en que se basa la consistencia en la programación de los instrumentos, que respetan el estándar SCPI, se encuentra en la reducción de los caminos para controlar funciones similares en los mismos. La filosofía del estándar SCPI es que las mismas funciones se controlen con los mismos comandos.

Adicionalmente el estándar SCPI provee distintos niveles de control. Por un lado se dispone de simples comandos que permiten al usuario una programación rápida y fácil; mientras que, existen otros comandos, más detallados, que darían lugar a una programación más tradicional.

Por último el estándar se puede expandir con el objeto de ampliar sus prestaciones a nuevas necesidades y requerimientos que presenten la tecnología a través de nuevos instrumentos o nuevas prestaciones disponibles en los ya tradicionales.

Uso del Estándar SCPI

La principal ventaja del uso del estándar es la disminución del tiempo de programación de un instrumento luego que se ha programado el primero. Este beneficio se extiende a programadores de lenguaje C, BASIC, FORTRAN, etc, aquellos que desarrollan manejadores de instrumentos para sistemas de medición automáticos y aquellos que implementan los paneles frontales a ser utilizados en un entorno de programación.

El estándar SCPI puede ser utilizado en la capa superior de una aplicación que se hace independiente del hardware y también se lo puede implementar en el controlador que se conecta con un instrumento.

Para lograr su cometido el estándar define los comandos, parámetros, datos y condiciones de estado.

A través del uso de la consistencia que presenta el estándar SCPI se "facilita" la tarea de reemplazar un instrumento en un sistema. Esto significa que el estándar SCPI no provee los medios para obtener el intercambio entre instrumentos, sino, más bien colabora definiendo comandos y respuestas. El estándar no define la exactitud, resolución, conexión y funcionalidad de los instrumentos que son necesarios para asegurar el intercambio y de esa manera no afectar el hardware y el software del sistema[3].

Manejadores de Instrumentos

Luego de aceptado el estándar SCPI como la arquitectura para intercambiar mensajes con los instrumentos se debe pasar a la etapa de implementación del mismo. Normalmente, las funciones de comunicación con los instrumentos, que hacen uso de

los mensajes y comandos establecidos en el estándar, se agrupan en módulos de programa. Estos módulos residen en una estación de trabajo y se los conoce con el nombre de Manejadores de Instrumentos (“Instruments Drivers”). Constituyen un nivel superior de encapsulamiento y de abstracción de hardware, mientras proveen un modelo fácil de usar para lograr el acceso completo a todas las facilidades del dispositivo y/o instrumento a través de un conjunto de APIs (“Application Programming Interface”).

Actualmente, los Manejadores de Instrumentos se presentan en dos versiones que intentan satisfacer distintas necesidades de instrumentación. Existe una versión que responde al estándar VISA –“Virtual Instrumentation Software Architecture”- mediante la cual se desarrollan manejadores que permiten controlar y comunicarse con un instrumento a través de un modelo sencillo de programación. Por otro lado el estándar conocido como IVI –“Interchangeable Virtual Instruments”- permite desarrollar manejadores más complejos para sistemas que requieren del intercambio y simulación de instrumentos. En este caso se han definido especificaciones para ocho clases de instrumentos con el objeto de maximizar la reutilización del software entre instrumentos pertenecientes a una misma clase, aunque pertenezcan a distintos fabricantes[2].

En el caso del estándar VISA los manejadores de instrumentos están constituidos por un conjunto de rutinas que controlan un instrumento. Cada rutina del manejador de instrumentos se corresponde con una operación programable del instrumento como ser “configuración”, “leer datos”, “escribir datos” o “disparar un evento”.

Por último, se puede agregar que en el caso del estándar IVI se dispone de una programación que permite aumentar el rendimiento a través de una “state-caching machine” que cambia el estado de los instrumentos en forma incremental y sólo cuando es necesario. Adicionalmente, estos manejadores permiten el intercambio de instrumentos, dentro de una misma clase, entre distintos fabricantes y la simulación de envío de comandos y la obtención de datos medidos.

La utilización de manejadores de instrumentos en el desarrollo de aplicaciones permitiría obtener los siguientes beneficios:

- APIs (“Application Programming Interfaces”) estándar, comunes, que reducirían el tiempo de aprendizaje para la escritura de nuevos manejadores.
- La simulación del instrumento le permitiría a los desarrolladores verificar el código sin la necesidad de disponer del hardware.
- Facilidad de conexión con ambientes de desarrollo a través de la disponibilidad de paneles frontales que permiten un acceso intuitivo a las funciones del manejador.
- El intercambio entre instrumentos reduce el tiempo y el esfuerzo necesario para integrar dispositivos en un sistema existente o nuevo.

Este último aspecto, el de intercambio de instrumentos, es muy importante por cuanto previene acerca de la obsolescencia del hardware. Adicionalmente se puede decir que se presenta en tres niveles: a) nivel de arquitectura que facilita la reutilización del código; b) el establecimiento de clases de dispositivos permite el intercambio sintáctico mediante la cual un dispositivo se puede reemplazar por otro con mínimos

cambios y c) el máximo nivel de intercambio que se logra a través del uso de señales estándar.

6. Sensores de Posición

Existe una gran variedad de sensores que se aplican para detectar y medir desplazamiento y movimiento. Entre los más utilizados se encuentran los siguientes:

a) Potenciométricos; b) Transformador Diferencial de Variación Lineal (LVDT) y c) Encoders.

Potenciómetros

Este tipo de sensor posee un eje que se desplaza o que rota, de acuerdo al tipo de movimiento que se requiera detectar o medir, provocando el desplazamiento o la rotación de un contacto deslizante sobre un material resistivo. En esencia este tipo de sensores está constituido por un potenciómetro en el cual el contacto deslizante se desplaza o rota con el movimiento del eje. El valor de la resistencia medido entre el contacto deslizante y uno de los extremos del potenciómetro es proporcional al movimiento, la posición o el desplazamiento.

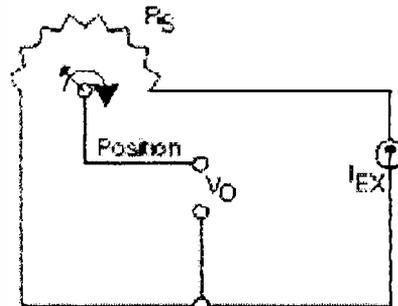


Figura 18: Sensor Potenciométrico.

Este tipo de sensores convierten un movimiento lineal o angular en un cambio en el valor de una resistencia variable que, a su vez, puede dar lugar a una señal eléctrica caracterizada por una corriente o un voltaje. Con el fin de no perder las características lineales del potenciómetro no se lo utiliza como divisor de tensión si no que se lo integra a la red que define la ganancia en un amplificador operacional.

Este tipo de sensores sufren por la acción mecánica sobre el eje y la fricción en el contacto deslizante. Además poseen baja resolución e introducen ruido eléctrico.

LVDTs

Este tipo de sensor basa su funcionamiento en un transformador con dos secundarios, cuyos arrollamientos se encuentran en serie y en oposición, y un núcleo de material magnético que se desplaza libremente entre el primario y los secundarios. El voltaje diferencial, V_o , entre ambos secundarios es nulo cuando el núcleo se encuentra en el centro y aumenta a medida que el núcleo magnético se desplaza hacia alguno de los extremos. Adicionalmente se produce un cambio de fase en el voltaje de salida, respecto al voltaje de entrada, cuyo signo

depondrá del extremo hacia el cual se desplaza el núcleo.

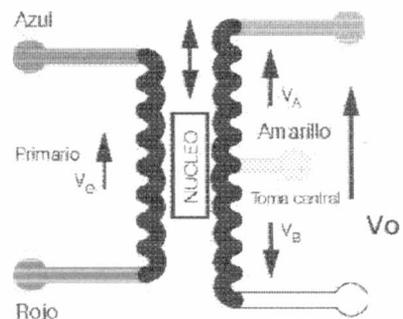


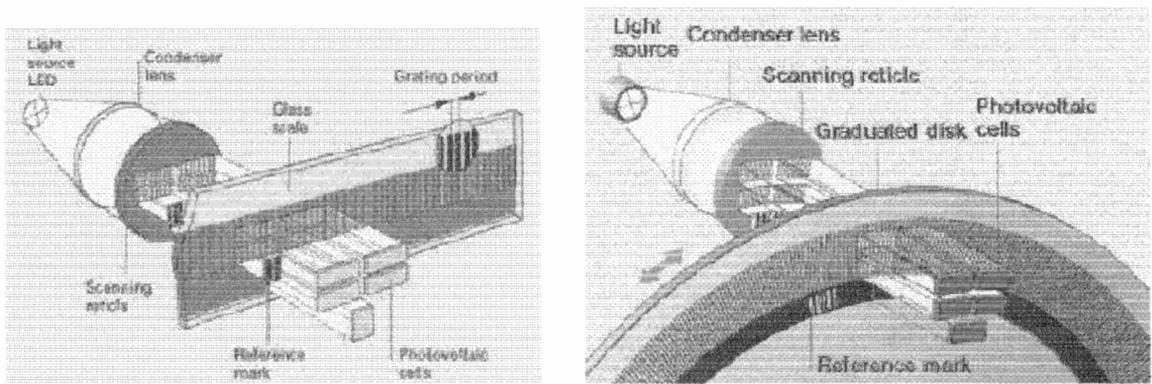
Figura 19: Sensor LVDT.

Este tipo de sensores pueden dar una salida con una linealidad del 0,25% y excelente resolución, sólo limitada por el circuito de acondicionamiento que lo acompañe.

Encoders

Encoders Incrementales y Absolutos:

Los **encoders incrementales** sirven para medir posición lineal o angular de un objeto que se está desplazando.[5] Generan impulsos al girar su eje y el número de impulsos por vuelta y por unidad de tiempo puede dar una medida de velocidad, longitud o de posición. Básicamente constan de un disco transparente, el cual tiene una serie de marcas opacas colocadas radialmente y equidistantes entre sí; de un elemento emisor de luz (como un diodo LED) y de un elemento fotosensible que actúa como receptor. El eje cuya posición angular se va a medir va acoplado al disco. [6]



Gráfica tomada de la página de www.heidenhain.com

El funcionamiento es el siguiente: cuando el sistema comienza a funcionar el emisor de luz empieza a emitir; a medida que el eje vaya girando, se producirán una serie de pulsos de luz en el receptor, correspondientes a la luz que atraviesa los huecos entre las marcas. Llevando una cuenta de esos pulsos es posible conocer la posición del eje. Sobre este esquema básico es habitual encontrar algunas mejoras. Por ejemplo se suele introducir una franja de marcas por debajo, desplazada de la anterior, para poder controlar el sentido de giro; además se suele incluir una marca de referencia que nos ayudará a saber si hemos completado una vuelta.[6]

Se dice que es relativo porque se calcula distancia relativa entre marcas. [5]

Realmente los Encoders incrementales miden la velocidad de giro, pero podemos extrapolar la posición angular. Como es lógico la resolución de este tipo de sensores depende directamente del número de marcas que podamos poner físicamente en el disco.[6]

Se pueden clasificar, según su función en **unidireccionales** (un solo canal de salida A), utilizados siempre que no es necesario detectar la dirección de rotación, tal como sumar o restar en contadores o tacómetros y **bidireccionales** (con dos canales de salida A y B), que permiten detectar el sentido de rotación del eje; en este caso el canal B está desfasado 90° eléctricos respecto al canal A. Se puede disponer de una tercera señal (canal 0) de referencia o cero que proporciona un impulso a cada vuelta del eje; adicionalmente esta señal puede sincronizarse respecto al canal A, B o respecto a ambos.[6]

La función de los **encoders absolutos** es similar a la de los anteriores, medir la posición angular, Sin embargo en este caso lo que se va a medir no es el incremento de esa posición, sino la posición exacta.[6] Se dice que es absoluto porque se determina la distancia con un código único para cada posición [5] . La disposición es parecida a la de los encoders incrementales. También se dispone de una fuente de luz, de un disco graduado y de un fotorreceptor. La diferencia estriba en la graduación o codificación del disco. En este caso el disco se divide en un número fijo de sectores (potencia de 2) y se codifica cada uno con un código cíclico (un código cíclico se puede definir como cualquier código en el que, para cualquier palabra de código, un corrimiento circular produce otra palabra de código, un ejemplo es el código Gray[4]), este código queda representado en el disco por zonas transparentes y opacas dispuestas radialmente. No es necesaria ninguna mejora para detectar el sentido de giro, ya que la codificación de los distintos sectores angulares es absoluta.[6] La resolución de estos sensores es fija y viene dada por el número de anillos que posea el disco, o lo que es lo mismo, el número de bits del código utilizado. Normalmente se usan códigos de 8 a 19 bits.[6]

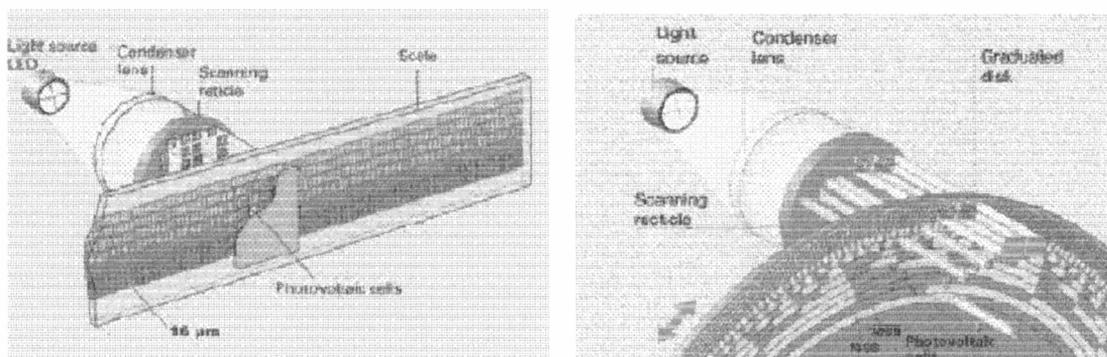


Figura 20: Gráfica tomada de la página de www.heidenhain.com

Tanto los encoders Absolutos como incrementales pueden presentar problemas debido a la gran precisión que es necesaria en el proceso de fabricación. además son dispositivos especialmente sensibles a golpes y vibraciones, estando su margen de temperatura de trabajo limitado por la presencia de componentes electrónicos. [5, 8] La conmutación ambiental puede ser una fuente de interferencias en la transmisión óptica. [7]

El código Gray

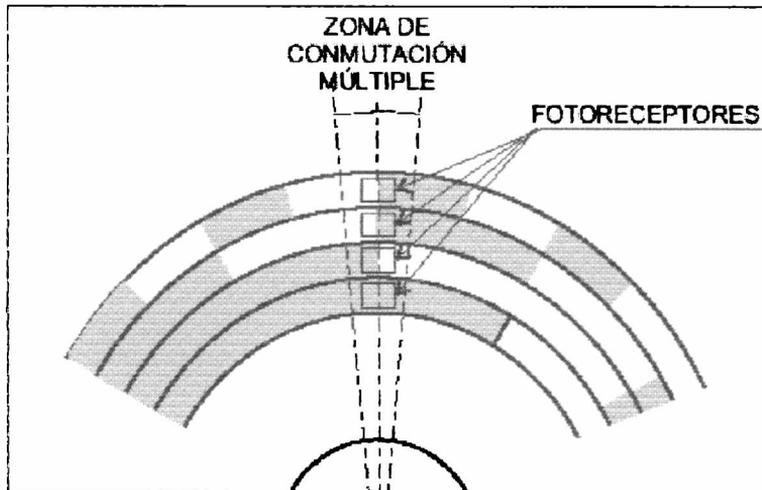
Como ya se vio antes los encoders absolutos presentan importantes diferencias desde el punto de vista funcional con respecto a los encoders incrementales. Mientras en los encoders incrementales la posición esta determinada por el computo del número de impulsos con respecto a la marca de cero, en los encoders absolutos la posición queda determinada mediante la lectura del código de salida, el cual es único para cada una de las posiciones dentro de la vuelta. Por consiguiente los encoders absolutos no pierden la posición real cuando se corta la alimentación (incluso en el caso de desplazamientos), hasta un nuevo encendido (gracias a una codificación directa en el disco), la posición esta actualizada y disponible sin tener que efectuar, como en el caso de los encoders incrementales la búsqueda del punto de cero.[4]

Analicemos ahora el código de salida que se deberá utilizar para definir la posición absoluta. La elección más obvia es la del código binario, porque fácilmente puede ser

manipulado por los dispositivos de control externo para la lectura de la posición, sin tener que efectuar operaciones particulares de conversión. En vista que el código se toma directamente desde el disco (que se encuentra en rotación) la sincronización y la captación de la posición en el momento de la variación entre un código y el otro se vuelve muy problemática. En efecto, si por ejemplo tomamos dos códigos binarios consecutivos como 7 (0111) 8 (1000), se nota que todos los bits del código sufren un cambio de estado: una lectura efectuada en el momento de la transición podría resultar completamente errónea porque es imposible pensar que las variaciones sean instantáneas y que se produzcan todas en el mismo momento. Debido a este problema se utiliza una variante del código binario: El **Código Gray**, el cual tiene la particularidad que al "pasar" entre dos códigos consecutivos (o desde el ultimo código al primero) solo cambia un bit.[4]

Los sistemas de posicionamiento van cambiando su posición y pueden hacerlo despacio o repentinamente; por ejemplo si un sensor de posición absoluta que utilizase el código binario natural estuviera leyendo 1000 (8) y la lectura siguiente fuera 1100 (12) quedará la duda si se trata de una transición a 7, que cambia todos sus bits, o que el sistema ha ido bruscamente a la posición 12. Si se utiliza el código Gray esta incertidumbre no existe.

| DECIMAL | BINARIO | GRAY |
|----------------|----------------|-------------|
| 0 | 0000 | 0000 |
| 1 | 0001 | 0001 |
| 2 | 0010 | 0011 |
| 3 | 0011 | 0010 |
| 4 | 0100 | 0110 |
| 5 | 0101 | 0111 |
| 6 | 0110 | 0101 |
| 7 | 0111 | 0100 |
| 8 | 1000 | 1100 |
| 9 | 1001 | 1101 |
| 10 | 1010 | 1111 |
| 11 | 1011 | 1110 |
| 12 | 1100 | 1010 |
| 13 | 1101 | 1011 |
| 14 | 1110 | 1001 |
| 15 | 1111 | 1000 |



El código Gray es 'no aritmético', es decir, no tiene pesos asignados a cada bit en una combinación. En efecto los términos 000, 101, etc. no denotan un valor matemático real (a diferencia de los demás códigos) sino uno de los X valores que puede tomar una variable. Por lo tanto partiendo de una variable que pueda tomar X valores, se toma un número n de bits que garantice $2^n > X$, se ordenan estos estados de la variable conforme a las normas de Gray de cambio entre dos estados sucesivos. Por ejemplo se necesitan, como se vio en la tabla anterior, cuatro bits para representar todos los números decimales del 0 al 15 y la conversión de código binario a código Gray se realiza con el siguiente procedimiento:

- El bit más significativo en el código Gray, es el mismo de código binario.
- Operando de izquierda a derecha, se suma cada par adyacente de los bits en código binario para obtener el siguiente bit en código Gray, descartando los acarros.

Para la conversión de código Gray a código Binario se siguen los siguientes pasos:

- El bit más significativo en el código Binario, es el mismo al correspondiente bit en código Gray.
- A cada bit del código binario generado, se le suma el bit en código gray de la siguiente posición adyacente. Se descartan los acarros.

El código Gray es un código reflejado, es decir que el código de n bits se puede generar en función del código de $n-1$ bits reflejándolo.

Gray dos bits

Gray tres bits

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 |

| | | |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 1 | 1 |
| 1 | 0 | 1 |

| | | |
|---|---|---|
| 1 | 0 | 0 |
|---|---|---|

Cuando el número de posiciones no es una potencia de dos, la propiedad de cambiar un solo bit se pierde al pasar el último valor al primero y viceversa. Para no perder la característica de la conmutación de un solo bit, se utiliza el código Gray quebrado o bien con exceso a N, haciendo corresponder a la posición 0 el código Gray correspondiente al valor N, donde N es el número que, substrayéndolo al código Gray convertido a número binario da el valor exacto de la posición. [4]

Ventajas del código Gray: [4]

- Es muy conveniente para fines de medición.
- Los dos dígitos mas significativos son los mismos tanto en el sistema binario como en el de Gray.
- En comparación con otros sistemas, la codificación Gray no genera grandes errores (eventualmente, en dispositivos mecánicos el error medio en la transmisión es menor cuando se hace uso de este código).

Un ejemplo de aplicación

Como ya hemos comentado, para que un robot realice su tarea de forma eficiente, rápida e inteligente es preciso que tenga conocimiento de una serie de parámetros o características internas. Tales características son muy diversas y se intentará controlar unas u otras en función de las necesidades. Por ejemplo puede ser necesario controlar la temperatura a la que está cierta parte del robot o la presión de sus ruedas. Sin embargo hay ciertas características que resulta interesante poder controlar para la mayoría de los robots: la posición, la velocidad y la aceleración. El problema de poder determinar la posición en la que se encuentra el robot en un momento determinado es uno de los más importantes e interesantes en el campo de estudio de la robótica. Actualmente no existe un método infalible y universal para calcular la posición, sino que, por el contrario, existen una serie de métodos basados en distintas técnicas que intentan resolver el problema. En la mayoría de los casos reales, la solución adoptada pasa por el empleo de varios de estos métodos. Son muy utilizados los Encoders incrementales y absolutos[8].

7. Entrada/Salida Digital

Conceptos Generales

El objetivo de las Unidades de Entrada /Salida (E/S), o de intercambio de información, es realizar la conexión, de la Unidad Central de Proceso (UCP) de un computador, con una gran variedad de dispositivos periféricos.

Esta conexión presenta unas características muy especiales, tales como las siguientes:

- Los periféricos tienen unas velocidades de transmisión que abarcan desde unos pocos bytes por segundo, hasta más del millón de bytes por segundo, sin embargo esta velocidad es sensiblemente inferior a la de la CPU.
- Los periféricos suelen usar un ancho de palabra de un byte, lo que no coincide con la palabra de la mayoría de los computadores.
- Algunos periféricos son de lectura, otros de escritura, y finalmente otros lo son de lectura y escritura simultáneamente.

El problema del intercambio de información entre UCP y periféricos es relativamente complejo.

Por un lado, hay que establecer un mecanismo que permita transmitir la información, mecanismo que debe considerar aspectos tales como direccionamiento o selección del periférico, forma de establecer el camino para el envío de datos, posible conversión serie/paralelo, posible conversión de códigos, etc.

Por otro lado hay que establecer un mecanismo de control que determine el origen y el destino de la información, la cantidad de ella a transmitir, los códigos de protección, etc.

Estos mecanismos se reparten entre el controlador del periférico, la unidad de control de la UCP y los programas de entrada/salida.

Por todo ello, hay que diferenciar entre lo que es una Transferencia elemental de información, una transferencia de Bloque y una operación de entrada/salida.

La **Transferencia Elemental de Información** tiene por objeto el envío o la recepción de una única unidad de información (un byte o una palabra), ya sea ésta un dato o una palabra de control. **La Transferencia de Bloque** se encarga de enviar o recibir un bloque de información, como puede ser el contenido de un sector de un disco. La operación de Entrada/Salida tiene por objeto la transferencia de un conjunto de datos, garantizando que ésta se produce correctamente[9],[10].

OPERACIÓN DE ENTRADA/SALIDA

!.2 Transfiere uno o varios bloques de datos

- Comprueba el estado del periférico
- Trata los posibles errores de la transferencia
- Se realiza por Software
- Se basa en los niveles de transferencia de bloque y elemental

!.1 TRANSFERENCIA DE BLOQUE

- Mueve un bloque de datos sincronizando con el periférico
- Se realiza por Software o automáticamente por DMA
- Su duración la marca la velocidad del dispositivo periférico
- Se basa en, transferencias elementales, que deben realizarse en el momento en que el periférico lo necesite
- Realiza almacenamiento temporal de la información

1. TRANSFERENCIA ELEMENTAL

- Mueve un dato o una palabra de control
- Se realiza por Hardware, entre la UCP y el controlador del periférico
- La UCP y el controlador del periférico se sincronizan mediante señales eléctricas
- Su duración es del orden de una instrucción y no depende de la velocidad del dispositivo periférico

Los aspectos mas importantes que caracterizan a cada tipo de transferencia son los siguientes:

- Transferencia elemental:
 - Se realiza a la velocidad de los circuitos electrónicos
 - Comunicación física entre el periférico y la UCP, para transmisión de un bit, un byte o una palabra. Habrá que considerar los distintos mecanismos físicos de comunicación, con sus respectivas ventajas e inconvenientes, así como las señales que son necesarias para realizar la comunicación, con sus respectivos cronogramas y métodos de sincronización.

- Control de los periféricos. Este control incluye la interrogación y modificación de su estado, por ejemplo su encendido, apagado, su disponibilidad, etc. Para ello la UCP debe ser capaz de leer y gobernar una serie de señales de control del periférico.
- Transferencia de Bloque
 - Sincronización de la UCP y del periférico. La UCP suele ser significativamente más rápida que el periférico, por lo que hay que establecer un mecanismo que permita que ésta sepa cuando puede enviar o debe recibir un dato. No debe confundirse esta sincronización con la requerida a nivel de transferencia elemental, para que una palabra o byte se transfieran correctamente.
 - Cuenta de los bytes o palabras transmitidas. Esta función es necesaria en transferencias múltiples, donde se tiene que llevar una cuenta de las palabras transmitidas, para finalizar la operación en su momento.
 - Detección de errores mediante códigos de paridad o polinomiales, con repetición de la transferencia en caso necesario.
 - Conversión serie/paralelo. La adaptación de los datos al distinto ancho de palabra de la UCP y del periférico, hace necesario realizar operaciones de paralelización y serialización.
- Operación de Entrada/Salida:
 - Gestión de las transferencias de bloques y supervisión del estado del periférico
 - Tratamiento de los errores y cuenta de los bloques transferidos
 - Almacenamiento temporal de información. Por razones de eficiencia y seguridad, los datos de la operación de entrada/salida no se manipulan directamente sobre la zona de memoria del programa que la solicitó, por el contrario se suele hacer sobre un buffer temporal
 - Conversión de códigos. Los datos pueden estar representados en el periférico de una manera distinta de la requerida por el programa, por lo que es frecuente tener que realizar una conversión de códigos[9], [10]

Comunicación entre la CPU y los Periféricos

La transferencia de un sector se puede ver como un bucle de sincronización o espera. Hay dos formas de realizar esta espera:

- El computador ejecuta un programa que realiza la espera, comprobando continuamente el estado del controlador del periférico, para detectar si puede enviar/recibir una palabra. En este caso la UCP esta en **Espera Activa**.
- El controlador del periférico realiza esta espera y es capaz de avisar a la UCP; mediante una interrupción; o es capaz de enviar directamente la palabra a la memoria del computador. En este caso la UCP es capaz de ejecutar otro programa por lo que no entra en espera.

El problema de sincronización, a nivel de UCP-Periférico, se concreta en la pregunta siguiente: Como puede un programa saber que un periférico tiene un dato disponible, o que puede recibir un nuevo dato?

La transferencia de información entre el controlador del periférico y la UCP se realiza de forma similar a las transferencias internas de la unidad de control. La diferencia sustancial es que en la Entrada/Salida la UCP no es soberana absoluta, puesto que

algunas de las señales de control, necesarias para la transferencia las ha de producir el controlador del periférico.

Existen dos alternativas básicas para hacer una transferencia elemental:

- Por ejecución de una instrucción (entrada/salida programada)
- Por acceso directo a memoria

Operaciones de entrada/salida

Las instrucciones de entrada/salida, en general, permiten realizar transferencias entre la Unidad Central de Proceso (UCP) y el controlador de un periférico. Estas instrucciones tienen dos argumentos, la dirección de un registro del controlador y un registro interno de la UCP o la dirección de una posición de memoria. Es evidente que para la ejecución de este tipo de operaciones es necesario que se establezca un camino físico entre la UCP o la memoria y los registros del controlador. Esta función es realizada por el "bus", que se puede definir como un conjunto de líneas conductoras de hardware utilizadas para la transmisión de datos entre los componentes de un sistema informático. Un bus es en esencia una ruta compartida que conecta diferentes partes del sistema, como el microprocesador, la controladora de unidad de disco, la memoria y los puertos de entrada/salida (E/S), para permitir la transmisión de información.

En el bus se encuentran dos pistas separadas, el bus de datos y el bus de direcciones. La CPU escribe la dirección de la posición deseada de la memoria en el bus de direcciones accediendo al registro de entrada/salida, teniendo cada una de las líneas carácter binario. Es decir solo pueden representar 0 o 1 y de esta manera forman conjuntamente el número de la posición dentro del mapa memoria o el mapa de entrada/salida. Cuantas más líneas hayan disponibles, mayor es la dirección máxima y mayor es la memoria o registros de entrada/salida a la cual puede dirigirse de esta forma. En el bus de direcciones original de una PC habían ya 20 direcciones que permitían acceder a un 1MB de memoria. En dicha arquitectura el mapa de entrada/salida se encontraba separado del mapa de memoria. Para acceder a los registros de entrada/salida se disponía de instrucciones especiales que permitían el acceso hasta 65536 puertos (16 bits). El bus de datos constituye el camino físico por donde circulan los datos y su ancho define la longitud de palabra del mismo. El bus XT tenía solo 8 bits con lo cual sólo podía transportar 1 byte a la vez. Si la CPU quería depositar el contenido de un registro de 16 bits o por valor de 16 bits, tenía que desdoblarlos en dos bytes y efectuar la transferencia de datos uno detrás de otro. Adicionalmente existe un tercer bus, denominado de control, que se encarga de arbitrar la actividad en los buses de datos y de direcciones. En el caso de las operaciones de entrada/salida define el tipo de operación, entrada o salida, y su ciclo de ejecución.

Los parámetros más importantes en las operaciones de transferencia entre la UCP y los puertos de entrada/salida son el tiempo de acceso y el tiempo de ciclo.

El tiempo de acceso es el que transcurre desde que se da una orden de lectura hasta que la información contenida en el puerto seleccionado aparece a la salida, según se puede apreciar en la Figura 21. Como se puede apreciar a partir del instante en que las variables de dirección permanecen estables, constituyendo una determinada combinación, se inicia la lectura conmutando la señal de control de escritura-lectura al nivel de lectura (en este caso se supone que es un cero lógico) y la señal de inhibición de escritura-lectura al nivel que permite ambas operaciones (en este caso se supone que la desinhibición se realiza con un uno lógico). Al cabo de un tiempo, denominado "t_{acceso}", aparece en los terminales de salida la información contenida en el puerto.

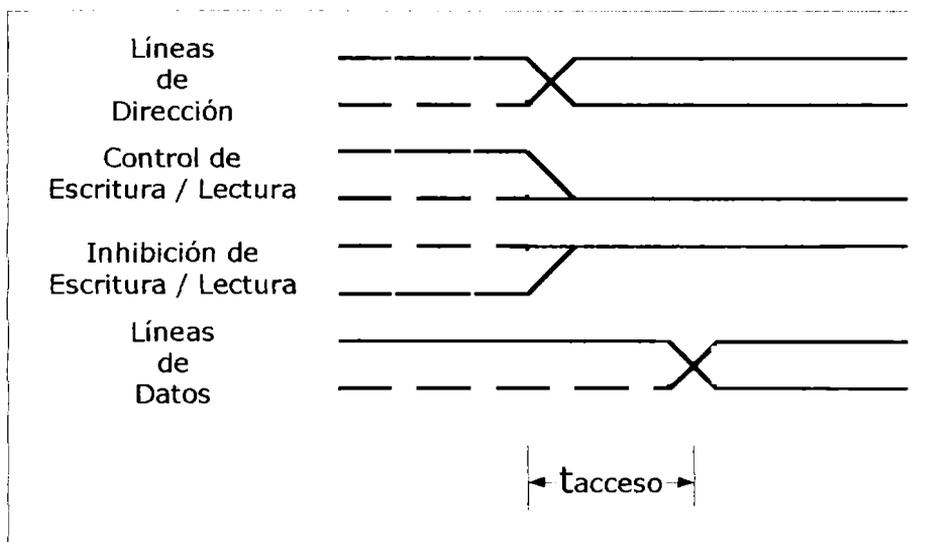


Figura 21: Tiempo de acceso a un puerto de entrada/salida

El tiempo de ciclo es el que ha de transcurrir como mínimo entre dos órdenes sucesivas de escritura o de lectura. En general, la operación de escritura necesita más tiempo que la de lectura y es la que condiciona el tiempo de ciclo. La Figura 22 muestra que la dirección ha de permanecer estable un cierto tiempo en el interior del cual se hace aparecer un impulso en la señal de control de escritura-lectura. Este impulso ha de tener una cierta duración mínima " t_e ", para que se realice con seguridad la operación de escritura, y además ha de aparecer un cierto tiempo " t_1 " después de iniciarse el ciclo de escritura y ha de desaparecer un cierto tiempo " t_2 " antes de que la dirección cambie y se dé por terminado dicho ciclo. Estos dos intervalos de tiempo " t_1 " y " t_2 " son necesarios para que la operación de escritura se realice con seguridad en el puerto seleccionado. Esto es debido a que la información se graba con el nivel alto de la señal de control de escritura-lectura. En otros casos los datos se graban con alguna de las transiciones -desde el nivel bajo al nivel alto o desde el nivel alto al nivel bajo- de la mencionada señal y los tiempos " t_1 " y " t_2 " se miden respecto a la misma.



BIBLIOTECA
FAC. DE INFORMÁTICA
U.N.L.P.

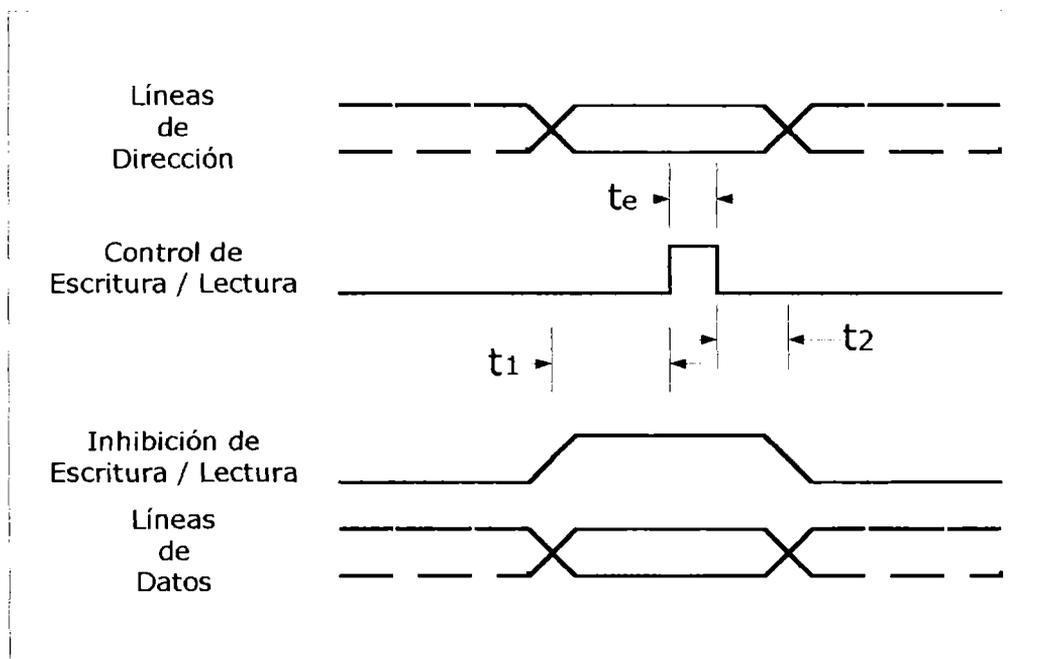


Figura 22: Ciclo de escritura en un puerto de entrada/salida

Direccionamiento de entrada/salida

Cualquiera que sea su naturaleza concreta, los dispositivos periféricos, se personalizan frente al sistema bajo una interfaz denominada **puerto E/S** ("I/O port"). Los intercambios de información con estos dispositivos (entradas/salidas), se realizan como en el resto de los casos utilizando direcciones; de modo que cuando el sistema necesita dirigirse a un "puerto", utiliza una dirección. En este sentido, el procedimiento es análogo al utilizado para intercambios de datos internos (entre el procesador y la memoria o entre direcciones de memoria), aunque en este último caso, el procesador dispone de un amplio abanico de instrucciones que permiten mover y operar entre registros y direcciones de memoria, frente a solo dos (las instrucciones **IN** y **OUT**) cuando estas operaciones se realizan entre registros y direcciones de puertos.

Tanto si se refiere a una transferencia interna (a una posición de memoria) como externa (E/S a un puerto), la forma de operar es siempre la misma; poner el dato en el bus de datos y la dirección en el de direcciones. El sistema sabe que se trata de una dirección de memoria o de puerto y cual es la acción esperada (lectura o escritura) porque se dispone señales adecuadas en las patillas del procesador, que tienen su prolongación en el bus de control. Aunque las direcciones de periféricos son coincidentes (en valor) con las de memoria, no hay posibilidad de confusión porque los elementos conectados al bus (memoria y periféricos) saben cuando un dato se dirige a unas u otros por el estado de las mencionadas líneas.

Direccionamiento dedicado

El sistema de direccionamiento de periféricos anteriormente descrito, se conoce como de ***direcciones independientes o dedicadas***. La dirección del puerto debe ser establecida por el fabricante del periférico, pero el 8088, y por consiguiente el PC, disponían de pocas de estas direcciones "independientes" para asignar a periféricos, por lo que tradicionalmente cada dispositivo montaba varias direcciones alternativas que debían ser seleccionadas mediante puentes o interruptores. Esta disposición permite que si una dirección está ocupada por otro dispositivo pueda ser cambiada, de forma que se adapte a una dirección libre, aunque la solución resultaba engorrosa y complicada a la hora de instalar el hardware.

Como se ha señalado, el acceso a puertos de direccionamiento dedicado se realiza exclusivamente mediante dos instrucciones, **IN** y **OUT**; la primera lee un byte del registro de salida del puerto y lo sitúa en un registro del procesador; la segunda escribe el contenido de un registro del procesador en un registro del puerto.

Desde la introducción del primer IBM PC era costumbre que determinados dispositivos muy comunes se conectasen mediante puertos de direcciones determinadas. También hay que reseñar que un mismo periférico puede tener varias direcciones (de hecho es rarísimo que tenga menos de dos). La razón es que estos dispositivos tienen varios registros (cada registro es una dirección de 1 Byte), normalmente uno o varios son direcciones de lectura/escritura de datos desde/hacia el dispositivo, mientras que otros (que pueden ser de solo escritura), sirven para programar el dispositivo. Por ejemplo, el controlador de puerto serie, que suele ser un chip 8250, dispone de 8 registros, aunque los datos se reciben y transmiten por uno de ellos, el resto sirve para programar y controlar el funcionamiento del dispositivo (poner el control de paridad, la velocidad de transmisión, el número de bits de datos, número de bits de parada, etc).

El sistema de direcciones dedicadas, aunque adecuado para dispositivos de mediana/baja velocidad, y de relativamente pocas direcciones, no resulta adecuado para dispositivos rápidos y/o que utilicen comparativamente muchos registros de programación o control.

Direccionamiento mapeado en memoria

Una alternativa al método anterior consiste en construir los dispositivos periféricos de forma que sus registros internos respondan como direcciones de memoria. Este sistema se denomina de ***direccionamiento mapeado en memoria***, y presenta indudables ventajas; la principal es que se dispone de todo el amplio arsenal de instrucciones para manipular datos entre estas posiciones y los registros del procesador. Es un método especialmente adecuado para manejo de periféricos rápidos o que tengan gran cantidad de registros internos. Un caso paradigmático lo constituyen los controladores de video, que deben almacenar grandes cantidades de información que además debe ser movida a gran velocidad.

Naturalmente esta solución requiere que los registros del dispositivo se comporten exactamente como lo hace la memoria. Es decir, deben responder como auténticas direcciones de memoria RAM. Observe que esto no es lo usual en determinados

periféricos. Por ejemplo; un ratón o un teclado no conservan la información que se pudiera escribir en ellos; además su contenido dependería del movimiento o de las pulsaciones de teclas realizadas por el operador[9].

Otra condición no menos importante es que los registros deben ser suficientemente rápidos para que puedan ser equiparadas a la velocidad de acceso de la RAM. Observe que en este caso, el procesador no distingue realmente entre la dirección de auténtica RAM y las direcciones que corresponden a un periférico. Por ejemplo, en el caso de un controlador de video el procesador escribe en unas determinadas posiciones de memoria y el resultado es mostrado automáticamente en la pantalla del monitor.

El sistema tiene la ventaja de que los periféricos se manejan de forma unificada dentro del espacio general de direcciones de memoria, y que se dispone de los recursos propios del manejo de la memoria, entre ellos el mecanismo de acceso directo **DMA**, con lo que determinados movimientos pueden efectuarse casi sin intervención del procesador (por ejemplo intercambios entre la memoria de video y el disco). Sin embargo, el sistema presenta también evidentes inconvenientes. El primero y principal es que estas direcciones, denominadas **direcciones I/O**, quedan excluidas de su posible utilización como auténtica RAM de propósito general. En general, para evitar fragmentar la memoria, se suele agrupar la zona dedicada a entrada/salida en porciones bien definidas del mapa, como puede ser la zona superior o bien la zona inferior. La Figura 23 muestra una partición del mapa de memoria donde la zona de entrada/salida se ha ubicado en la zona de direcciones bajas[9].

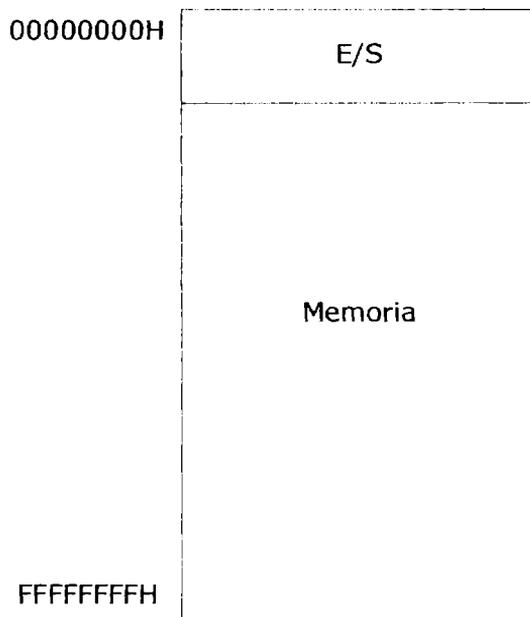


Figura 23: Entrada/Salida Mapeada en Memoria.

8. Comunicación Serie

Características de la Comunicación Serie:

Para comprender bien algunos aspectos de la comunicación serie en la Informática hay que tener en cuenta que este tipo de comunicación eléctrica precede en bastantes años a la aparición de los ordenadores electrónicos. La comunicación serie tiene sus raíces en el telégrafo, que posteriormente derivó en el Teleimpresor (Telex), donde se utilizaron diversos protocolos de comunicación; los más famosos el Morse, el Baudot y el ASCII.

Posteriormente las comunicaciones serie informáticas adoptaron algunos de los usos y costumbres existentes, adaptándolos a las nuevas necesidades y características de los equipos, empezando por su propio sistema de códigos (ASCII), pero conservando algunas reminiscencias de sus ancestros: Los Telégrafos.[12]

Tanto el sistema de codificación ASCII (American Standard Code for Information Interchange) como el Standard RS-232 nacieron pensando en las transmisiones serie utilizadas casi exclusivamente para tele-impresión. Las señales y protocolos empleados se referían a cuestiones tales como sonar una campana para recabar la atención del operador, y enviar un retorno de carro CR (Carriage Return) o un Salto de Línea LF (Line Feed), ya que se utilizaban impresoras electro-mecánicas. La comunicación Serie entre ordenadores no requiere tantas señales, sin embargo se siguieron utilizando la terminología y muchas de las señales del estándar primitivo.[12]

En la comunicación serie los bits se transmiten uno detrás de otro (de ahí el nombre), lo que hace que sean mucho más lentas que sus homólogas "paralelo" en las que se transmiten varios bits a la vez. **La ventaja es que pueden utilizarse un solo par de hilos, o incluso uno solo (sí el retorno se realiza por tierra).**

La comunicación serie ha permitido comunicar varios dispositivos al ordenador, como un ratón, una impresora u otro ordenador, aunque su aplicación principal han sido las telecomunicaciones a través de modems [12]

Nota: Al dispositivo que acepta como entrada los flujos de bits en serie, y al mismo tiempo produce una portadora como salida (o viceversa), se lo conoce comúnmente como Modem (derivado de los términos Modulador-Demodulador). El modem se inserta entre el Ordenador (dispositivo digital) y el sistema telefónico (dispositivo analógico).

Existen varias formas de transmisiones serie:

- **Simplex:** Un equipo transmite, el otro recibe.
- **Half-duplex:** Transmiten ambos equipos pero no simultáneamente; los equipos se alternan en la transmisión, uno transmite mientras el otro recibe.
- **Full-duplex:** Ambos equipos transmiten simultáneamente. Para ello se requieren dos líneas independientes, transmisión y recepción; la línea de transmisión de un equipo se conecta a la entrada de recepción del otro y viceversa. Los puertos serie del PC son capaces de utilizar este modo.
- **Sincronas:** Los dispositivos que se comunican se sincronizan en el momento inicial de la transmisión y constantemente intercambian información a una cadencia predefinida. En este tipo de transmisión, además de los datos, se envía el reloj con que el transmisor generó los datos. De esta manera el receptor recupera la información de reloj y se mantiene sincronizado con el Transmisor. Con objeto de mantener la sincronización, cuando no existen datos que enviar se transmiten caracteres sin valor (Idle Characters). Esta transmisión es más rápida que la asíncrona porque no es

necesario transmitir señales de inicio o fin de dato (no hay bits de Start y Stop); constantemente se reciben caracteres que pueden ser de datos o sin valor (de relleno).

Asíncronas: En este modo de transmisión no existe sincronización; No hay reloj común. Los datos deben ayudar al sincronismo. No es necesario enviar caracteres de relleno, pero hay que indicar cuando empieza un dato y cuando termina. Esto se hace incluyendo en la transmisión señales de inicio y de fin de dato (bits de "Start" y "Stop"). En la comunicación asíncrona, la información (cada carácter) es enviada en el interior de un cuadro ("Frame") de tamaño variable, que comienza con la mencionada señal de inicio y termina con la de final; es el tipo de comunicación utilizada en los puertos serie del PC.[12]

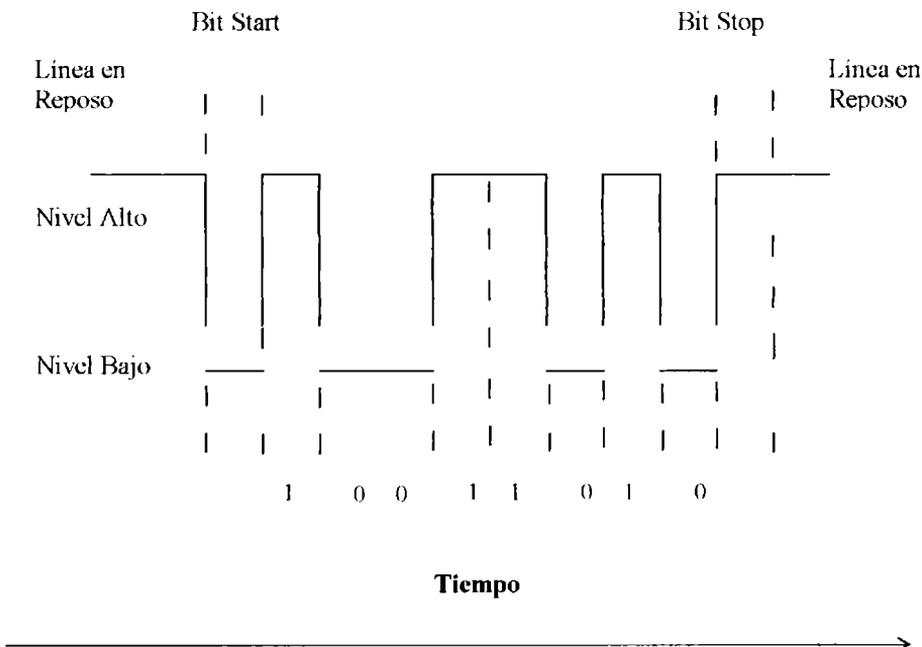
Por lo tanto los datos serie se encuentran encapsulados en tramas de la forma:



Primero se envía un Bit de Start, a continuación los bits de datos (primero el bit de mayor peso, puede ser de 5 a 8 bits de longitud), después puede haber un bit de paridad y finalmente 1 o 2 bits de Stop.

El número de los Bits de Datos y de Bits de Stop son parámetros configurables, así como el criterio de paridad par o impar para la detección de errores. Normalmente las comunicaciones serie tienen los siguientes parámetros: 1 bit de Start, 8 bits de datos, 1 bit de Stop y sin paridad.[15]

En el siguiente grafico se puede ver un ejemplo de la transmisión del dato binario 10011010 de acuerdo a la configuración mencionada en el párrafo anterior. La línea en reposo esta a nivel alto:



En este tipo de comunicación, el estado de reposo (cuando no se transmite nada) se identifica con un "1" (Marca). Cuando se recibe un bit de inicio, que es un "0" (espacio), el receptor toma nota que va a comenzar a recibir un dato.

Nota: Esta forma de asignación aparentemente ilógicas (llamar "1" a la ausencia de señal), tiene también sus orígenes en el telégrafo, donde se constató que había menos errores si se utilizaba siempre una tensión en línea (para indicar reposo) y se cambiaba para señalar el dato. Este Sistema tiene la ventaja adicional que el receptor puede distinguir una línea ociosa de una desconectada.

Los parámetros que caracterizan estas comunicaciones son: Velocidad, paridad, bits de datos y bits de parada. Es frecuente expresar estos datos en forma resumida por ejemplo:

1200 8 N 1

Donde se indica una transmisión de 1200 baudios con 8 bits de datos sin paridad y un bit de stop.

- I. **Velocidad de transmisión:** ("Connection Speed") En general es la cantidad de bits transmitidos en unidad de tiempo. En el caso de una transmisión serie se expresa en bits por segundo (bps). En las transmisiones serie a través de líneas telefónicas, en las que se emplean módems, era frecuente utilizar como unidad de medida de velocidad el Baudio ("Baud Rate"), en honor a Emile Baudot. Baudio se define como el número de veces que cambia la portadora en un segundo. La velocidad que puede emplearse depende en gran medida de la calidad del medio de transmisión (calidad de la línea), y de la distancia. Los primeros dispositivos serie operaban a velocidades muy bajas, del orden de 110 a 1200 baudios. Actualmente, las comunicaciones serie telefónicas están muy cerca del máximo teórico que pueden soportar los pares de cobre utilizados en la telefonía estándar.[12]

- II. **Longitud del carácter** ("Char Length") El conjunto de bits que componen un carácter se denomina bits de datos ("Data Bits"). En las transmisiones serie se han utilizado caracteres de 5, 6, 7 y 8 bits. Actualmente los datos son enviados como caracteres ASCII, por lo que pueden utilizarse 7 u 8 bits según se trate del juego de caracteres US-ASCII o el extendido.[12]
- III. **Bits de Parada:** ("Stop Bits") Después que se envía un carácter se envía un bit de parada, que tiene el valor "1" (marca); la duración de este bit puede ser 1;g 1.5 o dos periodos. [12]
- IV. **Características eléctricas** Como hemos señalado, las normas y terminología de las comunicaciones serie están muy influenciadas por sus orígenes telegráficos. Esto incluye algunas de las definiciones utilizadas por la Norma **RS-232**, que es la norma usada en nuestro trabajo de tesis. Una curiosidad de esta Norma es que la transmisión de los caracteres ASCII se realiza empezando por el bit menos significativo. Puesto que estamos acostumbrados a representarlos de derecha a izquierda (empezando por el más significativo), podría deberse que los bits son transmitidos al revés. Los datos son

transmitidos en binario, pero no es frecuente referirse a ceros y unos como en la Informática convencional, sino a espacios y marcas. En el caso del RS-232-C la especificación eléctrica considera que para decidir un 1 binario se debe tener un voltaje más negativo que -3 volts y que un 0 binario se tendrá cuando el voltaje positivo sea mayor a los + 4 volts [15] .

- V. Conectores:** El estándar RS-232 define un conector DB de 25 pines del que solo se utilizan 22; las señales están pensadas para la comunicación remota de teleimpresores, y actualmente no son realmente necesarias para comunicar ordenadores, por lo que se ha estandarizado el uso de 8 (las 8 grandes) que pueden ser utilizadas en un conector de 25 pines o de 9. [12]
- VI. Servicios BIOS de Puerto Serie:** Los servicios BIOS relacionados con el puerto serie se gestionan mediante el vector de interrupción 20 (14 h). Este servicio tiene cuatro subservicios numerados del 0 al 3:
- **Servicio 0:** Inicializar el Puerto. Permite ajustar los parámetros de transmisión.
 - **Servicio 1:** Enviar un carácter.
 - **Servicio 2:** Recibir un carácter.
 - **Servicio 3:** Obtener el estado del puerto, tanto de funcionamiento como estados de error.
- VII. Paridad:** ("Parity") La adición del bit de paridad al byte de datos es un mecanismo simple pero poderoso de aumentar la confiabilidad de los datos transferidos. Las posibilidades de paridad en la transferencia de datos es la siguiente:[11]
- **NONE:** No se incluye paridad en el byte transferido.
 - **EVEN:** Se agrega un bit al byte total, de modo que el carácter completo incluyendo los bits de partida, término, información y de paridad tengan un número par de bits.
 - **ODD:** El carácter completo ahora tiene un número impar de bits.
 - **ONE:** Se agrega siempre un bit 1 a todos los caracteres, el que es chequeado en cada carácter.
 - **CERO:** Idéntico al caso anterior, solo que ahora se agrega un cero a todos los caracteres transferidos.

Evidentemente estos dos últimos sistemas no aportan absolutamente ninguna información, por lo que son usados muy raramente. Se debe hacer notar que el agregado de paridad solo permite determinar la presencia de un error y no su corrección; sin embargo, puede resultar muy útil en un sistema que implemente una solicitud de re transmisión de un dato al producirse un error.

RS-232

La interfase localizada entre el ordenador, o terminal y el modem es un ejemplo de **capa física** en el que debe especificarse en forma detallada los aspectos mecánicos, eléctricos, funcionales y de procedimiento. RS-232 es un tipo de Norma bien conocido correspondiente a la capa física. [15]

RS-232-C corresponde a la tercera versión revisada de la norma original RS-232. Esta norma fue propuesta por la Asociación de Industrias Electrónicas, que es un organismo registrado de fabricantes de electrónica, y se le conoce propiamente como EIA-RS-232-C. La versión internacional se encuentra incluida en la **recomendación V.24 del CCITT(se describe en el apartado "Otras Normas")**, muy parecida y que solo se diferencia por el uso de algunos circuitos.[15]

La última de las revisiones ha sido la "E" realizada en Julio de 1991. [11]

El terminal o el ordenador se llaman oficialmente en las normas como DTE (Equipo Terminal de Datos, ETD), y al Modem también oficialmente se le conoce como DCE (Equipo Terminal de Circuito de Datos, ETCD). [15]

Nota: En la comunicación serie, como en cualquier otra comunicación informática, existen dos aspectos complementarios: uno relativo a las características físicas de la conexión; en este caso las características eléctricas y mecánicas, aspectos que están contemplados en el estándar RS-232, a estos elementos los denominamos **capa física**. Otros son los protocolos de comunicación, que incluyen los sistemas de codificación de la señal que se enviará por la capa física, a estos últimos los denominamos **capa lógica**. [12]

El estándar RS-232 es el más antiguo y más ampliamente usado en las comunicaciones seriales. Enlaza dos dispositivos conectando la línea transmisora de un dispositivo con la línea receptora del otro. Ambos dispositivos pueden "conversar" simultáneamente (Full-duplex). Además puede haber líneas de protocolo destinadas a controlar las comunicaciones, pero su implementación varía ampliamente entre fabricantes de equipos y no todos pueden llegar a utilizarlas. La circuitería de RS-232 envía señales de tensión por las líneas con referencia a tierra (single-ended), y tienen un alcance máximo de 50 pies entre equipos y una velocidad máxima de transmisión de datos de 20 Kilo baudios (20.000 bits/s), los que en la práctica normalmente se exceden.[11]

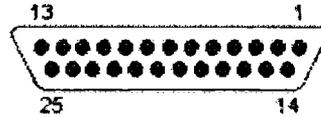
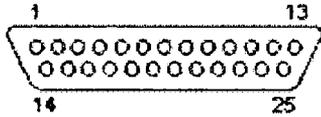
Algunos equipos RS-232 tienen una línea transmisora y receptora adicional, que implementa una interconexión cerrada de corriente no estándar. Estas interconexiones se utilizan en ambientes ruidosos desde el punto de vista electromagnético y cuando se requiere transmitir sobre largas distancias. En estos casos la velocidad de transmisión disminuye y se usa aislación óptica.[11]

El estándar define las características eléctricas que deben presentar los elementos de conexión para la comunicación serie entre ordenadores y equipos periféricos. Su título dice exactamente: "Conexión entre un Equipo Terminal de Datos y un Equipo de Comunicación de Datos empleando un intercambio de datos binario serie". Vemos que aparecen en el título del estándar los dos tipos de dispositivos antes mencionados: DTE y DCE. En la terminología de las comunicaciones serie se denomina "Sexo" de un equipo a su tipo; en este sentido DTE y DCE son "sexos contrarios". A efectos prácticos el sexo del equipo determina que asignación de señales tendrá su conector externo. [12]

Los aspectos contemplados en este estándar son:[12]

- Características eléctricas de la conexión
- Características mecánicas de la conexión
- Descripción funcional del intercambio, proporcionando nombres a las señales utilizadas
- Conexiones ejemplo para una selección de sistemas de comunicación

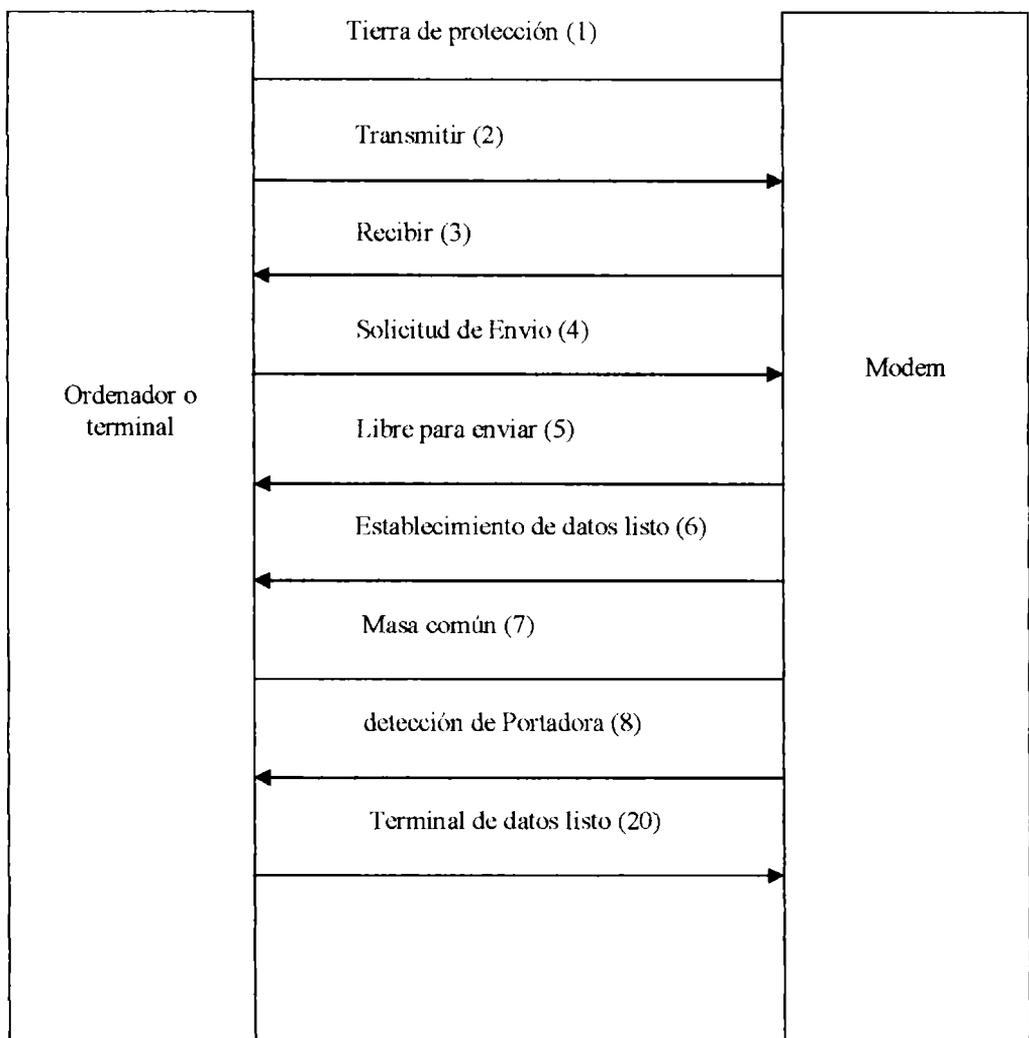
La especificación mecánica considera un conector con 25 patillas y 47.04 +- 0.3 mm de ancho (de centro de tornillo a centro de tornillo del conector), con todas las demás dimensiones igualmente bien especificadas. En la fila superior se numeran los pines de 1 a 13 (de izquierda a derecha); en tanto que en la fila inferior los pines están numerados del 14 al 25 (también de izquierda a derecha). [15]



DTE (Data Terminal Equipment) DCE (Data Communication Equipment)

La especificación eléctrica para el RS-232 como ya se mencionó anteriormente considera que para decidir un 1 binario se debe tener un voltaje más negativo que -3 volts, y que un 0 se tendrá cuando el voltaje positivo sea superior a +4 volts. Es posible tener velocidades de datos de hasta 20 Kbps, así como longitudes de cables de hasta 15 metros. [15]

La especificación funcional indica los circuitos que están conectados a cada una de las 25 pines, así como el significado de cada uno de ellos. En la siguiente figura se muestran 9 pines que casi siempre están soportadas, el resto de ellas con frecuencia se omiten, los números de las patillas se muestran en paréntesis:



Cuando el terminal u ordenador se enciende, esta activa (es decir pone un 1 lógico) la señal "Data Terminal Ready" (patilla 20). Cuando el Modem se enciende, se activa la señal correspondiente al "Data Set Ready" (patilla 6). Cuando el modem detecta una portadora sobre la línea telefónica, se activa la señal de "Carrier Detect" (patilla 8). El "Request to Send" (patilla 4), indica que el terminal quiere enviar datos. El "Clear to Send" (patilla 5), significa que el modem esta preparado para aceptar datos. Los datos se transmiten con el "Transmit Circuit" (patilla 2) y se reciben con el "Receive circuit" (patilla 3).[15]

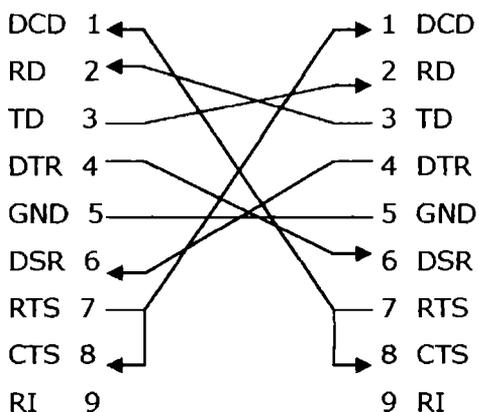
Se tienen a disposición otros circuitos con los cuales se puede seleccionar la velocidad de los datos, probar el modem, temporizar los datos, detectar las señales de llamada y enviar datos en la dirección opuesta, sobre un canal secundario. Difícilmente se llegan a utilizar todos estos circuitos en la practica. [15]

La especificación del procedimiento es el protocolo; es decir, el establecimiento de la secuencia legal de eventos. El protocolo esta basado en la definición de pares acción-reacción. Cuando en el terminal se propone el "request to send", por ejemplo, el modem contesta con un "clear to send", si tiene la capacidad para aceptar la

información. También existen otros pares de acción-reacción, parecidos a este, para otro tipo de circuitos. [15]

Es muy común que dos ordenadores quieran conectarse por medio de un RS-232. En este caso como ninguno de los dos es un Modem, surge el problema del Interfase. Este problema se resuelve al conectarlos bajo el esquema conocido como Modem Nulo, que conecta la línea transmisora de una maquina con la línea receptora de la otra maquina. También cruza las líneas de Handshaking en forma similar. [15]

Gráfica de Interconexión Módem Nulo:



Comunicación entre dispositivos: Cuando se intentan comunicar dos dispositivos usando la interfase RS-232, se deben cuidar 4 aspectos de compatibilidad entre los sistemas:[11]

- La designación funcional de los dispositivos (DTE o DCE).
- La velocidad de la transferencia de los datos (bit por segundo o baudios).
- La compatibilidad del formato de los datos, es decir bits de paridad, de partida y de final.
- Las líneas de control que usaran ambos dispositivos.

Velocidad de transmisión: Como ya se mencionó anteriormente, uno de los parámetros más importantes que se deben establecer correctamente entre los dispositivos que se comunican a través de una interfase, es el número de baudios, o la velocidad de transferencia de los datos a través del enlace. Por ejemplo las velocidades mas utilizadas en enlaces de comunicaciones con el computador son las de 300 y 1200 baudios, que se usan para comunicaciones telefónicas y las de 9600 y de 19200 baudios que se utilizan en redes internas entre computadores. [11]

Una regla nemotécnica que se usa en estos casos es que el número de caracteres por segundo que pueden ser transferidos entre dos dispositivos es aproximadamente el numero de baudios dividido por 10. La velocidad de transferencia exacta dependerá del formato de los datos que se utilice. [11]

Formato de los datos serie: Existe un grado de flexibilidad considerable en el formato de los datos que se pueden usar en enlace RS-232. Por ejemplo si se envía un solo bit de partida, 7 bits para el carácter y un bit de paridad seguido por un solo bit de termino. Esto generalmente se escribe como 7 + Even + 1. Los otros formatos que se usan habitualmente se muestran en la siguiente tabla:[11]

Nota: El bit de partida no se muestra explícitamente en la tabla.

| BITS POR CARÁCTER | BIT DE PARIDAD | BITS DE PARADA |
|-------------------|----------------|----------------|
| 7 | +PAR | +1 |
| 7 | +IMPAR | +1 |
| 8 | NINGUNO | +2 |
| 8 | NINGUNO | +1 |
| 8 | +PAR | +1 |
| 8 | +IMPAR | +1 |
| 7 | +PAR | +2 |
| 7 | +IMPAR | +2 |

Paridad: Se usa comúnmente un bit que es designado como bit de paridad.

Otras Normas:

RS-449:

Prácticamente incluye tres normas en una. Los procedimientos, mecanismos y funcionalidad de la interfase están considerados en la RS-449, en tanto que la interfase eléctrica esta establecida en dos normas diferentes. La primera de estas, la RS-423-A, es similar a la RS-232-C, en el sentido de que todos los circuitos comparten una tierra común. A esta técnica se la denomina transmisión asimétrica. La segunda norma eléctrica la RS-422-A, contraria a la primera utiliza una transmisión balanceada, en la que cada circuito principal necesita 2 hilos, sin tener una tierra común. Como resultado la RS-422-A, puede utilizarse en velocidades de hasta 2 Mbps, en cables de 60 metros, e incluso a velocidades más grandes, sobre cables de longitudes menores.

Se han agregado circuitos nuevos que no estaban presentes en la RS-232, particularmente circuitos que sirven para probar el modem, tanto en forma local como remota. Como consecuencia de la inclusión de varios circuitos de dos hilos es necesario tener mas patillas en la nueva norma. [15]

CCITT V.24:

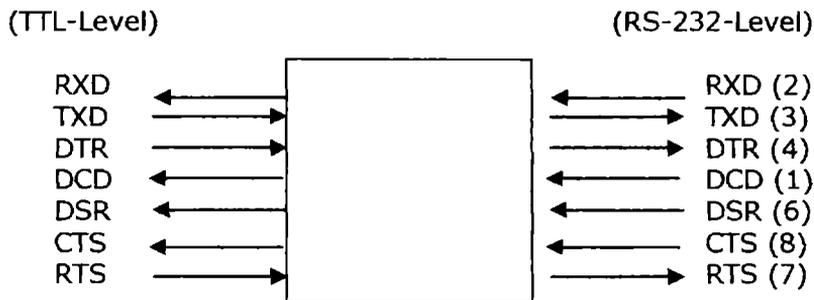
La capa física de de la recomendación V24 del CCITT (ahora ITU-T) es sustancialmente equivalente al RS-232-C.

La especificación mecánica considera un conector de 25 patillas y con sus dimensiones bien especificadas. La especificación eléctrica considera que para decidir un 1 binario se debe tener un voltaje más negativo que -3 voltios, y que un 0 binario se tendrá con un voltaje superior a los +4 voltios. Es posible tener velocidades de hasta 20 kbps, y longitudes de cables de hasta 15 metros. La especificación funcional indica los circuitos que están conectados a cada una de las 25 patillas y el significado de cada uno de ellos. La especificación del procedimiento es el protocolo, es decir, el establecimiento de la secuencia legal de eventos. A continuación se describe para cada patilla su abreviatura, dirección y función:

- 1 - -: Tierra de seguridad
- 2 TD: hacia DCE, Salida de datos DTE
- 3 RD: hacia DTE, Entrada de datos DTE
- 4 RTS: hacia DCE, DTE desea cambiar a modo transmisión
- 5 CTS: hacia DTE, DCE listo para transmitir

- 6 DSR: hacia DTE, DCE listo para comunicar con DTE
- 7 - -: Línea común (masa) del circuito
- 8 DCD: hacia DTE, Enlace de datos instalándose
- 20 DTR: hacia DCE, Pone a trabajar al módem
- 22 RI: hacia DTE, Anuncia una llamada
- 23 DSRD: ambas, Indicador de velocidad de datos

Gráfica de Interconexión:



Interfase digital X.21.

En 1976 el CCITT recomendó, para estimular la compatibilidad en el uso de las líneas digitales, una interfase de señalización digital llamada **X.21**. Esta recomendación especifica la manera en la que el ordenador del cliente, el DTE, establece y libera las llamadas mediante el intercambio de señales con el equipo del proveedor de servicios portadores, el DCE.

Líneas de señal utilizadas en el X.21.

- T (Transporte)
- C (Control)
- R (Recepción)
- I (Indicación)
- S (Señal; por ejemplo, debida a la temporización de bits)
- B (Octeto de temporización) opcional
- Ga (Vuelta común del DTE)
- G (Tierra)

El DTE utiliza las líneas T y C para transmitir los datos y controlar la información, respectivamente (la línea C es similar a la señal de colgar y descolgar el teléfono). El DCE hace lo propio con las líneas R e I. La línea S contiene la señal que permite conocer el momento en el que comienza un bit.

Protocolos de Comunicación (Handshaking Modes)

En una comunicación serie, como en toda transferencia de datos, es conveniente realizar un control sobre la transmisión y la recepción de datos a fin de que esta se

lleve a cabo en forma correcta, es decir, que no se pierda información. Uno de los principales problemas que se presentan es la vigilancia de los Buffers que destina el Puerto serie para la recepción y transmisión de datos. Estos Buffers son zonas de memoria reservada que funcionan de la siguiente manera:[11]

- Por una parte, se guardan los datos que van llegando por el puerto serie desde el periférico. Estos datos son almacenados hasta que son leídos por el programa que gestiona el puerto serie. Es posible que la llegada reiterada de información sin que esta sea leída acabe por llenar la capacidad del buffer de recepción. En este caso, los nuevos datos que se reciban tras la saturación del buffer son ignorados, es decir, toda nueva información se pierde. Para que se puedan volver a recibir datos nuevos, primero se han de leer algunos de los datos almacenados en el buffer, con lo que en éste se dejará espacio libre que podrá ocuparse con la nueva información que se reciba.
- Por otra parte, en el buffer de emisión se almacenan los datos que se desean enviar al periférico, en espera de que la transmisión sea posible, es decir en espera de que el periférico se encuentre preparado para recibir.

Los protocolos de comunicación (Handshaking Modes) se ocupan de evitar que el buffer de recepción de datos del puerto serie se sature y, por consiguiente, se pierda información. Con el uso de estos protocolos, el PC y el periférico se envían información sobre sus respectivos buffers de recepción. La mecánica de funcionamiento de los protocolos es la siguiente:

En caso que el buffer del receptor se llegue a saturar con datos recibidos, se envía una señal al emisor indicándole que ha de interrumpir la transferencia de datos, ya que de otro modo, se perdería información. El emisor recibe dicha señal y deja de transmitir, quedándose en espera de que el receptor vuelva a estar disponible para continuar con la comunicación. Una vez que el receptor haya leído parte de la información contenida en su Buffer, este volverá a disponer de espacio libre, con lo que se enviará una señal al emisor indicándole que ya puede continuar con la transferencia de datos.

Existen diferentes protocolos de comunicación para las transferencias. Por ejemplo LabView es capaz de utilizar "Software Handhaking" y "Hardware Handhaking". Es aconsejable, por simplicidad en aplicaciones en que la cantidad de datos a transferir es pequeña no usar ningún protocolo, ya que en estos casos la capacidad del buffer puede ser más que suficiente para que éste nunca llegue a saturarse. Ahora bien si el volumen de información a transmitir es considerable y/o si el periférico conectado al PC no es muy rápido, el uso de uno de los protocolos de comunicación está del todo recomendado.[11]

Software Handhaking – XON-XOFF

El protocolo XON-XOFF es un protocolo software para evitar la saturación de los buffers de comunicación serie. Por ejemplo en el caso de LabView el funcionamiento es el siguiente: cuando el buffer de recepción está a punto de ser saturado, el receptor envía el carácter XOFF (<control S>, decimal 19) para indicar al emisor que ha de detener la transferencia de datos para que no llegue a perderse información. Una vez que el receptor ha leído parte de la información contenida en el buffer de recepción, dispondrá de nuevo de espacio libre para recibir más datos, enviando entonces el

carácter XON (<control Q>, decimal 17) para indicar al emisor que puede reanudar la transmisión.[11]

Si se activa el protocolo XON/XOFF, los dispositivos que se están comunicando siempre interpretan los caracteres <control-S> y <control-Q> como los comandos XON y XOFF respectivamente, no como datos. Es por esto que no se ha de utilizar este protocolo al transmitir información binaria, ya que es posible que dichos caracteres se encuentren por casualidad entre los bytes enviados, con lo que se interpretaran erróneamente como los comandos XON y XOFF, alterando el curso de la comunicación, cosa que no es conveniente. En este caso se deberá desactivar el protocolo XON/XOFF, con lo que los caracteres <control-S> y <control-Q> se interpretaran como datos y no actuaran sobre la comunicación, ya que no son considerados comandos de control.

Hardware Handhaking – XON-XOFF

Al igual que el protocolo Software, su finalidad es la de evitar que llegue a saturarse alguno de los buffers de datos utilizados en la comunicación. Ahora bien en este caso no se controla la transferencia de información mediante el envío de caracteres, sino que se utiliza una serie de señales físicas que interconectan los dos dispositivos a comunicar. Estas señales, en el caso de LabView son: DSR, RTS, CTS y DTR, dado que son señales que necesitan un soporte físico para ser transmitidas, es necesario construir un cable de comunicaciones adecuado para el uso de este protocolo. Un ejemplo muy usual de conexión de este cable es la ya descrita configuración de modem nulo. Una vez conectados los dos dispositivos a comunicar con el cable adecuado, el protocolo podrá hacer uso conveniente de las señales de control antes indicadas[11]

9. GPIB (General Purpose Interface Bus)

Introducción

La mayoría de los instrumentos de laboratorio, constan de una interfase estándar para interconectarse entre sí o con sus correspondientes instrumentos virtuales llamada GPIB (General Purpose Interface Bus), también conocida como su denominación estándar IEEE 488.

Su nombre original es HP-IB (Hewlett Packard Interface Bus), diseñado por Hewlett Packard en 1965 para conectar sus líneas de instrumentos programables a sus computadores. El bus GPIB es estandarizado en 1975 por la IEEE para evitar la dispersión de características. El estándar IEEE 488-1975 hoy conocida como IEEE 488.1 era una norma a seguir por los fabricantes que deseaban utilizar el bus GPIB y definía las especificaciones eléctricas, mecánicas y funcionales del sistema de interfase. En 1987 se establece la norma IEEE 488.2 que define como los controladores e instrumentos deberían comunicarse. Esta norma fortaleció la norma 488.1 al adicionarle formatos de datos estándar, reporte de estados, manejo de errores, funcionalidad de controlador y comandos comunes que todos instrumentos debían responder de una manera definida. Sin embargo todavía no existía una estructura con una sintaxis común para instrumentos de una misma clase de diferentes fabricantes. En 1990 para evitar que cada fabricante eligiera los comandos de sus instrumentos se estructura las órdenes a los aparatos mediante la adopción del formato de comandos SCPI (Comandos comunes para instrumentos programables) respetando los principios del anterior 488.2. De esta manera un programa elaborado para un osciloscopio de un fabricante podía ser ejecutado con otro osciloscopio de otro fabricante con mínimos cambios, disminuyendo los costes de desarrollo y mantenimiento[15].

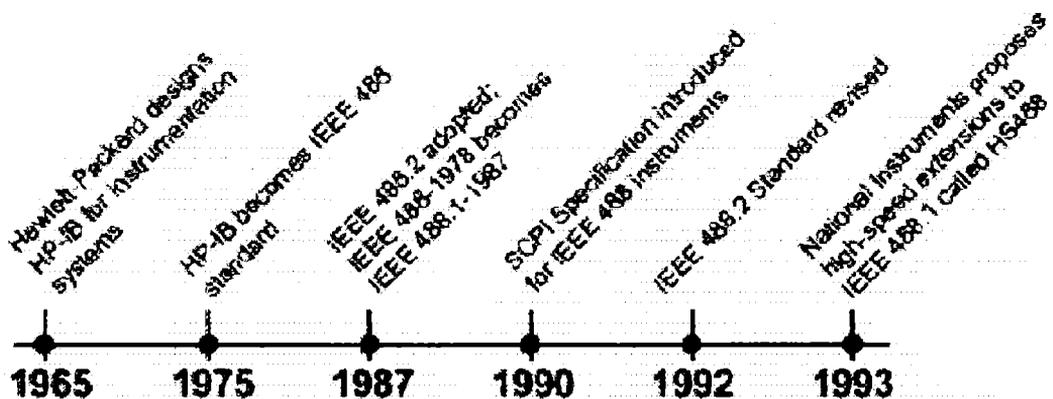


Figura 24: La cronología de eventos que han dado lugar al establecimiento de la norma IEEE 488.2

Conceptos Básicos GPIB

Características del bus GPIB

Las características generales del bus GPIB son:

- Bus de transmisión de datos de 8 bits en paralelo.

- Utiliza niveles TTL y lógica negativa.
- El número máximo de dispositivos conectados al bus es de 15 por limitaciones eléctricas, de los cuales uno de ellos es el controlador que maneja el flujo de información en el bus.
- Puede haber más de un controlador, aunque sólo uno puede estar activo en un mismo instante de tiempo.
- La velocidad máxima de datos es de 1MB por segundo.
- La longitud máxima permitida del bus es de 20 metros. El valor típico por dispositivo es de 2 metros. Se comercializan cables de 1, 2, 4 y 8 metros.
- Los instrumentos pueden estar interconectados mediante un cable en estrella, línea o cualquier combinación de ellas. (Ver Figura 25).
- La estructura del bus se compone de 24 líneas orientadas por byte. 8 bits para comandos, 8 bits para datos y 8 bits para tierra.
- Cada dispositivo conectado al bus puede enviar o recibir información hacia o desde cualquiera de los otros 14 dispositivos. Algunos como una impresora solo pueden recibir datos, otros como un contador sólo pueden enviar y otros están capacitados para enviar y recibir. Ej. un osciloscopio o una computadora.

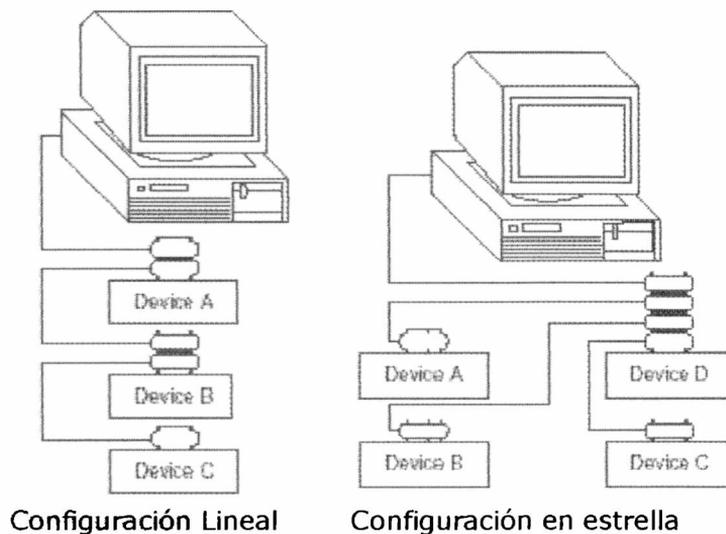


Figura 25: configuración de un sistema GPIB en forma lineal y estrella.

Para que los dispositivos conectados al bus de comunicaciones GPIB puedan comunicarse, debe existir alguna especie de orden. Deben cumplirse los siguientes criterios[16]:

- Los dispositivos conectados al bus deben hablar y comprender el mismo lenguaje.
- Debe establecerse algún tipo de jerarquía para el control de flujo en el bus.

Existen tres clases de dispositivos conectados al bus: controlador locutor y escucha.

1. Controladores (Controllers): El controlador es el dispositivo que controla a los demás instrumentos. Este gobierna el flujo de información en el bus. Generalmente el controlador suele ser una

computadora que coordina los demás dispositivos. En caso de haber más de una computadora conectada al sistema, es decir, varios controladores al mismo tiempo, solo uno puede ser el controlador activo o a cargo llamado (Controller In Charge). Es este quien puede pasar el control a otro que esté inactivo.

Su responsabilidad es definir las normas de comunicación, responder a los dispositivos que solicitan servicio, enviar comandos GPIB, pasar y recibir el control y regular los papeles de los locutores y escuchas.

2. Locutor (Talker): Los locutores envían mensajes al controlador o posibles escuchas. En el bus pueden existir uno o varios locutores, pero solo uno puede estar activo (establecido por el controlador) en un mismo instante de tiempo.
3. Escuchas (Listener): Los escuchas son dispositivos cuya función es la de recibir los datos que circulan por el bus enviados por otro dispositivo. En cada bus pueden existir varios escuchas al mismo tiempo. El controlador a cargo es el que establece a través de un comando que un equipo pasa o deja de estar en modo escucha.

Estructura del Bus GPIB

Los dispositivos en el bus se comunican enviándose mensajes. Estos mensajes son transferidos por señales que circulan por líneas que conforman el bus GPIB: 16 líneas de señal y 8 líneas de tierra[18].

La estructura del bus se organiza de la siguiente manera:

Bus de Datos

Ocho líneas de datos DIO1-DIO8 (Data input / output) bidireccionales orientados a la transferencia de bytes o caracteres ASCII de 7 bits en donde la línea DIO8 se utiliza como paridad. Transportan mensajes de datos y comandos.

Bus de sincronización de la transferencia de datos o handshaking

Dado que el bus GPIB es asíncrono, cuando un mensaje de múltiples bytes tiene que ser transferido de un instrumento a otro se utilizan 3 líneas de control que gestionan la transferencia de datos a través de un protocolo de acuse de recibo (HANDSHAKE) asegurando que no se emita si ningún instrumento está preparado para recibir o no se opere más rápido que el instrumento más lento que reciba los datos. De esta manera se garantiza la transmisión sin error entre dispositivos.[16]

La siguiente tabla describe las líneas de GPIB handshake:

| Línea | Descripción |
|---------------------|---|
| DAV (DATA Valid) | Se activa cuando el controlador envía comandos o el Talker envía mensajes de datos. Esta línea indica que los datos en la línea de datos es correcta y estable de manera que puedan ser |

| | |
|------------------------------|--|
| | aceptados por los receptores |
| NRFD (Not Ready For Data) | Línea controlada por los receptores. Cuando un receptor no está listo para recibir datos activa esta línea. Su valor es falso cuando todos los dispositivos direccionados están preparados para recibir un mensaje presente en el bus. Se realiza un OR cableado de todas las salidas NRFD de los receptores activos. De esta manera el emisor debe esperar hasta que todas las salidas NRFD de los emisores direccionados estén en falso (todos estén preparados) para poder transmitir. |
| NDAC (Not Data ACcepted) | Línea controlada por los receptores. Si su valor es verdadero indica que algún receptor activo no ha aceptado el mensaje presente en el bus. Si su valor es falso indica que todos los receptores han recibido un dato transmitido. Se realiza un OR cableado todas las salidas NDAC de los receptores direccionados. |

Bus de control

Está constituido por cinco líneas de hardware encargadas de manejar el flujo de datos a través del Bus GPIB.

La siguiente tabla describe las líneas de control de la interfase GPIB:

| Línea | Descripción |
|--------------------------|---|
| ATN (ATTention) | Cando el nivel lógico de esta línea está en verdadero, los datos que circulen por el bus serán considerados mensajes de interfase (mensajes remotos), en caso contrario se considerarán datos correspondientes a comandos del instrumento. |
| IFC (InterFace Clear) | Línea controlada por el controlador del bus. Sirve para inicializar el bus interrumpiendo el proceso que se estaba realizando, deshabilitando a todos los Listeners y Talker activos. El controlador que genera esta señal queda como CIC (Controlador a Cargo). Esta línea debe ser respondida por todos los instrumentos en cualquier instante de tiempo. |
| REN (Remote ENable) | Esta línea es usada por el controlador para indicar que dispositivos activos deben ponerse en modo remoto para atender al bus. Si esta línea no está activada los dispositivos se encuentran |

| | |
|--------------------------|--|
| SRQ (Service ReQuest) | en modo local. Esta línea será usada por cualquier dispositivo para solicitar servicio asincrónico desde el controlador. |
| EOI (End Or Identify) | Los locutores usan esta línea para indicar el fin del mensaje de datos avisándole a los escuchas y controlador. Con ATN a "1", cuando se activa esta línea indica que el controlador conduce un sondeo paralelo (solicitando identificador). |

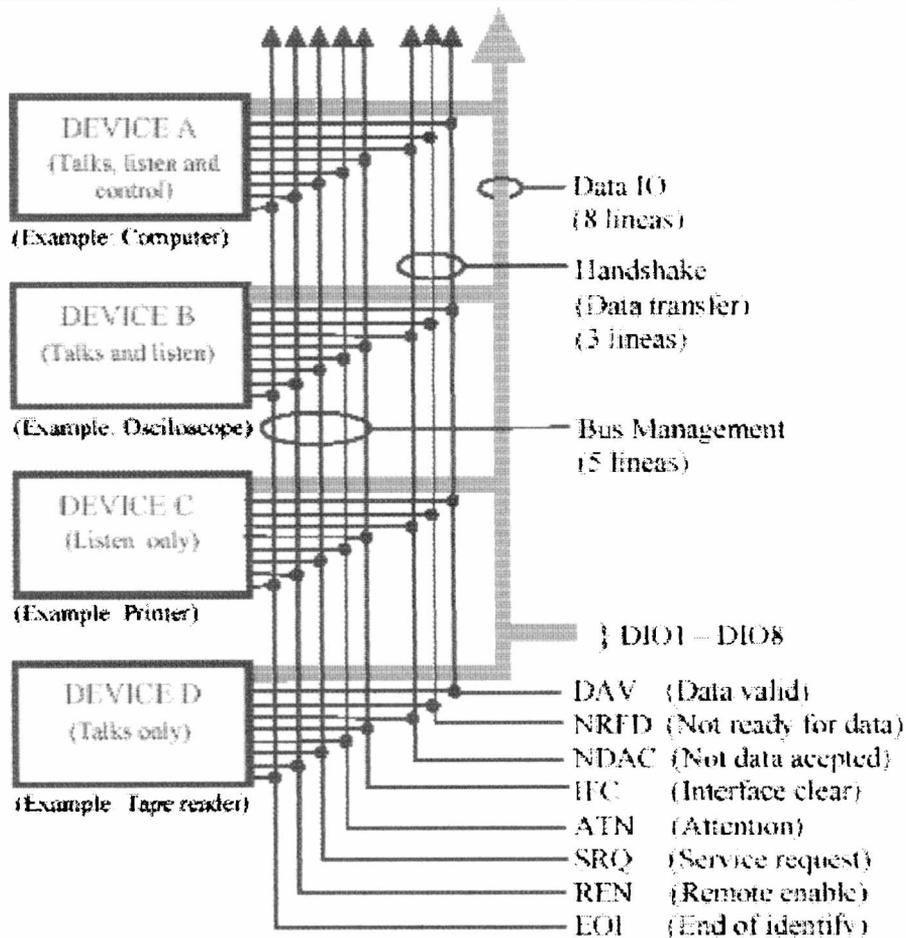


Figura 26: Líneas del bus.

Conector estándar de acceso al bus:

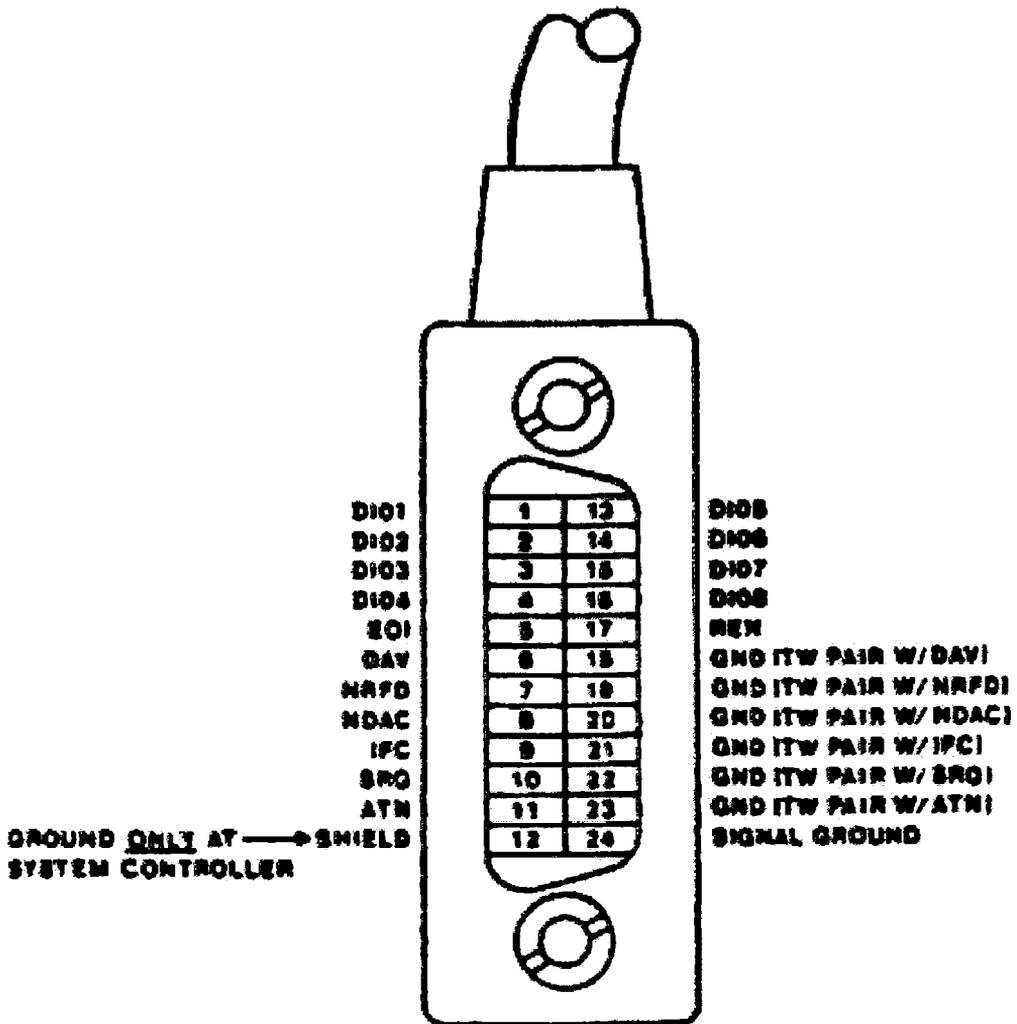


Figura 27: Conector para el bus GPIB.

Direccionamiento GPIB

Para asegurar la comunicación entre los instrumentos y el controlador, todos los dispositivos e interfaces GPIB deben tener asignado una dirección única de manera que las comunicaciones se realicen sin interferir un instrumento sobre el otro en bus. No debería haber dos dispositivos con igual dirección conectados al bus[16].

Las direcciones GPIB se dividen en dos partes: una dirección primaria y una dirección secundaria. La dirección primaria debe tener un valor entero en el rango de 0 a 30. El controlador usa esta dirección para direccionar los dispositivos como Talker o Listeners en la transferencia de datos. Para activar un dispositivo como Listener se activa en un byte de control de direccionamiento el bit 5 LA(Listen Active) incluyendo la dirección GPIB del dispositivo (bits 0 al 4). Para activar un dispositivo a Talker se activa el bit TA (Talk Active) de un byte de control de direccionamiento incluyendo en los bits 0

al 4 la dirección GPIB del dispositivo.

A continuación se detalla el byte de mensaje de direccionamiento:

| | | | | | | | | |
|-----------------|---|----|----|---|---|---|---|---|
| Bit de Posición | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Significado | 0 | TA | LA | Dirección Primaria GPIB (rango 0-30) | | | | |

Algunos dispositivos usan una dirección secundaria. Cuando se utiliza se direcciona enviando primero la dirección primaria del dispositivo y luego la secundaria. Su utilidad se encuentra en situar en un solo dispositivo varios instrumentos de manera de poder direccionar subunidades, cada una con una dirección diferente, dentro de un dispositivo. Dicho de otra forma, algunos instrumentos son como "clusters" de dispositivos, la dirección secundaria permite direccionar un dispositivo dentro de dicho "cluster".

Para desactivar a posibles Listener y al posible Talker activo en ese momento existen dos comandos para tal fin: Untalk Unlisten.

Reporte de estado

Con la norma IEEE 488.2 todos los dispositivos compatibles a esta norma disponen de una estructura de bytes donde guardan y actualizan su estado. El propósito de esta estructura es proveer información al controlador del estado del instrumento[17].

La siguiente figura muestra una estructura de reporte de estado:

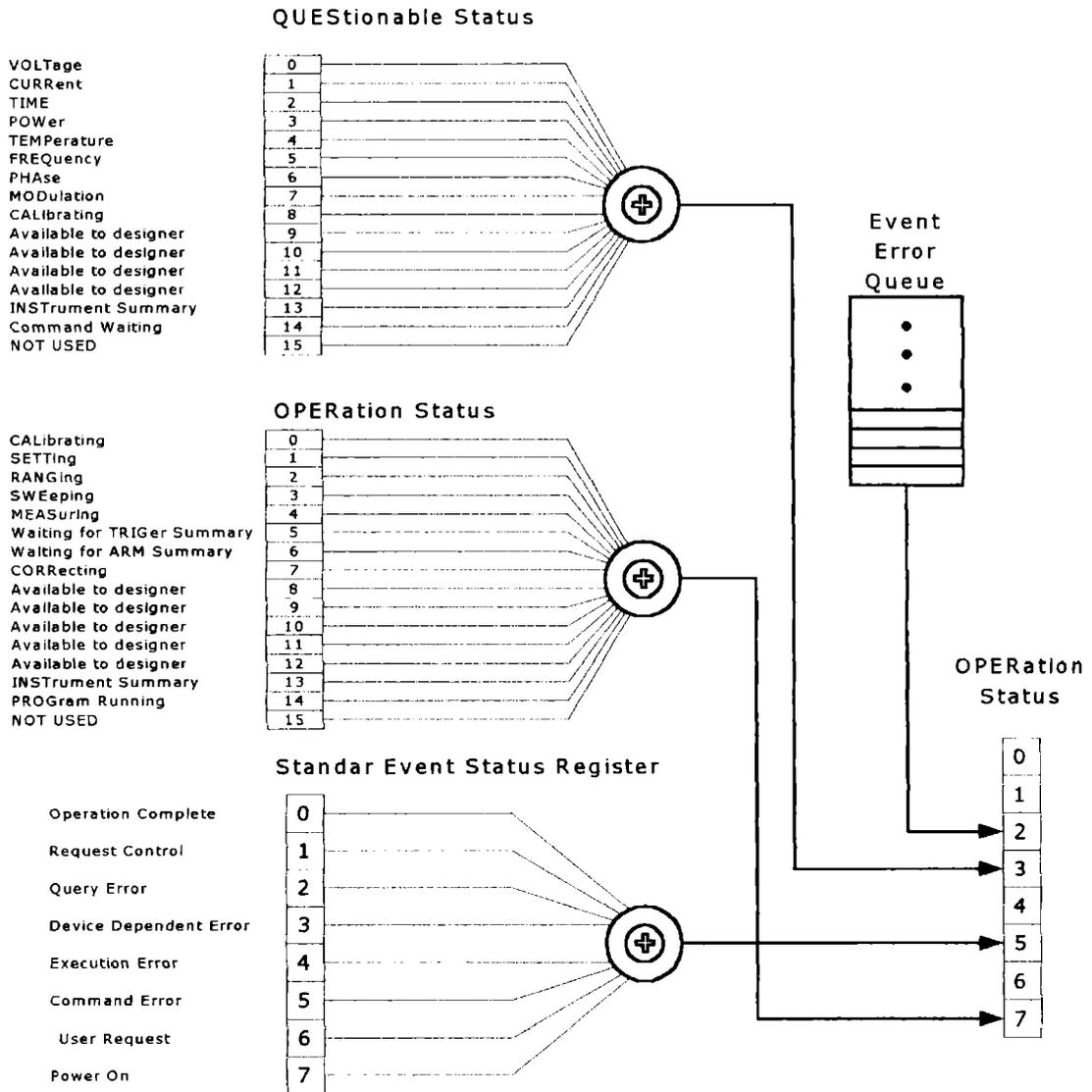


Figura 28: Estructura de reporte de estado.

Registros de estado

La norma IEEE 488.2 contempla tres registros de estado cada registro tiene un registro de habilitación / deshabilitación (Enable Register) que permite enmascarar ciertos bits[18].

Los registros son:

Standard Even Status Register (8 bits)

Este registro guarda información de los sucesos y errores más frecuentes en todos los instrumentos. Contiene la siguiente información:

- 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 disponemos de los comandos *ESE *ESE?. Para conocer todo el contenido del Standard Even Status Register disponemos del comando *ESR?.

Questionable Data (16 bits)

Muestra información propia del instrumento.

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

Para cambiar y conocer la máscara disponemos del comando combinado STAT:QUES:ENAB? y STAT:QUES:ENAB respectivamente. Para leer el contenido del registro se utiliza STAT:QUES:EVEN?.

Status Byte (8 bits)

Este registro existe desde la norma 488.1. Cada bit refleja la condición de otro registro o cola. Cuando un bit esta en "1", el correspondiente registro o cola debe ser leído para obtener información sobre el estado y limpiar la condición. El significado de sus bits es el siguiente:

- Questionable Data, indica si uno o mas de los bits presentes en el registro Questionable Data está activo y habilitado.
- Message Available, indica si existe un mensaje en la cola de salida donde se encuentran todos los posibles mensajes que un dispositivo desea enviar.
- Standard Event, indica si alguno de los bits del registro Standard Event Register está activo y habilitado.
- Request Service, indica si el dispositivo ha solicitado una petición al controlador. Este bit no tiene máscara.

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

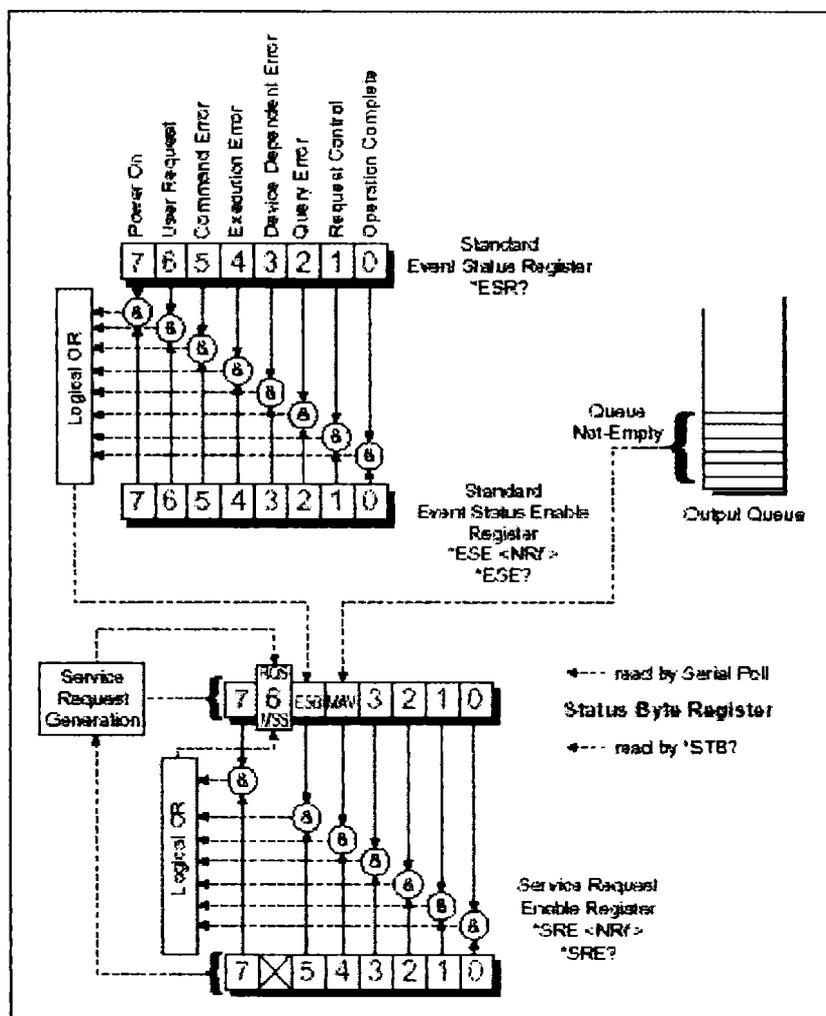


Figura 29: Registro de estado.

Comandos Comunes

La IEEE 488.2 estableció un conjunto de comandos comunes a todos los instrumentos. Los más conocidos se detallan a continuación[17]:

| Comando | Descripción |
|---------|--------------------------------------|
| *CLS | Clear Status Command |
| *ESE | Standard Even Status Enable Command |
| *ESE? | Standard Even Status Enable Query |
| *ESR? | Standard Event Status Register Query |
| *IDN? | Identification Query |
| *OPC | Operation Complete Command |
| *OPC? | Operation Complete Query |
| *RST | Reset Command |
| *SRE | Service Request Enable Command |
| *SRE? | Service Request Enable Query |

| | |
|-------|--------------------------|
| *STB? | Read Status Byte Query |
| *TST? | Self-test Query |
| *WAI? | Wait-to-Continue Command |

10. Proceso Paralelo

Se presentan a continuación algunas definiciones relacionadas:

- **Actividades**

Proceso: Es un programa en ejecución

Tarea: Son las distintas partes de un proceso que se ejecutan simultáneamente

- **Sistemas**

Multiprogramación: Admiten varias actividades que comparten el procesador, pero solo una puede estar ejecutándose en un momento dado.

Multiproceso: Las actividades se ejecutan en sus propios procesadores, compartiendo memoria común.

Proceso Distribuido: Las actividades se ejecutan en sus propios procesadores, conectados a través de una red de comunicaciones.

- **Paralelismo**

Es la ejecución de diversas actividades simultáneamente en varios procesadores. Si solo existe un procesador gestionando multiprogramación, se puede decir que existe Pseudo paralelismo. Se trata de un concepto físico producido por la existencia de varios procesadores.

- **Concurrencia**

Es la existencia de varias actividades ejecutándose simultáneamente y necesitan sincronizarse para actuar conjuntamente. Se trata en este caso de un concepto lógico, ya que solo hace referencia a las actividades, sin importar el número de procesadores presentes.

Para que dos actividades sean concurrentes es necesario que tengan alguna relación entre sí, como puede ser la cooperación en un trabajo determinado o el uso de información compartida.

En un sistema monoprocesador, la existencia de multiprogramación es condición necesaria pero no suficiente para que exista concurrencia, ya que los procesos pueden de forma totalmente independiente. Por ejemplo un editor y un compilador pueden estar ejecutándose simultáneamente en una computadora sin que exista concurrencia entre ellos.

Si un sistema es multiprocesador, también pueden presentarse situaciones de concurrencia siempre y cuando las actividades necesiten actuar entre si [19].

Exclusión mutua:

Debido a la concurrencia de las actividades pueden surgir problemas. Para abordar el estudio consideraremos los datos compartidos como un recurso que el sistema operativo debe gestionar para asegurar un uso correcto.

El principal problema con que nos encontramos, que es intrínseco a la concurrencia es el denominado Exclusión Mutua.

Supongamos que en el sistema existe un archivo formado por registros compuesto por cinco campos, la estructura que tiene el registro es:

| A | B | C | D | E |
|---|---|---|---|---|
|---|---|---|---|---|

El diseño del archivo indica que el contenido de los campos es el siguiente:

Campo A: Número del documento de identidad (DNI)

Campo B: Nombre

Campo C: Primer Apellido

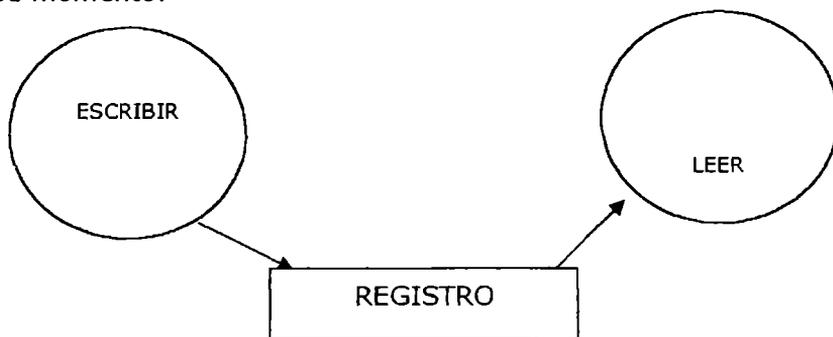
Campo D: Segundo Apellido

Campo E: Domicilio

Para que un registro sea valido, debe estar totalmente actualizado, es decir si se modifica el valor del campo A, el resto de los campos deben ser coherentes con el nuevo valor de dicho campo, ya que de otro modo el registro sería inconsistente.

Si en el momento en el que un proceso esta escribiendo o modificando un registro existente, otro proceso que realizase la lectura de ese mismo registro, y el primero de ellos solo hubiese tenido tiempo de escribir o modificar el campo A, la información obtenida por el segundo proceso sería inconsistente, para evitar esta situación se deben **sincronizar los procesos** de manera que mientras uno este escribiendo ningún otro pueda leer. Ahora bien, esto no ocurre en aquellos casos en que dos procesos tratan de leer en el mismo archivo, aun coincidiendo en el mismo registro.

Para generalizar el caso anterior, supongamos los dos procesos que coinciden en el mismo registro, uno escribiendo y el otro leyendo y los llamaremos ESCRIBIR y LEER, se encuentran en un sistema monoprocesador y multiprogramado y son los únicos presentes en ese momento.



En el momento de un cambio de proceso del uno al otro se pueden producir las siguientes situaciones:

- **Sin Sincronización de procesos:** Puede darse el caso en que ESCRIBIR este actualizando un registro y se quede a medias, sorprendiéndole el cambio de procesos, por tanto terminara de escribirlo cuando vuelva a hacer uso del procesador.

Con el cambio le tocará el turno al proceso LEER, que accederá a dicho registro pudiendo leerlo completamente. Es evidente que los datos leídos serán inconsistentes.

- **Con Sincronización entre procesos:** Supongamos algún mecanismo que prohíba la lectura (Bloqueo de Registro) a cualquier proceso mientras el proceso

ESCRIBIR este realizando alguna operación. En este caso LEER, al hacer uso del procesador y solicitar una operación de lectura sobre un registro que se encuentra bloqueado, quedaría en espera de que el registro quede totalmente escrito y se proceda a su desbloqueo. LEER pasaría a estado bloqueado, ESCRIBIR terminaría su trabajo sobre el registro y en el siguiente cambio LEER procedería a hacer el suyo.

Esta sincronización por la cual una actividad impide que otras puedan tener acceso a un dato mientras se encuentra realizando operación sobre el mismo es lo que se conoce como **Exclusión Mutua**.

La zona de código de un proceso que no puede ser interrumpida por otro por los motivos anteriores expuestos, se denominan como **Sección Crítica**.

Sincronización:

Si una actividad desea impedir que otra acceda a ciertos datos compartidos, mientras no se cumpla una determinada condición, debemos sincronizar las actividades con dicha condición. Por tanto la sincronización es un elemento necesario para asegurar la exclusión mutua.

Los algoritmos que se han diseñado para este fin se clasifican en tres grupos:

Espera Activa:

Son aquellos algoritmos que basan todo su funcionamiento en establecer la espera de entrada a la sección crítica con un bloque que será roto en el momento en que se cumpla una determinada condición. Se llaman de espera activa porque el proceso no queda bloqueado durante su ejecución, sino que estará compitiendo por el procesador constantemente. Por este motivo estos sistemas sobrecargan el sistema innecesariamente. Fueron los primeros en utilizarse y han sido sustituidos por otros más eficientes.

Entre los distintos tipos de estos algoritmos existentes podemos citar:

- **Espera con mutex:** Algoritmo que utiliza un switch (mutex) a través del cual se produce la sincronización.
- **Alternancia:** Ligeramente mejor que el anterior, utiliza también una variable TURNO para realizar el sincronismo entre los procesos.
- **Algoritmo de DEKKER:** Resuelve el problema mediante la solución propuesta por Dekker, basando su funcionamiento en una tabla unidimensional de dos elementos lógicos (switches).

Espera no activa:

Son los algoritmos que establecen la espera para entrar en la sección crítica bloqueando el proceso haciendo que deje de competir por el procesador hasta que se cumpla la condición de desbloqueo. Entre los algoritmos existentes citaremos los

siguientes:

- **Semáforos:** Para eliminar los problemas que se producen con los algoritmos de espera activa, fundamentalmente los referidos a la sobrecarga que producen en el sistema, E. W. Dijkstra (1965) diseñó un mecanismo basado en una variable entera utilizada como contador de peticiones de entrada a una sección crítica. Esta variable es compartida por todos los procesos del Sistema. Este nuevo tipo de variable se denomina semáforo. Por su capacidad de gestionar el tráfico de procesos que desean acceder a datos compartidos, se bloqueará de igual manera que cuando un proceso accede a un recurso que está ocupado.
- **Regiones críticas:** Son sistemas que permiten establecer protecciones contra una mala utilización de los usuarios para ello solo permiten que los datos compartidos se puedan acceder desde determinadas regiones quedando transparentes desde el resto. Tiene pequeños inconvenientes relacionados con la sincronización y no permite que las actividades puedan realizar operaciones de lectura simultánea.
- **Regiones críticas condicionales:** Consiste en una mejora del método anterior tratando de resolver algunos problemas de sincronización que se presentaban.
- **Monitores:** Uno de los problemas existentes en los mecanismos anteriores es que el programador tiene que proporcionar de forma explícita el modo de sincronización. Para evitarlo B. Hansen y C. A. R. Hoare desarrollaron un nuevo mecanismo denominado Monitor, que debe ser soportado por el lenguaje correspondiente. Un monitor permite compartir segura y eficientemente, datos entre varias actividades garantizando la exclusión mutua sin necesidad de que el programador tenga que suministrarla explícitamente. Se basa en dos premisas: la primera es la abstracción de datos consistente en una técnica capaz de separar las operaciones a ejecutar sobre los datos, de los detalles de diseño propios de los mismos (los lenguajes Modula y Ada soportan este tipo de estructuras). La segunda es que realizan la exclusión mutua de forma implícita. La finalidad más útil de los monitores es reunir todas las funciones que operan sobre un conjunto de datos compartidos en un solo módulo, de manera que todos los accesos a esos datos estarán forzados a utilizar dichas funciones.
- **Contadores de Eventos:** Es un mecanismo para sincronizar actividades sin que sea necesario forzar la Exclusión Mutua, ya sea porque no deseamos limitar la concurrencia de las actividades o simplemente porque no las necesitamos. Se basa en una variable entera cuya misión es contar determinadas operaciones.
- **Mensajes:** Es un mecanismo que permite a los procesos intercambiar aquella información que sea necesaria durante el desarrollo normal de su ejecución. Por tanto es más un mecanismo de cooperación que de sincronización. Esta cooperación se realiza por medio de mensajes que se envían entre sí los procesos. Se basa en una zona de memoria compartida que gestiona el Sistema Operativo directamente y que permanece oculta a los procesos. De esta manera el proceso que envía un mensaje a otro deposita la información en la zona de memoria compartida y el receptor lo lee de ella posteriormente. En este sistema las directrices de envío y recepción establecen una sincronización entre los procesos al tener que esperar por dichos mensajes, antes de continuar con su ejecución. Existen dos tipos de comunicación entre procesos:

- Directa: Los procesos envían y reciben los mensajes directamente entre sí, de manera que se asocia un enlace bidireccional único entre cada dos procesos.
 - Indirecta: Los mensajes son enviados y recibidos a través de mailboxes o buzones de correos. Con este método cada proceso puede estar relacionado con tantos buzones como desee consiguiendo comunicarse con tantos procesos como sea necesario.
- **Llamadas Remotas:** El paso de mensajes de un proceso a otro es un mecanismo muy utilizado para resolver los problemas de concurrencia entre procesos. Es análogo al paso de parámetros en una llamada a una rutina o procedimiento. Si unimos los conceptos de mensajes y paso de parámetros, tendremos lo que se conoce como rutinas o procedimientos remotos, creándose una copia del mismo cada vez que son ejecutadas (llamadas remotas), siendo dicha ejecución concurrente. Este tipo de interacción asegura que hasta que un proceso no termine determinadas operaciones, el siguiente permanecerá en espera.
 - **Rendez-Vous:** Es la culminación de todos los mecanismos anteriores, tratándose de una ligera modificación del método de las llamadas remotas, donde la llamada, en lugar de ser a todo un procedimiento, lo es solamente a un grupo de sentencias dentro de él. De igual forma, se trata de un procedimiento similar al Monitor, pero con más versatilidad y potencia. Este método se ha llevado a la práctica con el lenguaje Ada[10].

Mecanismos Hardware:

Son instrucciones hardware que aseguran la Exclusión Mutua. Entre las más utilizadas citaremos las siguientes:

- **Deshabilitar Interrupciones:** Los computadores actuales permiten que las interrupciones puedan ser deshabilitadas, de esta forma si un dispositivo generase una interrupción estando deshabilitada, el reconocimiento y tratamiento de la nueva interrupción se retardará hasta que habiliten de nuevo las interrupciones. Por este medio se pueden forzar la Exclusión Mutua deshabilitando las interrupciones mientras haya alguna actividad en la Sección Crítica; así dicha actividad no podrá ser interrumpida y por lo tanto no se podrá producir ningún cambio de proceso. En general la deshabilitación y nueva habilitación de interrupciones suele hacerse con una instrucción máquina, por lo que resulta ser una instrucción muy rápida. Las instrucciones máquina son aquellas que basadas en secuencias de ceros y unos (código binario) son entendidas directamente por el ordenador y por lo tanto ejecutables sin necesidad de traducción.
- **Instrucción TEST_AND_SET:** Muchas de las computadoras actuales suministran una instrucción máquina denominada Test And Set, cuya misión es la de forzar Exclusión Mutua. La ventaja de este mecanismo es que no puede ser interrumpida por ser una instrucción hardware y el inconveniente es que no son muchas las computadoras que la poseen. Esta instrucción por si sola no basta para asegurar la Exclusión Mutua, por lo que tendremos que basarnos en ella para construir los denominados locks.
- **Lock:** Se basa en la instrucción anterior y su cometido es permitir el acceso a la Sección Crítica a un proceso en caso de no existir otra actividad dentro de

su Sección Crítica, no permitiéndolo en caso contrario. En este caso el número de actividades y de procesadores compartiendo memoria principal puede ser cualquiera y además, son mecanismos simples y fáciles de probar y depurar.

Semáforos:

Podemos definir a un Semáforo como una instancia de un tipo de datos abstracto con solo dos operaciones P y V.

V: Señala la ocurrencia de un evento.

P: Se usa para demorar a un proceso hasta que ocurra un evento.

Los Semáforos permiten proteger la Sección Crítica y pueden usarse para implementar sincronización por condición.

En Semáforos Binarios la Semántica es la siguiente:

$P(S) : \langle \text{await } S > 0 \rightarrow S := S - 1 \rangle$

$V(S) : \langle \text{await } S < 1 \rightarrow S := S + 1 \rangle$

En Semáforos Binarios el valor de S debe variar entre 0 y 1

Semáforos en Sincronización Barrier:

La idea es usar un Semáforo para cada flag de sincronización.

- Un proceso setea un flag ejecutando la operación V.
- Un proceso espera que se setee un flag y lo limpia utilizando la operación P.

A manera de ejemplo implementaremos una barrera para dos procesos. Necesitamos saber cada vez que un proceso llega o parte de la barrera, para ello es necesario relacionar los estados de los procesos:

VAR

Arrive1:=0;depart1:=0;

Arrive2:=0;depart2:=0;

```
P1 :: do True → <Arrive1 :=Arrive1 + 1>
      <await depart1 < Arrive2 → Depart1 :=Depart1 + 1>
    od
```

```
P2 :: do True → <Arrive2 :=Arrive2 + 1>
      <await depart2 < Arrive2 → Depart2 :=Depart2 + 1>
    od
```

Ahora sean:

Barrier1 = Arrive1 - Depart2

Barrier2 = Arrive2 - Depart1

VAR

Barrier1 =Sem:=0

Barrier2 =Sem:=0

```
P1 :: do true →
    V(Barrier1)
    P(Barrier2)
od
```

```
P2 :: do true →
    V(Barrier2)
    P(Barrier1)
od
```

En el ejemplo anterior se ve que la idea es usar los semáforos como señales que indican la ocurrencia de eventos. En la nuestra solución cuando un proceso llega a la barrera señala ese evento ejecutando V sobre un Semáforo. El otro proceso espera el evento ejecutando P sobre el mismo semáforo.

Especificación de la Ejecución Concurrente:

Introducimos Notaciones de Programación para Concurrencia y Sincronización.

Cuando se ejecuta un programa secuencial, hay un único thread of control: el contador de programa comienza en la primera acción atómica del proceso y se mueve a través del proceso a medida que las acciones atómicas son ejecutadas. La ejecución de un programa concurrente resulta en múltiples threads of control, uno por cada proceso concurrente.

En nuestra notación especificaremos la ejecución concurrente con la sentencia **co** (con frecuencia llamada sentencia **coBegin**). Sea S_i un programa secuencial: una secuencia de sentencias secuenciales y declaraciones opcionales de variables locales. La siguiente sentencia ejecuta las S_i concurrentemente:

```
CO  $S_1$  //.....//  $S_n$  OC
```

El efecto es equivalente a algún interleaving de las acciones atómicas de las S_i . La ejecución del **CO** termina cuando todas las S_i terminaron.

Especificación de la Sincronización:

La Sincronización asegura el “**Orden correcto**” de las acciones que los procesos ejecutan. El objetivo de la Sincronización es restringir las historias o threads de un programa concurrente solo a las permitidas. Se la puede clasificar:

- Sincronización por Exclusión Mutua: Asegura que solo un proceso tiene acceso a un objeto compartido en un instante de tiempo. De este modo si un programa tiene Secciones Críticas que pueden compartir más de un proceso, la Exclusión mutua evita que dos o más procesos puedan encontrarse en la misma Sección Crítica al mismo tiempo.
- Sincronización por Condición: Permite bloquear la ejecución de un proceso hasta que se cumpla una condición dada.

Una acción atómica hace una transformación de estado indivisible. Esto significa que cualquier estado intermedio que podría existir en la implementación de la acción no debe ser visible para los otros procesos.

Con frecuencia necesitamos ejecutar secuencias de sentencias como una única acción atómica. Especificaremos acciones atómicas por medio de corchetes angulares $\langle Y \rangle$. Por ejemplo $\langle e \rangle$ indica que la expresión e debe ser evaluada atómicamente[20]. Especificaremos Sincronización por medio de la sentencia **await**:

$\langle \text{await } B \rightarrow S \rangle$

La expresión Booleana B especifica una condición de demora; S es una secuencia de sentencias secuenciales que se garantiza que termina. Una sentencia **await** se encierra entre corchetes angulares para indicar que es ejecutada como una acción atómica. En particular se garantiza que B es true cuando comienza la ejecución de S , y ningún estado interno de S es visible para los otros procesos.

Sincronización Barrier:

Nos planteamos el problema en el que es necesario computar sucesivas aproximaciones a una respuesta para converger.

Este tipo de sincronización consiste en que los procesos esperan en una "barrera" al final de cada iteración a que "llegue" el resto de los procesos y recién ahí podrán continuar.

Por ejemplo: Muchas veces es necesario realizar la misma operación sobre los distintos elementos de un array, y entonces podemos usar múltiples procesos que actúen en paralelo.

Se necesitan los resultados de cada operación sobre cada elemento antes de seguir adelante.

Planteamos a continuación una primera solución para el problema de la Sincronización Barrier. En el siguiente fragmento de código se itera creando n procesos.

```
Do True →  
  CO  $i: 1..n$  → Código Tarea  $i$   OC
```

El problema que aparece en esta solución pero es la poca eficiencia obtenida, ya que son generados n procesos en cada iteración.

En lugar de Crear y Destruir procesos, lo que nos llevó a una solución ineficiente, en la siguiente solución sincronizamos procesos al final de cada iteración:

```
Worker [ $j: 1..n$ ] :: Do True →  
  Código Tarea  $i$   
  Espera que se realicen las  $n$  tareas  
Od
```

Una vez encontrada una solución eficiente veamos diferentes implementaciones analizando sus ventajas y desventajas:

1) Contador Compartido:

En una primera versión de la implementación usamos n procesos "Workers" que necesitan encontrarse en una barrera. Cada proceso cuando concluye su tarea incrementa en 1 el valor de count, luego se queda bloqueado esperando hasta que el

valor de count sea n, es decir todos los Workers ya incrementaron count y por lo tanto finalizaron todos sus tareas. Recien en esta instancia cada proceso P_i podrá asignar a $passed[i]$ el valor True.

Var count := 0, Passed [1:n]:Bool := ([n] False)

Worker [i: 1..n] :: Do True →

```
Código Tarea i
< count := count + 1 >
< await count = n →
    passed [i] := True >
```

Od

Siendo invariante global: Para todo $i : 1 \leq i \leq n ; passed [i] = true \rightarrow count = n$

En la solución anterior es necesario resetear count cada vez que se llega a la barrera (valor 0), y hay que hacerlo antes de que algún proceso quiera incrementar. Una solución posible es que lo hagan todos los procesos Workers.

Otra alternativa para solucionar el problema planteado es tener dos contadores: uno que cuenta de uno a n y otro de n a uno, en cada iteración se intercambian los roles. En el peor de los casos, n-1 procesos pueden estar esperando al proceso n, y esto puede llevar a Memory Contention (varios procesos compiten por el acceso a una variable compartida y su acceso continuo es muy costoso).

2) Flags y Coordinadores:

En esta segunda versión de la implementación intentamos evitar Memory Contention distribuyendo la implementación de count al usar n variables.

Usaremos un arreglo $arrive [1..n]$ e inicializaremos sus elementos con valor 0

En el código que presentamos a continuación cuando un proceso $P[i]$ llega a la barrera cambiamos el valor de $arrive[i]$ ejecutando $arrive [i] := 1$, en lugar de incrementar count como lo hacíamos en la implementación anterior.

El invariante global ahora es $count = arrive[1] + \dots + arrive[n]$

La contención de memoria se evita si los elementos que pertenecen al vector arrive son almacenados en diferentes bancos de memoria.

Las siguientes líneas de código usadas en la implementación anterior:

```
<await count = 0 →  
passed [i] := True >
```

en la nueva versión del código cambian por:

```
<await arrive[1] + ....+ arrive[n] = n>
```

Este último cambio reintroduce Memory Contention y es ineficiente porque la suma se esta computando continuamente por cada proceso Worker que esta esperando.

Se plantea como Solución: Introducir un proceso Coordinador, que es el que controla que todos los workers lleguen a la barrera y les indica que pueden avanzar.

Agregamos otro array Continue [1..n] inicializado a 0, que les indica a los procesos cuando pueden avanzar:

```
Var count := 0, arrive [1:n]:Int := ([n] 0),  
Var Continue [1:n]:Int := ([n] 0)
```

```
Worker [i: 1..n] :: Do True →
```

```
    Código Tarea i  
    Arrive [i] :=1  
    <await continue[i] = 1 >  
    continue [i]:=0
```

```
    Od
```

```
Coordinador :: Do True →
```

```
    Fa i :=1 to n → < await arrive [i] = 1 >  
                    Arrive [i]:= 0
```

```
    Af
```

```
    Fa i := 1 to n → continue [i] := 1 Af
```

```
    Od
```

El Tiempo de ejecución de Coordinador es proporcional a Worker.

Como ventaja no existe Memory Contention y como desventaja requerimos de un proceso extra y de un procesador donde este se ejecute.

3) Barreras Simetricas:

Otra alternativa para la implementación es usar una barrera simetrica para n procesos, la cual se construye a traves de pares de barreras simples para dos procesos.

A continuacion se muestran un par de barreras simples para dos procesos, donde cada proceso P_i tiene un flag (Arrive[i]) que setea cuando llega a la barrera. Espera a que el otro proceso setee su flag (Arrive[j]) y luego "limpia" el flag del otro, asignandole valor 0.

```
P[i] :: <await arrive[i] = 0 >  
    Arrive [i] := 1  
    <await arrive[j] = 1 >  
    Arrive [j] := 0
```

```
P[j] :: <await arrive[j] = 0 >  
  Arrive [j] := 1  
  <await arrive[i] = 1 >  
  Arrive [i] := 0
```

No es simple implementar una barrera simétrica para n procesos, no siendo el código tan “simple” como los presentados en las implementaciones anteriores.

11. Descripción del Sistema de medida de Campo Lejano

Introducción:

La Comisión Nacional de Actividades Espaciales (CONAE), está desarrollando un nuevo satélite de manufactura argentina SAOCOM 1-A que constituye una nueva generación de satélites para la observación de la Tierra, con instrumentos que operan en el rango de microondas y dispone de sensores activos (radar).

El SAOCOM 1-A tendrá a bordo un radar de microondas de banda L (1.3 GHz) y una cámara de infrarrojo térmico. El radar es del tipo de apertura sintética, "SAR-Synthetic Aperture Radar", y emitirá señales de alta potencia en la banda de microondas a través de una antena gigante de 2,5m por 10m; su resolución espacial será de 10m y podrá penetrar hasta dos metros de profundidad. Estas características permitirán aportar información sobre la humedad de los suelos y la estructura geológica; mientras que la cámara de infrarrojo permitirá detectar incendios y erupciones. Sumadas estas capacidades se dispondrá de un satélite con excelentes prestaciones para la gestión de emergencias y monitoreo de los recursos naturales.

El Instituto Argentino de Radioastronomía (IAR) está trabajando en la construcción de los prototipos de la antena del radar que controlará dicho satélite. Como todo dispositivo que debe cumplir sus funciones en el espacio está sujeto a especificaciones y requerimientos muy exigentes. Fundamentalmente esta exigencia se encuentra asociada al amplio rango de trabajo en que deben mantenerse sus especificaciones dentro de las bandas de tolerancia. En el caso particular de la antena, interesa conocer con mucha exactitud y precisión su impedancia, diagrama de irradiación para ambas polarizaciones, ganancia, ancho del haz, ancho de banda, aislamiento de polarización y estabilidad con la temperatura, entre otros.

Paralelamente al esfuerzo profesional que implica el desarrollo de un componente que se encuentra en el "estado del arte", **surge la necesidad de establecer un sistema de medición** que permita verificar los parámetros característicos antes mencionados. En el caso especial del diagrama de radiación se deberá implementar un Sistema de Medición de Campo Lejano que dará como resultado el patrón de campo lejos de la antena.

El problema en cuestión consiste en medir las polarizaciones de una antena en una banda de frecuencia. Dicha medición era realizada en forma manual, pudiéndose obtener un Diagrama de irradiación con una resolución angular de 5°. Debido a la necesidad de medir antenas con un lóbulo principal muy angosto, cercano a 0,04° de resolución angular y a la demanda de una considerable cantidad de mediciones por día, surge la necesidad de automatizar el sistema de campo Lejano.

Como el tiempo es un factor determinante para una buena prestación del servicio donde aparece la necesidad de realizar aproximadamente 3600 mediciones por sesión y considerando 10 sesiones de medición diarias, dicho número de mediciones solo puede ser obtenido bajo un Sistema Automatizado de Medida.

La instalación de este nuevo Software permitirá al personal del IAR una mejora en la adquisición de información agilizando la toma de datos de los instrumentos de medida y controlar y monitorear la posición del eje de rotación en que se encuentra la antena bajo estudio en forma automática.

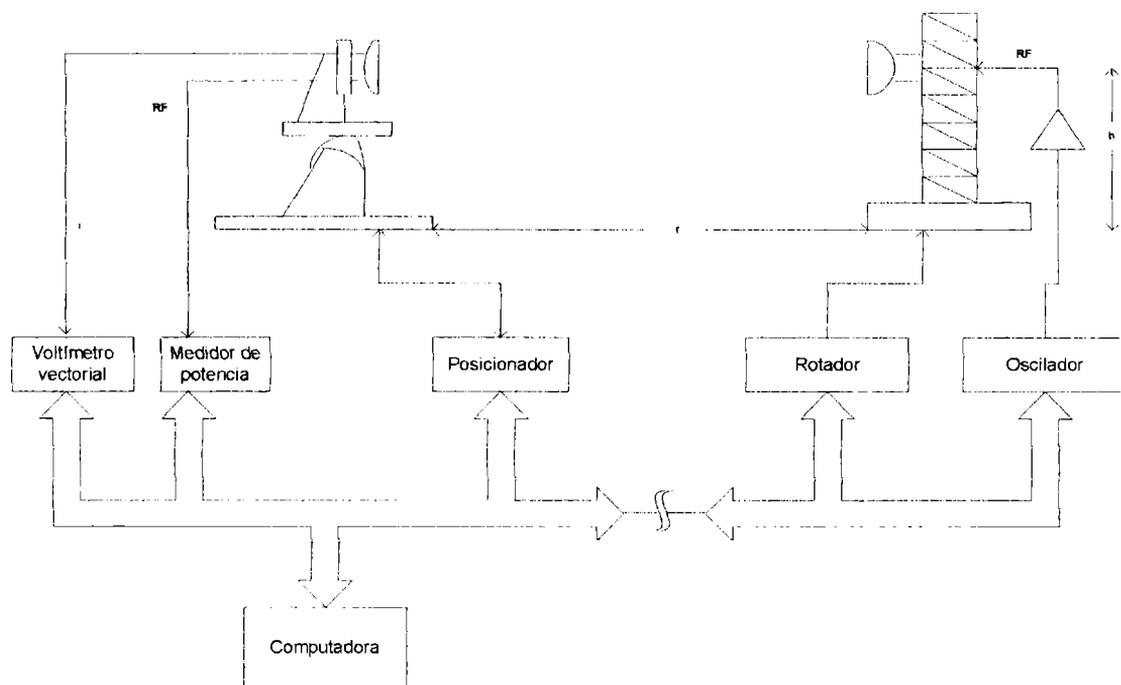


Figura 30

Restricciones del Sistema y del Proyecto:

Es importante considerar los requerimientos generales y particulares que el sistema de simulación deberá contemplar a fin de comprender en detalle el trabajo realizado. Se debe utilizar el instrumental disponible actualmente en el IAR: el Voltímetro vectorial 8085A, el medidor de potencia 438A y el Analizador de espectro 8593E. Estos instrumentos son relativamente lentos, si se tiene en cuenta su tiempo de respuesta de entre 30 y 120 ms., 100 ms., 160 ms respectivamente en comparación con el instrumental actuales en el mercado específicos para la medida de antena.

Ambiente de Desarrollo/ Operación/Mantenimiento:

El desarrollo fue realizado en las instalaciones del IAR, lugar donde también operará el Software implementado para las mediciones de antenas requeridas.

El sistema correrá bajo el entorno de LabWindows/CVI de National Instruments, corriendo en una plataforma Windows 2000, que introducirá el uso de herramientas orientadas a los Laboratorios Virtuales.

LabWindows/CVI es un ambiente de desarrollo completamente integrado, especialmente diseñado para la creación de sistemas de instrumentación basados en GPIB, PXI, VXI, tarjetas de adquisición de datos insertables y otros. El ambiente combina un interactivo ambiente de desarrollo de fácil uso con el poder y flexibilidad de programación de código compilado en ANSI C. LabWindows/CVI proporciona muchas características útiles que incrementan su productividad sin sacrificar la velocidad o fácil manipulación del código fuente compilado en C.

Funcionalidad del Sistema:

El sistema proporciona la siguiente funcionalidad:

- Controlar el movimiento del pedestal donde se monta la antena bajo estudio y obtener su ángulo de giro.
- Controlar y adquirir a través de la interfase GPIB los datos provenientes de los instrumentos de medida.
- Controlar remotamente a partir de comandos enviados desde la PC, los parámetros de frecuencia y amplitud del generador de señales, y que luego son transmitidos por la antena de referencia.

En los siguientes capítulos serán descritos en detalle cada uno de los ítems antes mencionados.

12. Control de Posición

Introducción

Con el fin de realizar las operaciones de adquisición y control en ambas coordenadas, azimut y elevación, en el Sistema de Medición de Campo Lejano se utilizó una interfase de entrada/salida digital. Las razones fundamentales de esta elección fueron las siguientes:

- Necesidad de realizar operaciones de control y monitoreo de baja velocidad, control de los motores, lectura de los sensores de posición límite, etc.
- La información de posición se encuentra disponible en sendos encoders de 13 bits que deben ser accedidos en forma serial con una velocidad mínima de 100kbts/seg.
- Evitar una sobrecarga no necesaria de la CPU con el uso de una interfase serie convencional.

La disponibilidad de LabView y LabWindows/cvi, ambos de la firma National Instruments, como entornos de desarrollo, llevó a la elección de una interfase de la misma firma con el fin de evitar la escritura del software de bajo nivel para manejarla. Efectivamente LabView se presenta como un excelente producto para la adquisición, análisis y presentación de datos, mientras que, LabWindows/cvi constituye una herramienta para la construcción de sistemas de adquisición y medición basados en GPIB, PXI, VXI e interfases que se insertan en el bus de la computadora personal.

Las características de baja velocidad y no necesidad de señales de "handshaking" han orientado la elección hacia una interfase de entrada/salida digital económica. Se prevé que las transferencias serán estáticas, que no requieren hardware de sincronismo y, se encuentran totalmente gobernadas por el software y la velocidad del procesador. La interfase PCI-DIO-96 fue seleccionada, pues, para cumplir con las tareas de control del sistema de posicionado.

Interfase PCI-DIO-96

La interfase PCI-DIO-96 es una tarjeta de entrada/salida digital con 96 líneas bidireccionales y programables. Su diseño se basa en el básico circuito integrado 8255 de INTEL, "PPI-Programmable Peripheral Interface". Cada uno de los cuatro "chips" que contiene se encuentran referenciados como PPI A, PPI B, PPI C y PPI D; a su vez, cada uno de éstos, aporta 24 líneas digitales de entrada/salida, repartidas en tres puertos denominados A, B y C, y un registro de control y estado. La interfase posee una gran flexibilidad a la hora de su programación por cuanto puede operar en modo unidireccional, bidireccional, de entrada o de salida, o de intercambio sincronizado por hardware[20].

El diseño de la tarjeta, que no contiene llaves y puentes, permite configurar su inserción en el bus PCI y su funcionalidad desde un programa. Adicionalmente puede interrumpir al procesador en aplicaciones críticas.

La Figura 31 muestra el diagrama funcional de la interfase PCI-DIO-96 donde se pueden apreciar los puertos de entrada/salida, la lógica de control y estado y su conexión con

el bus PCI.

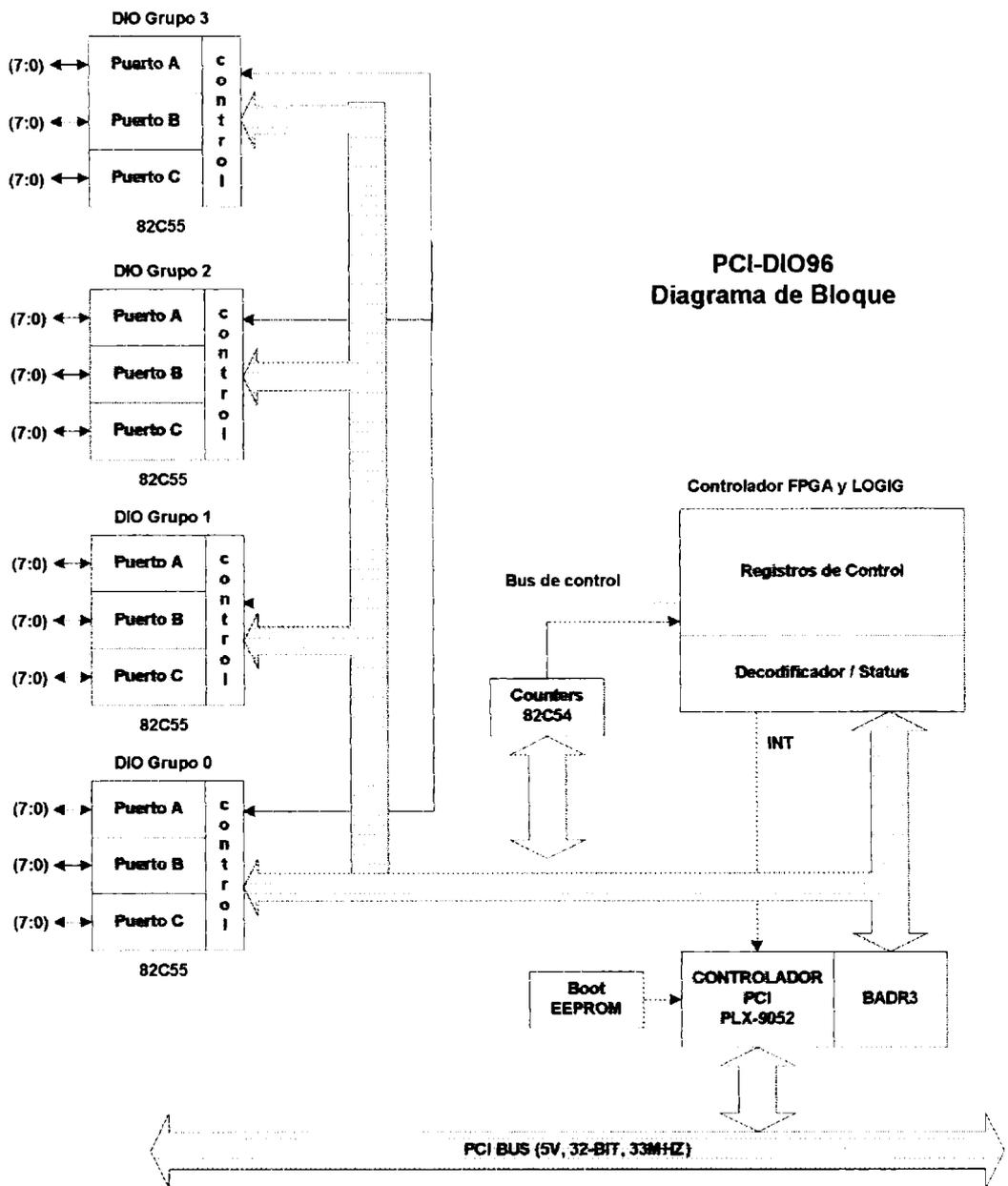


Figura 31: Diagrama Funcional de la interfase PCI-DIO-96

Mapa de los registros

Para programar y controlar la tarjeta la Unidad Central de Proceso escribe y lee sobre los puertos cuyas direcciones se encuentran incluidas en el espacio de memoria del bus PCI.

La Tabla 1 muestra el mapa de direcciones de los registros internos de cada uno de los

PPI. Cada uno de éstos ocupa cuatro direcciones, tres para los puertos de entrada/salida y uno para la lógica de control y estado. Según lo expresado en párrafos anteriores la tarjeta contiene cuatro PPI, de modo que, el espacio total de direcciones ocupado es de 16. Además cada uno de los circuitos integrados se encuentra referenciado por una de las letras A, B, C y D, comenzando con el que se encuentra mapeado en la dirección más baja[21].

| Dirección de E/S | Función |
|------------------|--------------------|
| BASE + 0 | Puerto A, grupo 0 |
| BASE + 1 | Puerto B , grupo 0 |
| BASE + 2 | Puerto C , grupo 0 |
| BASE + 3 | Control, grupo 0 |
| BASE + 4 | Puerto A, grupo 1 |
| BASE + 5 | Puerto B, grupo 1 |
| BASE + 6 | Puerto C, grupo 1 |
| BASE + 7 | Control, grupo 1 |
| BASE + 8 | Puerto A, grupo 2 |
| BASE + 9 | Puerto B, grupo 2 |
| BASE + 10 | Puerto C, grupo 2 |
| BASE + 11 | Control, grupo 2 |
| BASE + 12 | Puerto A, grupo 3 |
| BASE + 13 | Puerto B, grupo 3 |
| BASE + 14 | Puerto C, grupo 3 |
| BASE + 15 | Control, grupo 3 |

Tabla 1

Programación de la interfase

Básicamente existen tres opciones para programar la interfase PCI-DIO-96, como sucede con todas las interfases de adquisición de National Instruments:

- Programación de alto nivel haciendo uso de librerías genéricas disponibles en LabView y LabWindows.
- Programación de alto y bajo nivel haciendo uso de las librerías presentes en el software de adquisición NI-DAQ.
- Programación a nivel de registros.

LabView constituye un ambiente de desarrollo basado en una poderosa interfase gráfica que se programa con un lenguaje de programación, conocido como G, también gráfico, orientado al flujo de datos. Este entorno dispone de un conjunto muy completo de módulos denominados Virtual Instruments (VIs) que facilitan la comunicación con las tarjetas instaladas en el sistema.

LabWindows/cvi proporciona un ambiente de desarrollo, también con una poderosa interfase gráfica, y cuya programación se realiza con el lenguaje ANSI C. En este caso el fabricante pone a disposición de un conjunto de librerías que permiten la programación de las interfases de adquisición.

El software de programación NI-DAQ contiene los "drivers" necesarios para manejar el hardware de las interfases. Este paquete contiene un conjunto de librerías que

permiten controlar las interfases de dos maneras distintas: alto nivel, de muy fácil implementación y muy flexibles, y de bajo nivel, programadas a nivel de registro y optimizadas para ser ejecutadas en aplicaciones críticas. Cuando se utilizan las librerías disponibles en el paquete NI-DAQ para desarrollar un "driver" dedicado la aplicación se divide en dos partes: el "driver" para manejar el hardware y una interfase gráfica para controlar al "driver".

La última opción permite programar el hardware a nivel de sus registros y constituye el nivel más bajo de programación. Sólo se recomienda es opción cuando se quiere desarrollar un "driver" ajustado y optimizado para la realización de una determinada operación. En este caso, además de los conocimientos de programación, se requiere un muy buen conocimiento de la arquitectura funcional de la interfase que se programa.

La Figura 32 muestra la relación existente entre los distintos modos de programación y el hardware de adquisición.

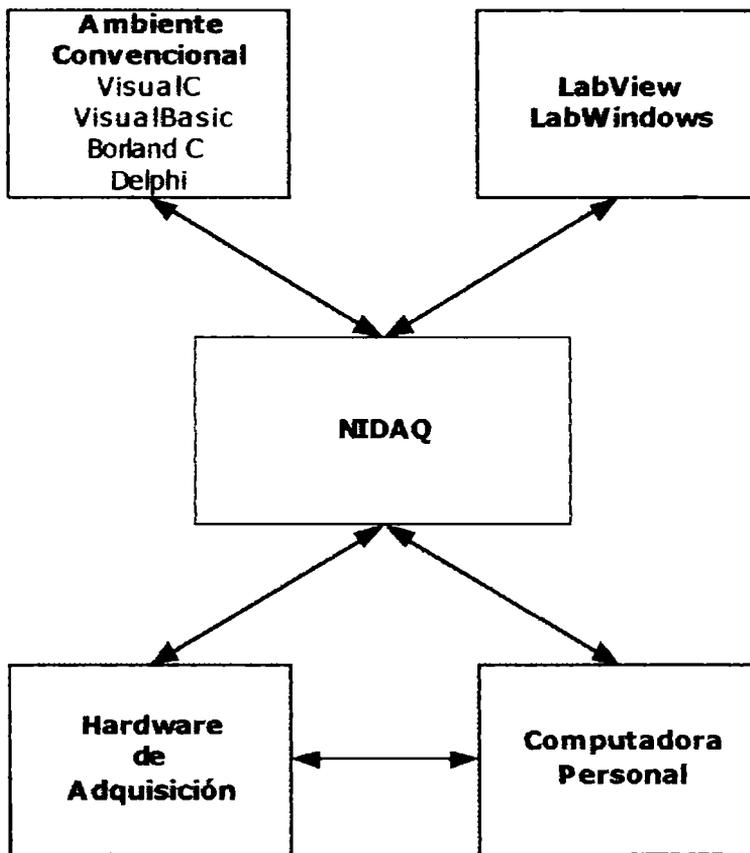


Figura 32: Software de Adquisición

Consideraciones de programación en el Programmable Peripheral Interface, 8255

De acuerdo a lo previsto en la introducción, las operaciones de adquisición y control en el sistema de posicionamiento se realizarán a través de operaciones de transferencia estática. En este sentido las transferencias se convierten en simples operaciones de entrada, adquisición de datos, y salida, control, sin necesidad de señales de

sincronismo.

Desde el punto de vista del PPI esta operatoria se logra configurando los puertos que se vayan a utilizar en "Modo 0", de entrada o salida según se realicen operaciones de adquisición o control respectivamente. Además se debe considerar que, luego de la inicialización del equipo, el PPI se configura para operar en modo cero con todos los puertos como entrada. Esta configuración resulta ser la más segura por cuanto no genera ninguna acción sobre los dispositivos externos hasta que no medie una orden bajo el control de un programa. Los tres puertos, A B y C, pueden ser programados como entrada o salida de 8(ocho) bits; mientras que el C presenta la posibilidad adicional de programar los "nibbles" inferior y superior por separado[22],[23].

La primera fase en el desarrollo de la conexión con el posicionador, consistió en la escritura de un programa que controlara el estado, de encendido y apagado, de un conjunto de 8(ocho) leds (diodos emisores de luz), conectados al PPI A Puerto A, y obtuviera el estado, abierto o cerrado, de un conjunto de 8(ocho) llaves, conectadas al PPI A Puerto B. La Figura 33 muestra el esquemático correspondiente. El objetivo de estos ensayos consistió en la obtención de experiencia en la realización de operaciones de transferencia sobre la interfase de entrada/salida digital.

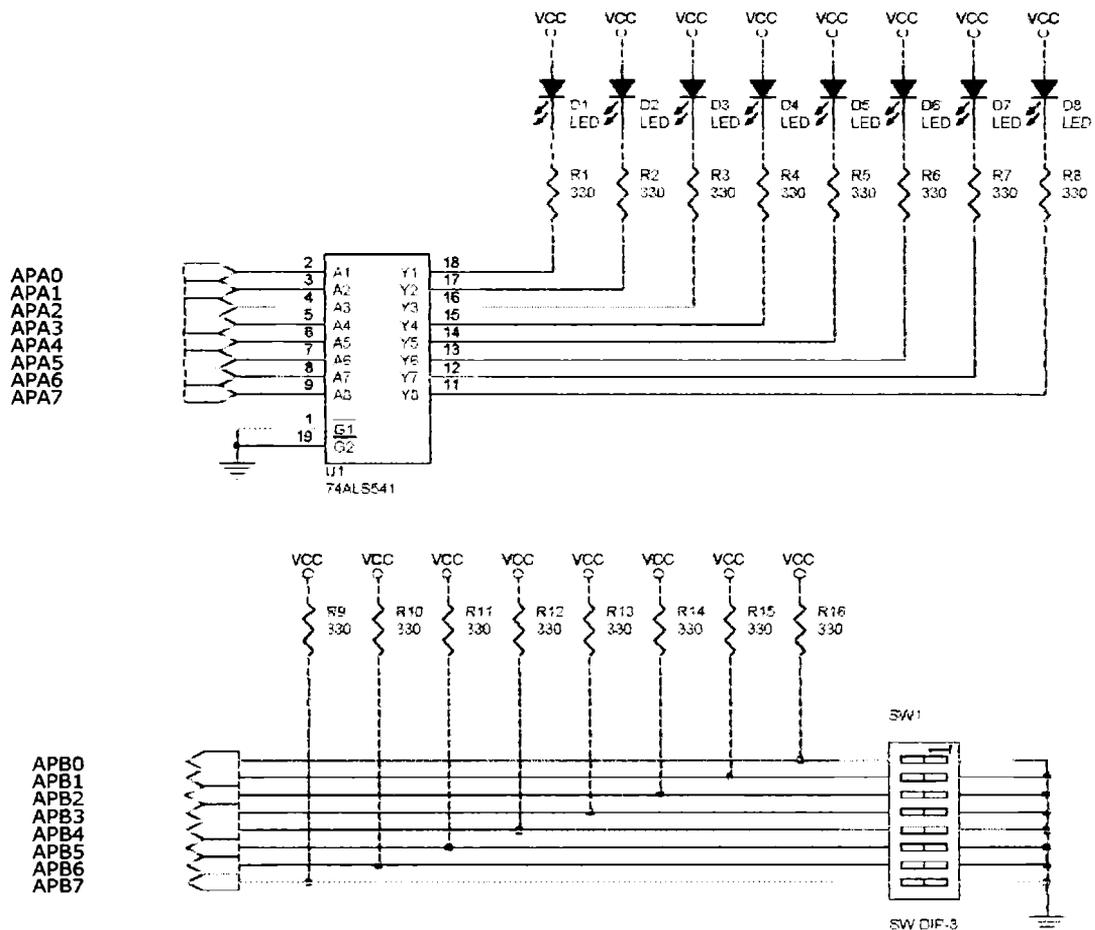
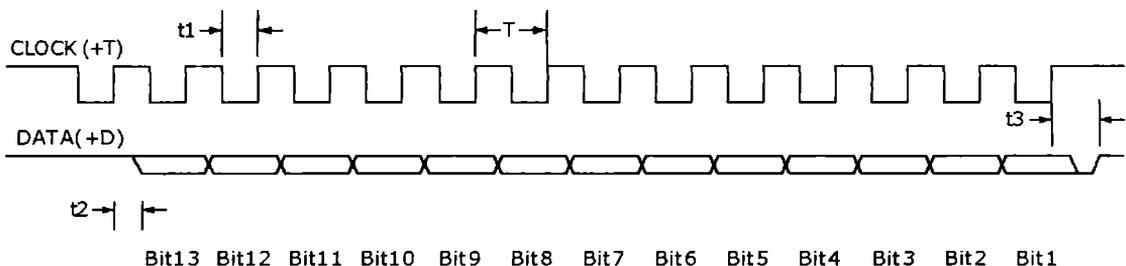


Figura 33: Circuito de Prueba I

Conexión con el Encoder

Antes de describir la conexión de la interfase de entrada/salida PCI-DIO-96 con el "encoder" se hará una descripción del tipo de interfase que dispone y los requerimientos del fabricante para obtener la información de posición. Se trata de un "encoder" marca Kubler, de procedencia alemana, modelo 585 del tipo absoluto de una vuelta. La interfase con el exterior es serie, compatible con el estándar RS485, de tipo sincrónica, "Synchronous Serial Interface-SSI", en que la obtención de los datos se realiza en forma sincrónica con el reloj según se ilustra en la Figura 34. En la misma se pueden apreciar los aspectos más importantes que deben ser tenidos en cuenta a la hora de implementar el hardware y el software:

- Las señales de dato de posición y reloj deben ser de tipo balanceado.
- El estado de reposo para las señales de dato y reloj es el uno lógico.
- El comienzo de la transmisión lo marca un START bit correspondiente a una transición hacia el estado bajo en el reloj y su permanencia en el mismo durante un tiempo mínimo de 1 us.
- Los valores de frecuencias mínima y máxima del reloj son 10 y 50 KHz respectivamente.
- La duración correspondiente al estado alto del reloj debe ser mayor o igual a 1 us.
- La duración correspondiente al estado bajo del reloj debe ser menor o igual a 0.5 us.
- Luego de 40 us de producirse la última transición positiva en la señal de reloj la línea de datos pasa al estado de reposo (uno lógico).



t1 // 1us

t2 ~ 0.5us(sincable)

t3 = max.40us

2us ■ T ■ 10us

Figura 34: Transferencia de los datos de posición a través de una interfase serie sincrónica, SSI.

La Figura 35 muestra el esquemático correspondiente a la conexión de la interfase PCI-DIO-96 con el "encoder" para la realización de los primeros ensayos. De acuerdo con el mismo el reloj se genera a partir del Bit 0 en Puerto C del PPI A y se transmite en forma balanceada a través de un excitador de línea, SN75176, cuyos niveles son totalmente compatibles con el estándar RS485. La recepción de los datos se encuentra optoacoplada (la conexión entre el "encoder" y la interfase se realiza por medio de un haz de luz dentro de un circuito integrado) a través del circuito integrado 6N136, que soporta una velocidad de transmisión de hasta 10 Mbits/seg, y cuya salida se conecta

al Bit 4 del Puerto A en el PPI A.

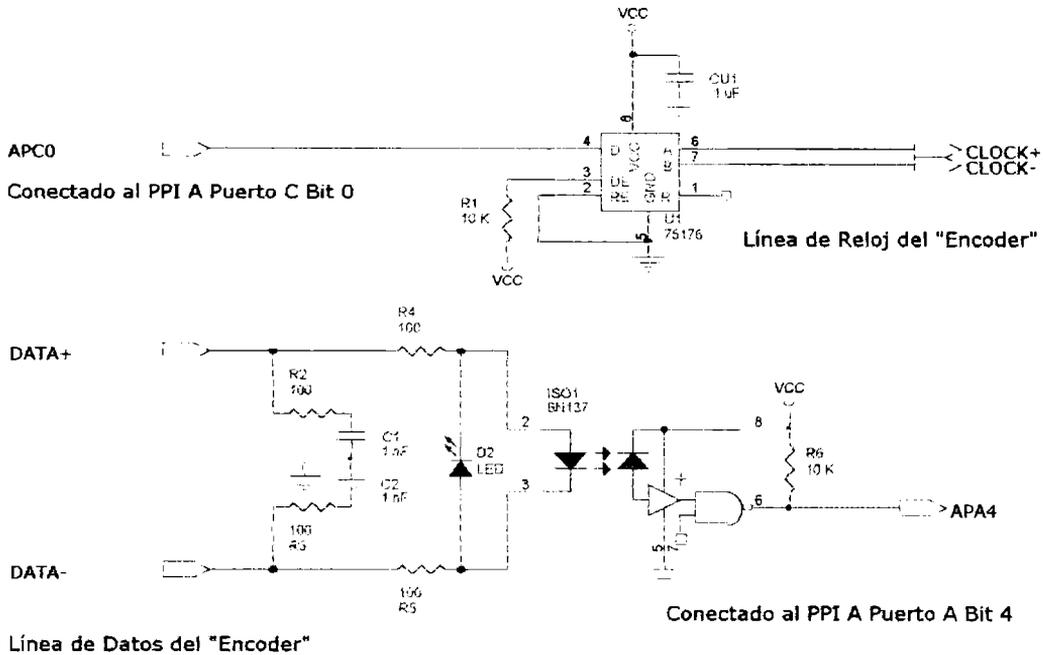


Figura 35: Conexión de prueba con el Encoder.

El programa desarrollado se basa en la experiencia anterior donde los diodos emisores de luz o "leds" han sido reemplazados por la señal de reloj y las llaves por los datos provenientes del "encoder".

Los resultados obtenidos no fueron satisfactorios por cuanto se obtenían distintos valores de posición con el "encoder" fijo. La medición de las señales con un osciloscopio dio la pista del problema: la señal de reloj no respetaba los tiempos sugeridos por el fabricante, que establecía un período máximo de 10us mientras en la realidad se obtenía 50 us. La explicación de este comportamiento fue la siguiente:

Las funciones de adquisición de LabWindows/cvi son genéricas y seguramente deben realizar demasiadas validaciones antes de ejecutar una acción. En el caso de la lectura del dato de posición se utiliza un lazo que se repite N veces correspondientes al número de bits del "encoder"; dentro del lazo se genera un pulso de reloj y sincrónicamente se lee el dato. Esto último significa que, al final del lazo, se han realizado "2*N" operaciones de escritura sobre la línea de reloj, correspondientes a sus N veces estados alto y bajo, y "N" operaciones de lectura de la línea de datos. Como estas transferencias son estáticas su tiempo de ejecución depende del programa y la velocidad del procesador y, de acuerdo a los expresado previamente, el uso de funciones genéricas sobrecarga innecesariamente al procesador estirando el tiempo de ejecución, disminuyendo la velocidad, de las operaciones de entrada/salida.

Estos resultados condujeron a la escritura de un módulo de adquisición donde las funciones de entrada/salida de LabWindows/cvi serían reemplazadas por otras, de bajo nivel, desarrolladas especialmente. Estas nuevas funciones serían implementadas mediante programación a nivel de registros de la interfase PCI-DIO-96 y se utilizaría la técnica de paginación; mediante ésta última se lograrían paginar las direcciones físicas

correspondientes a los registros de la interfase en la memoria virtual del proceso. Al escribirse y leerse en la memoria virtual se lo estaría haciendo directamente sobre la memoria física a través de la administración de un manejador o "handler".

La escritura del módulo de bajo nivel debería contemplar las siguientes acciones:

- Configurar los puertos de entrada/salida.
- Obtención de la dirección base, correspondiente a una dirección física, en que se encuentra implantada la interfase.
- Paginación de las direcciones de los registros físicos en el espacio virtual del proceso.
- Generar los pulsos de reloj.
- Leer los datos.
- Liberar los recursos asignados.

Configuración de los puertos: de acuerdo a la experiencia previa, la configuración de los puertos se haría a través del uso de las funciones genéricas provistas por National Instruments en su entorno de desarrollo LabWindows/cvi.

Obtención de la dirección base: los registros de la interfase se hallan mapeados en el espacio de memoria correspondiente al bus PCI y su valor se representa mediante una palabra doble (32 bits). El prototipo de la función que se utiliza para este propósito es la siguiente:

```
short Get_DAQ_Device_info( short Board,  
                           unsigned long Type_of_Information,  
                           unsigned long *Information_Value )
```

donde:

Board: es el número de dispositivo lógico.

Type_of_Information: tipo de información requerida.

Information_Value: valor correspondiente a la información requerida.

El número de dispositivo lógico es el asignado por el utilitario provisto por National Instruments con sus tarjetas de adquisición. Respecto a la información requerida existe una gran gama de posibilidades que permiten obtener una información completa de la configuración y su implantación en el sistema; al tratarse de la dirección base se utiliza el identificador ND_BASE_ADDRESS. El valor correspondiente a la información se almacena en una palabra doble cuya dirección se suministra a la función. Por último, la función retorna un valor nulo o negativo según su ejecución haya sido exitosa o no respectivamente; en el caso de haber fracasado el valor retornado contiene el código del error.

Paginación de las direcciones: las direcciones físicas correspondientes a los registros de la interfase se pagan en la memoria del proceso a través de una función cuyo prototipo es el siguiente:

```
int MapPhysicalMemory( unsigned int Physical_Address,  
                       unsigned int Number_of_Bytes,  
                       void *Ptr_to_Mapped_Address,  
                       int *Handle )
```

donde:

Physical_Address: dirección de comienzo de la zona a paginar.
Number_of_Bytes: tamaño en bytes de la zona a paginar.
Ptr_to_Mapped_Address: dirección de comienzo de la zona paginada.
Handle: instancia.

La instancia es un identificador que sirve para liberar los recursos a través de la función UnMapPhysicalMemory. La función retorna un valor nulo o negativo según su ejecución haya sido exitosa o no respectivamente; en el caso de haber fracasado el valor retornado contiene el código del error.

La generación del reloj y la lectura de los datos se obtiene escribiendo y leyendo sobre los bits correspondientes en los puertos respectivamente. Para escribir un cero lógico en un bit se aplica una operación AND, a nivel de bits, entre el contenido del puerto y la máscara, que contiene todos sus bits a uno menos el que se desea forzar a cero en el puerto, que contiene un cero. En el caso de escribir un uno lógico se utiliza la operación OR, a nivel de bits, entre el contenido del puerto y la máscara, que contiene todos sus bits a cero menos el que se desea forzar a uno en el puerto, que contiene un uno. El caso de la lectura de un bit es muy parecido sólo que ahora se aplica la operación AND, a nivel de bits, entre el puerto y la máscara, con todos sus bits puestos a cero a excepción del que se quiere leer en el puerto, que contiene un uno.

Aplicando lo expresado anteriormente tenemos:

```
#define deviceDIO96 1 // Dispositivo lógico
#define portW 2 // Puerto C
#define portR 0 // Puerto A

unsigned char *mapPORTW;
unsigned char *mapPORTR;
int memHandleW;
int memHandleR;

void InicializarTarjetaDIO96 ( )
{
    unsigned long devInfo;

    // Configura el puerto 2 de la tarjeta PCI-DIO96 como entrada y
    // deshabilita el handshaking
    DIG_Prt_Config (deviceDIO96, portW, 0, 1);
    // Configura el puerto 0 de la tarjeta PCI-DIO96 como salida y
    // deshabilita el handshaking
    DIG_Prt_Config (deviceDIO96, portR, 0, 0);

    MapeoDeRegistros (deviceDIO96, portW, &devInfo);
    // Mapea la dirección relativa del puerto contenida en devInfo en
    // un puntero a la dirección física del puerto
    MapPhysicalMemory ( devInfo, 1, &mapPORTW, &memHandleW);
    // clock 9
    *mapPORTW = *mapPORTW | 0x01;

    MapeoDeRegistros (deviceDIO96, portR, &devInfo);
    // Mapea la dirección relativa del puerto contenida en devInfo en
    // un puntero a la dirección física del puerto
    MapPhysicalMemory ( devInfo, 1, &mapPORTR, &memHandleR);
}
```

```

void mapeoDeRegistros ( unsigned long device, unsigned char puerto,
                        unsigned long *deviceInfo)
{
// Obtiene la dirección base de la tarjeta PCI-DIO96
  Get_DAQ_Device_Info(device, ND_BASE_ADDRESS, &devInfo);

  // Ejecuta el mapeo de registros

  // Verifica si el puerto pertenece al grupo 0
  if ( ( puerto >= 0 ) && ( puerto < 3 ) )
    //Mapea el puerto en el registro de direcciones. BASE + puerto
    devInfo += ( puerto + 0 );
  // Verifica si el puerto pertenece al grupo 1
  if ( ( puerto >= 3 ) && ( puerto < 6 ) )
    //Mapea el puerto en el registro de direcciones. BASE + puerto + 1
    // ( el 1 indica sumarle el control el grupo 0 )
    devInfo += ( puerto + 1 );
  // Verifica si el puerto pertenece al grupo 2
  if ( (puerto >= 6 ) && (puerto < 9 ) )
    //Mapea el puerto en el registro de direcciones. BASE + puerto + 2
    // ( el 2 indica sumarle el control del grupo 0 mas el control del grupo 1 )
    devInfo += ( puerto + 2 );
  // Verifica si el puerto pertenece al grupo 3
  if ( (puerto >= 9 ) && (puerto < 12 ) )
    //Mapea el puerto en el registro de direcciones. BASE + puerto + 3
    // ( el 3 indica sumarle el control del grupo 0 mas el control del grupo 1
    // mas el control del grupo 2 )
    devInfo += ( puerto + 3 );

  *deviceInfo = devInfo;
}

```

Lectura del encoder

Un "encoder" absoluto permite determinar la posición angular de un eje circular mediante un código único. Si el valor obtenido se le suma algebraicamente el valor del código para una posición de referencia se puede tabular un disco o visualizador que entregue el valor de la posición angular en forma invariante en el tiempo. A cada posición le corresponderá siempre el mismo valor respecto a dicha referencia aunque el sistema se apague.

En el módulo de lectura se obtiene la información de posición, compuesto por 13 bits de código Gray. El proceso de lectura parte del estado de reposo, con las líneas de dato y reloj en su estado alto, esperando la generación de un bit de comienzo, un cero lógico en la línea de reloj, para poder sincronizarse y comenzar con recepción de los 13 bits correspondientes a la posición angular. La sincronización con la señal de reloj se realiza de la siguiente manera: por cada bit de posición se deben generar dos pulsos en la línea de reloj, el primero positivo (uno lógico) y el segundo negativo (cero lógico) de modo que luego de 0.5 us de la transición positiva (de cero lógico a uno lógico) se dispone de información estable del bit de posición en la línea de datos. Una vez terminado el proceso de lectura de todos los bits la línea de reloj es llevada nuevamente a su estado de reposo, uno lógico, y el "encoder", luego de 40 us, responde con la línea de datos también en uno lógico. Por último, el dato de posición en código Gray se transforma a código binario para facilitar el manejo interno.

```

#define LINE_READ          0x10    // línea 4
int AnguloEncoder ( )
{
    unsigned long i;
    unsigned char bitCodigoGray;
    int posicion = 0x00;

    // Escribe un 0 en la línea 0 indicando bit de comienzo
    *mapPORTW = *mapPORTW & 0xFE;
    //Obtiene el dato de posición a partir de 13 bits de información recibidos
    for (i = 13;i>0;i--)
    {
        // Genera en la línea 0 un pulso con valor 1 y luego otro con valor 0
        // para indicar que va a leer un bit
        *mapPORTW = *mapPORTW | 0x01;    //escribe un 1 en la línea 0
        *mapPORTW = *mapPORTW & 0xFE;    //escribe un 0 en la línea 0

        // Lee el bit presente en el puerto de lectura en la línea LINE_READ
        bitCodigoGray = (*mapPORTR & LINE_READ)?TRUE:FALSE;
        posicion = ConcatenarEnCodigoBinario (posicion,i - bitCodigoGray);
    }

    // Pone la línea 0 ( línea de reloj) en estado de reposo
    *mapPORTW = *mapPORTW | 0x01;    //escribe un 1 en la línea 0
    //Retorna la posición angular
    return (posicion);
}

```

Para obtener la posición angular se debe seguir el siguiente procedimiento:

- Restar el offset correspondiente a la posición de referencia.
- Multiplicar el valor obtenido en el paso anterior por la resolución del encoder que se calcula a partir de la expresión:

$$\text{Resolución} = \Delta\emptyset / 2^N \quad \text{donde } \Delta\emptyset: \text{rango de medición}$$

N: número de bits del encoder

```

#define NRO_BITS          13        // número de bits de información del encoder
#define RESOLUCIÓN        360 / pow ( 2, NRO_BITS)

double PosicionAngular()
{
    double posicion;

    //Obtención de la posición angular
    posicion = AnguloEncoder() - offset;
    // Mantiene una variación sin sobresaltos en el código
    posicion = (posicion < 0)?posicion + 0x1fff:posicion;
    // Transforma la posición
    posicion = posicion* RESOLUCION;
    // Retorna la posición angular
    return(posicion);
}

```

Devolución de los recursos

Una vez utilizados los recursos de memoria se deben liberar para poder ser utilizados por otras aplicaciones.

```
void devolverRecursosPCI-DIO96 ( )
{
// Verifica que memHandleW no sea un puntero nulo
if(!memHandleW)
    {
        // Libera la memoria utilizada por el puerto de salida
        UnMapPhysicalMemory (memHandleW);
    }
// Verifica que memHandleR no sea un puntero nulo
if(!memHandleR)
    {
        // Libera la memoria utilizada por el puerto de entrada
        UnMapPhysicalMemory (memHandleR);
    }
}
```

La ejecución de estas funciones arrojó como resultado que la duración de los pulsos generados en la línea de reloj eran del orden de los 2 μ s, al igual que el tiempo utilizado para leer la información en la línea de dato. De esta manera se logró aumentar la velocidad de adquisición cumpliendo con las especificaciones de la hoja de datos del encoder y obteniéndose una lectura confiable de la posición angular.

Control de movimiento del pedestal

La antena bajo estudio, receptora, se monta sobre un dispositivo conocido con el nombre de rotador. La función del rotador es servir de montaje de la antena receptora, antena bajo estudio, y su funcionalidad incluye:

- *Puesta en servicio – On Line*
- *Puesta fuera de servicio – Off Line*
- *Movimiento a derecha en azimut*
- *Movimiento a izquierda en azimut*
- *Detención del movimiento en azimut*

Estas funciones se implementaron en Labwindows/CVI a través de la librería Traditional NI-DAQ, la cual provee entre otras de las siguientes funciones:

```
/*----- Prototipo -----*/
short DIG_Prt_Config (short Board, short Port,
                    short Handshaking, short Direction);
```

Configura el puerto especificado como entrada o salida, y habilita o deshabilita el handshaking.

```
/*----- Prototipo -----*/
short DIG_Out_Line (short Board, short Port, short Line,
                  short Line_State);
```

Inicializa o limpia el estado lógico de la línea digital del puerto especificado.

La implementación de las funciones requeridas para controlar el rotador se detallan a continuación:

```
void ConfigurarLineas()
{
    DIG_Prt_Config (1, 1, 0, 1);
}

void OnLine()
{
    DIG_Out_Line (1, 1, 6, 1);
    DIG_Out_Line (1, 1, 5, 1);
    DIG_Out_Line (1, 1, 1, 1);
}

void OffLine()
{
    DIG_Out_Line (1, 1, 6, 0);
    DIG_Out_Line (1, 1, 5, 0);
    DIG_Out_Line (1, 1, 1, 0);
}

void MoveRightAzimut()
{
    DIG_Out_Line (1, 1, 3, 1);
}

void MoveLeftAzimut()
{
    DIG_Out_Line (1, 1, 2, 1);
}

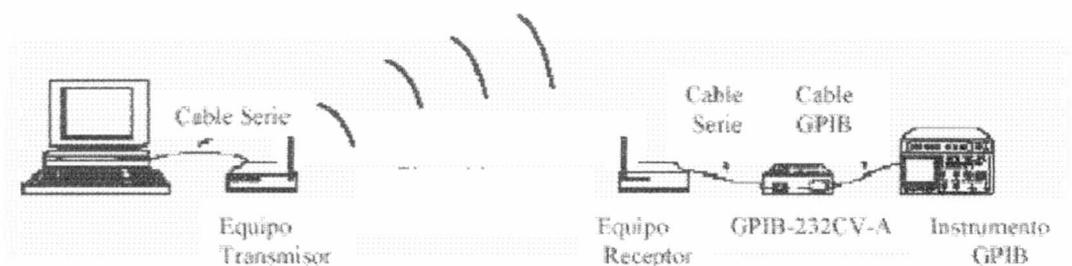
void StopMoveAzimut()
{
    DIG_Out_Line (1, 1, 3, 0);
    DIG_Out_Line (1, 1, 2, 0);
}
```

13. Control remoto del Generador de Señales en la antena de referencia

13.1 Descripción del Instrumental relacionado:

Los parámetros de frecuencia y amplitud del generador de señales, y que luego son transmitidos por la antena emisora, son controlados remotamente a partir de comandos enviados desde la PC.

Para efectuar la medición, se decidió comunicar la PC con el generador de señal a través de un enlace por aire, básicamente debido a la imposibilidad (costo y pérdida de señal) de efectuar un cableado. Para esto se requirió de los siguientes elementos: (como lo muestra el gráfico siguiente)



- Un equipo transmisor conectado a un puerto serie de la PC mediante la interfase RS-232.
- Un equipo receptor ubicado aproximadamente a 100 metros de la PC, conectado a un convertidor GPIB a través de un cable serie. El convertidor GPIB traduce comandos seriales a comandos GPIB. Este convertidor se conecta a su vez al generador de señales HP8590 D mediante un bus GPIB.

Debido a que se necesita probar la antena receptora sobre diferentes frecuencias, es importante establecer la configuración del emisor para poder efectuar las mediciones requeridas. Así, el programa de aplicación debe utilizar un puerto serie de la PC y configurarlo para sincronizarse en forma correcta con los equipos transmisor y receptor y el convertidor GPIB. Una vez realizada esta operación, se está en condiciones de establecer comunicación con el generador de señales. Los comandos de amplitud y frecuencia son enviados a través del protocolo de comunicación RS-232 al equipo transmisor. El equipo receptor recoge los datos y los envía al convertidor GPIB, quien traduce los comandos seriales a comando GPIB para que el generador de señales pueda entenderlos.

El módulo `Frecuencia_Amplitud` es responsable de inicializar la frecuencia y la amplitud del Generador de Señales.

La implementación de dicho módulo debería considerar las siguientes acciones:

- Como se necesita usar una función serial para establecer una comunicación con el Transmisor se debe abrir una "sesión" con el puerto serie 1 de la PC configurando algunos parámetros de dicho puerto, para ello se invoca a la

función `OpenComConfig`, utilizando como primer parámetro el valor literal 1 para indicar el puerto, especificamos también por medio de los parámetros que vamos a establecer una velocidad de transferencia de datos de 9600 baudios en el enlace de comunicación, con respecto al formato de los datos se utilizarán 8 bits para el carácter, 2 de parada y no utilizaremos bits de paridad, el tamaño de las colas de entrada y de salida será de 512.

- Como hace falta enviarle al aparato el comando "FREQ **freq** HMZ", donde **freq** es un parámetro de tipo `double` que indica la frecuencia requerida se debe realizar primero una conversión de tipos (`double` a `string`) y luego una concatenación para obtener el comando antes mencionado el cual es almacenado en la variable `buffer`. Dicha conversión es necesaria, para poder realizar luego la invocación a la función `ComWrt`. La cual recibe como parámetro los datos de la memoria especificada con formato `array` de caracteres.
- Se invoca a la función `ComWrt` para escribir el contenido de la variable `buffer` en el puerto COM 1. Como tercer parámetro de la función `strlen(buffer)`, ya que el valor retornado por la función `StrLen` puede ser usado para determinar este parámetro.
- Luego al tener que enviarle al aparato el comando "AMPL **ampl** dBm", donde **ampl** es un parámetro de tipo `double` que indica la amplitud solicitada se debe obtener el comando en la variable `buffer` procediendo de la misma manera que con la variable **freq**.
- Con el nuevo valor de `Buffer` se invoca a la función `ComWrt` para escribir su contenido en el puerto COM 1.
- Por último para liberar los recursos utilizados en esta sesión serial invocamos a la función `CloseCom` con parámetro 1, ya que este fue el número de puerto utilizado.

13.2 Implementación:

```
void setearFrecuenciaAmplitud8664A (double frecuencia, double amplitud)
{
    char dataS[256];

    OpenComConfig (1, "", 9600, 0, 8, 2, 512, 512);

    Fmt(dataS,"%s<%f",frecuencia);
    Fmt(buffer,"%s<%s%s%s","FREQ ",dataS,"MHZ\n");

    ComWrt (1, buffer, strlen(buffer));

    Fmt(dataS,"%s<%f", amplitud);
    Fmt(buffer,"%s<%s%s%s","AMPL ",dataS,"dBm\n");

    ComWrt (1, buffer, strlen(buffer));

    CloseCom (1);
}
```

14. Control de instrumentos de medida

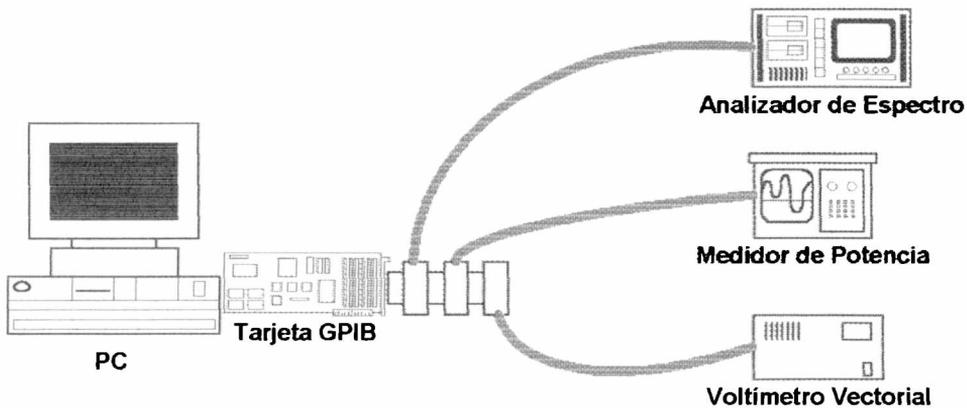
En este capítulo se verá como controlar los aparatos de medida utilizados para adquirir señales en el sistema de medida de campo lejano.

Los aparatos ha utilizar para la adquisición de datos en este sistema en particular son:

- Voltímetro vectorial hp 8085A
- Medidor de potencia hp 438A
- Analizador de espectro hp 8593E

Todos estos instrumentos deben poder ser controlados y automatizados simultáneamente desde una PC. Para esto se utilizó el protocolo de comunicación GPIB, donde a nivel físico, los instrumentos se conectan a través de un bus GPIB a una tarjeta controladora PCI-GPIB de Nacional Instruments ubicada dentro de la PC.

A continuación se muestra un esquema del subsistema de adquisición de señales de la antena bajo estudio.



En un sistema con estas características la PC se comunica con los instrumentos a través de la tarjeta GPIB mediante un software que contenga librerías para poder realizar la comunicación entre PC e instrumentos.

En particular, en este sistema se utilizó como entorno de desarrollo Labwindows/CVI, debido a que dispone de funciones para llevar a cabo una comunicación mediante el protocolo de comunicación GPIB.

Introducción a las funciones básicas GPIB

Para poder distinguir en el bus un instrumento de otro cada dispositivo se identifica con una dirección numérica, esta dirección está asignada en el instrumento y es leída por el driver GPIB. Esta dirección se maneja a través de un handle alfanumérico provisto por el lenguaje de programación. Así, uno de los primeros pasos para una correcta conexión es el direccionamiento de los instrumentos.

De esta forma para direccionar un instrumento con dirección 8 en el bus escribiremos en LabWindows el siguiente código:

```
int instrument;  
instrument = ibdev ( 0, 8, NO_SAD, T10s, 1, 0);  
if (instrument < 0) error ( );
```

donde los dos primeros argumentos de entrada se corresponden con la dirección de la tarjeta GPIB (en nuestro caso 0, porque disponemos de una sola) y la dirección del instrumento, 8. La función `ibdev` configura algunos de los parámetros del instrumento, así se observa que no hay dirección secundaria (`NO_SAD`), el tiempo fijado para el timeout es de 10 segundos, se elige mandar la señal `END` al final de cada operación de escritura y se elige 0 como caracter de final de cadena `EOS` (End Of String). La función devuelve un manejador (un handle) en la variable `instrument` para referirse al instrumento mientras dure la aplicación. Los handles válidos son enteros mayores que cero.

Cada vez que se ejecuta una función GPIB, se modifican tres variables globales de control definidas en el estándar GPIB (a través del archivo `gpib.h`) que podemos leer para verificar que todo ocurrió como se esperaba.

- variable de estado `ibsta`: En cada uno de sus bits se guarda información sobre aspectos de la ejecución. Por ejemplo si (`ibsta & ERR`) da como resultado verdadero, indica que se ha producido un error.
- Variable de error `iberr`: contiene el código de error producido luego de la ejecución de un comando GPIB
- Variable de cuenta `ibcnt`: contiene la cantidad de bytes transferidos en la última operación de comando de escritura o lectura.

Una vez obtenido el manejador del dispositivo antes de comenzar a enviarle órdenes al dispositivo es buena práctica limpiar el aparato de medida, inicializándolo a su estado de comienzo.

```
ibclr (instrument);
```

Las operaciones básicas para leer y escribir en los instrumentos son `ibrd` e `ibwrt`, respectivamente. Los comandos que se pueden enviar a cada instrumento GPIB se encuentran detallados en el manual de programación respectivo a cada dispositivo donde se explican los comandos que entiende.

Con la siguiente orden hacemos que el medidor de potencia 438A se resetee:

```
strcpy(comando, "IP");  
nbytes = strlen (comando);  
ibwrt (instrument, comando, nbytes);
```

La función `ibwrt` se compone de tres parámetros, el primero es el manejador del instrumento para indicarle a quien debe escribir en el bus, el segundo es el comando a enviar de bytes a transferir.

Luego de ejecutar `ibwrt`, como medida de seguridad debemos detectar posibles errores:

```
if ( ibsta & ERR ) printf ("Ocurrió un error");
```

En caso de presentarse un error, la variable `iberr` indica el error cometido. Ej:

```
If ( iberr & EARG ) printf ("Argumento no válido");
```

Cuando se quiere leer un dato de un dispositivo, primero hay que indicarle lo que se desea leer. De esta forma si queremos leer un dato del canal A del medidor de potencia 438A escribiremos:

```
ibwrt ( HP438A, "AP", 2);  
ibrd( HP438A, rd, 20);  
printf ("valor = %f dBm\n", atof(rd));
```

La función `ibrd` lee los datos del instrumento previamente especificados por la función `ibwrt`. Requiere de tres parámetros: el manejador del dispositivo sobre quien se leerá, una variable donde se contendrá los datos leídos y la cantidad de bytes que se desean leer.

Antes de salir de la aplicación debemos dejar el dispositivo en modo local para poder manejarlo desde la botonera, de lo contrario el instrumento quedará bloqueado.

```
ibonl (HP438A);
```

Comunicación con los instrumentos de medida del sistema de Campo Lejano a través de la interfase GPIB:

Para realizar la comunicación con los dispositivos de medida se consideraron las siguientes acciones

1. Apertura de los dispositivos como instrumentos virtuales y configuración de las características de medida de ese instrumento
2. Toma de datos de los dispositivos
3. Cierre de los dispositivos

Una vez inicializados los instrumentos, se estuvo en condiciones de realizar la toma de datos de medida de los instrumentos.

Los primeros pasos de lectura de medidas de los dispositivos se llevaron a cabo a través del voltímetro vectorial y el medidor de potencia dentro del laboratorio de desarrollo. La señal leída por cada instrumento era provista por un dispositivo generador de señales. Las primeras pruebas consistieron en medir la velocidad de adquisición de los instrumentos dando como resultado que el voltímetro vectorial medía a tiempos irregulares entre 30 ms y 120 ms y el medidor de potencia lo hacía de manera constante a una velocidad de 100 ms. Esto satisfizo con los requerimientos iniciales de partir de una cota máxima de 100 ms para obtener una resolución angular de 0, 04°. Avisado de las limitaciones de tiempo, el ingeniero Juan Sanz, director del proyecto del sistema de medición de campo lejano, modificó la cota máxima a unos 150 ms, de esta forma las limitaciones de tiempo quedaron solventadas dejando un amplio margen de tiempo ante cualquier eventualidad.

Luego de realizadas las pruebas de laboratorio, se estuvo en condiciones de realizar pruebas de campo para estudiar el comportamiento de los instrumentos por separado en una situación real ante la toma de datos de una medida de antena.

Las observaciones realizadas mostraron ciertos inconvenientes durante la toma de datos del voltímetro vectorial cuando la señal de entrada era muy débil y la sacaba fuera del rango de medición; bajo estas circunstancias el voltímetro tardaba en responder y generaba un enclavamiento del lazo de adquisición en la aplicación, los intervalos de adquisición se estiraban y en lugar de realizarse cada 150 ms lo hacían cada 250 ms o mas. Esta situación provocaba la pérdida de resolución angular, y algo mas importante, la pérdida de información en los alrededores de las singularidades (ceros) en el diagrama de la antena.

Para solventar este problema se hizo uso del conjunto de registros que tiene el instrumento a controlar. El problema en cuestión se debía a que se intentaba leer un mensaje de la cola de salida de mensajes del instrumento cuando ésta estaba vacía debido a una falta de señal. Dicha cola actúa sobre el bit 4, MAV : "MAV-Message Available" del Status Byte Register (STB), activando el bit 4 cuando se encuentran mensajes disponibles para ser leídos y desactivándolo en caso contrario. De esta manera, al preguntar previamente a cada lectura por el estado del bit MAV se logró resolver el inconveniente evitando leer el valor de una señal cuando ésta se encontrase fuera de rango poniendo un valor por defecto para indicar la situación. Como el tiempo de lectura de los registros de estado no es significativo y por lo tanto menor a los 150 ms, se determinó que el uso de estos registros era la mejor opción para evitar que los intervalos de adquisición se estiren superando la cota máxima establecida de 150 ms.

En el caso del medidor de potencia no hizo falta hacer uso de los registros de estados debido a que cuando este dispositivo se encuentra fuera de escala retorna el número +99990 en 5 ms., siendo este número un indicador de que el aparato no esta recibiendo datos.

Una vez resueltos los problemas antes mencionados que surgieron al implementar los módulos de medida el siguiente paso consistió en paralelizar la obtención de la posición angular de la antena bajo estudio y de los instrumentos de medida en cada toma de datos, ya que debían ser realizadas en el mismo instante de tiempo.

En ese momento se planteó un problema de sincronización Barrier, ya que todos los procesos debían comenzar a tomar sus respectivas medidas y luego esperarse mutuamente hasta que todos terminen de medir para continuar con la siguiente toma de datos.

Para que los procesos logren sincronizarse, se plantearon diferentes alternativas: Contador compartido, un proceso Coordinador y Barreras Simétricas.

La opción que utiliza un contador compartido fue descartada por generar Memory Contention ya que varios procesos compiten por el acceso a una variable compartida y su acceso continuo es muy costoso.

Fue descartada la opción de Barreras Simétricas por no ser legible su implementación para n procesos.

La alternativa elegida fue un proceso coordinador ya que requiere de un proceso extra pero no existe Memory Contention y el código resultó mucho mas legible que utilizando barreras Simétricas.

Conclusiones:

A partir del desarrollo realizado, se efectuaron un gran número de mediciones de simulación. Con los resultados obtenidos se logró mejorar el prototipo de la antena receptora. Para el IAR el desarrollo de estos prototipos resulta de vital importancia, dado que a partir de ellos se está construyendo la antena receptora del satélite SAOCOM 1-A.

Además, con los resultados que se obtuvieron y que se siguen realizando el IAR adquiere experiencia para futuros trabajos que el Instituto pueda desarrollar. En particular, está la posibilidad concreta de nuevos desarrollos vinculados a otros programas de la CONAE.

15. Referencias:

- [1] Procesamiento de señales analógicas y digitales. Ambardar Ashok. Thomson Internacional. 2002.
- [2] A Beginner's Guide to Scpi (Hewlett-Packard Press series) by Barry Eppler
- [3] SCPI Syntax & Style, 1999
- [4] <http://www.silge.com.ar>
- [5] <http://www.heidenhain.com>
- [6] <http://jdlope.tripod.com/encoders.html>
- [7] <http://www.ab.com/catalogs/C114-CA001A-ES-P/7encoder.pdf>
- [8] Digital Displays and Applications: Decoders and Encoders, Memories, Text Display, Text Data Acquisition, Viewdata
- [9] Introducción a los Sistemas Operativos, E Alcalde, J Morera, J A Perez, Campanero, McGraw-Hill, 1997
- [10] Fundamentos de los computadores, Pedro de Miguel Anasagasti, Parainfo 1992
- [11] Manual del HP425A Características y funciones. Configuración de la Interfase Remota (Capítulo 3)
- [12] www.redes.upv.es/rc1/practicas/rs232.pdf Transmisión serie RS-232
- [13] www.learobotics.com/proyectos/cuadernos/ct1/ct1.html Introducción RS-232
- [14] Redes de Ordenadores - 2 Ed - by Andrew S. Tanenbaum
- [15] The IEEE 488 bus (GPIB);: An introduction to the 488 bus from chips to debug - a self study subject by Eugene R Fisher
- [16] www.ni.com/pdf/manuals - GPIB Hardware Guide, March 2004
- [17] LabVIEW Programación gráfica para el control de instrumentación. Antonio Manuel Lázaro. Parainfo 1996.
- [18] IEEE 488.2 INTERNATIONAL STANDARD, IEC 60488-2 ,First edition 2004-05
- [19] Foster I. Designing and Building Parallel Programs. Addison-Wesley. 1994.
- [20] Hwang, K., Briggs, F. Computer architecture and parallel processing. McGraw-Hill. 1985.
- [21] PCI-DIO96 DIGITAL INPUT/OUTPUT User's Manual Revision 2 November, 2000
- [22] Sistemas en Tiempo Real. Interfaces para Sistemas en Tiempo Real doc. pdf
- [23] www.omega.com/manuals/manualpdf/MPCIDIO96.pdf
- [24] www.omega.com/manuals/manualpdf/M3892.pdf