



45  
Zejan  
**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**  
**Facultad de Ingeniería**

FALLA DE ORIGEN

**RED DE MONITOREO DE  
SIGNOS VITALES**

**TESIS PROFESIONAL**

Que para obtener el título de:  
**INGENIERO MECANICO ELECTRICISTA**

presentan:

**LUIS FERNANDO CANTO RAMIREZ  
MINERVA PATRICIA RUIZ GUERRERO**



Director de Tesis:  
**ING. ROMAN OSORIO COMPARAN**

México, D.F.

Mayo de 1995

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*A la Gloria del Gran Arquitecto del Universo.*

*A mis padres y hermanos:*

*Que con su apoyo y enseñanzas me permitieron dar este  
pequeño gran paso.*

*A mami:*

*Por que su sonriza siempre estará conmigo,*

*A mis hermanos de G7 que me enseñaron:  
Que con voluntad de ser se puede llegar a la conciencia,  
que con la conciencia se puede llegar al conocimiento,  
que con el conocimiento se puede alcanzar la libertad.*

*Luis Fernando Canto Ramirez*

*A mis hijos Mariana y Marvin:*

*Por estar conmigo, por que nunca perdieron la fé en mi,  
por mostrarme un cariñoso entusiasmo, su apoyo para no  
darme por vencida y seguir siempre adelante.*

*Por ustedes y para ustedes que son mi todo, gracias.*

*Los quiero mucho.*

*A mi mamá Doña Nevada:*

*Gracias por ser una columna de apoyo y cariño en toda  
mi vida y en todo momento. Para tí y para los que ya  
no están con nosotros este pequeño logro. Te amo.*

*A Gela:*

*Que ha sido un ejemplo para mí y por darme su apoyo  
incondicional.*

*Muchas gracias.*

*A Luis a Ricardo y a Román:*

*Les agradezco que sigamos siendo un equipo hasta el final.*

*Los amo mucho.*

*Minerva Patricia Ruiz Guerrero.*

*A mi madre:*

*Que gracias a su incondicional apoyo, a sus interminables  
estimulos, y en especial a su ejemplo a seguir con todo mi  
amor y respeto.*

*A mis compañeros Pato y Luigui:*

*Por que no solo fueron compañeros de trabajo y escuela,  
si no que fueron más que amigos, gracias por su amistad y  
apoyo.*

*Ricardo Duran Boccardo.*

**INDICE**



---

Introducción	1-1
--------------	-----

**CAPITULO 1**

Sistemas de comunicación	1-1
1.1 Comunicación de datos	1-3
1.2 Comunicación en serie y en paralelo	1-5
1.3 Tipos de transmisión serie	1-6
1.4 Unidades de interfase	1-6
1.5 Transmisión asíncrona	1-7
1.6 Transmisión síncrona	1-8
1.7 MODEMS	1-9
1.8 Procesadores de comunicaciones	1-10
1.9 Protocolos	1-13
1.10 Configuraciones de enlace de datos	1-14
1.11 Normas de transmisión	1-16
1.12 Norma RS-232C	1-17
Requisitos para el transmisor RS-232c	1-20
Requisitos del receptor RS-232C	1-21
1.13 Norma RS-422A/423A	1-22
1.14 Norma RS-485	1-23
1.15 Circuitos de comunicacion de datos (UART).	1-23
1.16 ACIA	1-25
1.17 USRT	1-26
1.18 USART	1-26

---

**CAPITULO 2**

Conceptos médicos	2-1
2.1 Mediciones electrocardiograficas [ECG]	2-2
Parámetros gráficos de un electrocardiograma	2-2
2.2 Electrofisiología celular	2-4
2.3 Secuencia de la despolarización y la repolarización	2-9
2.4 Terminología de las deflexiones electrocardiograficas	2-10
2.5 Mediciones de las deflexiones electrocardiograficas	2-14
2.6 Determinacion de la frecuencia cardiaca [FC]	2-17
2.7 Determinacion de temperatura [TEMP]	2-20
2.8 Determinación de presión o tensión arterial [PA]	2-20
2.9 Determinacion de frecuencia respiratoria [FR]	2-22
2.10 Análisis biomédico	2-22
2.11 Electrocardiografo	2-23
2.12 Frecuencia cardiaca	2-23
2.13 Temperatura	2-24
2.14 Presión arterial	2-24
2.15 Frecuencia respiratoria	2-25

**CAPITULO 3**

Red de monitoreo de signos vitales	3-1
3.1 Descripción del sistema	3-1
3.2 Operación del sistema de transmisión y recepción	3-5
3.3 Adquisicion de datos analogicos	3-9
3.4 Manejo de los actuadores	3-12
3.5 Utilización de las líneas de control	3-13
3.6 Oscilador	3-15
3.7 Selector de velocidad de comunicación	3-16
3.8 La interfase RS-232	3-17



3.9	Uso del MAX232	3-20
3.10	Convertidor de RS232 A RS422.	3-22
3.11	Programación del puerto de serie comunicaciones	3-24
3.12	Operación del sistema en lenguaje basic	3-24
3.13	Operación del sistema en lenguaje C + +	3.13

#### CAPITULO 4

	Sistema en red de adquisición y transmisión de datos	4-1
4.1	Conexión entre el sistema y la PC	4-2
4.2	Direccionamiento	4-4
4.3	Selección de la velocidad de recepción/transmisión	4-6
4.4	Canales de entrada analógicos y salidas digitales	4-7
4.5	Programación de la pc para la comunicación con el nodo	4-5
4.6	Definición del conector DB25.	4-11
4.7	Transmisión del sistema a la pc.	4-12
4.8	Especificaciones.	4-13
	Generales:	4-13
	Entradas Analógicas:	4-13
	Salidas Digitales:	4-13
	Entradas Digitales:	4-14

#### CAPITULO 5

	Diseño del programa (software)	5-1
5.1	Instrucción	5-1
5.2	Estructura del sistema de monitoreo de signos vitales y operación	5-29

	CONCLUSIONES	C-1
--	--------------	-----

<b>APENDICE</b>	<b>A-1</b>
Especificaciones del MC14469	
Especificaciones del MAX232	
Especificaciones del ADC0809	
Plantilla del circuito impreso del Nodo	
<b>GLOSARIO</b>	<b>G-1</b>
Terminos médicos	<b>G-3</b>
<b>BIBLIOGRAFIA</b>	
	<b>B-1</b>
Manuales	<b>B-2</b>
Hemerografía	<b>B-3</b>

## **INTRODUCCION**

---

---

En las últimas décadas se ha venido desarrollando en forma exponencial la tecnología de las computadoras y con esto también los diversos campos de aplicación. Inicialmente sólo fueron ocupadas como hojas de cálculo o procesadoras de textos, actualmente prácticamente no existe campo de actividad en el cual no se utilicen.

El área médica es una de las más exploradas y que con mayor frecuencia se va mejorando por su importancia para la supervivencia del ser humano. La automatización de los equipos médicos surge como una respuesta al volumen y diversidad creciente de los servicios que se solicitan y se brindan.

La RED DE MONITOREO DE SIGNOS VITALES se creó con el fin de facilitar y automatizar (hasta donde sea posible), la observación y el control de los instrumentos médicos que miden los signos vitales de los pacientes en estado crítico desde un lugar remoto, dando como resultado mayor rapidez y seguridad a la atención de las áreas donde el equipo es utilizado. Una de estas áreas es la unidad de cuidados intensivos, donde se necesita la vigilancia ininterrumpida de algunos parámetros fisiológicos, a diferencia de los pabellones de hospital o consultorios, en los que se hacen sin prisa las mediciones. Otros de los departamentos para los casos agudos son el quirófano, y la unidad de cuidados intensivos neonatal.

Para el diseño se consideraron varios elementos para mantener el mayor grado de confiabilidad del sistema. Uno de ellos fue el ritmo al que se hacen las mediciones, y que depende del parámetro por medir y el medio en que se valora. También es importante la frecuencia con que se obtienen las muestras, ya que constituye un componente crítico de la vigilancia al enfermo y que dicho intervalo debe ser lo suficientemente breve para detectar cualquier perturbación en el parámetro fisiológico que se vigila. Un ejemplo lo constituye el electrocardiograma. En el caso de los adultos, conviene que cada año el médico haga en su consultorio un electrocardiograma para detectar cambios menores del ritmo cardíaco, en tanto que la unidad de cuidados intensivos el procedimiento debe hacerse en forma continua, para detectar cualquier perturbación mortal en el ritmo, a consecuencia de infarto del miocardio. De este modo, el intervalo es de un año en el primer caso, y diez milisegundos en el segundo.

En el quirófano por lo general está presente un médico, en tanto que no siempre se cuenta con él en la unidad de cuidados intensivos, donde un personal especializado en cuidado crítico es el encargado de la asistencia inmediata del enfermo. Comunmente se da el caso que se confunda el funcionamiento defectuoso de un aparato y una anormalidad del paciente. Los aparatos de monitoreo se diseñaron con la idea de que deben ser sencillos tanto en su acceso (entrada) como en la información obtenida (salida), y contar con alarmas que señalen el momento en que los parámetros médicos excedan los límites aceptables, a su vez que tengan la mayor confiabilidad posible para evitar cualquier error de interpretación en los datos y poder detectar cualquier posible desperfecto.

Se han desarrollado diversos equipos para la detección y procesamiento de datos de signos vitales como los siguientes: electrocardiógrafos, detectores digitales o analógicos de frecuencia cardíaca, respiratoria, temperatura, presión arterial, encefalógrafos, etcétera y que al ser probados en diferentes centros de salud han dado buen resultado. Los datos que manejan estos instrumentos son datos analógicos, por lo que es necesario transformarlos a datos digitales para que una computadora pueda procesar la información que se genera. Esta transformación de datos, si no se hace de manera constante, repercute irremediabilmente en la pérdida de información valiosa. La confiabilidad del sistema puede verse disminuida si algunos de los datos muestreados son erróneos, o si por inducción de ruido o cualquier otro medio se leen datos irreales, por lo que con los elementos utilizados en el circuito se buscó disminuir esta posibilidad al máximo, debido a que existe un manejo de datos clínicos delicados, y el tener una alta incertidumbre podría ocasionar un mal diagnóstico médico.

La programación es lo más amigable posible, porque este sistema va dirigido a personas que seguramente han tenido muy poca o ninguna relación con equipos de cómputo. En su mayor parte se utilizan menús y ventanas, para evitar errores de interpretación y manejos incorrectos. El lenguaje C ofrece las facilidades para el manejo de esto, además que el tiempo en que tardan los procesos es bastante más corto que el que ofrecen otros lenguajes de alto nivel.

Los elementos principales de LA RED DE MONITOREO DE SIGNOS VITALES son tres, una computadora personal o PC, uno o varios sistemas de adquisición y transmisión de datos (nodos), y los dispositivos sensores de signos vitales. La computadora no requiere de mayores adaptaciones, únicamente es necesario que cuente con un puerto serie de comunicaciones RS-232. Los sistemas de adquisición y transmisión de datos, son el corazón de toda la RED DE MONITOREO DE SIGNOS VITALES, a través de éstos se maneja todos los datos analógicos que posteriormente son enviados hacia la PC como datos digitales. Los elementos que lo forman se seleccionaron buscando el grado óptimo de funcionamiento y confiabilidad, además de cubrir los requerimientos de las normas establecidas.

Aparte de ser un proyecto confiable en el proceso de datos, también se logró que su costo no fuera muy elevado y los elementos que lo constituyen sea posible conseguirlos sin grandes dificultades. Por su tamaño ocupa poco espacio y su manejo no implique riesgos ni problemas de conexión.

**Capitulo 1**

---

**SISTEMAS DE COMUNICACION**

Los sistemas de comunicación eléctrica se encuentran en todas partes donde se transmite información de un punto a otro. El teléfono la radio y la televisión han venido a ser factores de la vida diaria. Los circuitos de larga distancia cubren el globo terráqueo llevando, texto, voces e imágenes. Los sistemas de radar y telemetría desempeñan papeles importantes, vitales, en navegación, defensa e investigación científica. Las computadoras hablan con otras computadoras por medio de enlaces transcontinentales de datos. Los logros son muchos y la lista es interminable. ciertamente se han logrado grandes avances desde los días en que Samuel Morse enviara el primer mensaje telegráfico, también es cierto que en las décadas próximas veremos muchas nuevas hazañas en la ingeniería de la comunicación, aunque las aplicaciones potenciales sólo están limitadas por las necesidades, aspiraciones e imaginación del hombre.

Durante la primera mitad del siglo XX, la industria de la comunicación se dedicó a la transmisión y recepción de voz, ésta se basaba en sistemas analógicos (las señales son continuas en el tiempo) y sólo hasta los últimos años de este siglo se ha dado importancia a la comunicación de información digital (la señal se transmite y recibe por medio de muestreos en el tiempo), como tendencia de crecimiento dominante. Esta situación se hace evidente en la gran demanda por el uso de servicios remotos que ha sido resultado de las necesidades de las organizaciones públicas y privadas de tener información exacta y oportuna. Se han desarrollado sistemas complejos para el intercambio de información digital, estos sistemas incluyen la comunicación entre personas, personas-máquinas y entre máquinas, mediante el uso de dispositivos que permiten hacer más eficientes los canales de comunicación.

En términos de la informática, comunicación se define como el proceso por medio del cual la información se transfiere de un punto llamado fuente, en espacio y tiempo, a otro punto que es el destino o usuario. Un sistema de comunicación es la totalidad de mecanismos que proporcionan el enlace para la información entre fuente y destino.



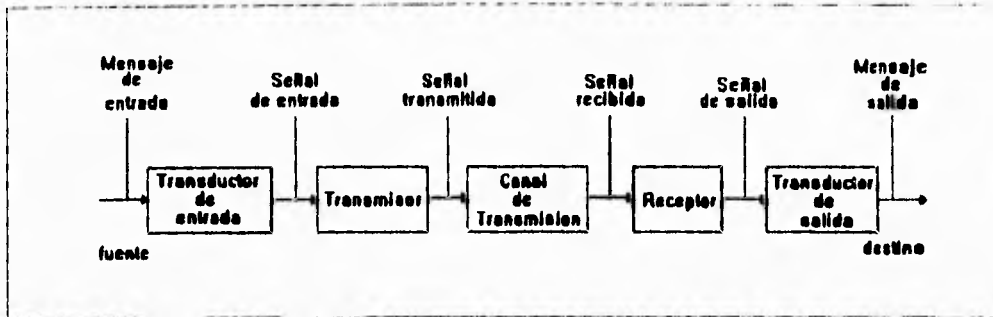


Figura 1.1 Elementos de un sistema de comunicación.

La figura 1.1 muestra los elementos funcionales de un sistema completo de comunicación. El sistema se aísla en entidades distintas para facilitar su esquemización, aunque en los sistemas reales la separación no resulta ser tan obvia.

Omitiendo los transductores, hay tres partes esenciales en un sistema de comunicación: el transmisor, el canal de transmisión y el receptor. Cada uno tiene una función característica:

El transmisor pasa el mensaje al canal en forma de señal. El canal de Transmisión o medio, es el enlace eléctrico entre el transmisor y el receptor, siendo el puente de unión entre la fuente y el destino. El receptor obtiene la señal deseada del canal de transmisión y la entrega al transductor de salida.

El canal de transmisión o medio es el enlace entre el receptor y el transmisor, siendo el puente de unión entre la fuente y el destino. Puede ser un par de alambres, un cable coaxial, una onda de radio o un rayo laser. Pero sin importar el tipo, todos los medios de transmisión eléctricos se caracterizan por la atenuación, la disminución progresiva de la potencia de la señal conforme aumenta la distancia. La magnitud de la atenuación puede ser pequeña o muy grande. Generalmente es grande, y por lo tanto, es un factor que debe ser considerado.

El receptor tiene la función de extraer del canal la señal deseada y entregarla al transductor de salida. Como las señales son frecuentemente muy débiles como resultado de la atenuación, el receptor debe tener varias etapas de amplificación. En todo caso, la operación clave que ejecuta el receptor es la demodulación (o detección), el caso inverso del proceso de modulación del transmisor, con lo cual vuelve la señal a su forma original.

### 1.1 Comunicación de datos

En la actualidad una de las principales áreas donde se aplican los conceptos de comunicación es en la computación, principalmente en lo que se refiere el manejo de datos desde sistemas remotos o redes locales. En la figura 1.2, se puede observar un sistema sencillo de comunicación de datos. La aplicación (AP) es el programa o proceso de información controlado por el usuario en una computadora. Ejemplo de esto son los programas de contabilidad, de nóminas, los sistemas de reservación de boletos de avión, los paquetes de control de inventarios, o los sistemas de gestión de personal.

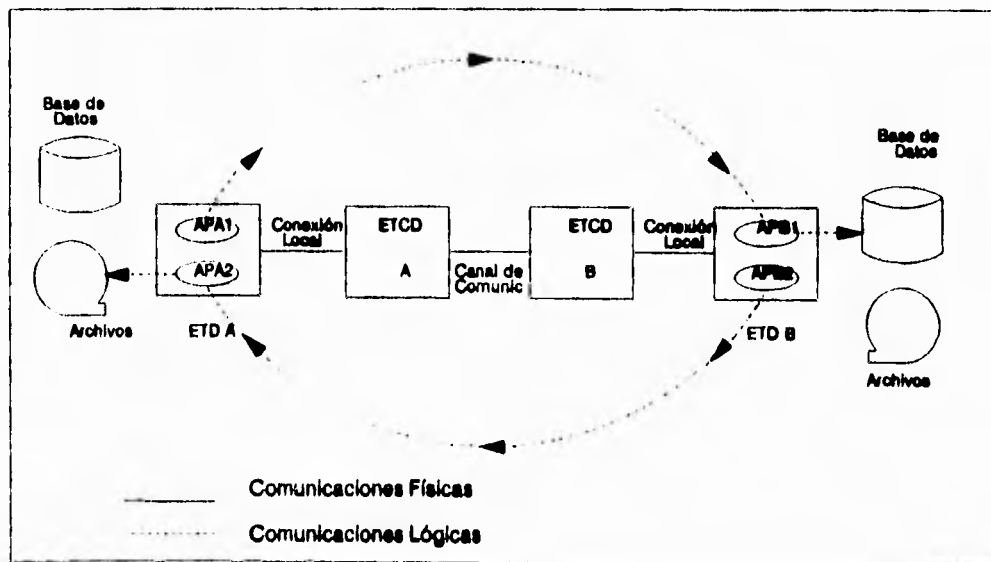


Figura 1.2 Sistema de Comunicaciones.

En la figura 1.2, el nodo A ejecuta un proceso de aplicación (APA1) en forma de programa para acceder otro proceso de aplicación situado en el nodo B, que en este caso incluye un programa (APB1) y una base de datos. También se observa un programa en el nodo B (APB2), que accesa al nodo A a través de un programa de aplicación (APA2).

La aplicación reside en el "Equipo Terminal de Datos" o ETD. Estas siglas suelen emplearse de forma genérica para definir a la máquina que emplea el usuario final. Algunos ejemplos de ETD son:

- Las terminales de trabajo para el control de tráfico aéreo.
- Los cajeros automáticos de los bancos.
- Las terminales punto de venta de un supermercado.
- Los dispositivos que muestran la calidad del aire.
- Las computadoras o terminales del correo electrónico.

En la actualidad la forma más usual para transmitir información entre equipos terminales de datos (ETD's) es el empleo del sistema telefónico, cuya organización típica se muestra en la figura 1.3 .

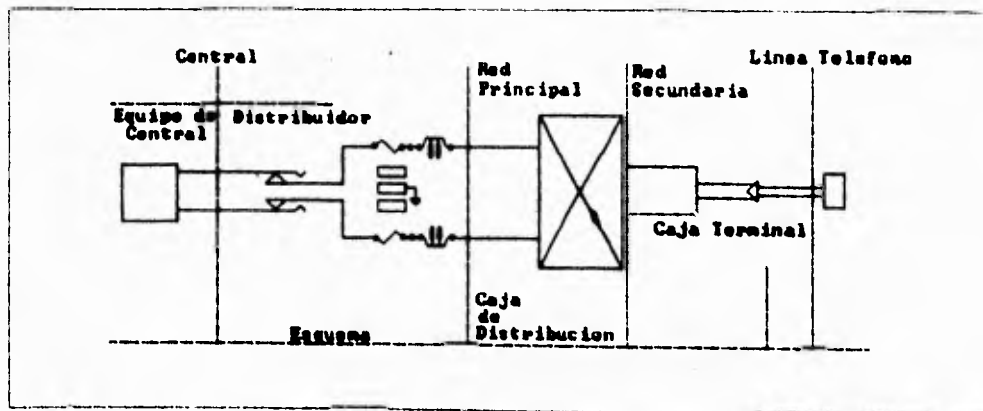


Figura 1.3 Organización típica de un sistema telefónico.

Otro elemento que se muestra en la figura 1.2, es el "Equipo de Terminación del Circuito de Datos" (ETDC), también llamado equipo de comunicación de datos. Su función es la de conectar los equipos ETD a la línea o canal de comunicaciones. Los ETDC diseñados en los años sesenta y setenta eran dispositivos exclusivamente de comunicación, para posteriormente ir evolucionando hasta incorporar más funciones de usuario, y algunos de ellos contienen parte de los procesos de aplicación.

## 1.2 Comunicación en serie y en paralelo

La transferencia de datos entre dos unidades se puede realizar de dos formas, en paralelo o en serie. En la transferencia en paralelo de datos, cada bit del mensaje (un bit es la mínima unidad de información), tiene una trayectoria propia y el mensaje total se transmite al mismo tiempo. Esto significa que un mensaje de  $n$  bits se transmite en paralelo a través de  $n$  trayectorias conductoras independientes. En la transmisión serie cada bit del mensaje se envía en secuencia, uno a la vez. Este método requiere de un par de conductores o un conductor y una tierra común. La transmisión paralela es más veloz que la serie pero requiere de más conductores; se usa para distancias cortas y donde la velocidad no es importante. La transmisión serie es más lenta pero más económica por que sólo requiere de dos conductores. La figura 1.4 muestra un ejemplo de cada uno de los tipos de transmisión.

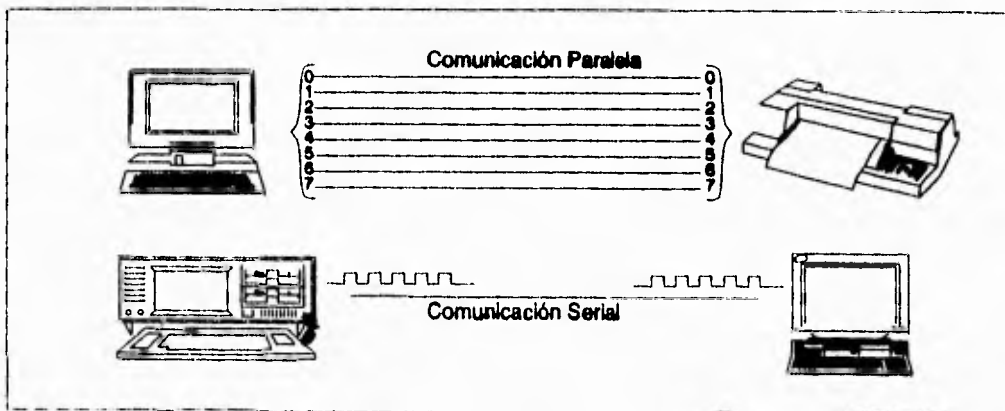


Figura 1.4 Ejemplos de los tipos de comunicación.

### 1.3 Tipos de transmisión serie

Los datos en serie se pueden transmitir entre dos puntos en tres modos distintos: **SIMPLEX, SEMIDÚPLEX O DÚPLEX COMPLETO**. En una **LÍNEA SÍMPLEX** se transmite información sólo en una dirección. Este modo rara vez se utiliza en la comunicación de datos porque el receptor no le puede comunicar al transmisor la presencia de errores. Un ejemplo es la transmisión de señales de radio y televisión.

Un sistema de **TRANSMISIÓN SEMIDUPLEX** transmite información en ambas direcciones, pero los datos sólo se pueden transmitir en una dirección a la vez. Para este método se necesita un par de conductores. El tiempo que se requiere para conmutar una línea semidúplex se denomina tiempo de cambio de dirección.

Una **TRANSMISIÓN DUPLEX COMPLETA** puede enviar y recibir datos en ambas direcciones en forma simultánea. Esto se logra por medio de un enlace tetraalámbrico (de cuatro conductores), con un par de conductores diferentes destinados a cada dirección de transmisión. En forma alternativa, un circuito biálambrico puede soportar la comunicación dúplex completa si el espectro de frecuencias se subdivide en dos bandas de frecuencia no superpuestas para crear canales de recepción y transmisión separados en el mismo par de conductores.

### 1.4 Unidades de interfase

Se dice que los dispositivos que tienen control directo de la unidad de procesamiento están conectados en línea. Estos dispositivos transfieren información binaria dentro y fuera de una unidad de memoria cuando se reciben comandos del procesador principal.

Para resolver diferencias como conversión de valores de señales, velocidades de transmisión, códigos o formatos de palabra o modos de operación, los sistemas de comunicación requieren de dispositivos especiales entre el procesador y algún dispositivo más, que supervizan y sincronizan todas las transferencias de entrada y salida de datos. Estos componentes se conocen como unidades de interfase (o enlace) porque sirven de

medio de enlace entre un bus de un procesador a un dispositivo en particular. La interfase seleccionada responde a un código y procede a ejecutarlo. Si se tienen que transferir datos, la interfase se comunica con el bus de datos del dispositivo y del procesador para sincronizar la transferencia.

La transmisión de datos serie puede ser síncrona o asíncrona. En la transmisión síncrona, las dos unidades comparten una velocidad de frecuencia de reloj común y los bits se transmiten en forma continua a la velocidad adaptada. En la transmisión en serie a larga distancia, las unidades transmisora y receptora son excitadas por un reloj independiente de la misma frecuencia. Se transmiten señales de sincronización en forma periódica entre las dos unidades para mantener sus frecuencias de reloj en sincronía. En la transmisión asíncrona, la información binaria se envía sólo cuando esta disponible, y la línea se mantiene desocupada cuando no hay información que transmitir.

### 1.5 Transmisión asíncrona

El formato más popular para la transmisión de datos es el asíncrono. Este formato especifica que cada palabra (por lo general consta de datos de cinco a ocho bits) debe estar precedida por un bit de arranque o un espacio, definido como un 0 lógico y seguida por uno o más bits de detención o marca definido como un 1 lógico. Entre una palabra y la siguiente se transmiten marcas, o sea bits de detención, en forma continua. Como cada carácter está enmarcado por los bits de arranque y detención, el receptor se resincroniza a cada palabra



Figura 1.5 Formato de un código asíncrono.

recibida, permitiendo intervalos diferentes de tiempo entre los caracteres. Es decir la información binaria se envía sólo cuando está disponible y la línea se mantiene desocupada cuando no hay información que transmitir.

El dato se transmite empezando por el bit menos significativo (LSB) acabando en el bit más significativo (MSB). El último bit de datos puede ir seguido por un bit de paridad, que es opcional y la paridad puede ser par o impar. En la figura 1.5 se puede observar el formato de un código asíncrono.

### **1.6 Transmisión síncrona**

En este tipo de transmisión la unidad transmisora y la unidad receptora comparten una velocidad de frecuencia de reloj común y los bits se transmiten en forma continua a la velocidad adoptada. En los casos donde la transmisión serie es a larga distancia las unidades transmisora y receptora son excitadas por un reloj independiente de la misma frecuencia, por lo tanto, se envían señales de sincronización en forma periódica entre las dos unidades para mantener sus frecuencias de reloj en sincronía. La transmisión de bits debe ser continua para mantener sincronizadas las frecuencias de reloj en ambas unidades.

Con los protocolos síncronos, el receptor obtiene la señal de reloj de los propios datos de entrada. Este reloj tiene la misma frecuencia que la de los datos. La longitud de la palabra de datos, la paridad y el número mínimo de bits de detección entre las palabras puede variar de uno a otro sistema, al igual que la velocidad de transmisión de bits (medida en baudios).

Los protocolos asíncronos son de realización sencilla pero ineficientes debido al exceso impuesto por los bits de arranque y detención para cada palabra. Son utilizados regularmente a velocidades bajas de transmisión.

## 1.7 MODEMS

La desconcentración del procesamiento de información involucra la existencia de terminales de datos localizadas en lugares remotos y por lo mismo, la necesidad de un medio de comunicación y de los dispositivos necesarios de transmisión-recepción, que permitan usar este medio de comunicación para el envío y recuperación de la información generada por los equipos terminales de datos. Las computadoras y terminales distantes se conectan entre sí a través de líneas telefónicas y otras instalaciones de comunicación públicas y privadas. La información se genera en forma de bits o señales discretas que presentan grandes requerimientos de ancho de banda, los cuales no pueden ser satisfechos por las redes de servicio público de transmisión de voz, que por naturaleza, fueron diseñadas para permitir la transmisión de la gama de frecuencias necesarias para hacer inteligible la voz humana y para reconocer al que habla. De esta manera podemos inferir muy fácilmente, la necesidad de modificar o transformar las señales binarias en señales analógicas y viceversa, con la ayuda de las técnicas de modulación más apropiadas. Los convertidores se conocen como acopladores acústicos o modems (de modulador-demodulador, también es común referirse a ellos como "Data-sets"). Un modem convierte señales digitales en tonos de audio que se transmiten a través de líneas telefónicas y también convierte tonos de audio de la línea en señales digitales para usarse en computadoras. Los pulsos de datos generados por las terminales, modulan a una señal de audiofrecuencia (portadora), de tal modo que al final de la línea, la señal modulada es demodulada, permitiendo así recuperar los pulsos de datos transmitidos. En la figura 1.6 se muestra los elementos funcionales principales de un modem que opera en modo duplex completo.



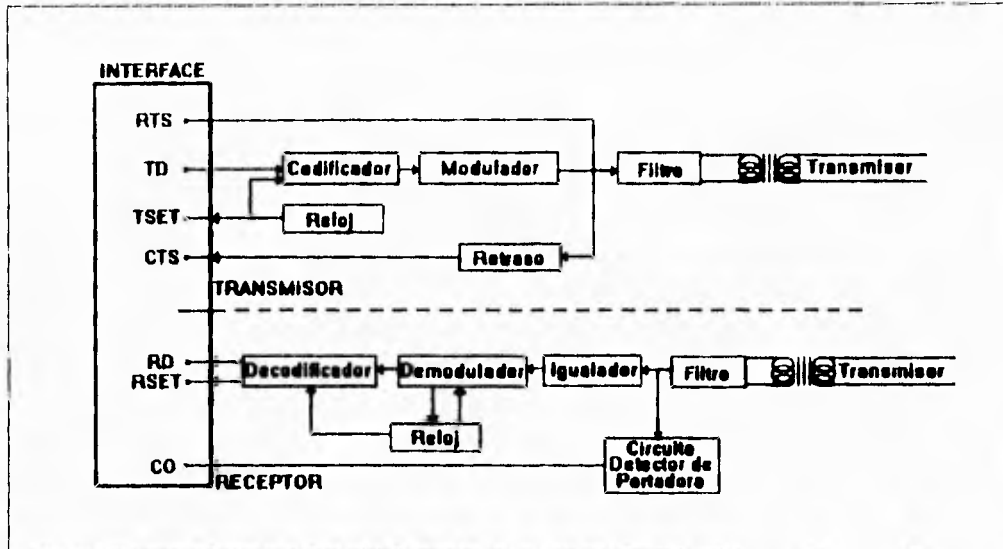


Figura 1.6 Elementos funcionales de un modem en modo Duplex.

### 1.8 Procesadores de comunicaciones

Al nacimiento de los sistemas de Teleproceso, surgió también la necesidad de realizar un control estricto sobre los dispositivos involucrados en el sistema, para lo cual se diseñaron funciones que permitieran tener control sobre la red, estas funciones se implementaron como parte del sistema operativo de la computadora y se realizaban dentro de ella, esto hacía que el proceso de comunicación de datos resultara en cierta manera deficiente; además con el crecimiento desmedido de los sistemas de Teleproceso, se incrementó aún más, el problema debido a la incompatibilidad y diversidad de equipo en el mercado. Esto dio motivo a que el diseño de dispositivos de transmisión de datos desarrollara un mecanismo capaz de resolver los problemas de control y compatibilidad, de este estudio surgieron los procesadores de comunicación inteligentes, mejor conocidos como procesadores frontales, en la figura 1.7 se ilustra el diagrama funcional de un procesador frontal en particular. Algunos ejemplos de estos dispositivos lo son el IBM-370X y CDC-2550X.

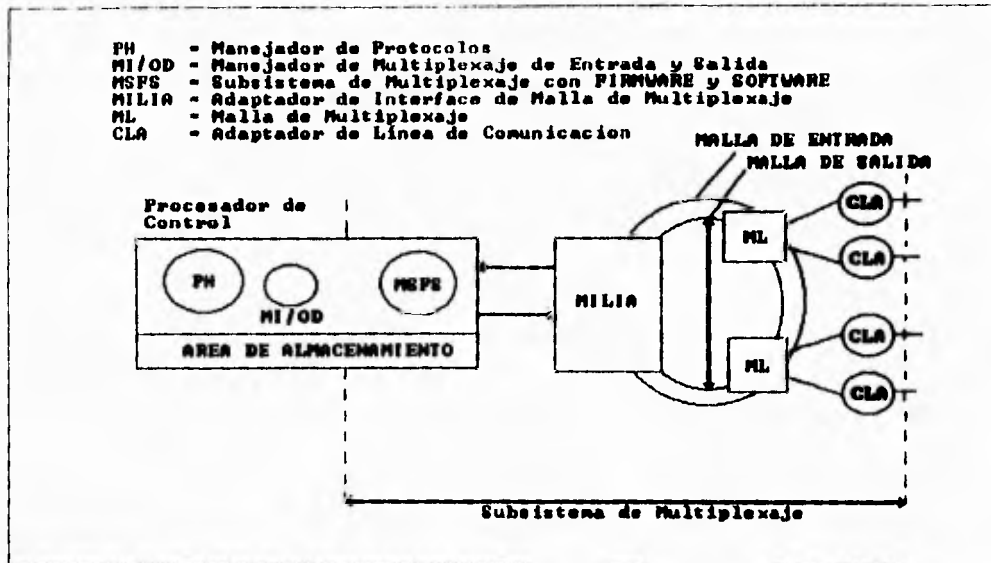


Figura 1.7 Diagrama de un procesador de comunicaciones.

Las funciones de control desempeñadas por los procesadores de comunicaciones frontales son las siguientes:

**Señalización de interconexión.** Es el manejo de señales individuales que interaccionan entre el procesador frontal o las terminales y el resto del circuito que facilita la transmisión.

**Sincronización.** Este proceso permite detectar la distribución del flujo de información a través de los medios de comunicación.

**Generación de caracteres de sincronía.** Para permitir la identificación de cada mensaje.

**Ensamblado y Desensamblado de caracteres.** Para proporcionar acoplamiento entre la computadora y el resto de la red para el proceso de los datos generados por las terminales.

**Llamado.** Es un proceso de direccionamiento secuencial que permite que las terminales conectadas al procesador sean invitadas a transmitir la información que tengan disponible.

**Ensamblado de mensajes.** Este proceso consiste en la acumulación, en áreas de trabajo, de caracteres con el objeto de formar un mensaje coherente para su recepción y verificación.

**Detección y control de errores.** Esta función se realiza por medio de una serie de circuitos lógicos y su objetivo es permitir confiabilidad durante la transmisión, la técnica utilizada es generalmente chequeo de paridad o redundancia cíclica.

**Conversión de códigos.** Esta actividad es una necesidad en las redes de Teleproceso, de tamaño considerable, en las cuales se trabaja indistintamente con varios códigos de transmisión.

**Control y monitoreo de la red.** Esta función proporciona al usuario informes acerca de la situación real de funcionamiento de la red, esta información es de tipo estadístico y además permite realizar una planeación para el crecimiento de la red.

**Enrutamiento por direcciones.** Esta función permite que puedan realizarse conexiones múltiples en un mismo puerto del procesador de comunicaciones, ya que cada mensaje lleva un encabezado de dirección.

**Manejo de colas de espera.** Mediante este proceso se organizan los mensajes de acuerdo a un orden establecido, particularmente cuando estos son recibidos de varias terminales.

**Asignación de prioridades.** Esta función involucra la aplicación de un criterio de prioridades en el manejo de colas de espera, es decir, los mensajes más importantes se envían primero.

**Conversión de formatos.** Mediante este proceso se logra la reestructuración de los mensajes para su correcta interpretación.

Las funciones anteriormente mencionadas se realizan dentro del procesador frontal mediante la programación de éste por software y están enfocados a:

- Proporcionar la flexibilidad necesaria para la interconexión de un rango bastante amplio de equipo y servicios de comunicación.
- Reducir los altos costos de comunicación mediante concentradores de control remoto.
- Proporcionar preproceso local de la información enfocada en el establecimiento de comunicación.

### **1.9 Protocolos**

Un protocolo de comunicaciones define la inicialización de la comunicación, el control de intercambio de datos, la terminación de la comunicación y lo más importante para el usuario; tiene implementadas dentro del mismo, técnicas para la detección y en su caso de recuperación de condiciones anormales de proceso de transmisión de datos; las cuales deben garantizar la integridad de la información en tránsito.

Los procedimientos para control de líneas o sistemas de control para enlaces de comunicación de datos, son protocolos usados para la transferencia de datos y control de información entre dispositivos de cómputo separado. Para lograr este objetivo, un protocolo de comunicaciones debe realizar las siguientes funciones: Sincronización (entre las partes a comunicar), control (para el acceso de los equipos), intercambio de datos y actividades de interrupción y desconexión, detección y control de errores.

Los protocolos de comunicación han sido tradicionalmente con orientación a caracter; es decir, utilizan singularmente o en secuencia, estructuras definidas de caracteres de un código dado, para convertirlos en información supervisora. Aunque los protocolos con orientación a caracter, (BYSYNCHRONOUS, Modo 4 de CDC, RMSI de Univ etcétera) representan la mayoría de los protocolos en uso actualmente.

Para organizar los diferentes tipos de protocolos establecidos por diversos comités de normas, se ha establecido una jerarquía de niveles.

El nivel 1 contiene los intercambios físicos, eléctricos y funcionales para establecer el enlace físico con el equipo terminal.

El nivel 2 contiene los controles de enlace de datos, tal como el BISYNC (IBM), SDLC (IBM), DDCMP (DEC), ADCCP (ANSI). Los parámetros clave de estos protocolos de enlace de datos aparecen en la siguiente tabla.

El nivel 3 establece los controles de trayectoria, o el formato y los procedimientos de control para las conexiones en una red. Estos controles incluyen los controles de ruta y de tráfico.

El nivel 4 es el control de sistema. Ejemplos de esto son DECNET (DEC) y SNA (arquitectura de redes de sistemas IBM).

El nivel 5 y niveles superiores no han sido definidos con claridad. Un comité de ANSI estableció una jerarquía de cinco niveles en la que el quinto nivel se refería al control de usuario. Otro de los comités de ANSI se pronunció por una jerarquía de seis niveles. La International Standardization Organization (ISO) ha establecido un protocolo de siete niveles

#### **1.10 Configuraciones de enlace de datos**

Un protocolo para el control de enlace de datos (Data Link Control) es un conjunto de reglas y procedimientos bajo los cuales los datos entre equipos terminales de datos (ETD'S) son intercambiados entre sí mediante un circuito de comunicaciones. Los equipos terminales de datos pueden ser terminales, concentradores de datos, procesadores de comunicaciones o computadoras.

Cuando los centros de conmutación fueron automatizados (por los años 50's), surgió la necesidad de usar caracteres de control, de manera que el equipo pudiese diferenciar entre texto y dirección; adicionalmente, de acuerdo a diferentes aplicaciones específicas, se usaron muchos otros caracteres de control que conformaron finalmente la cobertura del protocolo tal como lo conocemos actualmente.

La evolución continuó, dirigida principalmente por las necesidades de la industria de la conmutación de mensajes, tuvieron lugar algunas estandarizaciones, se empezaron a tratar los problemas de detección y corrección de errores, así como la necesidad para el control de dispositivos, ya con un enfoque para el establecimiento de comunicación entre equipos.

Un gran paso fue la implementación en forma dominante del protocolo síncrono binario (BYSINC) de IBM; pero retrospectivamente, en la actualidad, sólo podemos ver fallas en este protocolo como es la inexistente transparencia al usuario, ya que en dicho protocolo se emplean caracteres de código con aplicación a caracteres de control; esto impide que todos los caracteres del código sean empleados como parte de un texto. Además el control de dispositivos está mezclado con el control de la transmisión, de tal manera que fueron necesarias diferentes implementaciones para diferentes dispositivos y aplicaciones. Si bien este protocolo contiene deficiencias, de bastante gravedad como las enunciadas, es muy usado todavía en la actualidad.

Desafortunadamente también en la actualidad, cada fabricante de equipo de cómputo y equipo terminal pensó que podría fabricar un protocolo eficiente y consecuentemente existe una proliferación de protocolos, algunos ejemplos de ellos son: TM2 y TM4 de CDC; UCCP de UNIVAC; Protocolo Control de Enlace de Datos en modo Síncrono (SDLC) de IBM, conocido también como Protocolo con Orientación a Bit; Protocolo Estándar Control de Enlace de Alto Nivel (HDLC) de la Organización de Estándares; Control de Procedimientos Avanzado para Comunicación de Datos (ADCCP) del Instituto de Estándares Nacional Americano. Las características de los protocolos anteriores se resumen en la Tabla 1.1.

PROTOCOLO					
Característica	DDCMP	BISYNC	SDLC	ADCCP	HDLC
Sincrono/Asincrono	Sí	No	No	No	No
Dúplex	Sí	No	Sí	Sí	Sí
Semidúplex	Sí	Sí	Sí	Sí	Sí
Transparencia Datos	Cont.	Inser. ceros xCarac.	Inser. ceros xBit	Inserc. ceros xBit	Inserc. ceros xBit
Detección de error	CRC-16	CRC-16	CRC-CCITT	CRC-CCITT	CRC-CCITT
Cálculo de CRC	Texto y encabez.	Sólo texto encabez.	Texto y encabez.	Texto y encabez.	Texto y encabez.
Punto a punto	Sí	Sí	Sí	Sí	Sí
Multipunto	Sí	Sí	Sí	Sí	Sí

Tabla 1.1 Características de algunos protocolos de Comunicación.

### 1.11 Normas de transmisión

Para asegurar que dos o más estaciones se están transmitiendo los datos entre ellas en forma coherente es necesario establecer además de los protocolos una serie de normas de transmisión creadas por los comités de normas. La figura 1.8 resume cuatro normas definidas por la Asociación de Industrias Eléctricas (EIA) de uso común hoy en día.

La necesidad de normas para la transmisión de datos se hizo evidente a medida que maduró la industria. Al principio cada fabricante de sistemas establecía sus propias normas para sus productos. Al adoptar normas comunes, los fabricantes permitieron elegir la norma adecuada para su aplicación para después mezclar y acoplar computadoras y equipos periféricos conformes a dicha norma, independientemente de sus fabricantes, para configurar el mejor sistema posible.'

Las consideraciones clave en la selección de una norma de transmisión de datos son la longitud de la línea, velocidad de transmisión, condiciones ambientales, ruido, etc., y el sistema tendrá que interconectarse o no con otros sistemas, existentes o futuros.

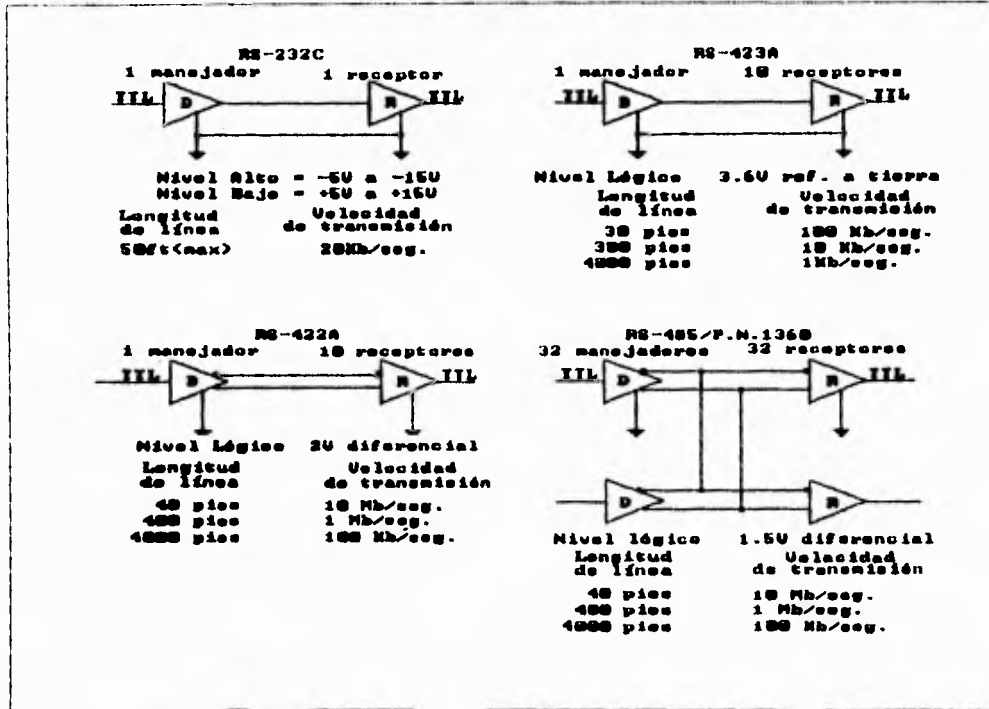


Figura 1.8 Normas de comunicaión.

### 1.12 Norma RS-232C

Para transmitir datos a través de la interfase serial, se debe hacer una especie de sorteo, mutuamente compatible. Como los datos seriales envían un bit por tiempo, debe haber un modo de sincronizar la operación de envío de datos entre el DTE y el DCE. La primer cosa que se debe hacer es acordar el rango de velocidad para la transferencia de datos. La velocidad en una línea de comunicación serial se expresa en rango de baudios. En forma más sencilla, el rango de baudios es el número de bits por segundo que ha sido transferido entre el DTE y el DCE. Por ejemplo, dos dispositivos seriales están transfiriendo datos a razón de 2400 bits por segundo.



Con respecto a esto, se llega a la conclusión de que los dispositivos seriales están transfiriendo datos en un rango de 300 caracteres por segundo de  $(2400 \text{ baudios} / 8 \text{ bits}) = 300 \text{ bytes por segundo}$ . Desafortunadamente, esta aceptación será incorrecta. Además de tener compatibilidad en el rango de baudios, los dos dispositivos deben formatear los datos de tal modo que sean posibles de sincronizar cuando éste se finaliza. Se llama caracter en lugar de byte porque no todos los datos ligan el envío total de bytes de datos. Para arreglos esta sincronización de transmisión de datos, los dos dispositivos seriales usarán bits de comienzo, paridad y paro con cada caracter de dato.

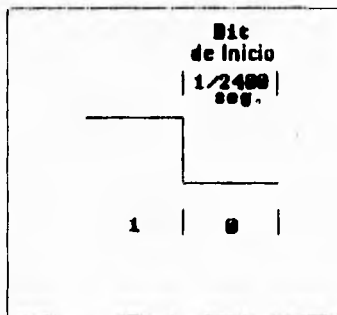


Figura 1.9 Bit de inicio.

**Bit de Inicio.** Cuando un dispositivo no está transmitiendo, éste aserta (lógicamente 1) con la línea TD. Para comenzar la transmisión del caracter de dato, el dispositivo serial envía un bit de comienzo. El bit de comienzo es un 0 lógico y dura la parte proporcional al tiempo del baud  $(1/2400 \text{ segundos})$ . Con el envío del bit de comienzo el DCE está disponible para sincronizarse con el transmisor esperando simplemente para que el DTE de la línea vaya de un estado lógico 1 a un estado lógico 0. El DCE entonces hace una pausa durante la mitad del tiempo de un bit y entonces empieza a

ejemplificar el dato que va llegando uno cada tiempo de un bit hasta que el caracter ha sido completamente transferido. Haciendo pausas cada medio período de bit, el DCE se dispone a mantener sincronización, siempre ejemplificando la cadena de datos entrantes aproximadamente a la mitad de cada bit. Si el DCE no provee esta pausa, puede causar una confusión para la cadena de bit entrante por la diferencia entre el DTE y el rango del reloj del DCE. La figura 1.9 muestra cómo un bit de comienzo es enviado a 2400 baudios. Se observa que el bit de comienzo dura  $1/2400$  segundos.

Después de que el dispositivo de transmisión serial, ha enviado el bit de inicio, éste enviará de cinco a ocho bits de datos. Para la mayoría de las transmisiones de datos, ya sean de siete u ocho bits de datos, son enviados. La figura 1.10 ilustra como el caracter 4 hexadecimal puede ser transmitido utilizando ocho bits.



Figura 1.10 Carácter transmitido utilizando ocho bits de datos.

**Paridad.** Algunas ligaduras de datos requieren un bit de paridad para la detección de errores. Las formas para detectar errores se detallan en la parte concerniente a protocolos, por lo pronto se mencionará que existen cinco formas básicas de paridad: sin paridad (none), par (even), impar (odd), marca (mark) y espacio (space). Como su nombre lo dice, "sin paridad" significa

que el carácter es transmitido sin un bit de paridad. Esta es una forma muy común de hacerlo cuando ocho bits de datos han sido transmitidos. Con la paridad par, el dispositivo serial de transmisión se enviará un bit de paridad 1 lógico después de los caracteres de los bits de datos, si es un número impar de bits de lógica 1 en los caracteres de los bits de datos. De otro modo el dispositivo de transmisión serial enviará un bit de paridad de 0 lógico después de los caracteres de los bits de datos. Así la paridad par asegura que existirá siempre un número par de bits de lógica 1 enviados con cada carácter e indicará un error si un número impar de bits de lógica 1 es recibido por el dispositivo de recepción serial.

Con la paridad impar, el dispositivo de transmisión serial enviará un bit de paridad 1 lógico después de los caracteres de los bits de datos. De otro modo, el dispositivo de transmisión serial enviará un bit de 0 lógico después de los caracteres de los bits de datos si hay un número impar de bits 1 lógico en los caracteres de los bits de datos. Así la paridad impar asegura de que haya siempre un número impar de bits de lógica 0 enviados en cada carácter e indicará un error si un número par de bits lógica 1 es recibido por el receptor del dispositivo.

Con la paridad marca el dispositivo de transmisión serial enviará un bit de paridad de 1 lógico después de cada carácter de los bits de datos.

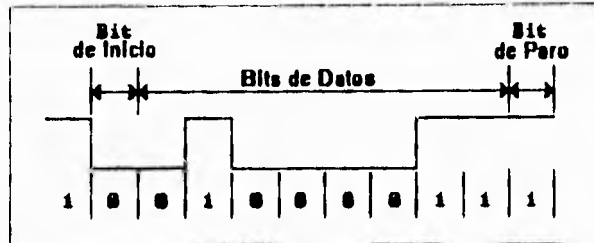


Figura 1.11 Carácter transmitido utilizando 7 bits de datos y 1 de paridad.

Con la paridad espacio, el dispositivo de transmisión serial enviará un bit de paridad de lógica 0 después de cada carácter de los bits de datos. La figura 1.11 muestra cómo el carácter 43 hexadecimal será transmitido por el dispositivo utilizando siete bits de datos de paridad par..

Como el bit de inicio significa el carácter de inicio, el bit de paro significa el fin del carácter. El dispositivo de transmisión serial siempre envía un 1 lógico como el bit de paro. Este además tiene que coincidir con la línea TD. Entonces la interfase serial se pone en un estado propio para el bit de paro. La figura 1.12 muestra como un carácter 43 hexadecimal es completamente transmitida usando ocho bits de datos y sin paridad.

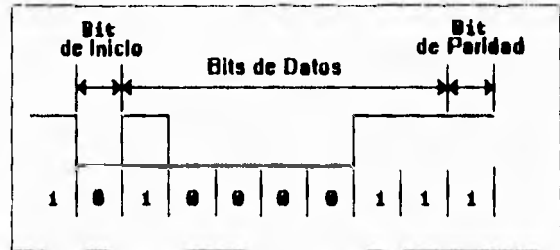


Figura 1.12 Transmisión de un carácter sin bit de paridad.

### Requisitos para el transmisor RS-232c

Los requisitos básicos de la RS-232C para transmisiones y receptores se resume a continuación:

La salida debe soportar un circuito abierto o corto circuito a tierra, a cualquier alimentación o con cualquier otro conductor en el cable.

- La impedancia en reposo (ausencia de alimentación) debe ser mayor a 300 ohms.
- El voltaje máximo en circuito abierto debe ser más o menos 25 volts.
- La impedancia máxima de salida en corto circuito debe ser 500 mA.
- Un 1 lógico binario es de +5 volts a +15 volts para una línea de salida de -3 volts a -15 volts para líneas de entrada con cargas de 3000 a 7000 ohms.
- Los tiempos de subida y bajada de la salida, dentro de los límites tradicionales de más o menos 3 volts, no deben exceder a 1 milisegundo.
- La pendiente máxima de salida no debe exceder a 30 volts/milisegundo.
- La velocidad máxima de transmisión de datos es de 20000 bits/segundo.

#### **Requisitos del receptor RS-232C**

- La impedancia de entrada debe ser mayor a 3000 ohms y menor a 7000 ohms.
- La capacitancia máxima de derivación para la entrada del receptor y cable conector debe ser menor a 2500 pF y no presentar componentes inductivas.
- Voltajes de entrada de circuito abierto menor a 2 volts.
- Límites de voltaje de entrada más o menos 25 volts.

### **1.13 Norma RS-422A/423A**

La norma **RS-423A**, propuesta en 1965, se refiere también a aplicaciones de un sólo hilo, como la **RS-232C**; sin embargo, la velocidad de transmisión y la longitud de la línea son notablemente mayores. La norma estipula también las formas de onda, que dependerán de la longitud del cable y de la velocidad de transmisión, para controlar reflexiones y emisión radiada o diafonía.

Para aumentar la longitud de la línea y la velocidad de transmisión de datos en la **RS-423A** se utilizan receptores muy sensibles con entradas diferenciales. El uso de estos dispositivos mejora considerablemente el rendimiento de la **RS-423A** respecto de la **RS-232C**. El paso siguiente es el empleo de una salida diferencial en el manejador y una entrada diferencial en el receptor, con lo que se obtienen aún mejores rendimientos. La configuración diferencial anula los efectos de cualquier desplazamiento de tierra o señales de ruido que aparecen como voltajes de modo común tanto a la salida del manejador como a la entrada del receptor.

La norma **RS-422A** se aplica al modo diferencial de operación. De esta forma se cubren transmisiones de hasta 10 Megabaudios para longitudes de línea de 40 pies y de 100 kilobaudios para longitudes de línea de hasta 4000 pies.

Los manejadores diseñados conforme a **RS-422A** son capaces de transmitir una señal diferencial de 2 volts a un par trenzado terminado en 100 ohms. Los receptores son capaces de detectar una señal diferencial de -200 milivolts en presencia de una señal de modo común de -7 volts a +7 volts. Ambas normas, **RS-423A** y **RS-422A**, contemplan la conexión simultánea de 10 receptores a la línea de un transmisor.

### **1.14 Norma RS-485**

Es relativamente nueva la norma RS-485, ésta tiene por objeto superar algunas de las limitaciones del RS-423A con respecto a la multiplicidad de transmisores y receptores en la misma línea. Inicialmente, esta norma se conocía como P.N.1360, pero hace poco se cambió su designación a la P.N.1488. Sin embargo, en Texas Instruments y otros catálogos aún se habla de P.N. 1360. Sus principales diferencias con respecto a la RS-422A son:

Voltaje de salida de modo común del manejador, -0.25 a +6 volts en RS-422A, ampliada al intervalo de -7 a + 12 volts, con o sin alimentación.

La entrada de voltaje al receptor en modo común, de -7 a +7 volts en RS-422A, ampliada a -12 a + 12 volts.

Impedancia de entrada al receptor aumentada de un mínimo de 4 Kilohms a un mínimo de 12 Kilohms.

El RS-422A y RS-423A permiten únicamente un transmisor y diez receptores en la misma línea. El RS-485 permite hasta 32 transmisores y 32 receptores en la misma línea.

Los dispositivos que satisfagan los requisitos RS-485 contarán con circuitos de protección internos para evitar fallas en caso de conectarse simultáneamente 2 transmisores. A esto se le llama protección de contención.

### **1.15 Circuitos de comunicación de datos (UART).**

Uno de los primeros y hasta ahora el dispositivo más popular asíncrono de recepción-transmisión de datos es el Receptor Transmisor Asíncrono Universal (UART). Este dispositivo combina un receptor y un transmisor independientes, cada uno con su propio bus de datos, puerto serie y reloj. La lógica de control que determina el número de bits de datos y si la paridad es impar o par (o si no hay paridad), es común a ambas, receptor y transmisor. Esta arquitectura permite operar en modo dúplex con velocidades de

transmisión diferentes. La lógica que controla el número de bits de detección afecta únicamente al transmisor. Esto se debe a que el receptor sólo busca un bit de detención. El transmisor el que necesita conocer el número mínimo de bits de detención que debe insertar.

El UART puede conectarse al microprocesador en una configuración interruptiva o en modo de escrutinio periódico. En el primer caso, la señal de registro de transmisión vacío solicita una interrupción del microprocesador, quien la atiende cargando nuevos datos y enviando el pulso de control correspondiente. Si el microprocesador no envía estos nuevos datos el UART sigue marcando la línea (transmitiendo bits de detención) indefinidamente.

Como el transmisor, el receptor también tiene un registro de corrimiento y un buffer como bloques fundamentales. Los datos son desplazados en forma serie en el registro de corrimiento serie-paralelo. Cuando se completa una palabra se transfiere al registro buffer de retención, quedando libre el registro de corrimiento para recibir la siguiente palabra. Cuando la palabra está disponible en el bus de datos, la señal de dato disponible sube a nivel alto y el registro de estados se carga con tres bits indicadores de error.

El primero es el indicador de paridad y señala la recepción de un bit de paridad errónea. El indicador de error de cuadro (framing) señala la recepción de un bit 0 de detención. Finalmente, el indicador de desbordamiento (overrun) señala que no se ha establecido la señal de dato disponible. Puesto que la entrada de restablecimiento de dicha señal se activa, por lo general, cuando se lee el dato, el indicador de desbordamiento normalmente señalará que se ha recibido una nueva palabra y se ha escrito encima de la que todavía estaba en el registro buffer.

El atractivo de la comunicación asíncrona radica en que el receptor y el transmisor pueden operar con relojes independientes. Para evitar la desviación de frecuencia y las fluctuaciones, y para proporcionar inmunidad al ruido, se utiliza un reloj con una frecuencia múltiplo exacto de la de datos, generalmente 16. Cuando el receptor detecta un bit de arranque por una transición de 1 a 0 en la entrada serie, comienza a contar los pulsos de reloj. Después de ocho pulsos (nominalmente el centro del bit), se muestrea la entrada serie. Si se detecta un 0, se supone que el bit de arranque es válido. Si se detecta un 1, se

supone que la primera transición era ruido, y el receptor es restablecido. Una vez detectado un bit válido de arranque, los datos se muestrean cada 16 pulsos de reloj.

Algunos de los UART más recientes utilizan relojes de 32 o 64 veces la frecuencia de entrada de datos para incrementar la precisión del centro de muestreo, incrementando así los márgenes de distorsión de datos.

### 1.16 ACIA

De uso menos general que los UART son los dispositivos de comunicación con estructura de bus desarrollados por algunos fabricantes de microprocesadores. Estos dispositivos pueden facilitar en gran manera la conexión al microprocesador ya que su bus de datos y las señales de sincronización necesarias han sido diseñadas especialmente para ese microprocesador.

Un ejemplo puede ser el adaptador interfase de comunicaciones asíncronas (ACIA). El ACIA (6850) está diseñado para conectarse directamente al bus del microprocesador 6800. Los bus de datos tanto del transmisor como del receptor están multiplexados en el propio circuito integrado para permitir la conexión directa del bus de datos del 6800. Los indicadores de estado y las líneas de control, también se multiplexan sobre el mismo bus.

Para el microprocesador, el ACIA no es sino dos localidades bidireccionales de memoria formadas una por el transmisor-receptor, y la segunda por los registros de control-estado. La selección del registro se efectúa por medio de un bit de dirección conectado a la selección del registro (RS) y la señal de control de lectura-escritura (R/W).

El ACIA dispone de tres entradas de selección que permiten elegir uno u otro circuito integrado en caso de conexión múltiple.

Además de las funciones del UART, el ACIA incluye las señales de control para modem RS-232C; petición de envío (RTS), detección de portadora de datos (DCD), y listo para enviar (CTS).



### 1.17 USRT

Los primeros circuitos integrados receptores-transmisores síncronos fueron básicamente UART, modificados para operar CCP síncrono. La palabra de sincronización queda almacenada en un registro especial. Cuando el receptor detecta la secuencia de bits de sincronización se inicia la recepción de la primera palabra. El transmisor envía los datos del buffer de datos o del registro de sincronización según se le ordene. Los USRT realmente controladores de comunicaciones; más bien son interfases síncronas.

### 1.18 USART

Más recientes que el UART o el USRT, los transmisores-receptores síncronos-asíncronos universales (USART) combinan un UART y un USRT con estructura de bus en un sólo chip. Por lo general incluyen también controles para modem.

**Canales de grado de voz.** Esta clasificación abarca todo el sistema telefónico público de transmisión de voz, el cual bajo ciertas condiciones es utilizado para transmisión de datos, este servicio se encuentra disponible en calidad de líneas privadas (LP'S). En México es el medio que se emplea comunmente para la transmisión de datos y se puede trabajar a velocidades hasta de 9600 bps con mucha confiabilidad.

**Canales de banda ancha.** Estos canales se caracterizan por manejar rangos mayores de velocidades, normalmente estos sistemas reciben un tratamiento especial para aplicaciones específicas como lo es en la aplicación de redes de computadoras. El rango de velocidades empleado en la actualidad en canales de este tipo es del orden de megabits/segundo (Mbps).

Hasta aquí se realiza el análisis y descripción de los terminos y elementos de la informatica aplicados a la RED DE MONITOREO DE SIGNOS VITALES. En el capítulo siguiente se realizan la descripción de los terminos médicos y características de la señales de los signos vitales de un paciente.

**Capítulo 2**



**CONCEPTOS MEDICOS**

Actualmente las inovaciones se presentan en cualquier área de trabajo dando lugar a que lo existente se mejore o se sustituya, pero siempre pensando no sólo en la optimización de los resultados, sino también en el costo. Esto se logra con el uso de varios dispositivos electrónicos existentes que sirven de herramientas ideales para estos propósitos y en combinación con una computadora personal y su respectiva programación.

En el área médica los cambios son todavía más exigentes debido a que es una área muy delicada y a la vez solicitada, por eso se pretende que las mejoras sean altamente confiables y no costosas para ser utilizadas con buena demanda por el usuario.

Los análisis computacionales de algunos signos vitales, como los del electrocardiograma, han sido muy estudiados y aplicados con resultados positivos. En este capítulo se describen brevemente los signos vitales más importantes para su aplicación al sistema en cuestión.

Este desarrollo ha sido provechoso tanto para el paciente como para el médico. Para el médico porque esto le ha restado tiempo al tedioso y lento análisis manual de la señal del electrocardiograma, así como de algunos otros signos como la presión arterial, la temperatura, etc. Para el paciente porque le permite realizarse exámenes con mayor frecuencia y con un factor de costo muy reducido.

Algunos sistemas existentes en el mercado pueden sensar algunos signos vitales además de analizar e interpretarlos, pero no lo hacen para varios signos vitales al mismo tiempo. Con el sistema RED DE MONITOREO DE SIGNOS VITALES se puede sensar un total de ocho diferentes signos vitales al mismo tiempo y desplegar toda la información necesaria de hasta cuatro pacientes simultáneamente, en un monitor de computadora personal sin dar una interpretación, pero sí con la capacidad de activar alarmas sonoras o visuales en casos de situaciones críticas. Estas alarmas se establecen con parámetros que sirven de límites superior e inferior, y estos límites los establece el médico o personal autorizado que utilice el sistema. Además el monitoreo es constante y en caso necesario la información de cada paciente puede ser almacenada en unidades de disco.

El algoritmo que se emplea detecta todos los datos que registran los sensores, porque en función de esos datos se hace la evaluación correspondiente, y también despliega la información de forma tal que pueda ser interpretada fácilmente.

Para evaluar adecuadamente los datos que proporcionan los sensores de los signos vitales, es necesario establecer la terminología aceptada y los parámetros médicos correspondientes. Por lo cual, a continuación se realiza el análisis de los diferentes parámetros que se utilizan para este caso.

### **2.1 Mediciones electrocardiográficas (ECG)**

El electrocardiograma (ECG) es un registro cronometrado y amplificado de la magnitud y dirección de las fuerzas bioeléctricas generadas por la despolarización y repolarización de las células miocárdicas durante cada ciclo cardíaco.

El uso de sistemas microcomputarizados dedicados al estudio de la señal del (ECG) permite obtener una solución de bajo costo, alta eficiencia y disponibilidad. Pero la captación de todos los datos para poder mostrar la señal correspondiente debe ser correcta y sin perder ningún latido, sea normal o anormal.

#### **Parámetros gráficos de un electrocardiograma**

Para medir las amplitudes y duraciones, primero se identifica el nivel isoeléctrico de la onda cardíaca y, centrado en este nivel, se define una ventana que representa el nivel de ruido en cada derivación; luego se identifican los cruces por cero que señalan los inicios y términos de las ondas, complejos y segmentos.

**Nivel Isoeléctrico.** Este se determina como el promedio de los valores medios de los puntos comprendidos entre el inicio de cada onda QRS y cinco muestras antes de dicho punto. En caso de haberse detectado una onda T, el valor recién obtenido se promedia con

el promedio de los valores medios de los cinco puntos inmediatamente posteriores a cada onda T. El promedio final se calcula eliminando el valor con mayor dispersión, ver figura 6.

**Nivel de Ruido.** Se determina en forma análoga al nivel isoelectrico, pero en vez de usar el valor de la señal se utiliza la distancia de cada punto al nivel isoelectrico.

**Cruces por Cero.** Se define como los instantes en que se producen cruces desde 0 hacia la zona comprendida entre el nivel isoelectrico y el nivel de ruido, ver figura 1.

**Duraciones.** Se define como la diferencia temporal entre el inicio y término de cada onda o segmento. La duración de cada onda se evalúa como el promedio de las duraciones medidas en cada derivación.

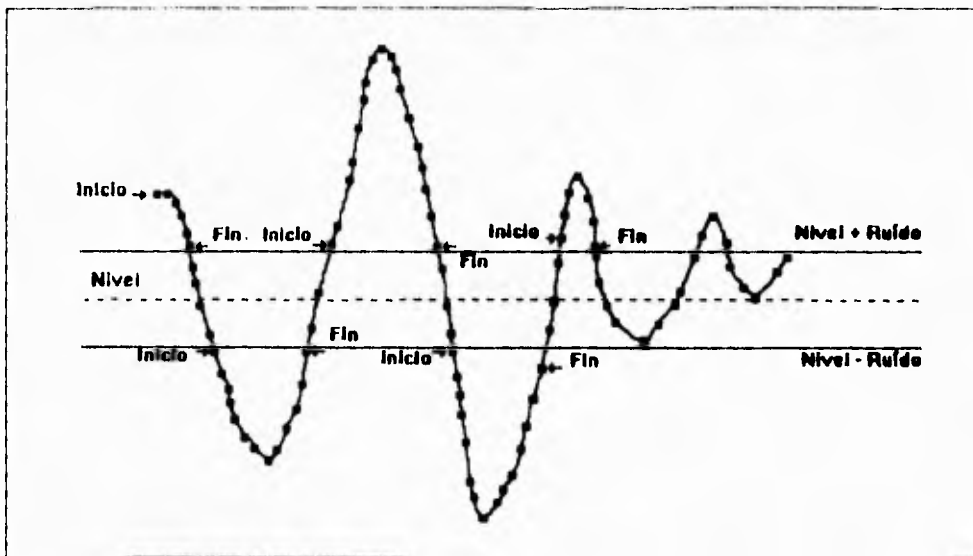


Figura 1

**Amplitudes.** Se definen como la distancia entre el nivel isoelectrico y el valor absoluto máximo de la onda. La amplitud de cada onda se evalúa como el promedio de las amplitudes medidas en cada derivación.

**Segmento ST.** Para calcular su elevación se analiza el trazado de los diez puntos ubicados cinco muestras después del final de cada complejo QRS, mediante mínimos cuadrados, y se mide la altura desde el nivel isoeléctrico hasta el centro de la recta.

**Derivación.** Consiste de dos polos conectados a un galvanómetro sensible que realiza el registro y es componente básico del electrocardiógrafo (ver figura 2). La línea imaginaria entre ambos polos de cada derivación se conoce como eje de la derivación. Se acostumbra usar doce derivaciones para efectuar el registro: seis en el plano frontal y seis en el plano horizontal.

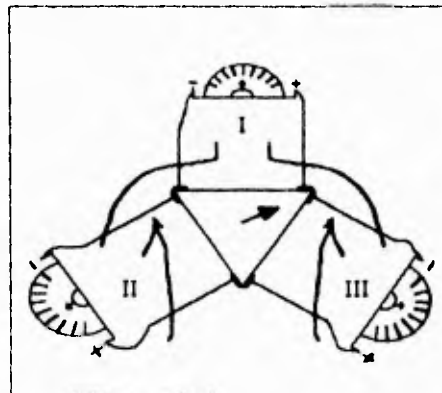


Figura 2

## 2.2 Electrofisiología celular

Una explicación sencilla de la electrofisiología celular proporcionará un conocimiento básico de los conceptos de vectores eléctricos. En reposo, el interior de la célula contiene una alta concentración de potasio y baja de sodio; lo contrario sucede en el exterior de la membrana celular (ver figura 2.1). Durante este periodo de reposo la membrana celular evita que los iones de sodio la atraviesen, pero es permeable a los movimientos de los iones de potasio. Debido a que los iones de potasio llevan cargas positivas al exterior de la célula y provocan en forma relativa, cargas más negativas en el interior, se establece una diferencia eléctrica (potencial) entre el interior y el exterior de la célula. Esta diferencia de potencial ---de alrededor de 90(mV)--- se conoce con el nombre de potencial en reposo de trasmembrana. La célula podría conceptualizarse como una línea doble de cargas positivas y negativas separadas por la membrana celular (ver figura 2.1). Cada lugar en el que se encuentra una carga positiva y otra negativa se llama dipolo. Se considera que la célula está polarizada (cargada) en estado de reposo, cuando los iones de sodio atraviesan la membrana. La despolarización ocurre cuando la membrana celular

recibe un estímulo eléctrico y se vuelve permeable a la entrada de iones de sodio, lo cual provoca un cambio rápido en la diferencia de potencial de la membrana celular de  $-90(\text{mV})$  a  $+20(\text{mV})$  (ver figura 2.2). La recuperación de las cargas eléctricas a los valores que tenía en su estado de reposo de  $-90(\text{mV})$  se conoce con el nombre de repolarización.

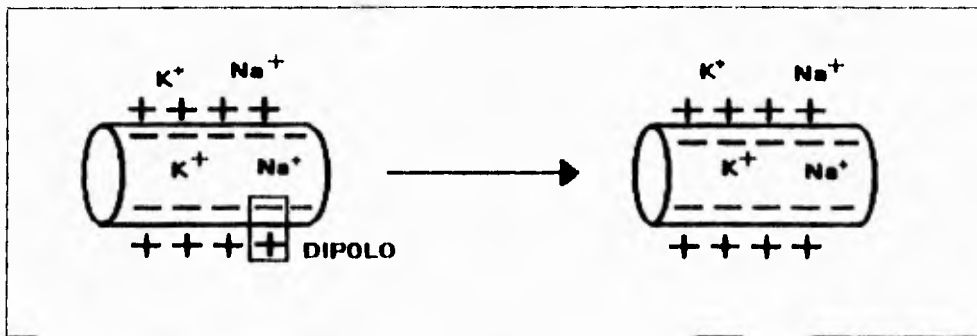


Figura 2.1

Debido a que el ion sodio ingresa rápido al interior de la célula, ocurren una serie de cambios eléctricos en los cuales la superficie celular se vuelve negativa desde el punto de vista eléctrico, mientras que el interior se vuelve positivo (ver figura 2.1). Esta actividad sigue una secuencia compleja de movimientos de otros iones, que incluyen potasio, calcio y cloro, los cuales provocan una serie de cambios eléctricos que al final conducen a la repolarización completa. La curva electrofisiológica de estos cambios se conoce como potencial de acción transmembrana (ver figura 2.2).

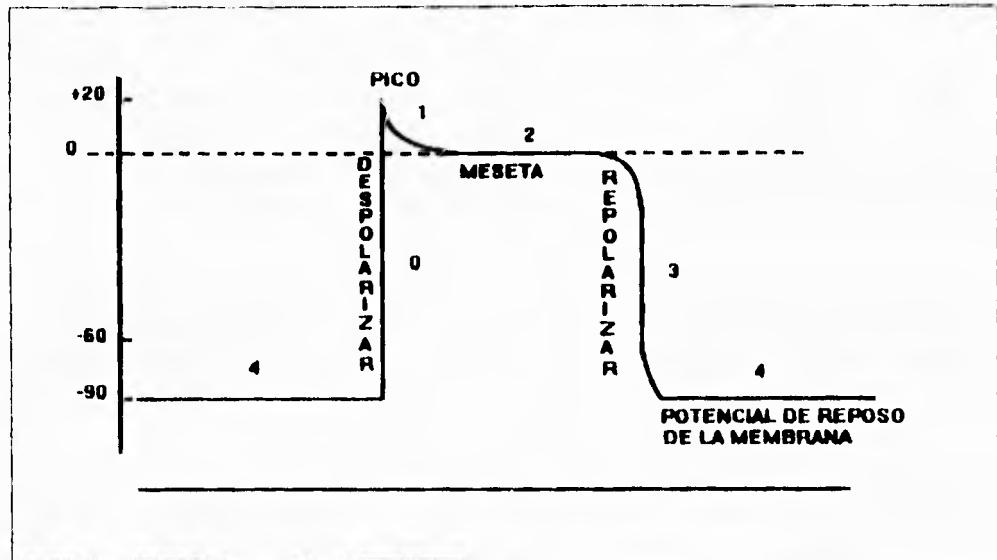


Figura 2.2

Pueden identificarse cinco fases:

**La fase 0** es la despolarización en la cual un estímulo permite la entrada rápida de iones de sodio con carga positiva al interior de la célula. El potencial transmembrana sube en forma rápida desde su nivel de reposo de  $-90(\text{mV})$  y en los niveles críticos de potencial umbral de aproximadamente  $-60(\text{mV})$ , ocurre una excitación abrupta y definitiva provocando que el potencial alcance niveles de  $+20(\text{mV})$ . Si no se alcanza un potencial umbral de  $-60(\text{mV})$  debido a que el estímulo es muy débil, el potencial de acción no se presenta.

**La fase 1**, mediada en forma primaria por el movimiento de iones de potasio y cloro es una fase temprana y breve de la repolarización desde  $+20(\text{mV})$  hasta  $0(\text{mV})$ .



La fase 2 se conoce como meseta debido a que la diferencia de potencial queda nivelada o se presenta una repolarización muy lenta. Esta fase es provocada en su mayor parte por la entrada de calcio que trata de contrarrestar la salida de potasio. Esta prolongada y bien desarrollada fase 2 provoca el periodo refractario largo del tejido miocárdico, el cual es básicamente diferente de otros tejidos musculares o nerviosos.

Durante el periodo de repolarización rápido de la fase 3, el potasio continúa saliendo. Cuando se ha alcanzado el potencial de reposo de la membrana de  $-90\text{(mV)}$ , la célula de nuevo se encuentra totalmente polarizada y a esto se le denomina estado de reposo de la fase 4.

En términos del electrocardiograma, el cual representa la suma de los potenciales de acción de todas las células, el complejo QRS coincide con la fase 0, el segmento ST con la fase 2, la onda T con la fase 3 y la línea de base isoeléctrica con la fase 4.

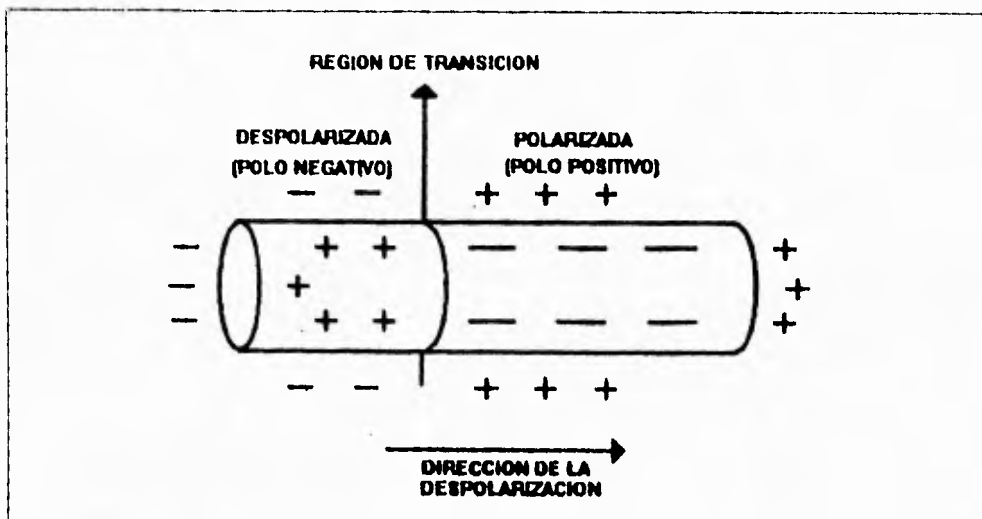


Figura 23

Las células del marcapaso del nodo sinusal difieren de otras células miocárdicas, ya que tienen un potencial de reposo de la membrana más bajo (menos negativo), una fase 4 ligeramente ascendente que puede en forma espontánea alcanzar el potencial umbral y desencadenar una despolarización, una fase 0 más lenta y una curva más redondeada de repolarización lenta que no permite separar en fases 1, 2 o 3.

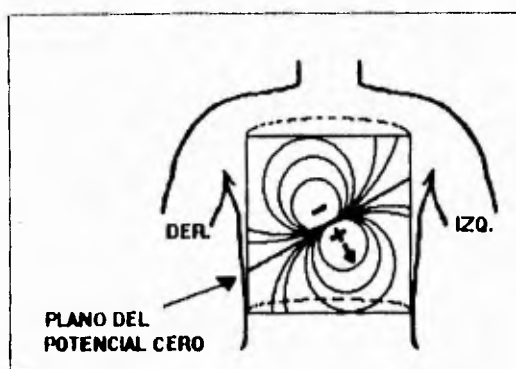


Figura 2.4

Una vez que se alcanza el umbral, el impulso se propaga de un dipolo al siguiente mientras se extiende una onda de despolarización a través del eje de la célula hacia las células adyacentes. Al mismo tiempo, la célula se encuentra dividida en un polo positivo (que aún no ha sido activado) y una región negativa (despolarizada) con un campo eléctrico intenso que se extiende alrededor de cada uno de los polos

positivos y negativos en todas direcciones (ver figura 2.3). La región de transición que separa estos dos campos posee un potencial cero que no es positivo ni negativo y es siempre perpendicular a la dirección de la corriente eléctrica.

Einthoven consideró que el corazón actúa como un dipolo simple desde el cual se origina un campo eléctrico. Se discute si este modelo simple es apropiado en todas las condiciones; sin embargo, la teoría del dipolo simple no aporta un concepto de utilidad al propósito de la discusión. Se considera que el campo del dipolo simple sale del centro del tórax como un campo tridimensional con una serie de capas de fuerzas equipotenciales positivas y negativas alrededor de cada polo (ver figura 2.4). Como estas fuerzas se extienden cada vez más lejos de su origen, su potencial eléctrico disminuye. Como se mencionó antes, un plano de transición de potencial cero es siempre perpendicular a la dirección de esta fuerza eléctrica. Estos campos eléctricos positivos y negativos y el plano de potencial cero se extienden hacia afuera a través de los tejidos del tórax que actúan como medios de conducción hasta alcanzar la superficie del cuerpo donde pueden ser registrados por medio de las derivaciones del electrocardiograma.

Durante cada ciclo cardiaco se generan muchas fuerzas eléctricas mientras la electricidad se dispersa a través del corazón. Debido a que esas fuerzas cambian de magnitud y dirección, la orientación de los campos positivos y negativos y el plano de potencial cero también varían. Desde su posición fija cada derivación electrocardiográfica registra estos movimientos y suministra una gráfica del curso de la electricidad con relación a la derivación en cuestión.

### 2.3 Secuencia de la despolarización y la repolarización

El ciclo cardiaco se inicia en forma normal con la despolarización eléctrica espontánea de las células marcapaso del nodo sinusal. Estas células se localizan en la unión de la vena cava superior con la aurícula derecha y la orejuela de la aurícula derecha (ver figura 2.5a). La electricidad es diseminada desde esta área hacia el nodo auriculoventricular (AV) al parecer por vías de conducción preferencial en el septum interauricular y por los haces de Bachmann que unen las dos aurículas. A medida que la onda de despolarización se dirige hacia el nodo AV, se produce una despolarización concéntrica del miocardio auricular. La onda P del electrocardiograma es la representación electrocardiográfica del fenómeno de esta despolarización auricular (ver figura 2.5b). El nodo AV que se localiza en la porción inferior de la aurícula derecha, entre el seno coronario y la hojuela tricuspídes septal, retarda el impulso eléctrico antes que dicho impulso viaje a través del haz de His hacia el sistema de conducción ventricular (ver figura 2.5a). El intervalo PR mide el tiempo que el impulso tarda en viajar desde el nodo

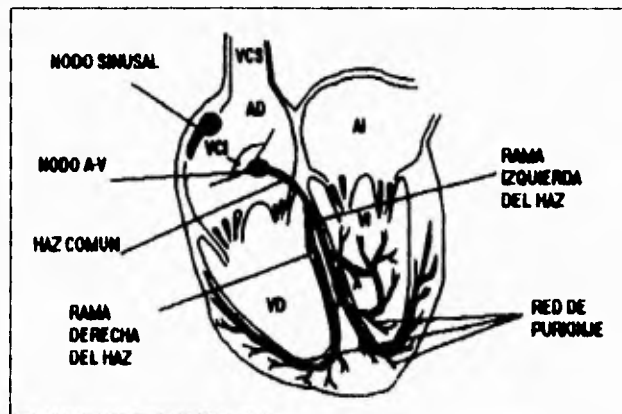


Figura 2.5

sinusal. El nodo AV que se localiza en la porción inferior de la aurícula derecha, entre el seno coronario y la hojuela tricuspídes septal, retarda el impulso eléctrico antes que dicho impulso viaje a través del haz de His hacia el sistema de conducción ventricular (ver figura 2.5a). El intervalo PR mide el tiempo que el impulso tarda en viajar desde el nodo

sinusal hasta el principio de las ramas del haz. Este período incluye el retraso que sufre el nodo AV.

Las dos ramas del haz se originan en el vértice del tabique muscular ventricular (ver figura 2.5a). La rama derecha es delgada y cursa en el lado derecho del tabique ventricular hasta el músculo papilar anterior del ventrículo derecho, donde se divide en una red de Purkinje de fibras conductoras especializadas. La rama izquierda se divide ampliamente en el lado izquierdo del tabique ventricular en un mínimo de dos y posiblemente tres divisiones mayores (fascículos). A nivel de los músculos papilares del ventrículo izquierdo, la rama izquierda se ramifica en una compleja red subendocárdica de fibras de Purkinje. El tercio medio del lado izquierdo del tabique ventricular es el área que se excita primero. Entonces, la despolarización se disemina en el endocardio al epicardio de ambos ventrículos, subiendo desde la región apical hacia la basal de los ventrículos. El complejo QRS se produce por la propagación de la electricidad a través del miocardio ventricular (ver figura 2.5b). Después de la despolarización ventricular, existe un breve período de mínima actividad eléctrica que corresponde al segmento ST del electrocardiograma. Este período termina con la repolarización ventricular que procede del epicardio al endocardio, desde la punta hacia la base, e inscribe una onda T en el electrocardiograma. La onda U, una deflexión de baja amplitud y de significado incierto, sigue a la onda T (ver figura 2.5b).

#### 2.4 Terminología de las deflexiones electrocardiográficas

**Onda P.** Primera deflexión del ciclo cardíaco y refleja la activación de la aurícula por un impulso auricular en el nodo sinusal o ectópico. De acuerdo a la derivación, la onda P puede ser positiva (arriba de la línea base), negativa (abajo de la línea base), o difásica.

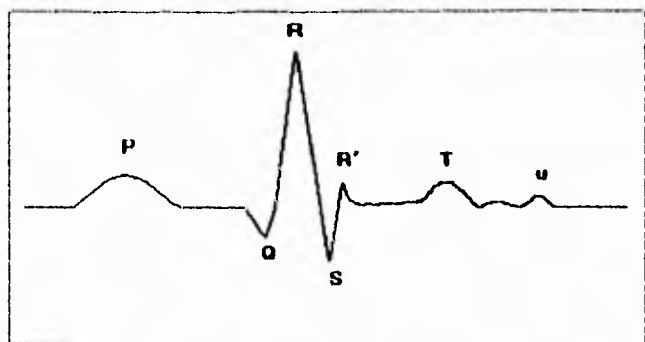


Figura 2.5b

**Complejo QRS.** Representa la despolarización ventricular. De común acuerdo, el complejo se clasifica según las diferentes ondas características a continuación:

**Onda Q.** Es una deflexión negativa al inicio del complejo QRS.

**Onda R.** Es la primera deflexión positiva ascendente, exista o nó una onda Q. Si existe una segunda deflexión positiva recibe el nombre de onda R'.

**Onda S.** Es la primera deflexión descendente bajo la línea isoeleétrica después de una onda S recibe el nombre S'.

Un complejo QRS monofásico consiste de una onda R simple (positiva) o un complejo QS (negativo). Se dice que un complejo es bi- o difásico cuando existe una deflexión que se encuentra en forma parcial por encima y por debajo de la línea isoeleétrica. Un complejo trifásico es aquel en el cual el complejo QRS tiene una configuración RSR'. Las letras mayúsculas (Q, R y S) se utilizan cuando las deflexiones individuales son iguales o mayores de 5(mm). Las letras minúsculas (q, r y s) se aplican a las deflexiones individuales menores de 5(mm).

En el siguiente esquema se clasifica el complejo QRS en ocho familias que representan las variaciones morfológicas más significativas desde el punto de vista de la detección de arritmias. Esta división, que se muestra en la figura 3, considera tanto la amplitud de las ondas Q, R y S, como la diferencia de amplitud entre los puntos Q y S.

**PAGINACION VARIA**

**COMPLETA LA INFORMACION**

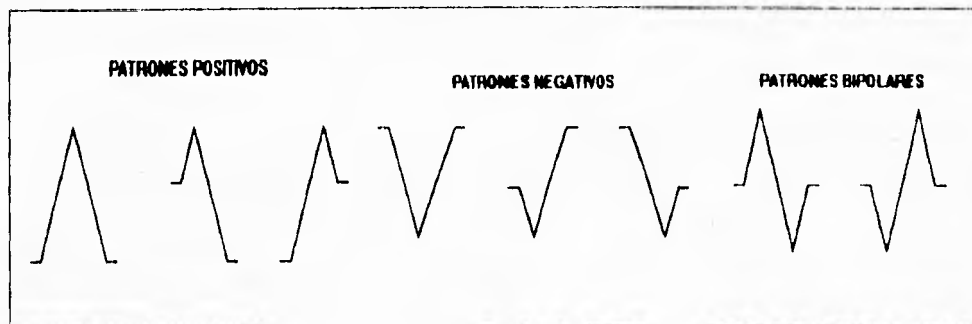


Figura 3

Las primeras cuatro familias se definen en base a puntos Q y S de igual amplitud. Las familias restantes representan complejos con dos ondas principales, en donde existe una diferencia de amplitud entre los puntos Q y S.

Se pueden reconocer dos etapas de segmentación: en la primera, se recorren los puntos leídos que se encuentran entre los puntos Q y S, almacenando aquellos donde la pendiente presente un cambio de signo. Luego se representa el complejo QRS como un conjunto de rectas que unen los puntos de cambio de signo almacenados.

En la segunda etapa, se eliminan los segmentos que no sean de interés, de tal forma que quede totalmente descrita la morfología, sin considerar detalles innecesarios. Esta etapa lleva implícita la selección de primitivas que deben construirse a partir de los segmentos encontrados.

Para seleccionar las primitivas, el complejo QRS se divide en tres zonas como se muestra en la figura 4.

La zona II representa siempre al segmento de mayor amplitud ubicado entre el máximo y el mínimo del QRS. Un QRS tendrá una zona II, pero no necesariamente deberá poseer ambas zonas I y III. La zona II se define como positiva o negativa, y las zonas I y III en base a segmentos que poseen o no un mínimo relativo. Esta división es válida para todos

los tipos de QRS a encontrar en la derivación II del ECG, manteniendo las características morfológicas de interés.

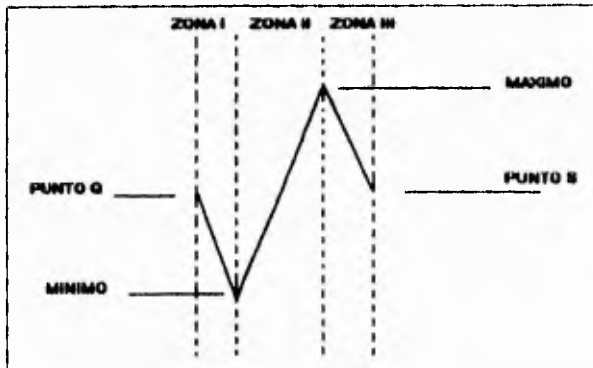


Figura 4

De la segmentación se obtiene una secuencia de rectas caracterizadas por su ubicación temporal, pendiente y amplitud. Como cada recta debe ser clasificada en primitivas y el número de primitivas es finito, ello lleva a una cuantización de amplitudes en niveles.

La amplitud del segmento principal en valor absoluto,  $A_{mm}$  es dividida en tres partes, utilizándose una de éstas como unidad básica de clasificación. La amplitud  $A_{s1}$  de cada segmento  $S1$  se calcula como la diferencia entre el punto de fin y comienzo del segmento, y se clasifica cada segmento  $S1$  según el siguiente segmento (ver figura 5).

**Onda T.** Es una deflexión provocada por la repolarización ventricular, sigue al complejo QRS. Puede ser positiva, negativa o difásica, de acuerdo a la derivación.

**Onda U.** Es una deflexión de baja amplitud que puede seguir a la onda T.

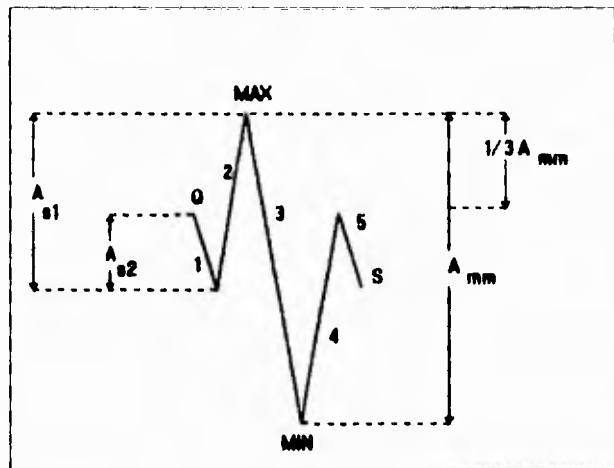


Figura 5



### **2.5 Mediciones de las deflexiones electrocardiográficas**

Un electrocardiograma estándar está dividido por tenues líneas horizontales y verticales, separadas entre sí 1 [mm]. Tanto en sentido horizontal como vertical aparecen líneas más oscuras cada cinco divisiones. El voltaje se mide de acuerdo al desplazamiento vertical, en milímetros [mm] hacia arriba y hacia abajo de la línea isoelectrica (línea media en sentido vertical del cuadrículado). Todos los electrocardiógrafos son de fabricación estándar de tal manera que 1 milivolt [mV] de fuerza eléctrica provoca un desplazamiento vertical de la aguja de registro de 10 [mm]. Por lo tanto, cada [mm] vertical es equivalente a 0.1 [mV].

El tiempo se mide en el eje horizontal. En un registro estándar la velocidad es de 25 [mm/segundos], la duración entre dos líneas verticales tenues es de 0.04 segundos y entre dos líneas oscuras es de  $5 \times 0.04$  segundos es decir 0.2 segundos. Un minuto de registro abarca 300 divisiones de mayor tamaño ( 60 segundos / 0.2 segundos ).

En el siguiente diagrama de un electrocardiograma se ilustra el nombre, las deflexiones y los intervalos electrocardiográficos para un ciclo cardíaco, utilizando el formato estándar de rejilla para medir la amplitud de dichas deflexiones y la duración de los intervalos. Considerando que la velocidad del rollo es de 25 [mm/segundos].

#### **Determinación de los intervalos electrocardiográficos.**

Los intervalos electrocardiográficos pueden medirse en cualquier derivación, ya sea frontal o precordial, que presente el intervalo más largo y en cual puedan delinearse con seguridad tanto el principio como el final de dicho intervalo. La elección de la derivación puede ser diferente para cada intervalo.

Los siguientes intervalos deben medirse en forma rutinaria (ver figura 6):

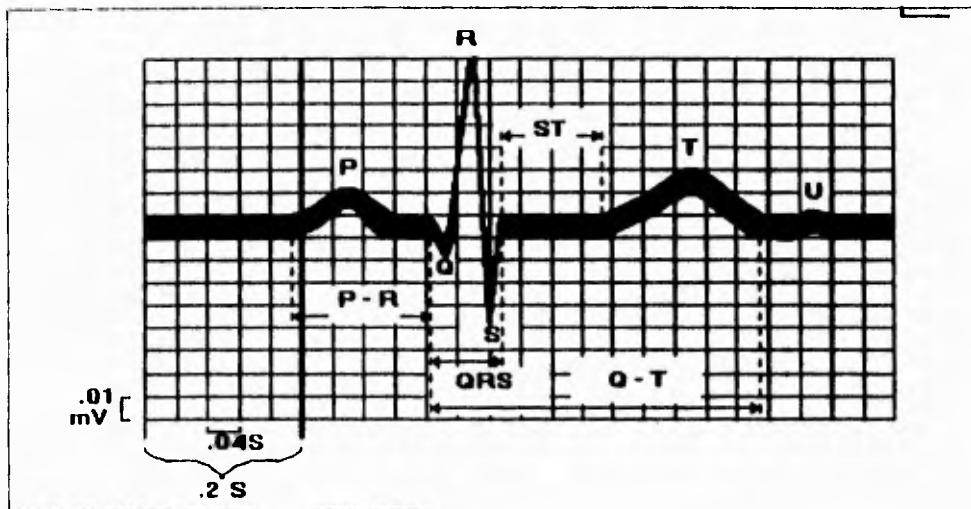


Figura 6

**Intervalo PR.** Se mide desde el inicio de la onda P hasta el inicio del complejo QRS (onda Q o R) (ver figura 6). Las oscilaciones normales para un adulto pueden variar desde 0.12 hasta 0.20 segundos. En forma usual el intervalo es más corto en los niños y cuando la frecuencia cardiaca es más rápida (tabla 1). Un porcentaje muy pequeño de personas tienen un intervalo PR superior a 0.20 segundos.

**Intervalo QRS.** La duración de este intervalo se mide desde el inicio (onda Q o R) hasta la terminación del complejo QRS (ver figura 6). Para medir el QRS se selecciona la derivación donde este complejo tenga una mayor amplitud. EL intervalo QRS normal en los adultos es de 0.06 a 0.10 segundos. Los niños tienen un QRS con una duración más corta.

**Intervalo QT.** Este intervalo se mide desde el inicio del intervalo QRS hasta la terminación de la onda T (ver figura 6). A menudo se selecciona una derivación precordial media, en la cual la onda T se encuentra bien definida. Este intervalo, representa la despolarización y la repolarización ventricular total, y varía con la frecuencia cardiaca, la

edad, el sexo, y las influencias autonómicas. En la tabla 2, se presentan los valores normales corregidos para la frecuencia cardiaca y el sexo. El intervalo QT puede corregirse dividiendo la frecuencia cardiaca entre la raíz cuadrada del intervalo RR. Este tipo de corrección se conoce como QTc; el límite superior de lo normal es de 0.43 segundos.

**Eje eléctrico.** Por el método mencionado con anterioridad, se determinan y colocan los vectores medios de QRS y T en el sistema de referencia hexaxial. El eje medio normal de QRS para un adulto oscila entre  $-30$  y  $-90$ , (ver figura 6.1). Un eje eléctrico desviado a la izquierda indica que el vector QRS medio se encuentra entre  $-30$  y  $-90$ , mientras que un eje eléctrico desviado a la derecha se encuentra entre  $+90$  y  $-150$ . El vector medio normal de T es paralelo en forma relativa al vector de QRS (ver figura 6.2)

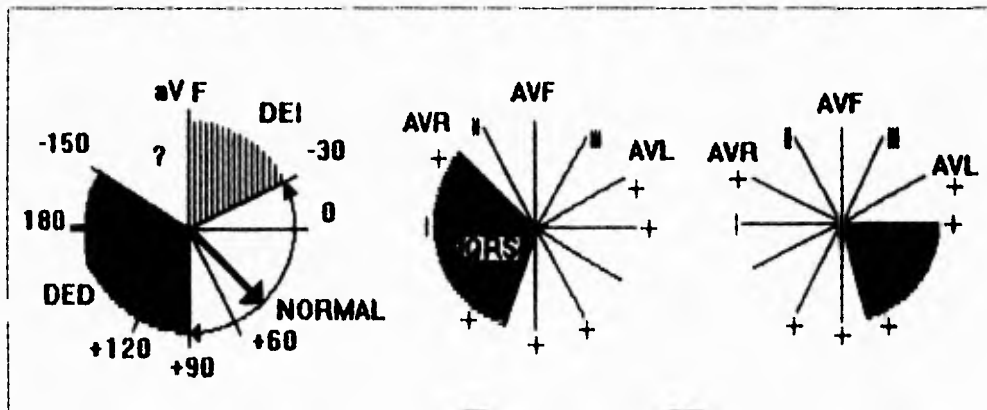


Figura 6.2

Un electrocardiograma es una medición muy compleja en la cual pueden influir la interacción de factores biológicos o técnicos, lo mismo que diferentes enfermedades. Nunca se ha podido establecer los valores normales verdaderos para cada grupo etario, sexo, peso corporal, tamaño del tórax y otras muchas variables que pueden alterar de manera significativa el resultado de un electrocardiograma. Por lo tanto, diferenciar lo normal de lo anormal resulta muy difícil.

Dentro de las consideraciones biológicas no se puede profundizar porque se carece de la información que por ejemplo tiene un médico o el personal autorizado para la interpretación de un electrocardiograma a nivel de diagnóstico médico, esto significa que se tiene un sin número de enfermedades y de síntomas que pueden variar el electrocardiograma. Con lo que respecta a las consideraciones técnicas, éstas incluyen: la posición de las derivadas precordiales con relación a la posición anatómica del corazón, amplitud del esternón, fidelidad del electrocardiógrafo, sustancias que se utilizan para la conducción por debajo de los electrodos, errores de las mediciones corporales, cambios en la línea base y los cambios normales y cotidianos.

Sin embargo, lo que interesa en este análisis es cómo se van a presentar esas fuerzas eléctricas que determinan las diferentes deflexiones electrocardiográficas y los diferentes intervalos según sea el caso. Para esto podemos tomar una tabla que muestre los valores normales según las características del paciente.

FRECUENCIA	menor de 70	71-90	91-110	111-130	mayor de 130
Adultos corpulentos	0.21	0.20	0.19	0.18	0.17
Adultos delgados	0.20	0.19	0.18	0.17	0.16
Niños de 14 a 17 años	0.19	0.18	0.17	0.16	0.15
Niños de 7 a 13 años	0.18	0.17	0.16	0.15	0.14
Niños de 1.5 a 6 años	0.17	0.165	0.155	0.145	0.135
Niños de 0 a 1.5 años	0.16	0.15	0.145	0.135	0.125

Tabla 1

## 2.6 Determinación de la frecuencia cardiaca (FC)

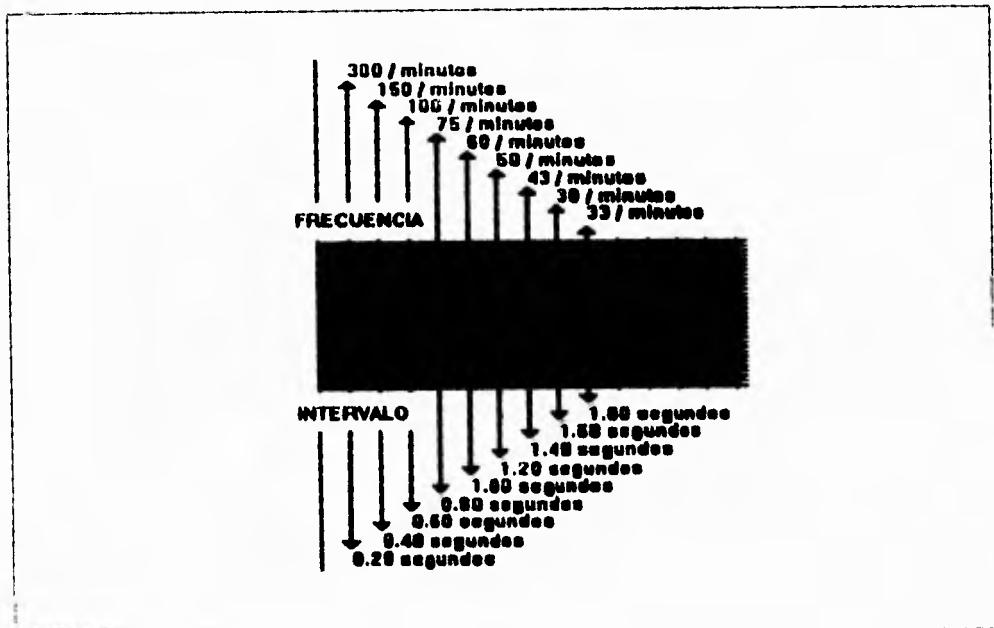
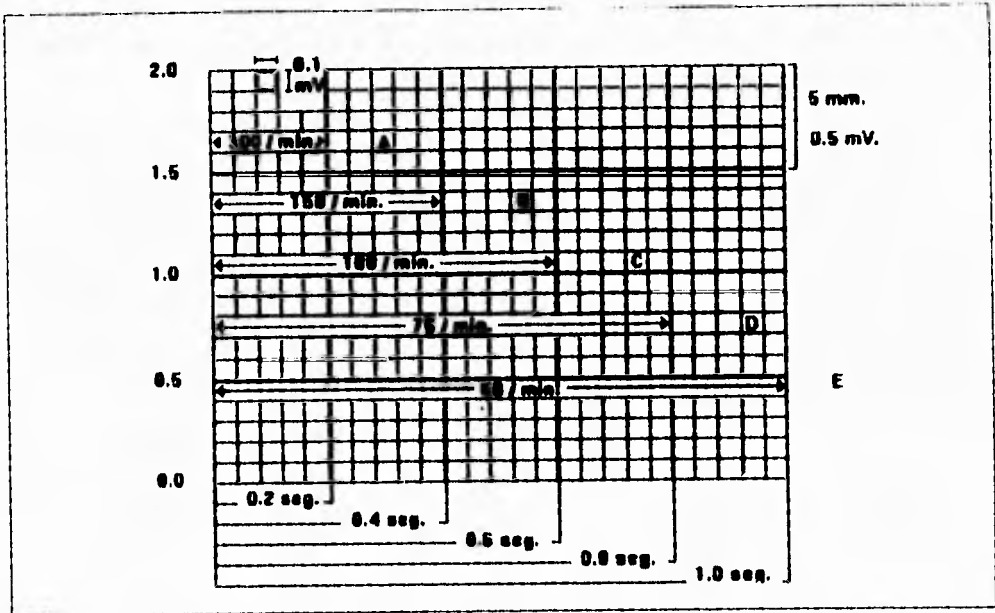
En un electrocardiograma también se puede determinar la frecuencia cardiaca en forma aproximada. El método utilizado para dicha medición se hace contando el número de divisiones mayores (5 [mm]) entre dos ciclos cardiacos consecutivos y dividir esta cifra entre 300. Con este método simple una frecuencia cardiaca de 300 por minuto será aquella en la cual un ciclo cardiaco esté separado de otro por una división, en la de 150 por minuto, los dos ciclos cardiacos estarán separados por dos divisiones, y así sucesivamente. Si se dá que entre los ciclos cardiacos existe una fracción de división, se realiza un ajuste apropiado: 3.5

divisiones se encuentran en un punto intermedio entre 100 y 75, por tanto, la frecuencia cardiaca es de 88 por minuto.

En el siguiente diagrama se muestra la relación entre los intervalos y la frecuencia cardiaca. El tiempo de calibración estándar es de 1 [mm] = 0.04 segundos y 5 [mm] = 0.02 segundos. El voltaje estándar es de 0.1 [mV/mm]. Un evento que se repite cada 5 [mm] x 0.2 segundos, (A) en el eje del tiempo ocurre 300 veces por minuto. Un evento que se repite cada 10 [mm] x 0.4 segundos, (B) ocurre 150 veces por minuto. Los incisos C, D y E indican el evento que se repite cada 0.6, 0.8 y 1.0 segundos ocurre con frecuencia de 100, 75, y 60 veces por minuto respectivamente.

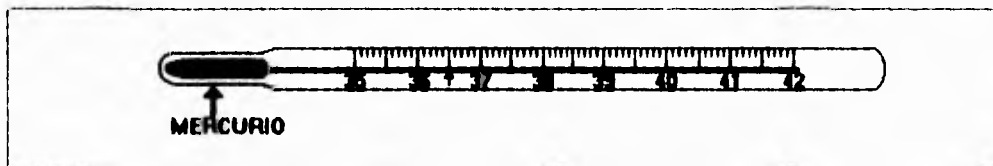
Longitud del ciclo [seg]	Frecuencia cardiaca [min]	Hombres y niños [seg]	Mujeres [seg]	Límites superiores de lo normal	
				Hombres y niños [seg]	Mujeres [seg]
1.50	40.0	0.499	0.461	0.491	0.503
1.40	43.0	0.438	0.450	0.479	0.491
1.30	46.0	0.426	0.438	0.466	0.478
1.25	48.0	0.420	0.432	0.460	0.471
1.20	50.0	0.414	0.425	0.453	0.464
1.15	52.0	0.418	0.445	0.445	0.456
1.05	54.5	0.400	0.411	0.438	0.449
1.00	57.0	0.393	0.404	0.430	0.441
1.00	60.0	0.386	0.396	0.422	0.432
0.95	63.0	0.378	0.388	0.413	0.423
0.90	66.5	0.370	0.380	0.404	0.414
0.85	70.5	0.361	0.371	0.395	0.405
0.80	75.0	0.352	0.362	0.384	0.394
0.75	80.0	0.342	0.352	0.374	0.384
0.70	86.0	0.332	0.341	0.363	0.372
0.65	92.0	0.321	0.330	0.351	0.360
0.60	100.0	0.310	0.318	0.338	0.347
0.55	109.0	0.297	0.305	0.325	0.333
0.50	120.0	0.283	0.291	0.310	0.317
0.45	133.0	0.268	0.276	0.294	0.301
0.40	150.0	0.252	0.258	0.275	0.282
0.35	172.0	0.234	0.240	0.255	0.262

Tabla 2



## 2.7 Determinación de temperatura [TEMP]

La temperatura interna del organismo es de aproximadamente 37 grados centígrados (la externa es aproximadamente medio grado menor) con oscilaciones más o menos importantes según el individuo. Unas décimas por encima o debajo de 37 grados es normal en algunas personas; una diferencia mayor indica algún trastorno. La temperatura del cuerpo no es constante a lo largo del día. Hacia las cuatro de la tarde suele aumentar aproximadamente medio grado respecto al valor basal, y disminuye durante la noche; aumenta también durante la digestión y en el ejercicio físico. Por esta razón, es conveniente tomar la temperatura al menos tras 20 minutos de reposo en la cama. La temperatura ambiente influye también sobre el organismo, sobre todo en los niños, y muy particularmente en los recién nacidos (en los que la temperatura normal puede variar a lo largo del día incluso dos grados, según el momento). Si un recién nacido se encuentra en un ambiente cálido o está muy tapado, la elevación de su temperatura puede ser importante como para provocarle un estado de deshidratación. El adulto, por lo contrario, se adapta mejor a las variaciones externas de la temperatura, sobre todo cuando hace calor, gracias a la pérdida de agua por medio de la transpiración. Durante la ovulación la mujer presenta un aumento de temperatura de aproximadamente medio grado que la tomada por la mañana antes de levantarse. En el caso de que una mujer esté embarazada, su temperatura continuará experimentando ese aumento, siempre y cuando no exista enfermedad, por supuesto.



## 2.8 Determinación de presión o tensión arterial [PA]

Son los sistemas de presión quienes aportan la mayor parte de la información clínica que se obtiene en la unidad de cuidado crítico. Las presiones cardiovasculares de interés pueden dividirse en los del sistema venoso a baja presión (-5 a 60 torr) y el sistema arterial a

alta presión (0 a 300 torr). La fuerza motora proviene de una bomba de cuatro cámaras, que es el corazón. Otras fuerzas son la presión intratorácica negativa durante la inspiración y la contracción de músculos de las extremidades que comprimen los vasos periféricos.

Las cifras normales de la presión o tensión arterial varían con el sexo (algo superior en el masculino al principio; a partir de los 45 años se invierten los términos), raza (menor tensión en los orientales), constitución, herencia ligada al sexo y la edad.

Las cifras tensionales para cada sujeto "normal o sano" no es mayor de 160/90 [mm] de mercurio hasta los 60 años, a partir de esta edad pueden ser mayores sin significado patológico.

EDAD en años	HOMBRES		MUJERES	
	Sistólica	Diastólica	Sistólica	Diastólica
Prematuros	42	21	42	21
1 a 2 Kg.	50	28	50	28
2 a 3 Kg.	60	37	60	37
1 mes	80	46	80	46
6 meses	89	60	89	60
1 - 5	75 - 90	50 - 60	75 - 90	50 - 60
7 - 8	85 - 95	50 - 60	85 - 95	50 - 60
10 - 12	95 - 105	50 - 70	95 - 105	50 - 70
14 - 15	97 - 120	60 - 72	95 - 115	57 - 72
17 - 18	106 - 122	60 - 75	93 - 110	57 - 73
20 - 24	108 - 132	65 - 77	100 - 120	55 - 70
25 - 29	110 - 132	65 - 80	100 - 120	55 - 75
30 - 34	110 - 132	65 - 80	100 - 120	60 - 75
35 - 39	110 - 132	65 - 80	105 - 125	60 - 80
40 - 44	110 - 140	65 - 85	110 - 130	60 - 80
45 - 49	110 - 140	65 - 90	110 - 140	65 - 85
50 - 59	115 - 142	65 - 90	110 - 145	65 - 90
60 - 69	115 - 145	65 - 90	110 - 145	65 - 90
70 - 79	120 - 160	65 - 90	115 - 165	70 - 90
80	120 - 155	65 - 90	115 - 169	70 - 90

Pero para los requerimientos del proyecto, el enfoque va dirigido a lo que se llama hipotensión arterial, puesto que es la que registra los valores más bajos de la tensión



arterial. En la hipotensión arterial, las tensiones máxima y diastólica se encuentran por debajo de los límites mínimos, estimados en 100/60 [mm] de Hg (según Baruch) y de 90/50 [mm] de Hg (según Hamilton). La tabla anterior muestra los valores medios de la tensión arterial normal.

### 2.9 Determinación de frecuencia respiratoria (FR)

La frecuencia respiratoria (número de actos respiratorios por minuto) varía por múltiples causas, entre las que se cuentan esfuerzos físicos, excitaciones psíquicas, trabajo digestivo, tono vegetativo del sujeto y sobre todo la edad.

Según Quetelet, en condiciones basales, la frecuencia respiratoria es:

EDAD	FRECUENCIA RESPIRATORIA respiraciones por minuto
al nacer	44
1 mes	24 - 35
6 meses	24 - 35
1 año	20 - 30
5 años	26
15 - 20 años	20
20 - 25 años	18
25 - 30 años	16
arriba de 40 años	18

### 2.10 Análisis biomédico

Como conclusión a lo anterior, los datos que se requieren para hacer un monitoreo adecuado deben ser reales, verídicos y lo más exacto posible debido a la importancia de éste en el campo de la medicina.

---

Con el análisis anterior se pueden fijar los parámetros y los límites necesarios para cada signo vital. Así como los tiempos requeridos por cada sensor para que la adquisición de los datos no presente pérdidas y se omita algún dato importante.

A continuación se presentan los tiempos estimados para cada sensor según los casos críticos, que son aquellos que requieren de un tiempo de adquisición determinado.

### **2.11 Electrocardiógrafo**

Basándose en la información anterior, se realiza que para predeterminar un patrón de cómo va a ser un ciclo cardíaco, es prácticamente imposible debido a las causas que lo definen. Pero sí es posible establecer los tiempos críticos de dicho ciclo cardíaco. Esto permite saber el tiempo necesario para estar sensando el ciclo cardíaco completo y que no haya pérdida de ningún tipo de información. El caso crítico máximo (ciclo cardíaco más largo o más lento) según las estadísticas es de 1.5 segundos duración de ciclo cardíaco completo, este caso se presentará en los casos donde los pacientes tengan problemas de estados de coma, arritmias, etcétera.; el caso crítico mínimo (ciclo cardíaco más rápido o más corto) según las estadísticas es de 0.35 segundos duración del ciclo cardíaco completo, este caso se presentará en pacientes que tengan problemas de arritmias, hipertensos, etcétera. Por supuesto que los datos estadísticos obtenidos se generalizaron sin importar el sexo, edad, complexión, etcétera, ni casos clínicos específicos.

### **2.12 Frecuencia Cardíaca**

Al igual que el electrocardiograma, este signo es difícil de predeterminar con respecto a la patología que lo produce, pero sí se pueden obtener los tiempos críticos máximos y mínimos de muestreo. También por datos estadísticos se obtuvo que el caso crítico máximo es de 33 latidos por minuto siendo equivalentes a 1.8 segundos por latido, en este caso tendremos el registro de un número menor de latidos en un cierto periodo; el caso crítico mínimo es de 300 latidos por minuto siendo equivalente a 0.2 segundo por cada latido, en este caso tenemos un número mayor de latidos en cierto periodo.

### **2.13 Temperatura**

La temperatura es el parámetro vigilado más sencillo de medir y consagrado. La interpretación de sus tendencias indica una infección incipiente, pérdida del control humoral o metabólico, tipo de infección, órgano afectado y la eficiencia del tratamiento.

Para la temperatura, hay transductores de mercurio en cristal, termistores y diodos de polarización inversa. En ocasiones se emplean termopares. La vigilancia del paciente anestesiado o hipotérmico suele hacerse con una sonda rectal o esofágica acoplada al medidor digital o analógico, que no señala tendencias. En personas que no están en fase aguda, la medición se puede hacer fácilmente con termómetros sublinguales o rectales de columna de mercurio. En muchas instituciones se usan sondas digitales manuales, pues son más rápidas y menos frágiles. Para la valoración precisa del sujeto en estado crítico es indispensable medir la temperatura central por medio de un termistor en el extremo distal de un catéter dirigido por balón (de Swan-Ganz). Dada la enorme diferencia entre las temperaturas de la piel y de las vísceras internas debido a la constricción o dilatación de los vasos periféricos, las sondas autoadherentes de contacto superficial tienen una utilidad limitada en la unidad de cuidado intensivo, pero se usan en investigación vascular.

Para este caso la temperatura variará en un rango relativamente pequeño, por lo tanto si se está sensando durante tiempos muy largos se podrían presentar oscilaciones en los datos que se obtengan, por eso con sólo tomar algunas muestras en cierto tiempo y luego promediarlas será suficiente.

### **2.14 Presión arterial**

Además del método usual de auscultación (el medidor de columna de mercurio o baumanómetro), se puede medir con gran exactitud colocando un catéter a permanencia en la arteria unido a tubos conectores, un transductor, amplificador y filtro. El mismo método puede utilizarse para medir presiones venosas colocando el catéter en el sistema venoso central. El equipo debe medir presiones de -5 a 300 torr, con una frecuencia de 0 a 5 Hz y

armónicos de 100 Hz. El transductor puede ser de cerámica piezoeléctrica, de manómetro o del tipo de resistencia.

Otro equipo es el vernier de equilibrio para que el nivel de la presión atmosférica sea de cero, con el orificio abierto y el transductor en medio del tórax del enfermo. Asimismo, a veces se cuenta con la opción de posiciones de ganancia, rango, filtro y límites de alarma.

La salida de datos se verá en una pantalla o monitor y puede mostrar las ondas de presión y las lecturas digitales o analógicas.

Por lo tanto el muestreo en condiciones no internas es decir que el medidor no se encuentre dentro del paciente, podría mostrar oscilaciones en los datos que se obtengan si el tiempo de sensado es muy prolongado, por lo tanto al igual que la temperatura, con sólo tomar algunas muestras durante cierto tiempo y promediarlas será suficiente.

### **2.15 Frecuencia respiratoria**

La monitorización de los parámetros respiratorios en la unidad de cuidados intensivos y el quirófano ha sido siempre importante, pero hasta años recientes se había limitado a la simple vigilancia mecánica o electrónica de la frecuencia respiratoria, de crisis de apnea o de presiones de las vías respiratorias.

Existen métodos intracorporales para dicha medición, pero aunque el método es exacto y confiable en la obtención directa de muestras, genera problemas y riesgos técnicos tales como los de la colocación del catéter arterial, infección, embolia y trombosis del vaso sanguíneo. Otros métodos son a través de dispositivos automáticos como el monitor trascutáneo de oxígeno (P O ) y los monitores de gas aspirado del tipo infrarrojo y espectrómetros que han eliminado muchos de los problemas de muestreo y la monitorización intracorporal. Dichos monitores pueden generar valores reales, pero en ocasiones sólo indica una tendencia cambiante. También se usa la espectrometría de masas para analizar la concentración de los gases inspirados y espirados, cada gas tiene su propio espectro de masa y se analiza en base a él, cada paciente puede ser controlado

contínuamente o en intervalos. Las velocidades de flujo dentro de los tubos de muestreo son del orden de 10 ml/minuto. La extracción de muestras en cada enfermo debe hacerse durante 15 a 30 segundos para lograr un perfil respiratorio adecuado, este caso se apoya en la información de datos estadísticos obtenidos de diferentes pacientes de distinta edad, sexo, complexión, etcétera.

Los tiempos críticos: máximo (es donde se presentan menos respiraciones por minuto) es de 18 respiraciones por minuto siendo equivalente a 3.33 segundos por respiración; mínimo ( es donde se presentan más respiraciones por minuto) es de 44 respiraciones por minuto siendo equivalente a segundos por respiración.

PARAMETRO	Min	Max
	[segundos]	
Frec. Cardiaca	0.20	1.80
Electrocardio	0.35	1.50
Temperatura	0.10	0.10
Presión Arterial	0.10	0.10
Frec. Respiratoria	0.33	0.73

## **Capitulo 3**

---

### **RED DE MONITOREO DE SIGNOS VITALES**

La utilización de las computadoras personales (PC) es cada vez mayor. Su uso no es exclusivamente como procesadora de palabras, análisis de hojas de distribución, o el manejo de bases de datos, su utilización se ha convertido en necesaria para cualquier actividad humana.

La red de monitoreo de signos vitales parte de la necesidad que existe en ciertos hospitales de reunir en un sólo aparato y en forma remota las diversas señales que son utilizadas en el cuidado de los pacientes de terapia intensiva, como son: electrocardiogramas, frecuencia cardíaca, frecuencia respiratoria, presión arterial y temperatura. La forma de lograrlo óptimamente es mediante el uso de una PC apoyada en los aparatos que obtienen estas señales, recolectarlos y analizarlos. Esto se logra mediante la utilización de una tarjeta de comunicación conectada a través del puerto RS232 serial de la PC.

### 3.1 Descripción del sistema

El sistema se observa en la figura 1, cuenta con tres partes principales: una PC con software de control, una interfase serial RS-232, y uno o varios nodos o sistemas de adquisición y transmisión de datos (hasta 128) sobre una línea de transmisión/recepción de tres conductores (Tx, Rx, GND). Se denomina sistema de adquisición y transmisión o nodo al circuito responsable de controlar las comunicaciones entre la computadora y los dispositivos sensores y actuadores (considerados como instrumentos externos al sistema) a través de la interfase serial RS-232. Los sensores son dispositivos médicos que realizan el muestreo de datos de donde el sistema de adquisición y transmisión los va recolectando y enviando a la PC que va de procesandolas por medio de "software" especializado. Los actuadores son los elementos que responden cuando existen situaciones que en el "software" son consideradas como críticas o peligrosas y requieran alguna medida preventiva, como el activar alguna alarma.

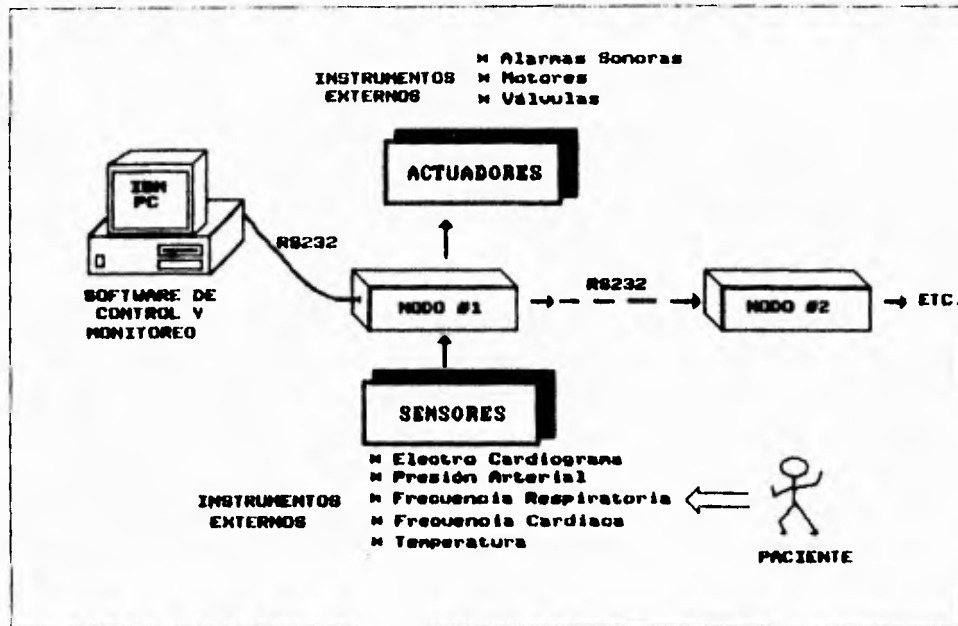


Figura 1. Diagrama Representativo de la Red de Monitoreo

La base su función principal en un UART (Universal Asynchronous Receive Transmitter), este dispositivo fue uno de los primeros y hasta ahora el más popular de los dispositivos de transmisión y recepción de datos, ya que combina ambas funciones (recepción y transmisión) en forma independiente, contando cada una con su propio bus de datos, puerto serie y reloj. El protocolo de comunicación que determina el número de bits de datos y el tipo de paridad (o si no existe paridad), es común a ambos, receptor y transmisor. Esta arquitectura permite operar en función "dúplex" con velocidades de transmisión diferentes. La lógica que controla el número de bits de detención afecta únicamente a la parte del transmisor. Esto se debe a que la parte receptora sólo busca un bit de detención y una vez encontrado no importa lo que venga a continuación. En la parte de el transmisor se define el número exacto de bits de detención que se insertan, de no ser así, se generan problemas de comunicación.



**PAGINACION VARIA**

**COMPLETA LA INFORMACION**

El UART puede conectarse a un microprocesador en una configuración interruptiva o en modo de escrutinio periódico. En el primer caso, la señal de "registro de transmisor vacío" solicita una interrupción del microprocesador, quien la atiende cargando nuevos datos y enviando el pulso de control correspondiente. Si el microprocesador no genera los nuevos datos, el UART sigue marcando la línea (transmitiendo bits de detección) indefinidamente.

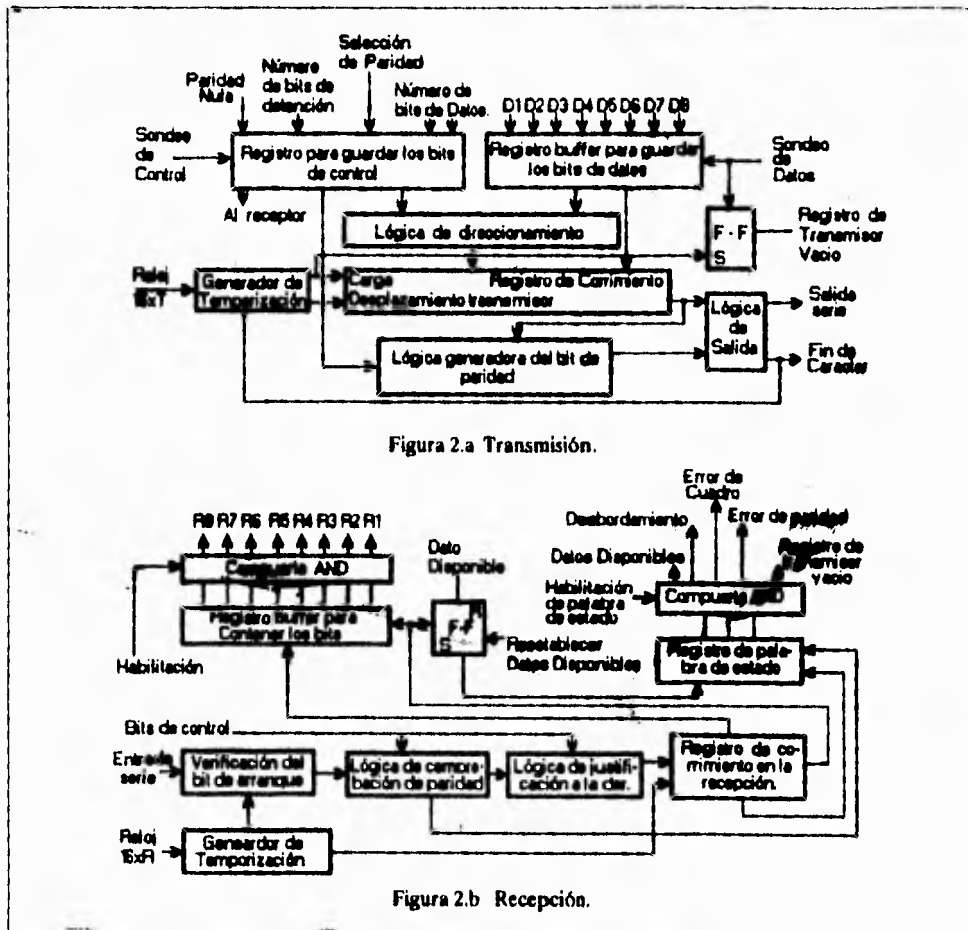


Figura 2. Diagramas de Bloques de la Recepción y la Transmisión de un UART.

En la figura 2.a, se muestra el diagrama de bloques del transmisor. En él observa que se opera en forma de reconocimiento e intercambio (handshaking) con el microprocesador. Los dos bloques principales que lo constituyen son el registro de almacenamiento temporal (buffer) del transmisor y el registro de corrimiento de paralelo a serie.

Como el transmisor, el receptor (figura 2.b) también cuenta con un registro de corrimiento y otro de almacenamiento temporal como bloques fundamentales. Los datos son desplazados en formato serie en el registro de corrimiento serie-paralelo. Cuando se completa una palabra se transfiere al registro de almacenamiento, quedando libre el registro de corrimiento para recibir la siguiente palabra. Cuando la palabra se encuentra disponible en el bus de datos, se genera un pulso de nivel alto y el registro de estados es cargado con tres bits indicadores de error.

El atractivo de la comunicación asíncrona radica en que el receptor y el transmisor pueden operar con relojes independientes. Con el fin de evitar la desviación de frecuencia, las fluctuaciones y proporcionar inmunidad al ruido, se utiliza un reloj con una frecuencia múltiplo exacto de la velocidad a la que fluyen los datos, generalmente dieciseis. Cuando el receptor detecta un bit de arranque por una transición de "1" a "0" en la entrada serie, se inicia el conteo de los pulsos de reloj, después de ocho pulsos (nominalmente al centro del bit), es muestreada la entrada serie. Si se detecta un "0", se considera que el bit de arranque es válido. Si el bit es un "1" se considera que la primera transición es ruido y el receptor es reinicializado. Una vez detectado un bit válido de arranque, los datos se muestrean cada dieciseis pulsos de reloj.

Algunos de los UART más recientes como el HM6402 (fabricado por Harris Semiconductor) y el IM6402 (fabricado por Intersil) utilizan relojes de 32 a 64 veces la frecuencia de entrada de datos para incrementar la precisión del centro de muestreo, incrementando así los márgenes de distorsión de datos.

3.2 Operación del sistema de transmisión y recepción

El sistema de transmisión y recepción es el medio por el que se realiza y controla la comunicación entre la PC y los instrumentos externos, es posible utilizar varios dentro de la red de monitoreo, cada uno controlando un grupo de instrumentos de medición. La forma de identificar a los diversos nodos es mediante una palabra de control de siete bits que es enviada desde la PC a través del puerto de comunicaciones serial (RS232) y que debe coincidir con la especificada físicamente en el sistema de recepción y transmisión como dirección de selección.

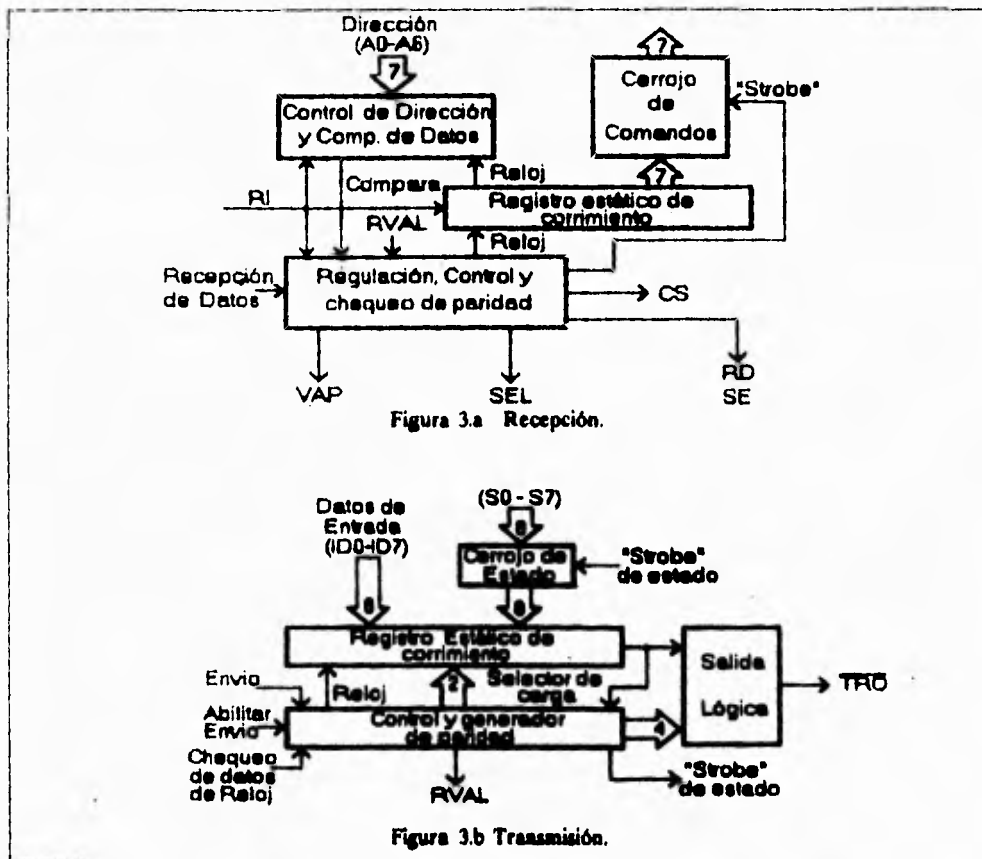


Figura 3. Diagrama esquemático del MC14469.

El componente central del sistema es un UART fabricado por Motorola, el MC14469. Este es un receptor/transmisor asíncrono direccionable hecho con la tecnología CMOS y está diseñado especialmente para la recolección de datos en forma remota y control de salida. Es en este circuito donde se define la dirección para seleccionar el nodo, ésta se establece por medio de un microinterruptor, el cual está conectado a las siete líneas de dirección A0 - A6 del MC14469.

La operación del circuito se resume en los diagramas de bloques de la figura 3, que como se puede observar es muy similar al diagrama de la figura 2. Para la recepción de datos de entrada se debe enviar al menos una dirección válida a RI (Receive Input, del MC14469) para establecer las condiciones internas necesarias para que el transmisor trabaje. Una vez realizada la recepción de una dirección válida se recibe una palabra de comando. El transmisor comenzará a operar cuando detecte un cambio de nivel de bajo a alto en la terminal de envío (SEND, del MC14469). Es en ese instante cuando los puertos de entrada de datos del UART ID0-ID7 (Input Data) y los puertos de estado S0-S7 (Status) son enviados hacia la PC.

El formato de datos asíncronos consiste de cuatro palabras de once bits, dos para la recepción y dos palabras para la transmisión. Para la recepción (Figura 4a), la primera palabra contiene un bit de arranque, siete de dirección, un identificador de dirección (validado alto), uno de paridad y otro de detección, la segunda está formada por un bit de inicio, siete de control, un identificador de control (validado bajo), uno de paridad y otro de detención. El tipo de paridad manejado por el circuito es par. La recepción de una dirección y un byte de control establecen las condiciones internas necesarias para la transmisión. El ingreso de una cadena de datos de dirección (Figura 4c) genera un pulso válido de recepción (VAP), de igual manera cuando se recibe una cadena de datos de control se genera un pulso de "Comand Strobe" (CS). La terminal de envío (SEND) debe detectar un cambio de nivel lógico de bajo a alto para transmitir, éste debe ocurrir en un tiempo no mayor al equivalente a ocho bits después de haberse generado el pulso de CS de lo contrario no ocurrirá la transmisión. Al alambrar directamente CS con SEND se garantiza que esto ocurra, además de que se asegura que la transmisión se realice después de recibir un comando válido. Cuando SEND cambia a alto, inicia la transmisión de datos, definidos en el puerto de ingreso de datos ID0 a ID7 y en el puerto de estado S0 a S7, las palabras transmitidas contienen cada una un bit de inicio, ocho de datos, uno de paridad par y uno de



dirección definida en las líneas de dirección A0-A6 con el microinterruptor del sistema, si los valores corresponden entonces el circuito se prepara para recibir la palabra de control que es enviada inmediatamente después de la palabra de dirección por la PC.

La palabra de control que es transmitida aparecerá en el MC14469 en el puerto de control (C0-C6) después de ser recibida, ésta se compone por siete bits de datos, de los cuales los tres primeros (C0-C2) son utilizados para seleccionar el canal de entrada del convertidor A/D y adquirir el dato del dispositivo médico que se requiere sensar o direccionar el dispositivo actuador que se requiera activar, el cuarto bit (C3) es utilizado para activar cuando está en estado lógico alto "1" el convertidor A/D, si éste se encuentra en estado lógico bajo "0" no se produce la adquisición de datos, los bits C4 y C5 se requieren para el control de los actuadores, C4 indica la activación (estado lógico alto) o la desactivación (estado lógico bajo) de los dispositivos actuadores, C5 indica cuando se encuentra en estado lógico alto que la activación o desactivación se puede llevar a cabo, de no ser así estas dos acciones no se realizarán.

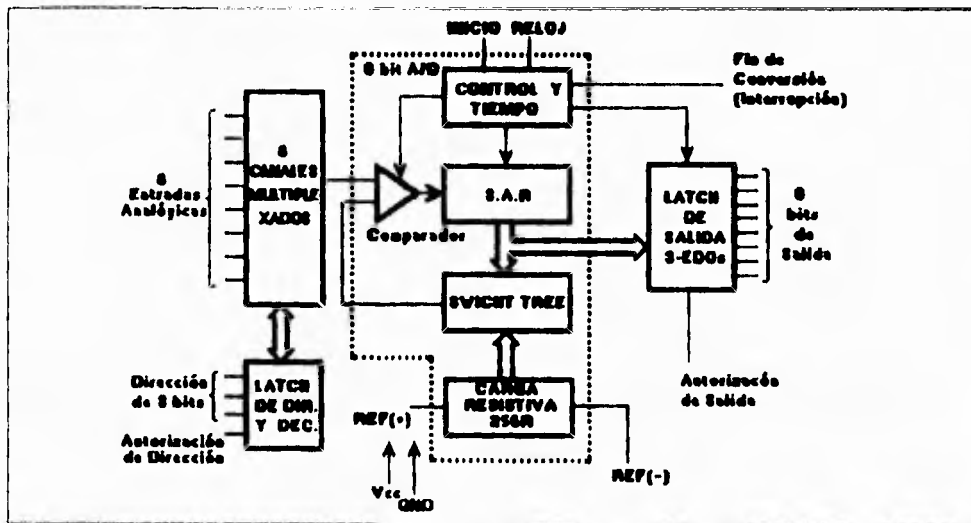


Figura 5. Diagrama de Bloques del ADC0809.

### 3.3 Adquisición de datos analógicos

La adquisición de datos se controla directamente por el programa de la PC, el cual ha sido diseñado buscando el grado de operación y funcionalidad óptimo para obtener una velocidad adecuada.

Como se mencionó anteriormente, la parte central del sistema de adquisición y transmisión de datos es el UART (Receptor/Transmisor direccionable asíncrono MC14469), que es el responsable de controlar todas las señales utilizadas para la comunicación, estas señales deben ser digitales ya que son el tipo de señales manejadas por la PC. Debido a que los instrumentos de medición de tipo médico operan con señales analógicas, no es posible conectar directamente al MC14469 a los dispositivos sensores, por lo cual, es necesario utilizar un convertidor analógico digital (el ADC0809) como complemento al sistema.

El ADC809 es un componente de adquisición de datos monolítico fabricado con tecnología CMOS que convierte una señal analógica a una señal digital de ocho bits, con ocho canales de entrada multiplexados (como se observa en el diagrama de bloques de la figura 5), su funcionamiento es el de mayor velocidad y se conoce como de ADC de ráfaga.

Canal analógico Seleccionado	Dirección		
	A2	A1	A0
IN0	0	0	0
IN1	0	0	1
IN2	0	1	0
IN3	0	1	1
IN4	1	0	0
IN5	1	0	1
IN6	1	1	0
IN7	1	1	1

Tabla 1. Direccionamiento del ADC0809.

La conversión analógica a digital utiliza la técnica de aproximaciones sucesivas, ésta es adecuada porque puede combinar resoluciones útiles hasta 12 bits y más, con un tiempo bastante breve de conversión (menos de 12 microsegundos para la conversión de 12 bits). Otra de las ventajas es que el tiempo de conversión es fijo e independiente de la magnitud de la señal de entrada, permitiendo un enlace eficiente con los microprocesadores.



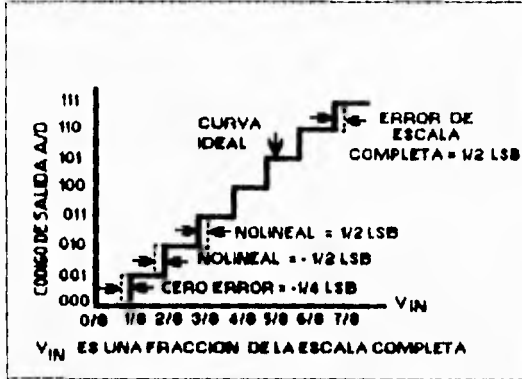


Figura 6.A Curva de transferencia de un convertidor de 3 bits (conversión)

La entrada básica de control es una línea de conversión, puesto que los datos de salida no son válidos hasta que finalice ésta; una línea de estado (o fin de conversión EOC) indica que el convertidor está ocupado, mientras se encuentra en proceso.

Como características principales el convertidor tiene una alta impedancia, un comparador estabilizador de corte (chopper), un divisor de voltaje 256R con un árbol de interruptores analógicos

(Swich Tree) y un registro de aproximaciones sucesivas (S.A.R). Los ocho canales multiplexados pueden ser conectados a cualquier señal analógica. La selección de un canal en particular se realiza mediante el direccionamiento mostrado en la tabla 1.

El divisor de voltaje fija niveles de referencia para cada comparador de forma tal que existan 256 niveles correspondientes al valor del LSB (bit menos significativo). La entrada analógica, se conecta a la otra entrada de cada comparador. Cuando el voltaje analógico es menor al voltaje de LSB, dará como resultado que todas las salidas serán altas, cuando es mayor, una o más de las salidas del comparador serán bajas. Por ejemplo, en el caso de que el convertidor fuera de 3 bits como se muestra en las figuras 6A y 6B, si el voltaje analógico está entre 3/8 y 4/8 del nivel máximo, los primeros 3 bits

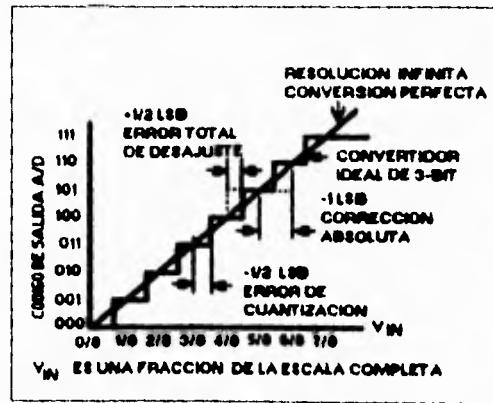


Figura 6B. Curva de transferencia de un convertidor de 3 bits (resolución).

serán bajos y el resto serán altos. De acuerdo a la tabla de verdad se responderá solamente a la salida baja y producirá una salida binaria 011, que será el equivalente digital dentro de la resolución.

El ADC0809 está diseñado para proporcionar una rápida y exacta conversión en un amplio rango de temperaturas; está se divide en tres partes principales: la red de cargas 256R, el registro de aproximaciones sucesivas y el comparador.

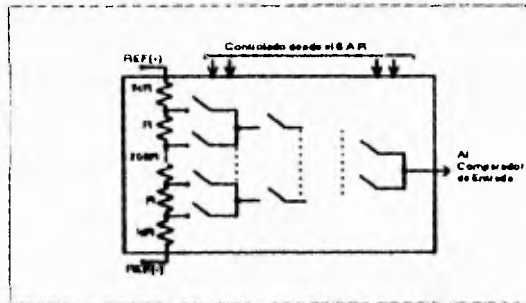


Figura 7. Arbol de Interruptores ("Switch tree").

La resistencia inferior y superior en la red de cargas 256R de la figura 7, no tienen los mismos valores que el resto de las resistencias de la red. La diferencia en estas resistencias provoca que la característica de salida sea simétrica entre el cero y los puntos de escala completa (full scale) en la curva de transferencia. La primera transacción de salida ocurre cuando la señal ha sido extendida al LSB (bit

menos significativo), y la siguiente transición de salida ocurre siempre después de un LSB completo sobre la escala completa.

El registro de aproximaciones sucesivas (S.A.R) ejecuta ocho iteraciones para aproximar el voltaje de entrada. Para cualquier tipo de convertidor SAR, "n" iteraciones son requeridas para convertir "n" bits. Para esquematizar esto de manera simplificada y no llenar de unos y ceros una tabla; la figura 6A muestra un ejemplo típico de un convertidor de tres bits en lugar del real que es de ocho. En el ADC0809 la conversión es extendida a ocho bits utilizando la red 256R.

El SAR del convertidor A/D es reinicializado con el flanco de subida del pulso de inicio de conversión (SC). El inicio de proceso interrumpirá la recepción de nuevos pulsos de CS. Una conversión continúa se realizará cuando la salida de fin de conversión (EOC) y la entrada de CS sean unidas. Si es utilizado en este modo un pulso de CS externo puede ser

aplicado después del encendido. La señal EOC permanecerá baja ocho pulsos después de iniciada la conversión.

La sección más importante del convertidor analógico digital es el comparador. Esta sección es responsable de la última corrección de la conversión. La cual tiene mayor influencia en la capacidad de repetición del dispositivo. Un comparador estabilizador de corte (Chopper), provee el más efectivo método para satisfacer todos los requerimientos de conversión, éste convierte una señal de entrada DC en una señal AC, la cual es saturada a través de un amplificador de AC.

Estas características hacen del convertidor ADC0809 un elección adecuada para ser implementado en sistema de adquisición y transmisión de datos.

### 3.4 Manejo de los actuadores

Para el manejo de las señales de error y con el fin de evitar variaciones de potencial en el circuito y mantener constante una señal hasta que sea desconectada por software, es necesario utilizar un cerrojo (latch) o un flip-flop. El MC14099 es un cerrojo direccionable de ocho bits (8-bit Addressable Latches) hecho con tecnología CMOS, su funcionamiento es sumamente sencillo, éste se explica con ayuda del diagrama de bloques de la figura 8.

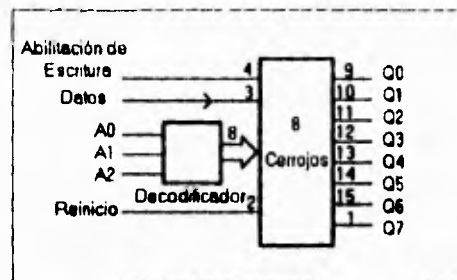


Figura 8. Diagrama de bloques del MC14099

El MC14099 cuenta con ocho bits de salida, cada uno de estos es direccionado por las terminales A0, A1 y A2, donde cada una de las posibles combinaciones de estas tres terminales corresponde a un bit (por ejemplo, la señal 000 corresponde al bit Q0, la señal 001 al bit Q2 y así sucesivamente), al activar el circuito aplicando un cero lógico en la terminal WD (Write Disable terminal 4) la señal contenida en la terminal de dato (Data

terminal 3) es puesta en el bit direccionado manteniéndose el resto sin variaciones. La tabla 2 muestra la operación del MC14099.

Las direcciones de las terminales (A0, A1 y A2) están definidas de tal forma que cada uno de los actuadores corresponde a un dispositivo sensor. Lo anterior se realiza con el fin de evitar confusiones y errores al interpretar la señal de los actuadores.

ESCRITURA (WD)	REINICIO	CERROJO DIRECCIONADO	CERROJOS NO DIRECCIONADOS
0	0	DATO	Q <sub>n</sub>
0	0	DATO	REINICIO
1	1	Q <sub>n</sub>	Q <sub>n</sub>
1	1	REINICIO	REINICIO

Tabla 2. Manejo de las salidas del MC14099.

### 3.5 Utilización de las líneas de control

Las líneas de control, del MC14469, son utilizadas para la operación de las diversas líneas de entrada de datos analógicos (a través del ADC0809) y las señales a los actuadores (a través del MC14099). Cada línea de entrada o dispositivo médico está relacionado con un actuador, por lo que la acción a tomar de éstos dependerá de la lectura en la línea de control. Al recibirse un byte de comando o control válido el MC14469 lo sitúa en las

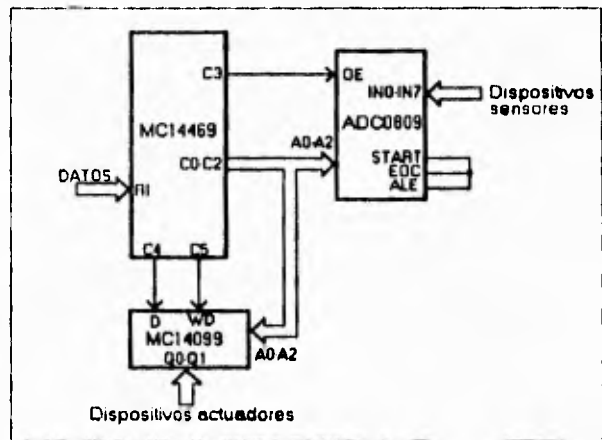


Figura 9. Relación del MC14469 con el ADC0809 y el MC14099.

terminales de control C0-C6, estos siete bits son utilizados para la operación y la comunicación con los dispositivos externos, como se aprecia en el diagrama de bloques de la figura 9. Los primeros tres bits (C0, C1 y C2) son los que definen el direccionamiento tanto de los dispositivos médicos, como de los actuadores por lo que éstos están alambrados a las

terminales respectivas del convertidor A/D (A0, A1 y A2 del ADC0809) y del cerrojo (A0, A1 y A2 del MC14099), el bit C3 cuando está en alto activa el convertidor A/D puesto que se encuentra alambrado a la terminal de habilitación de salida OE (output enable) del

TERMINAL DEL MC14469	SELECTOR DE SENSORES CANAL DE ENTRADA	SELECTOR DE SALIDA ACTUADORES
C0	A0	A0
C1	A1	A1
C2	A2	A2
C3	OE	----
C4	----	D
C5	----	WD

ADC0809, el bit C4 se utiliza para activar (en alto) o desactivar (en bajo) el actuador direccionado en ese momento y está alambrado a la terminal de dato D (data) del MC14099, la terminal C5 está alambrada a la terminal WD del MC14099 y esta misma indica si se debe desactivar o no algún actuador, C6 no es utilizado. En la tabla 3 se resume lo anterior.

Tabla 3. Relación de las terminales de Comando del

Las terminales de entrada al MC14469 ID0-ID7 y las de salida digital del convertidor analógico (ADC0809) digital (IN0-IN7) son conectadas directamente entre si. Con el convertidor A/D se realiza la adquisición de datos analógicos provenientes de los dispositivos médicos y éste es activado por el MC14469. Las terminales de comando están alambradas para direccionar el canal analógico que es sensado, y así realizar la conversión. Para este caso se utiliza el método de lectura continua, conectando al mismo punto, las señales ALE, START y EOC del ADC0809 ya que es necesario mantener la entrada de voltaje durante el proceso de conversión, que dura algunos microsegundos.

El sistema tiene la capacidad de operar hasta ocho líneas de entrada de datos analógicos, éstas son conectadas a las diversas interfaces de acondicionamiento médico, por el momento sólo se utilizan cinco (frecuencia cardiaca, presión arterial, frecuencia respiratoria, temperatura y electrocardiograma).

Teóricamente con un direccionamiento de siete bits es posible direccionar hasta 128 nodos, cada nodo consume aproximadamente 50 mA y está alimentado por un eliminador de baterías de 9 volts. El límite práctico está en función de la velocidad de adquisición de Datos, que por tratarse de variables muy críticas no debe ser muy baja y con cada nodo se disminuiría.

### 3.6 Oscilador

Para una correcta sincronización entre el nodo y la PC es necesario controlar el circuito del nodo mediante la utilización de un cristal oscilador. Este por sus características piezoeléctricas, es decir, cuando se aplica un voltaje a través de las caras paralelas de un cristal se expande y contrae dependiendo de la polaridad del voltaje que se aplique, y a la inversa, es un elemento que se comporta como un circuito resonante con un alto factor de calidad. Por tanto las frecuencias producidas por los cristales son mucho muy estables, ya que están diseñados para mantener una tolerancia a la frecuencia de 0.0005 % o mejor.

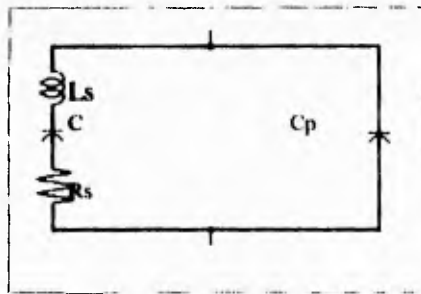


Figura 10. Circuito equivalente del cristal Oscilador.

Los cristales osciladores pueden ser modelados con el circuito mostrado en la figura 10. La combinación de los elementos en serie  $L_s$ ,  $C_s$  y  $R_s$  representan el equivalente eléctrico de las características vibratorias propias del cristal. La inductancia  $L_s$  es el equivalente eléctrico de la masa del cristal que es efectiva en la vibración,  $C_s$  es el equivalente mecánico de la permitividad efectiva y  $R_s$  representa el equivalente de la fricción mecánica. Este circuito se modifica cuando el cristal se monta en el

circuito. Como resultado de esto, en el circuito equivalente del cristal de la figura 10 aparece montado un capacitor  $C_p$ , éste representa la capacitancia electrostática entre los electrodos del cristal y la capacitancia asociada con el circuito cuando el cristal no está vibrando.

Para el correcto funcionamiento del oscilador en el circuito, la frecuencia deberá estar dividida por sesenta y cuatro para que puedan ser perfectamente sincronizados la recepción de datos y el reloj de rango de datos. Por lo tanto el período de bits es sesenta y cuatro veces el período del oscilador. Como se requiere un baudaje de 4800, el cristal oscilador deberá tener una frecuencia de 307.2 KHz. Debido a que éste no es un valor comercial, se obtiene a través de un cristal oscilador de 4Mhz, éste es dividido entre 13, con la ayuda de un contador (74LS161) lo que proporciona como resultado una frecuencia de 307.69 KHz, que es un valor aproximado por lo que el margen de error es bajo. Además en

caso de necesitar un baudaje de 2400 se instaló un flip-flop para reducir la frecuencia a la mitad y contar con opción de seleccionar cualquiera de los dos baudajes con la ayuda de un microinterruptor.

Para dar lugar a una fase máxima de recepción de datos del MC14469 está centrado en la mitad de cada bit de datos. La recepción de un bit de comienzo inicia el muestreo de recepción de datos y se sincroniza para recibir una cadena de bits, desde que los datos son enviados asncronamente el oscilador de transmisión y el oscilador de recepción deben de estar a la misma frecuencia para garantizar que el muestreo de recepción ocurra en la mitad del periodo de bits. La variación máxima permisible en la frecuencia del oscilador entre una unidad de transmisión y una de recepción puede ser tal que sea mayor que el tiempo total de la recepción de la palabra de datos. Como cada palabra que se recibe consiste de once bits, la variación en el oscilador no puede ser mayor a la mitad del pulso del tiempo de un bit dividido entre once pulsos de bit o sea 4.5 %.

### **3.7 Selector de velocidad de comunicación**

Para el control de la frecuencia es posible utilizar otro arreglo en el circuito, éste no se alabró debido a que en esta aplicación no se requiere cambiar constantemente el baudaje, sin embargo en caso de ser necesario se puede realizar. El arreglo consiste de tres elementos principales, un cristal a 19.6608 Mhz, un circuito Oscilador/Divisor CMOS (el 74HC4060) y un microinterruptor como selector (referirse a la Figura 11). Con este arreglo es posible seleccionar siete distintas frecuencias, permitiendo elegir la que mejor se adapte a la aplicación. La selección de frecuencias se deberá considerar también dentro de la configuración del software para evitar discrepancias y errores en la comunicación, esta selección se hará pensando en las características propias del sistema para obtener la mayor confiabilidad en la adquisición de datos y el óptimo control en la operación de los actuadores.

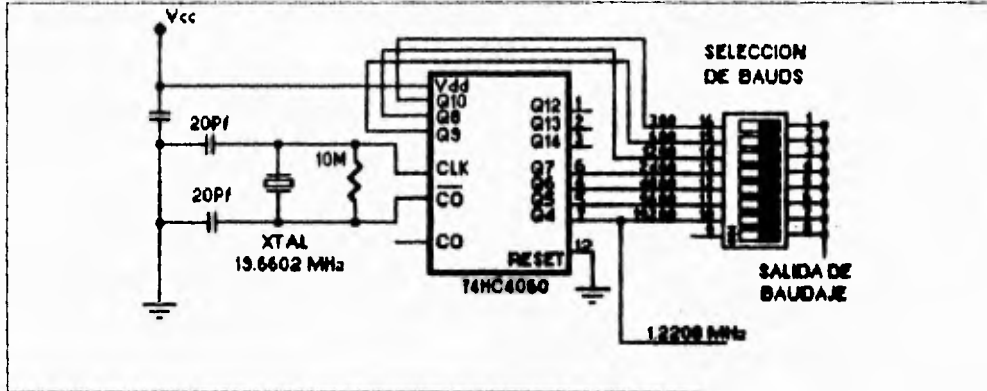


Figura 11. Circuito Selector de Baudaje.

### 3.8 La interfase RS-232

El sistema se basa en la norma RS-232 (capítulo de Métodos de comunicación). Esta norma que actualmente es la más popular define niveles de voltaje, cargas, tipos de señal, etc. El diseño de un circuito RS-232, debe cumplir con todos los requerimientos de esta norma. Se considera que dentro de la norma se especifica que todas las señales de voltajes deben ser leídas con respecto a la línea de tierra, un voltaje de 3 a 25 volts representa un "0" lógico y un voltaje de -3 a - 25 volts representa un "1" lógico; por lo tanto se trata de una lógica negativa ya que el valor lógico alto corresponde al valor de voltaje más negativo.

La RS-232 fue introducida en 1962 y se utiliza ahora en la industria. Esta norma se destina específicamente a transmisiones de datos a un solo hilo a distancias cortas, con bajas velocidades de transmisión. La interfase serial RS-232 está formalmente definida como Recomendación Standar #232 revisión C. Este es el nombre estándar para un conector hembra en un equipo de comunicación de datos (DCE) y para un conector macho en el equipo terminal de datos (DTE). Para las computadoras XT la norma define las líneas para un conector de 25 terminales (DB25, Tabla 4) y para las computadoras AT la norma define las líneas para un conector de 9 terminales (DB9, Tabla 5). Las líneas utilizadas por la norma RS-232 del DB25 se especifican a continuación:

- Clear to send (CB). La terminal CB es alta (estado lógico "1") por el DCE cuando éste está listo para recibir datos.



- Data Signal Rate Detector (CH/CI). La terminal CH/CI es baja siempre y cuando no exista un dato no aceptado.
- Data Set Ready (CC). La terminal CC es alta por el DCE cuando está listo para comunicarse con el DTE.
- Received data (BB). La terminal BB se utiliza para enviar datos al DTE.
- Ring Indicator (CE). La terminal CE es cambiada a estado alto por el DCE cuando un timbre sea detectado.

Pin	Abreviación	Nombre:
1	AA	Tierra de protección.
2	BA	Transmisión de datos
3	BB	Recepción de datos
4	CA	Requerimiento de envío
5	CB	Limpiar envío
6	CC	Datos listos
7	AB	Señal común
8	CF	Detección de señal de línea de recepción
9	---	Reservado para pruebas
10	---	Reservado para pruebas
11	---	No asignado
12	SCF	Detección de señal secundaria de recepción
13	SCB	Limpiar datos secundarios
14	SBA	Recibir datos secundarios
15	DB	Tiempo de la señal de transmisión
16	SBB	Recepción de datos secundarios
17	DD	Tiempo de la señal de recepción
18	---	No asignado
19	SCA	Requerimiento secundario para envío
20	CD	Terminal de datos lista
21	CG	Detector de calidad de señal
22	CE	Indicador de Alarma
23	CH/CI	Detector velocidad de señal de datos
24	DA	Tiempo de la señal de transmisión
25	---	No asignado

Tabla 4. Definición de las terminales del DB25.

- Request to Send (CA). La terminal CA es cambiada a estado alto por el DTE cuando éste requiere transmitir daos al DCE.

- **Transmit Data (BA).** La terminal BA es utilizada por el DTE para enviar datos al DCE.

El uso de las líneas para el conector DB9 de acuerdo a la norma RS-232 se explica a continuación.

- **Data Carrier Detect (DCD).** Esta terminal se define en estado alto ("1" lógico) siempre y cuando se encuentre la liga de datos en progreso.
- **Received Data (RD).** Esta terminal es utilizada por el DCE para transmitir datos al DTE.
- **Transmit Data (TD).** La terminal es utilizada por el DTE para enviar datos al DCE.
- **Data terminal Ready (DTR).** El DTR se define en estado alto (1 lógico) por el DTE cuando está listo para comunicarse con el DCE.
- **Data Set Ready (DSR).** La terminal de DSR se define alta ("1" lógico) por el DCE cuando está listo para comunicarse con el DTE.
- **Clear to Send (CTS).** La terminal CTS es validada alta por el DCE cuando está lista para recibir datos.
- **Ring indicator (RI).** La terminal de RI es validada alta por el DCE cuando una alarma es detectada.

Pin	Abreviación	Nombre
1	DCD	Detector de acarreo de datos
2	RD	Recepción de datos
3	TD	Transmisión de datos
4	DTR	Terminal de datos lista
5	---	Señal común
6	DSR	Asignación de datos lista
7	RTS	Requerimiento de datos
8	CTS	Limpiar el envío
9	RI	Indicador de alarma

Tabla 5. Definición de las terminales del DB09.

Para la comunicación con el puerto serie RS232 de la PC se requiere una línea de tres hilos más la tierra común, ya que el resto de las señales definidas en la norma no son utilizadas. La forma en que éstas son conectadas a través de un conector DB9 o un conector DB25 (dependiendo si la PC es XT o AT), se muestra en la figura 12. La longitud de la línea, desde la computadora hasta el último nodo de operación dependerá del grado de ruido eléctrico en el ambiente de la red. Las pruebas realizadas fueron desarrolladas con una longitud superior a 80 metros y con una velocidad de 4800 bauds, logrando excelentes resultados de operación.

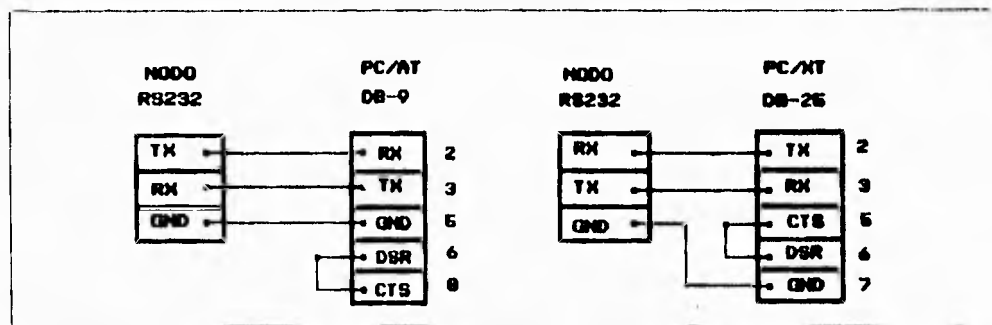


Figura 12. Utilización de los conectores DB9 y DB25.

### 3.9 Uso del MAX232

Para simplificar el diseño y la comunicación serie RS-232 es utilizado un circuito integrado fabricado especialmente para cubrir los requerimientos de esta norma; el MAX232, el cual cuenta con dos transmisores y receptores RS-232, esto otorga la posibilidad de controlar dos dispositivos, ya sea en forma independiente, o como maestro y esclavo. Al alambrear el MAX232 como maestro y esclavo proporciona la posibilidad de recibir y retransmitir una señal, que en el caso de la red de monitoreo es de gran importancia, debido a que con este sistema se logra la comunicación desde la PC a todos los nodos. Esto se muestra en la Figura 13 donde TS1 esquematiza al primer nodo, el cual es considerado como el maestro que retransmite la señal para TS2 por lo que este último es considerado como el esclavo, al recibir TS2 la señal la retransmite para TS3 lo que convierte a TS2 en el nodo maestro y a TS3 en el nodo esclavo, esto continúa así sucesivamente,

quedando el número de nodos limitado teóricamente a la capacidad de la direccionabilidad del UART que con una dirección de 7 bits es de 128, y como límite real a la velocidad requerida por el proceso, ya que entre mayor sea el número de nodos, mayor será el tiempo de respuesta.

El MAX232 está dividido en tres partes dobles, transmisor, receptor y convertidor de voltaje. El convertidor doble de voltaje tiene dos secciones, la primera utiliza un capacitor externo C1 (Figura 14) para duplicar los +5V de entrada a +10V, con una impedancia de salida de aproximadamente 200 ohms. La segunda emplea un capacitor externo C2 para convertir los +10V a -10V con una impedancia de 450 ohms. En él se utilizan los capacitores de C1 a C4 con valor de 10 microfaradios, aunque estos valores no son críticos. Si se incrementan los valores de C1 y C2 disminuirá la impedancia de salida de las fuentes (duplicadora e inversora). Si C3 y C4 son aumentados, disminuye la frecuencia en las fuentes.

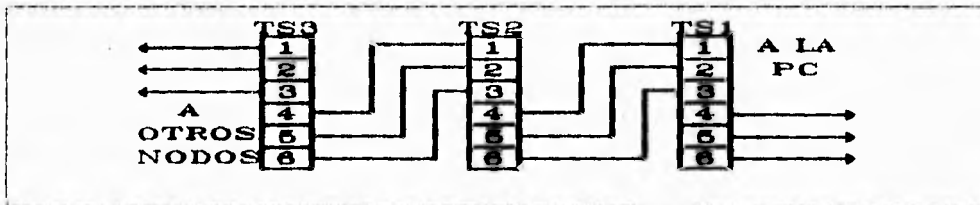


Figura 13. Conexión en cascada de los nodos.

La parte transmisora contiene un inversor de voltaje interno a 10V. La entrada es compatible con TTL y CMOS, con un umbral lógico del 26% aproximadamente (1.3 V para 5 Vcc). Si una de las entradas del transmisor no es utilizada puede ser desconectada, ya que contiene una resistencia interna de "pull-up" conectada entre Vcc y la entrada que provoca que una señal alta al transmisor no utilizada otorgue una salida baja. Los dos receptores tienen una impedancia de entrada de entre 3 ohms y 7 kilohms, los umbrales de entrada están entre  $\approx 3V$ , lo cual es compatible con cualquier entrada RS232 TTL o CMOS.

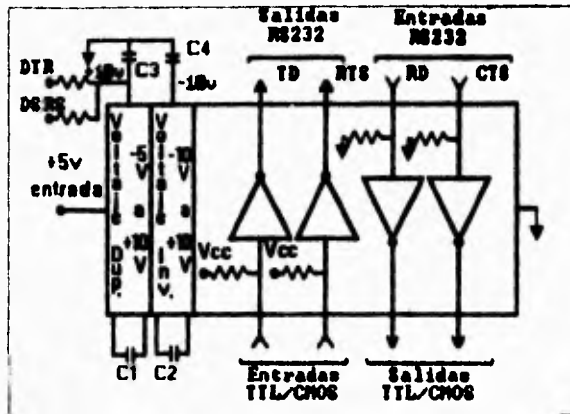
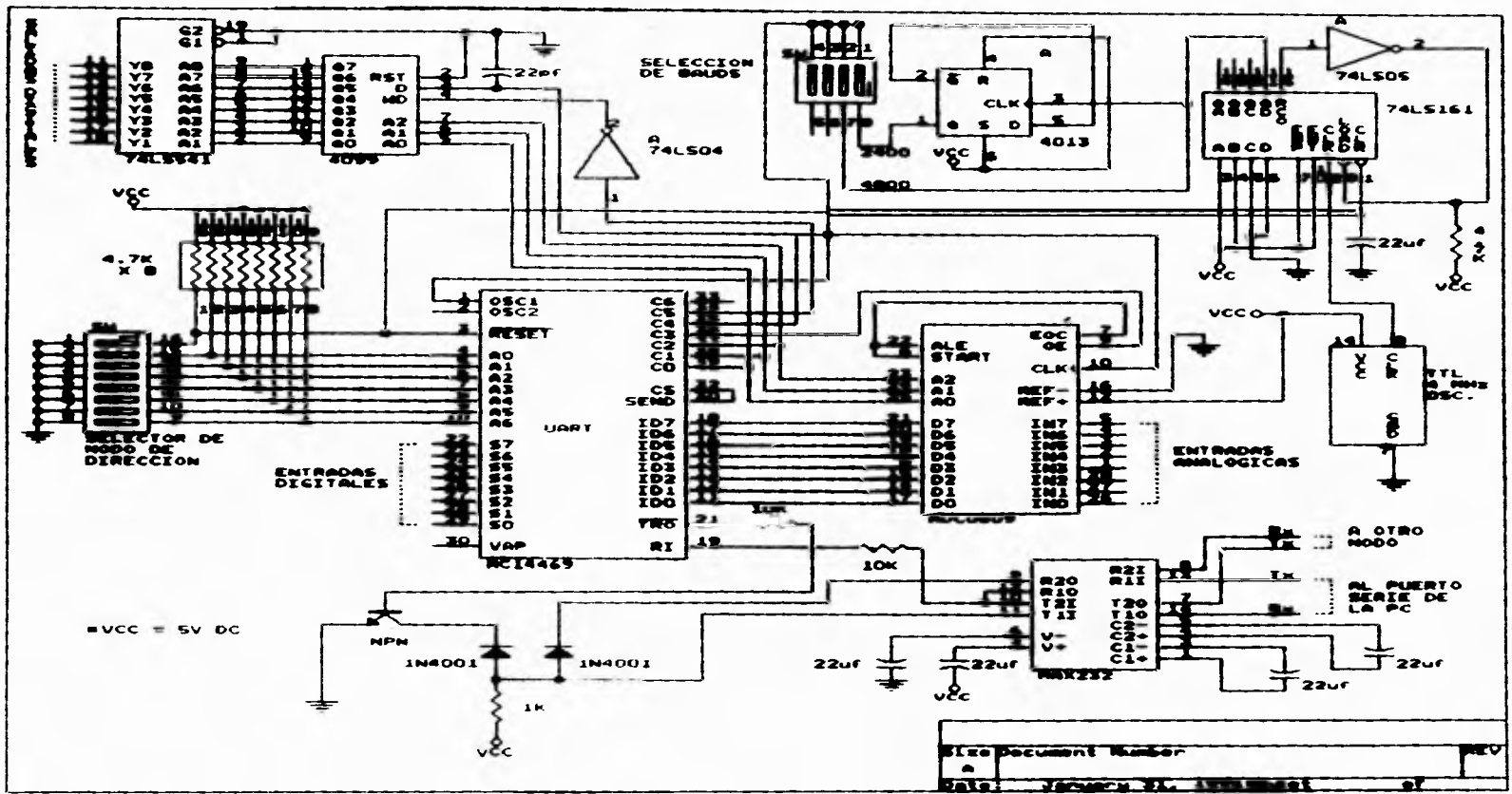


Figura 14. Esquema del MAX232.

Los receptores tienen un voltaje de entrada bajo de 0.8V y un voltaje de entrada alto de 2.4V. La salida del receptor será baja siempre que la entrada sea mayor a 2.4V. El receptor de salida será alto cuando la entrada de voltaje fluctúe u opere entre +0.8V Volt -30V.

Anteriormente al MAX232 se realizaron pruebas para el circuito de acondicionamiento de la señal serial con otros dos integrados también construidos con tecnología CMOS; éstos son el 1488 que es una NAND manejador de línea (line driver) y el 1489 que es un inversor receptor de línea (line receiver), ambos diseñados para ser utilizados con una interfase RS-232, lográndose con éstos resultados satisfactorios. Sin embargo se prefirió el MAX232, no obstante su costo superior, ya que otorga mayores facilidades de implementación, debido a que éste funciona con un voltaje de 5 volts y únicamente requiere de cuatro capacitores, en cambio para poder utilizar el 1488 y el 1489 es necesaria una fuente de  $\pm 12$  volts para su correcto funcionamiento, por lo que se tiene que instalar en el circuito una fuente especial para alimentar sólo a la interfase con los voltajes requeridos.

El circuito armatado con todos sus elementos se muestra en el diagrama siguiente:



Size	Document Number	REV
A		
DATE	JANUARY 21, 1988	01

### 3.10 Convertidor de RS232 a RS422.

Como se vió en el capítulo correspondiente la norma RS-232 es limitada, debido a que restringe la longitud máxima de transmisión a algunos metros de distancia, además que la señal debe estar entre +5 y +15 volts para el nivel alto y entre -5 volts y -15 volts para el nivel bajo. Además de una transferencia de datos promedio arriba de 20,000 (20K) bits por segundo (bps). En caso de necesitar aumentar la longitud de la línea y la velocidad de transmisión, se tendrían serios problemas, ya que es posible que en el laboratorio si se realizan algunas pruebas con miras a satisfacer los nuevos requerimientos se tenga éxito, sin embargo los fabricantes de los elementos del que se forma el circuito de interfase no garantizan el funcionamiento de éstos a valores superiores de la norma.

En 1975 se creó la norma RS422, ésta permite un promedio de transferencia de datos de hasta 10M bits por segundo hasta a 50 pies y hasta aproximadamente 100K bps a una longitud hasta de una milla. Como se ve, esta norma nos da mayor velocidad y mayor longitud. En caso de ser necesario se plantea un arreglo alternativo, para convertir la salida RS232 a una salida RS422. El arreglo se muestra en la figura 15 y como se ve únicamente se requiere de un circuito integrado más (el DS-8922 de NATIONAL), este circuito tiene la capacidad de cambiar un nivel TTL a señales RS422 y viceversa, éste, acoplado al circuito MAX232 otorga la facilidad de operar sin mayores dificultades cualquiera de las dos normas. El DS-8922 se alimenta con un voltaje de 5 volts que se obtiene de la misma fuente del circuito.

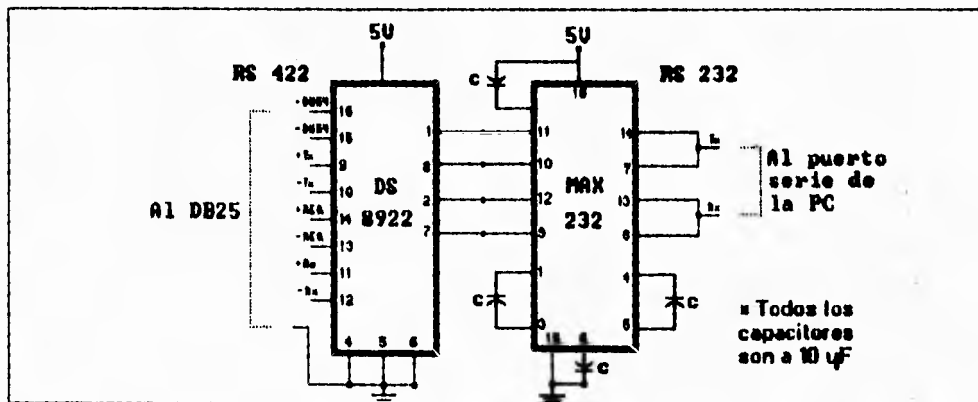


Figura 15. Convertidor RS232 a RS422.

### 3.11 Programación del puerto de serie comunicaciones

El objetivo primordial del software es el manejo de la información digital que es recibida por el puerto de comunicaciones sobre el estado que presentan los sensores del monitoreo de los signos vitales, así como establecer el protocolo adecuado de comunicación entre el sistema externo y la computadora. Una característica principal del programa es otorgar al usuario (médico) un medio amigable de la representación de los datos en la PC, por medio de pantallas especiales de monitoreo, graficación en tiempo real e impresión de reportes.

### 3.12 Operación del sistema en lenguaje BASIC

Para que el software pueda procesar los datos correctamente, es necesario que opere los dispositivos de comunicación adecuadamente, la forma en que esto se realice dependerá del lenguaje que se utilice para programar. En BASIC por ejemplo, para garantizar la comunicación se deberá realizar lo siguiente. Primero se informa a la PC que se va a trabajar con un puerto serie de comunicación, esto se hace mediante la utilización de la instrucción "OPEN COM", al utilizarla se define en ella las características del protocolo de comunicación, como es velocidad, tipo de paridad, número de bits de información que se van a utilizar, número de bits de paro, etc. Así como el nombre que se dará al puerto, como se aprecia en el siguiente ejemplo.

```
OPEN "COM1: 4800,E,8,1" AS # 1
```

Aquí se está abriendo el puerto de comunicación serie COM1, a una velocidad de 4800 bauds, con paridad par (Equal), con 8 bits de datos, un bit de paridad y el puerto se definió como #1.

Una vez abierto el puerto para evitar posibles errores por basura en el almacenamiento temporal (buffer), es necesario limpiarlo. Con esto se evita que se saturé. Lo anterior se realiza como se muestra a continuación.



```
WHILE B$ " "  
B$=INPUT$(1,#1)  
WEND
```

Una vez listo el almacenamiento temporal para la comunicación con el nodo es necesario enviar un byte por el puerto abierto, como se explicó en la parte de "Operación del nodo", éste contendrá la dirección del nodo con el cual se solicitará la comunicación y no deberá ser mayor a 127 (7F en hexadecimal), y un segundo byte que se denominó de comando, en el cual en los tres bits menos significativos se especificará la dirección del canal analógico o del actuador, los cuatro bits restantes estarán especificados como ya se explicó en la parte correspondiente a la "Utilización de las líneas de control", el bit más significativo deberá ser siempre un valor mayor o igual a 128 (80 en hexadecimal) hasta 255 (FF en hexadecimal). Para el envío de estos dos bytes en lenguaje BASIC se utiliza la instrucción PRINT como se muestra a continuación:

```
PRINT#1,CHAR$(address)  
PRINT#1,CHAR$(command)
```

Donde "address" y "command" son dos variables que estarán definidas con anterioridad dentro del programa como del tipo entero.

Después de haber activado con estas dos instrucciones el puerto serie, para leer los datos que se están enviando desde el nodo el BASIC define la instrucción INPUT, en esta se indicará el puerto donde se tomará la lectura, así como el número de bits. Un ejemplo se muestra a continuación:

```
B1=INPUT(1,#1)  
B2=INPUT(1,#1)
```

Se debe tener en cuenta que aunque el segundo byte no es utilizado se debe de leer, de lo contrario, éste quedará dentro del almacenamiento temporal provocando que en la siguiente lectura sea el dato obtenido, confundiendo con la lectura del canal analógico direccionado y provocando un dato falso.

Con estas instrucciones se puede realizar el manejo de los datos entre el puerto de comunicaciones de la PC y el nodo en cuestión, así como su interpretación de acuerdo a los requerimientos médicos, además del manejo de los actuadores. Al finalizar el programa, para garantizar que no se presenten problemas con la PC, después de utilizado el programa, se deberá cerrar el puerto de la forma siguiente.

CLOSE #1

### 3.13 Operación del sistema en lenguaje C++

La operación del sistema en lenguaje C++, es un poco más complicada, debido a que éste lenguaje fue diseñado para estar lo más cercano posible al lenguaje de máquina. Se conoce como lenguaje a nivel medio por que combina elementos de lenguajes de alto nivel con la funcionalidad del lenguaje ensamblador. Para el caso de la comunicación a través del puerto serie, existen en las librerías de C++ instrucciones que manejan transmisión y recepción de datos a través del puerto serie, sin embargo, es necesario crear otras que funcionen de forma similar, pero que consideren la información y parámetros como son manejados por el sistema de adquisición y transmisión de datos.

Para poder comprender esto, se definen algunos conceptos relativos a las computadoras personales y su funcionamiento a nivel máquina. Internamente las computadoras para poder comunicarse con el exterior tienen normalmente conectado al microprocesador un UART, similar al utilizado en el sistema de transmisión y recepción de datos. Todos estos UART's tienen uno o más registros para leer o escribir datos, a través de los puertos de entrada/salida. Las direcciones que se utilizan comúnmente para estos puertos son las siguientes:

Puerto Serie COM1	3F8H
" " COM2	2F8H
" " COM3	3E8H
" " COM4	2E8H

No es muy común que algún tipo de computadora maneje más de cuatro puertos de comunicaciones por lo que no se establece un estándar más allá de COM4.

Los registros utilizados por los UART's definen una serie de señales que le indican al microprocesador el estado de los puertos de comunicación, por ejemplo, el UART 8250 que es de uso frecuente en las computadoras IBM PC y compatibles. Este utiliza 10 registros definidos de la siguiente forma:

**El Registro 0 (RBR).** Se define como registro receptor de almacenamiento temporal (buffer). Cuando un carácter es recibido por el UART 8250, este ensambla y lo almacena en el RBR. El RBR es el primer registro del UART que puede ser direccionado y leído por el registro base del puerto serie. Para COM1 el RBR puede ser leído buscando un valor desde el puerto 3F8H. La Figura 16 ilustra como el carácter de datos es regresado por el UART. Si se requiere utilizar menos de ocho bits, es posible utilizar una máscara para eliminar cualquier bit no utilizado.

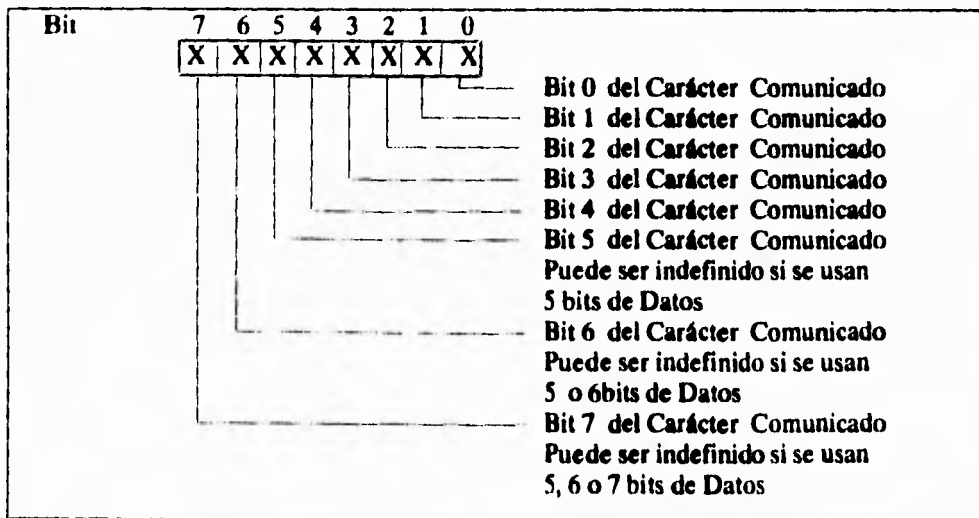


Figura 16. Registro Receptor de Almacenamiento Temporal del UART 8250.

El Registro 0 (THR). También se define como registro retenedor transmisor, este es utilizado por el UART 8250, para transmitir un carácter de datos fuera del puerto serie. Este comparte el primer registro del UART y puede ser direccionado escribiendo la dirección en el registro base del puerto serie. Para COM1, el THR puede ser habilitado para enviar escribiendo el valor 3F8H. La Figura 17 ilustra como el caracter de datos es enviado por el UART.

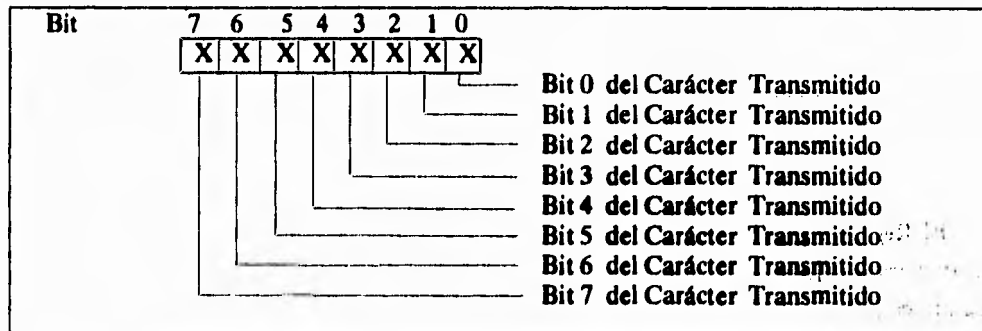


Figura 17. Registro Retenedor Transmisor (THR) del UART 8250.

El Registro 1 (IER). Se define como registro habilitador de interrupciones y puede ser activado para escritura o lectura con la dirección del puerto más uno. Para COM1, el IER para escribir un dato fuera o leer un dato recibido la dirección del puerto serial sera 3F9H. La Figura 18 muestra las funciones definidas en el IER.

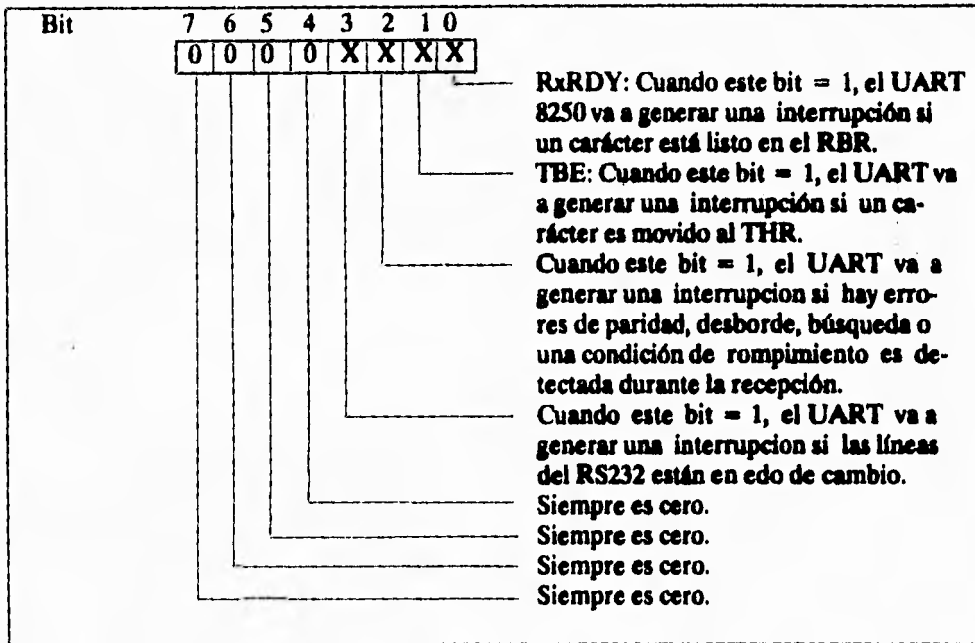


Figura 18. Registro Habilitador de Interrupciones (IER) del UART 8250.

**El Registro 2 (IIR).** Se define como registro identificador de interrupciones y es utilizado para determinar si alguna interrupción ha ocurrido. Este puede ser habilitado para su lectura con la dirección del puerto serie más dos. Para COM1, el IIR puede ser leído con la dirección 3FAH. La Figura 19 muestra las funciones definidas en el IIR.

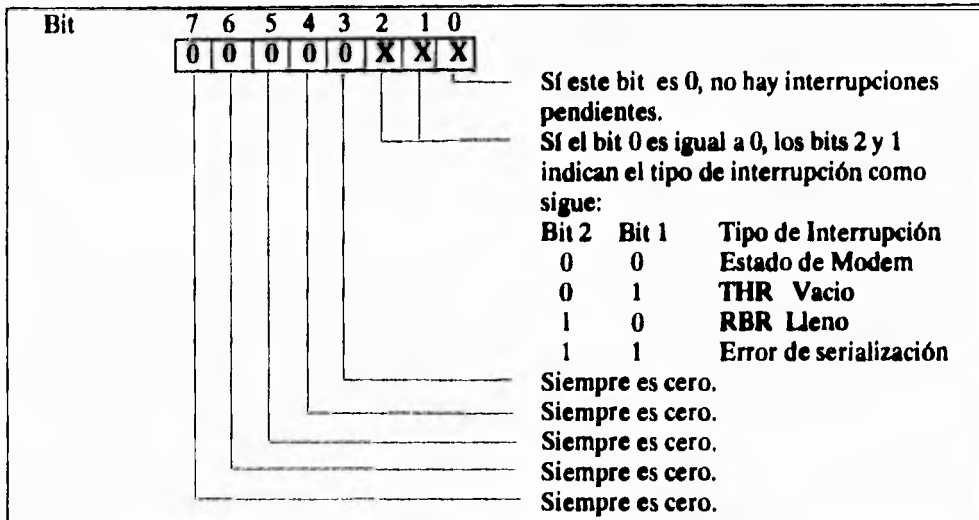


Figura 19. Registro Identificador de Interrupciones (IIR) del UART 8250.

**El Registro 3 (LCR).** Se define como registro de control de línea, en el se definen el número de bits de datos, el número de bits de paro y el tipo de paridad. Este puede direccionarse para su lectura y escritura con la dirección del puerto serie más tres. Para COM1, el LCR puede ser leído o escrito para buscar o enviar un valor con la dirección 3FBH. La Figura 20 muestra las funciones definidas en el LCR.

**El Registro 4 (MCR).** Se define como registro de control de MODEM, en el se definen estados de las líneas de comunicación con el RS232, como la línea DTR, la RTS y las interrupciones para habilitar el UART. Este puede direccionarse para su lectura y escritura con la dirección del puerto serie más cuatro. Para COM1, el MCR puede ser leído o escrito para buscar o enviar un valor con la dirección 3FCH. La Figura 21 muestra las funciones definidas en el MCR.

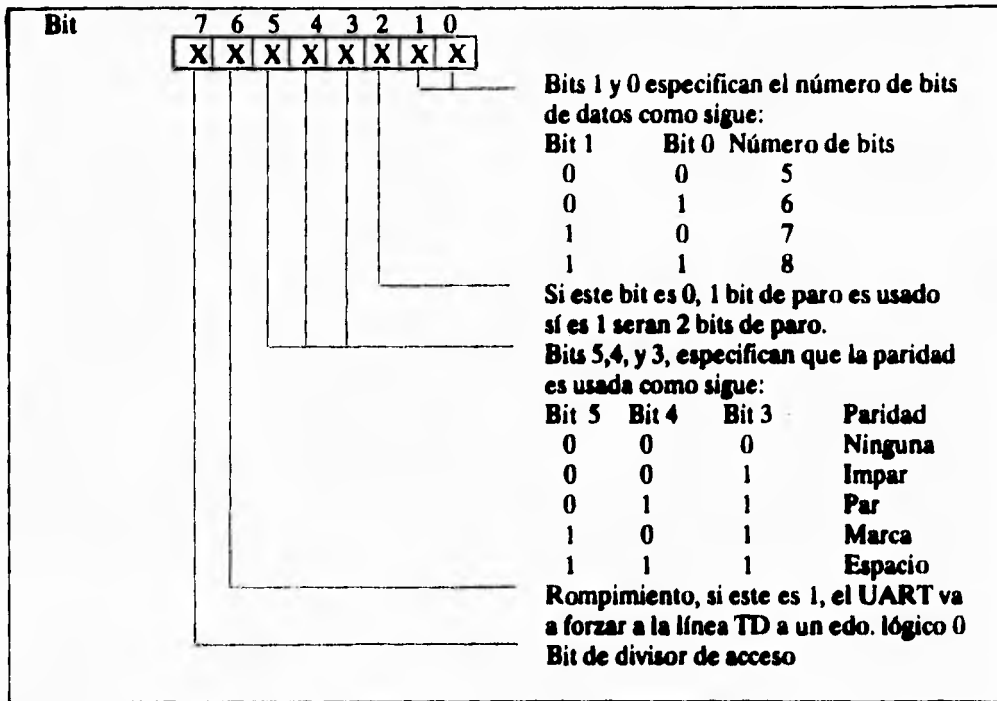


Figura 20. Registro de control de Línea (LCR) del UART 8250.

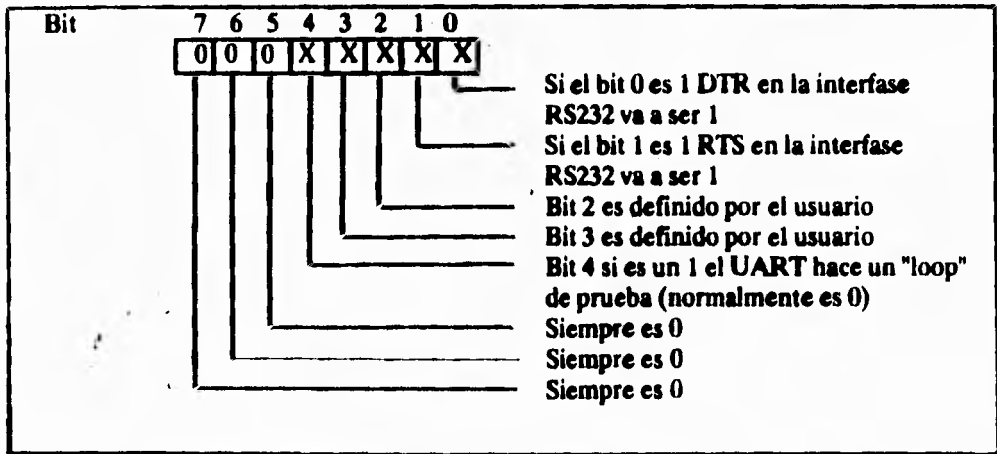


Figura 21. Registro de Control de MODEM (MCR) del UART 8250.

**El Registro 5 (LSR).** Se define como registro de estado de línea, se utiliza para reportar si se realizó correctamente la recepción, la transmisión o si existió algún error. Este puede direccionarse para su lectura con la dirección del puerto serie más cinco. Para COM1, el LCR puede ser leído o escrito para buscar o enviar un valor con la dirección 3FDH. La Figura 22 muestra las funciones definidas en el LSR.

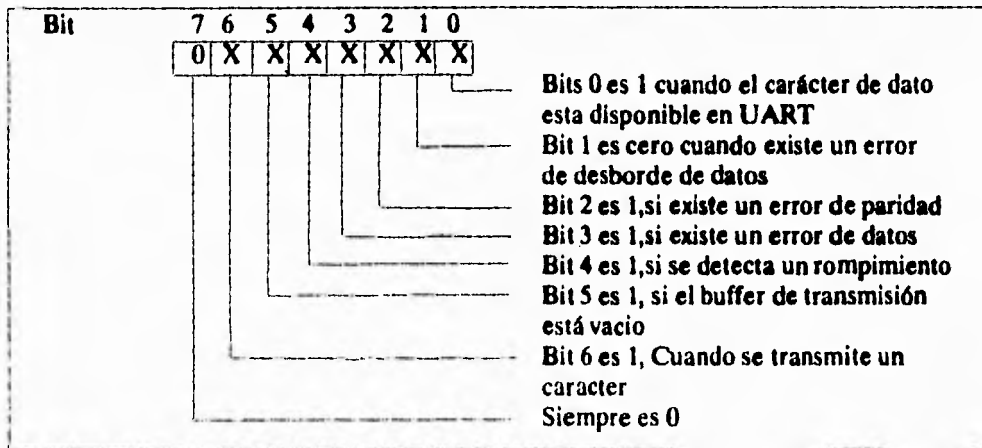


Figura 22. Registro de estado de Línea (LSR) del UART 8250.

Los Registros 6, 8 y 9, son utilizados preferentemente para la comunicación a través de modem's y no son utilizados para la red de monitoreo, por lo cual no es necesario definirlos. El registro 7 no es utilizado por el UART, y no existe en todas las versiones del 8250.

Otra parte importante para la comunicación a través del puerto serie de la PC son las cuatro rutinas que la ROM BIOS ofrece para la comunicación serial, estas son:

- Inicialización de el puerto serie.
- Escritura de un carácter en el puerto serie



- Lectura de un caracter del puerto serie, y
- Estado del puerto serie

Como todas las otras rutinas de la ROM BIOS, la comunicación serial llama a una interrupción al software. En este caso, el llamado a la comunicación serial se realiza por medio de la interrupción 14H. Dependiendo de la rutina en cuestión, el registro del CPU es cargado con ciertos parámetros antes de hacer el llamado a la interrupción. Después de completar la tarea, la rutina de comunicación regresa valores en el registro del CPU.

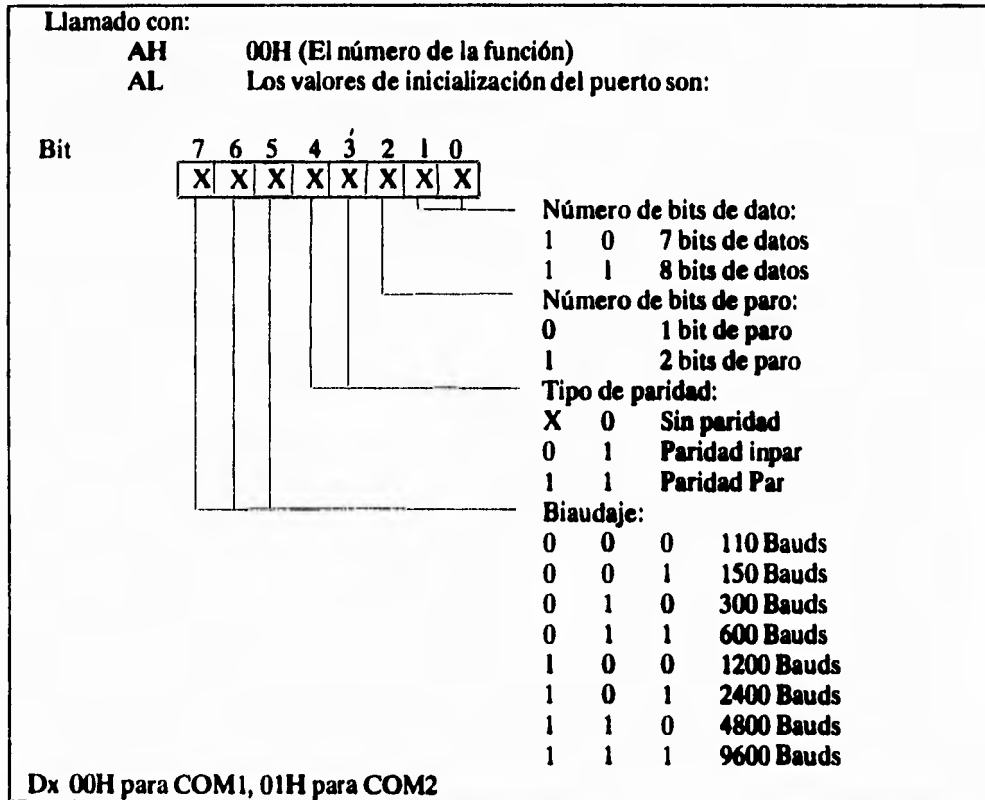


Figura 23a. Interrupción 14H, función 00H, inicialización del puerto serie.

La primera rutina de la ROM BIOS inicializa el puerto serial, en la figura 23a y 23b se muestra como ésta realiza el llamado y cualquier valor que es regresado. Los valores que se definen en el puerto serial son el baudaje, el número de bits de datos, tipo de paridad y el número de bits de paro y regresa el estado corriente del puerto.

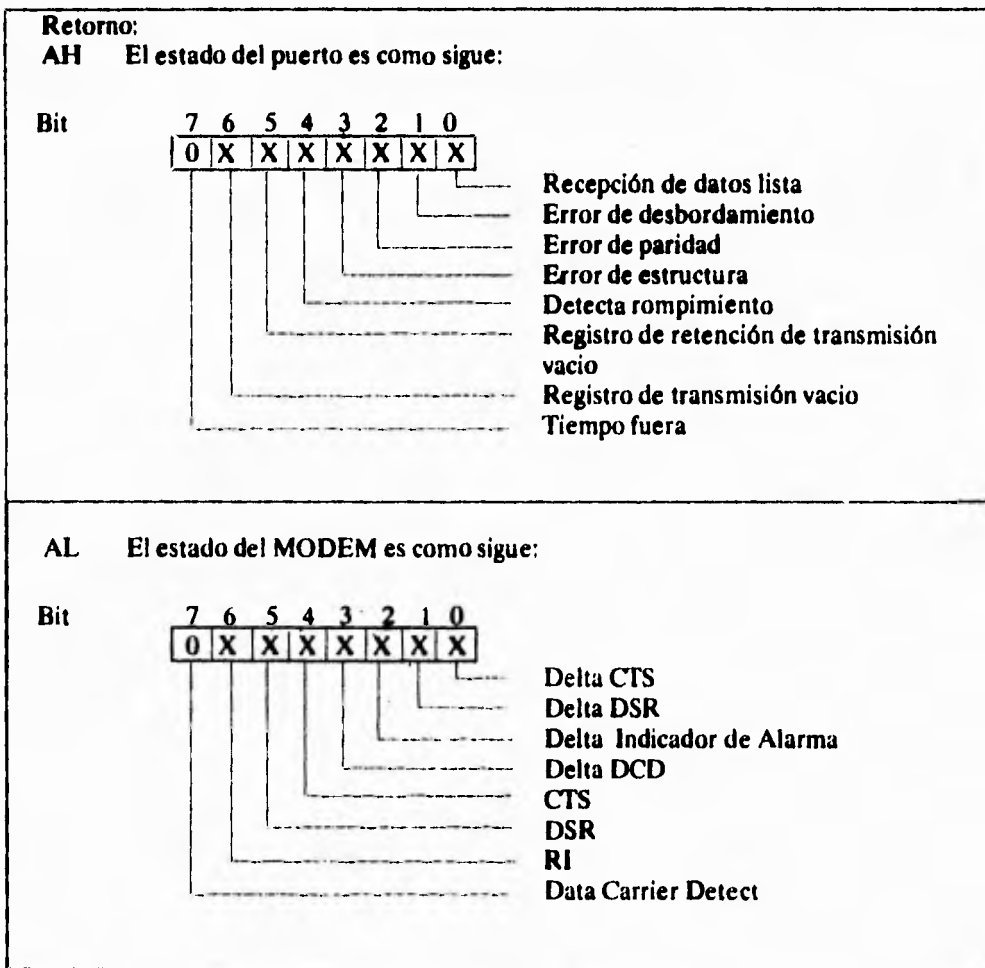


Figura 23b Interrupción 14H, función 00H (continuación).

La segunda rutina de la comunicación serial de la ROM BIOS es la de escritura de un caracter al puerto serie. Esta se realiza mediante el envío desde la ROM BIOS de un caracter fuera del puerto especificado y el regreso su estado corriente. Esto se observa en la figura 24.

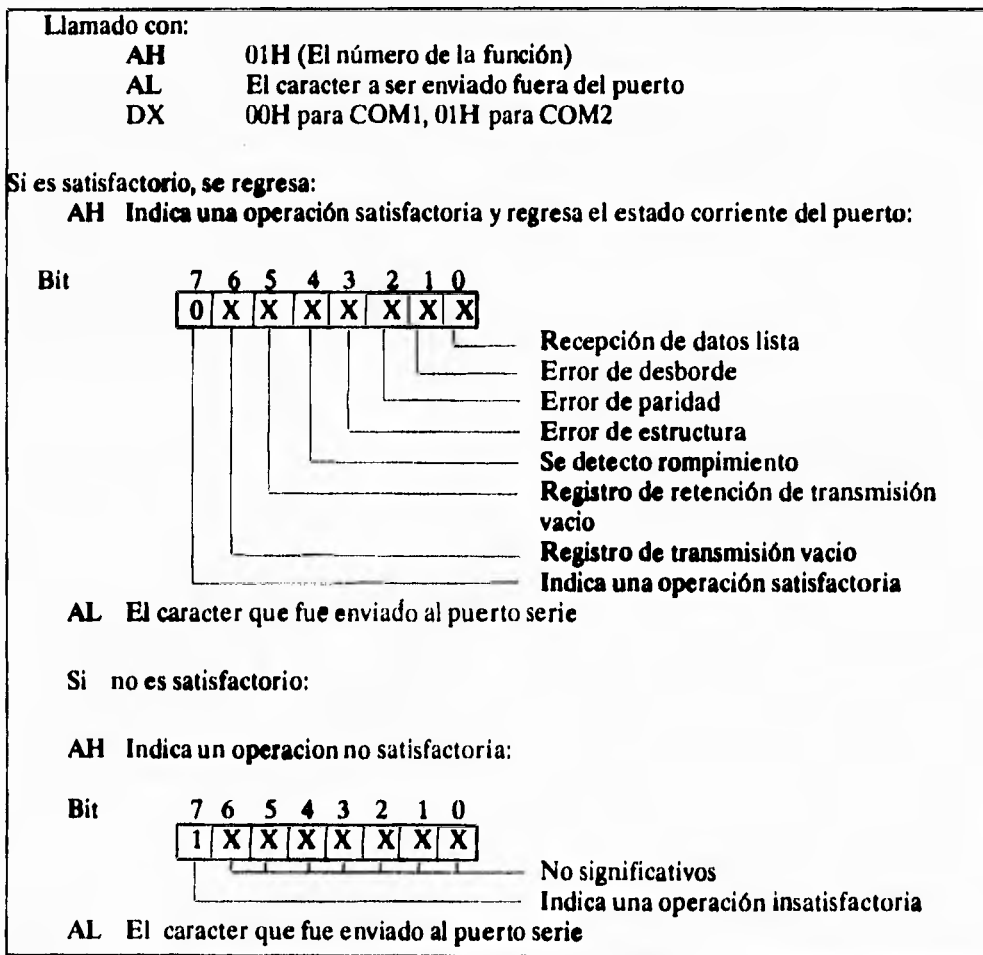


Figura 24 Interrupción 14H, función 01H.

La tercera rutina de comunicación de la ROM BIOS es la de lectura de un caracter desde el puerto serie especificado, esta rutina indica cualquier error que haya ocurrido. La Figura 25 indica como es llamada esta rutina y sus valores de retorno.

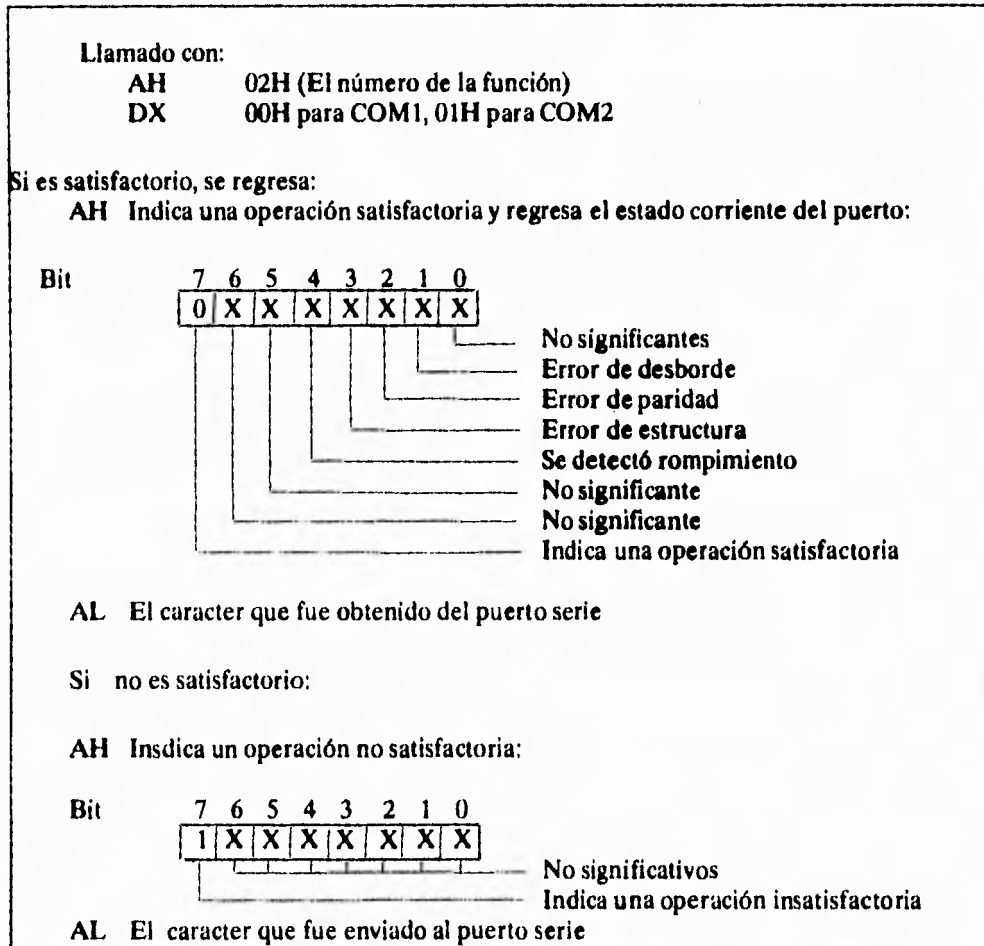


Figura 25 Interrupción 14H, función 02H.

La cuarta y última rutina de comunicación serial es la obtención del estado del puerto serie. La Figura 26 muestra como esta rutina es llamada y los valores que regresa.

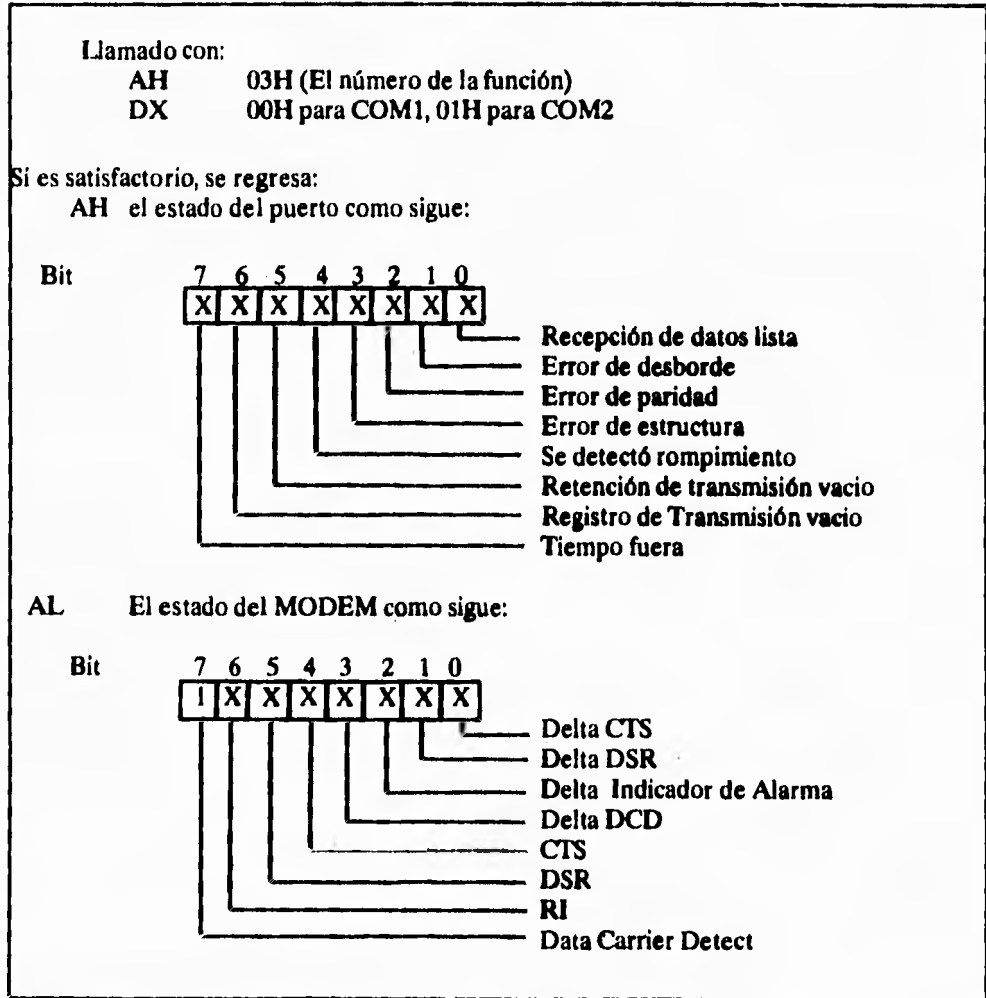


Figura 26 Interrupción 14H, función 02H.

Una vez definidos estos conceptos básicos, se aplicaran modificando la libreria "SERIAL.H" que esta contenida en las rutinas de comunicación en C++ de Microsoft (esta es independiente del software de Borland C++ y viene incluida en un disco de 5 1/4 que acompaña al libro "Serial Communications in C and C++" de Mark Goodwin). Las rutinas que utiliza "SERIAL.H" son definidas a traves de "TSERIAL.LIB" y estan codificadas en lenguaje ensamblador en el programa "SERASM.ASM".

Este último programa esta diseñado para la comunicación serial utilizando todas las señales de los conectores DB25 y DB9. En el caso del "Sistema en red de monitoreo de signos vitales" por razones de optimización no son utilizadas todas estas señales. Para evitar problemas y mensajes de error todas las instrucciones que contengan señales que no sean utilizadas deberan ser eliminadas.

Un ejemplo de la forma en que se realiza esto se muestra a continuación donde se observa la codificación de la instrucción "get\_serial"; una vez que ya han sido eliminadas todas las partes de la instrucción que hacen referencia a cualquier terminal del DB9/DB25 y se incluyen la inserción de un (-1) cuando no existen datos en el buffer (todos estos cambios son marcados con las letras LFC). Las instrucciones que llevan un punto y coma al principio (;) no son ejecutadas.

```

_get_serial proc far
    push    di                ;salva di
    push    es                ;salva es
    mov     ax,-1            ;sin datos en el buffer (LFC)
    push    ax                ;salva este en el stack
    mov     bx,_sibuff        ;siguiente caracter en el ptr
    cmp     bx,_eibuff        ;buffer vacio? (LFC)
    je      gs4              ;va al final (LFC)
    pop     ax                ;elimina el valor de retorno (LFC)
    les     di,_inbuff        ;es:di = input buffer ptr
    mov     al,es:[di][bx]    ;al = next character
    xor     ah,ah            ;ax = next character
    push    ax                ;save it on stack
    inc     bx                ;bump the next character ptr
    cmp     bx,_ilen         ;wrap ptr?
    jne     gs1              ;jump if not
    xor     bx,bx            ;point it to start of buffer gs1:

```

```

mov     sibuff,bx      ;save the new ptr
cmp     _rx_flow,TRUE ;recevive flow?
jne     gs4           ;jump if not
call    chars_in_buff ;ax = number of chars in buffer
cmp     ax_fon        ;turn back on?
jg      gs4           ;jump if not
mov     _rx_flow,FALSE ;flag receive flow off
;cmp    _rx_rts,TRUE  ;RTS/CTS? Esta instrucción se inhibe (LFC)
;jne    gs2           ;jump if not (LFC)
mov     dx,_base      ;dx = base ptr
add     dx,MCR        ;dx = modem control register
in      al,dx         ;get current value
or      al,RTS        ;assert RTS
out     dx,al         ;send new value to UART
gs2:
;cmp    _rx_dtr,TRUE  ;DTR/DSR? (LFC)
;jne    gs3           ;jump if not (LFC)
mov     dx,_base      ;dx = base ptr
add     dx,MCR        ;dx = modem control register
in      al,dx         ;get current value
or      al,DTR        ;assert DTR
out     dx,al         ;send new value to UART
gs3:
cmp     _rx_xon,TRUE  ;XON/XOFF?
jne     gs4           ;Jump if not
cli     ;disable the interrupts
mov     dx,_base      ;dx = xmit register
mov     al,XON        ;al = XON value
out     dx,al         ;send it to remote
sti     ;enable the interrupts
gs4:
pop     ax            ;get the character
pop     es            ;restore es
pop     di            ;restore di
ret     ;return _get_serial
endp

```

Ya que se realizaron las adaptaciones necesarias al programa SERASM.ASM para que no existan discrepancias con el sistema se genera la librería TSERIAL.LIB, esto se logra dando las siguientes instrucciones dentro del subdirectorio de borlandc\bin:

```

> tasm /mx serasm
> tcc -ml -c serial.c protocol.c ansi.c ansicpp.cpp
> tlib tserial.lib +serasm +serial +protocol +ansi +ansicpp

```

Una vez que se cuentan con todos los elementos necesarios, para la comunicación en C++, se puede compilar el programa de comunicaciones que se muestra a continuación:

```
#include <Stdio.h>
#include <Stdlib.h>
#include <Dos.h>
#include "serial.h"

#define PORT 1          /* serial port */
#define BAUDRATE 4800  /* baud rate*/

void main(void)
{
    int c;
    int d;
    int e;
    int f;
    int i,ii;
    int z;
    int chars;

    c = 0xff;
    d = 0x28;

    open_port(PORT, 1024);
    set_port(BAUDRATE, 8, EVEN_PARITY, 1);

    for (ii = 0; ii < 10; ii++)
    {
        outportb(0x3f8, c);
        delay(5);
        outportb(0x3f8, d);
        delay(5);
        c = get_serial();
        delay(5);
        f = get_serial();
        printf("Lectura en canal analógico No. %d",ii);
        printf(" = %d\n",e);
        printf("Estado de los actuadores %d\n",f);
        delay(10);
        d = d + 1;
    }
    close_port();
}
```



Como en todo programa escrito en lenguaje C++, primero se definen las librerías que son utilizadas para el programa con la instrucción `#include`, a continuación posteriormente se definen las constantes necesarias, en este caso se definen el puerto de comunicaciones y el baudaje. Posterior a todas las definiciones viene lo que es el programa de comunicaciones.

Como primer paso se abre el puerto de comunicaciones y se asigna el protocolo con las funciones `"open_port"` y `"set_port"`. Posteriormente se activa el nodo con el cual se requiere la comunicación, en este caso se define la dirección 128 decimal que equivale a la dirección 000 0000 binario ya que como se vio anteriormente el octavo bit del byte es utilizado por el UART para validar la dirección por lo que siempre en este caso deberá ser 1, el segundo byte es el de comando que es utilizado por el nodo para activar los actuadores y las entradas analógicas, para enviar estos dos bytes se utiliza la función `put_serial( )`, que está definida para enviar un caracter de datos.

Una vez activado el nodo, debemos obtener los datos para procesarlos, esto se realiza con la función `get_serial()`, esta función lee un caracter del puerto serie. Como el nodo envía dos bytes de datos se deben leer ambos a fin de evitar errores en la comunicación, aunque existan aplicaciones en las que solo se utilice el primer dato. Y finalmente se cierra el puerto con la función `close_port()`.

La forma en que se compila este programa no es directa, si no se tiene que realizar con la instrucción `BCC` incluida en el compilador de Borlandc, dandola de la siguiente forma para que incluya la biblioteca que fue generada:

```
> BCC -ml NAME.C TSERIAL.LIB
```

La comunicación con el puerto serie de la PC mediante el lenguaje de programación C++ no presenta mayores dificultades y debido a la potencia de C++ en lo que se refiere a velocidad de proceso y facilidades para la programación, lo hace el lenguaje más óptimo para realizar este tipo de aplicaciones.

## **Capitulo 4**

---

# **SISTEMA EN RED DE ADQUISICION Y TRANSMISION DE DATOS**

Un nodo del sistema en red de adquisición y transmisión de datos se esquematiza en la figura 1, este se conforma, de una interfase RS-232 (a través de un circuito integrado MAX232), que se conecta a la computadora personal; un convertidor analógico digital (ADC0809), que recibe las señales analógicas de los dispositivos sensores (electrocardiógrafos, medidores de frecuencia cardiaca, termómetros, etc.); un dispositivo ("latch" MC14099), que mantiene las señales de control para los actuadores (motores, alarmas sonoras, válvulas, etc.); y un UART (MC14469), que maneja el control de los actuadores y la colección de datos de los sensores, además de la comunicación de estos con la interfase RS-232. Los actuadores son retroalimentados a las entradas digitales del UART con el fin de que el programa de la computadora determine el estado en el que se encuentran.

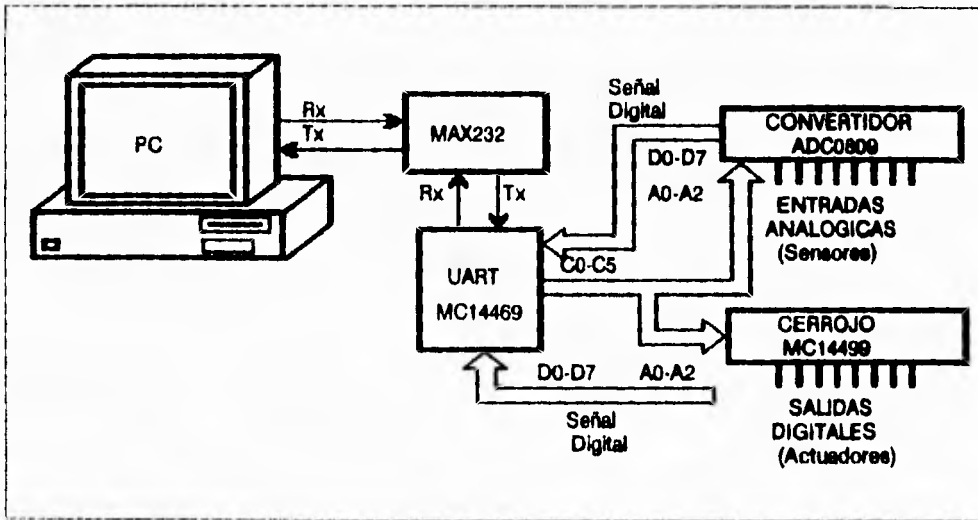


Figura 1. Diagrama esquemático de la Red de monitorco .

El sistema de adquisición y transmisión de datos, es la parte central de la red, gracias a él es posible la comunicación entre la computadora y los diversos dispositivos sensores y actuadores. Este es conectado a cualquier tipo de computadora personal que cuente con un

puerto serie de comunicaciones, permitiendo además un gran número de aplicaciones, siendo el sistema de monitoreo de signos vitales tan sólo un ejemplo de éstas. Cuenta con ocho canales de entrada analógicos y ocho salidas digitales. Es posible seleccionar entre dos frecuencias de transmisión-recepción (4800 y 2400 baudios) a través de un microinterruptor. La red de adquisición y transmisión de datos permite la operación de más de un nodo conectados en cascada, controlando cada uno en forma independiente mediante la utilización de una dirección (propia al nodo) de siete bits.

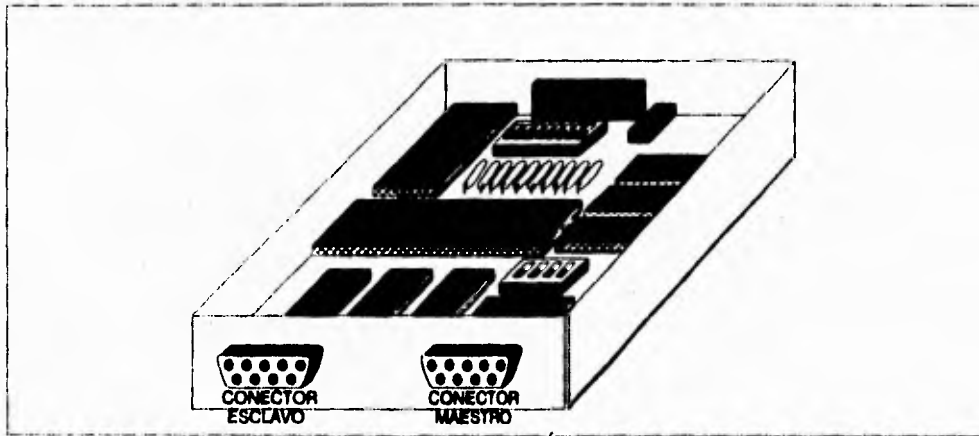


Figura 2. Esquema de las líneas de comunicación del sistema.

### 3.1 Conexión entre el sistema y la PC

En la figura 2 se muestra un esquema de los dos conectores tipo DB9 del sistema de adquisición y transmisión de datos, cada uno cuenta con tres líneas de comunicación, transmisión (Tx), recepción (Rx) y tierra (GND). Estas tres líneas se conectan a la PC por medio de otro conector DB9 o un DB25 (dependiendo del tipo de computadora que se utilice XT o AT), hacia el puerto serial.

Uno de los conectores DB9 recibe el nombre de maestro, ya que a través de éste se controla las señales propias del nodo. El otro conector llamado esclavo es utilizado para controlar las señales que se utilizan para comunicar otros nodos. Esto significa que si se

tiene un sólo nodo el control de sus señales sería únicamente a través del conector maestro. En el caso de haber más de un nodo, el primero se controla a través del conector maestro y el conector esclavo correspondiente se conecta al maestro del siguiente nodo, y así sucesivamente hasta llegar al último nodo. Por ejemplo en el sistema de la figura 3 se muestran tres nodos, la conexión de la PC entra por el conector maestro de TS1 (conectores TXM, GNDM y RXM), el conector esclavo (conectores TXE, GNDE y RXE), de éste se conecta al maestro de TS2 y así sucesivamente hasta un número "n" de nodos.

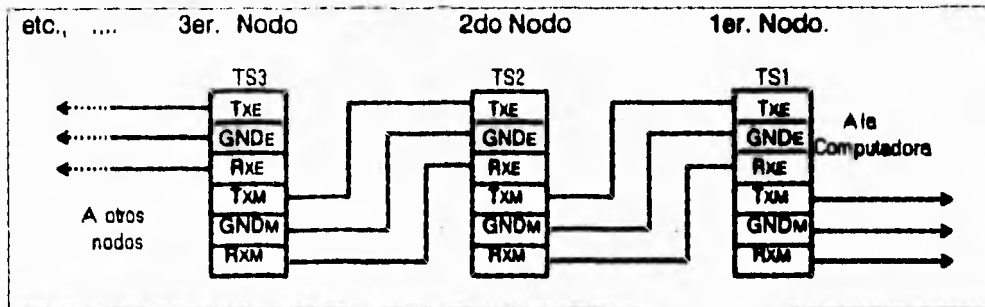


Figura 3. Conexión en cascada de varios nodos.

Para iniciar la comunicación de la computadora con el sistema, el programa en la PC debe de enviar dos palabras (bytes), la primera contiene la dirección del nodo que se requiere habilitar, la segunda contiene el comando que lleva definida la dirección de entrada analógica y/o el canal del actuador del cual (o de los cuales) se requiere tener el control (ver figura 4a).

Cuando es habilitado el sistema de transmisión y recepción con estas dos palabras, este contesta enviando dos palabras hacia la PC, la primera contiene la información adquirida por el canal analógico en operación y la segunda, que no es utilizada en todos los casos, se dá el estado los dispositivos sensores (ver figura 4b), esto se puede ver con mas detalle en la parte de este capítulo donde se trata la comunicación del sistema hacia la PC .



Dir	Hex	3			6			
	Bin	0	1	1	1	0	1	0
Correspon.		A6	A5	A4	A3	A2	A1	A0
Microinter		2	3	4	5	6	7	8

Esta correspondencia indica que los microinterruptores marcados con los números 2, 6 y 8 están cerrados (ya que estos corresponden a los bits A6, A2 y A0), los marcados con los números 3, 4, 5 y 7 están abiertos (A5, A4, A3, A1).

Con siete bits de datos es posible direccionar hasta 128 nodos, siendo éste tan sólo un límite teórico, ya que cuando existe gran cantidad de nodos en la red el tiempo de respuesta va aumentando, por lo que el muestreo de datos es menos frecuente. Debido a lo anterior el número de nodos utilizados está directamente en función de la aplicación.

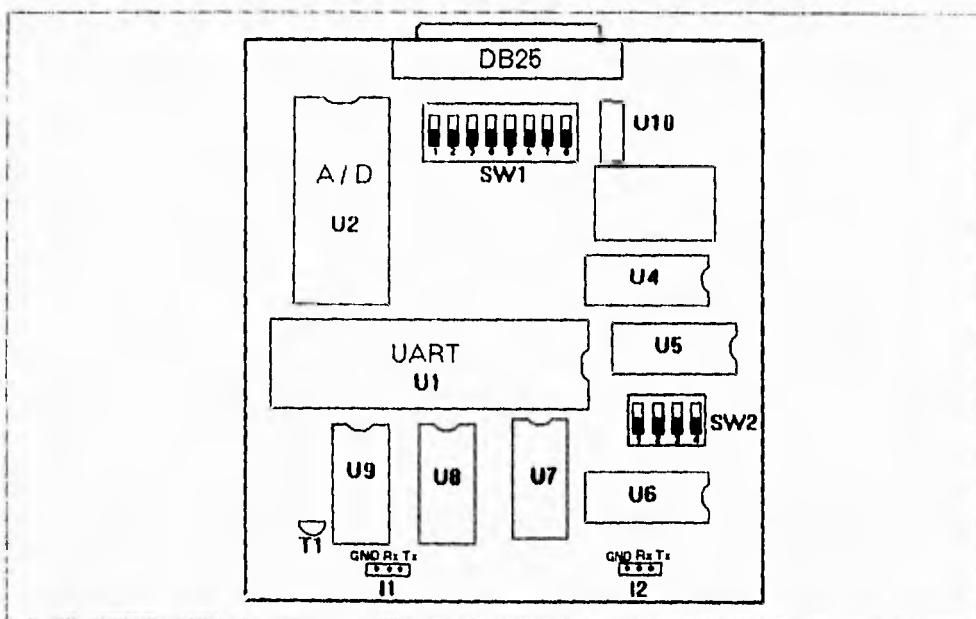


Figura 5. Ubicación de los componentes en la tarjeta.

La dirección debe ser manejada por el programa de la PC cada vez que requiera comunicarse con el sistema para adquirir datos o controlar los actuadores asociados al nodo; el formato de ésta para el programa se observa en la figura 4a, y debe cumplir con el siguiente protocolo:

Parámetro	Descripción	Valor
Velocidad	Tipo de baudaje (bits por seg.)	2400, 4800
Paridad	Tipo de chequeo de paridad	Par
Bits de Datos	Número de bits utilizados para dir.	7 (A0-A6)
Bits de Paro	Número de bits que marcan el fin de dat.	1
Bits de inicio	Número de bits que marcan el inic. de dat.	1
Bits Indicad.	Número de bits indicadores de dirección	1

El programa utiliza un byte para definir la dirección, el octavo bit de éste que no es considerado en el microinterruptor, es utilizado por el programa para indicar al sistema que se trata de una dirección cuando esta encendido, por lo que a la definida en el microinterruptor se le debe sumar 128 (1000 0000 binario) al momento de programar, por ejemplo: si en los microinterruptores esta definida la dirección 30 (011 0000 binario), el byte enviado por el programa sera 158 (1011 0000 binario). Más adelante se presenta esto en un programa realizado en lenguaje BASIC.

#### 4.3 Selección de la velocidad de recepción/transmisión

Otro elemento que es importante considerar en la transmisión y recepción de datos con la PC es la velocidad (Baudaje) a la que los datos están viajando. El sistema de adquisición y transmisión tiene la posibilidad de manejar datos a 2400 y 4800 baudios. Para lo cual se utilizan los microinterruptores SW2 de la figura 5, el marcado con el número 1 es para la velocidad de 2400 y el marcado con el número 2 es para la velocidad de 4800, es importante mencionar, que sólo uno de estos dos interruptores se encuentra en estado de encendido, de lo contrario no se generarán las velocidades requeridas en la comunicación y se podría dañar alguno de los circuitos integrados. Los microinterruptores de SW2 marcados con los números 3 y 4 no se utilizan.



#### 4.4 Canales de entrada analógicos y salidas digitales

Se utilizan ocho canales de entrada analógicos y ocho salidas digitales con nivel TTL, los cuales son direccionados por tres bits (A0, A1 y A2), éstos se definen dentro de la segunda palabra que es transmitida desde la PC (palabra de comando), inmediatamente después de enviar la dirección. Esta palabra como se observa en la figura 4a tiene las mismas características que la dirección, esto es, siete bits de comando (C0-C6) y uno como identificador, con la única diferencia de que el bit más significativo de la palabra siempre debe ser "0" lógico. La dirección de los canales de entrada analógicos y de las salidas digitales se define en los tres primeros bits (C0, C1 y C2), el cuarto bit (C3) indica cuando se debe de activar el canal analógico direccionado para la adquisición de datos (estando con un valor de "1" lógico). El quinto bit (C4) indica la acción a realizar por los actuadores ("1" habilitación y "0" deshabilitación). El sexto bit (C5) indica que la acción especificada en el bit anterior se realiza (con un valor de "1" lógico). Lo cual se resume en la tabla 1.

BIT	Comando	FUNCION
0	C0	AO (Direccion del actuador y/o canal analógico)
1	C1	A1 (Direccion del actuador y/o canal analógico)
2	C2	A2 (Direccion del actuador y/o canal analógico)
3	C3	OE Activa la adquisición de datos Analógicos
4	C4	D Acción a realizar por los actuadores
5	C5	WD Permite la acción especificada en el bit anterior
6	C6	No es utilizado
7	C7	Bit indicador de comando (Siempre debe de ser uno)

Tabla 1. Utilización de los bits de comando.

A continuación se muestra un ejemplo de la utilización de la palabra de comando. Para el comando 45 (2A hexadecimal = 010 1101 binario) cuyos bits corresponden de esta forma:

Com	Hex	2			A			
	Bin	0	1	1	1	0	1	0
Correspon.		C6	C5	C4	C3	C2	C1	C0
Función		---	WD	D	OE	A0	A1	A2

Este comando indica lo siguiente:

- C2 = "1", C1 = "0", C0 = "1" indican la dirección del canal analógico y/o del actuador que se desea habilitar.
- C3 = "1" Se realiza una "lectura" del canal analógico seleccionado.
- C4 = "0" Se deshabilita el actuador seleccionado (la acción indicada por este bit, para que se realice, depende de que el bit C5 esté encendido, de no ser así no se lleva a cabo).
- C5 = "1" indica que efectivamente se realiza la acción señalada por el bit anterior.
- C6 = no se utiliza.

#### 4.5 Programación de la PC para la comunicación con el nodo.

Un ejemplo de un programa en BASIC para activar el nodo se muestra a continuación.

```
50 REM *****
51REM*  DEFINICION DE CONSTANTES  *
53REM*****
100  DIREC = 128    /* Dirección */
200  COMANDO=8     /* Comando */
250 REM *****
251REM*  APERTURA DEL PUERTO DE COM. *
252REM*****
300  OPEN"COM1: 4800,E,8,1" AS #1
400  WHILE B$=""   /* CON ESTA RUTINA */
500  B$=INPUT$(1,#1) /* SE LIMPIA EL BUFER*/
600  WEND         /* DE COMUNICACIONES */
650 REM *****
651REM* SE HABILITA EL NODO Y SE INDICA*
652REM*   ACCION A REALIZAR          *
652REM*****
700  PRINT #1,CHAR$(DIREC) /* DIRECCION */
800  PRINT #1,CHAR$(COMANDO) /* COMANDO */
850 REM *****
851REM*   SE LEEN LAS DOS PALABRAS  *
852REM*   ENVIADAS POR EL NODO     *
852REM*****
900  B1=INPUT(1,#1) /* SE LEE LA PAL. 1*/
1000 B2=INPUT(1#1) /* SE LEE LA PAL. 2 */
1100 C1=ASC$(B1)   /* SE CAMBIA A COD. ASCII*/
1200 C2=ASC$(B2)
1250 REM *****
1251REM* SE IMPRIMEN LOS DOS CARAC.  *
1252REM*****
1300 PRINT "DATO1=";C1,"DATO2=";C2
1400 CLOSE #1
1500 END
```

En la variable **DIREC** se define la dirección del sistema de adquisición y transmisión de datos como 128, esto en binario es 1000 0000, como el bit más significativo es "1", indica al nodo que efectivamente se trata de una dirección, los siete microinterruptores dentro del sistema estarán todos en ceros. El byte de comando es un 8 equivalente en binario a 0000 1000, con esto se indica al nodo que el canal analógico 000 es seleccionado (los tres primeros bits están en "0" y el bit 4 está en "1"), por lo que la señal existente en éste será transmitida hacia la PC. El actuador con la dirección 000 también se selecciona, pero no se toma ninguna acción ya que el bit 6 se encuentra en "0".

Como primera acción el programa abre el puerto de comunicaciones serial **COM1** con la instrucción "OPEN COM". En ésta se define la velocidad en baudios a la que se realiza la comunicación, el tipo de paridad (E = Par), la longitud en bits de la palabra (8 bits) transmitida y recibida, así como el número de bits de paro (1).

Una vez abierto el puerto se limpia el "buffer", para después enviar los dos bytes para iniciar la comunicación, con la instrucción **PRINT**. Al recibir estos el nodo realiza la adquisición en el canal analógico seleccionado y envía instantáneamente dos bytes con la información correspondiente. El programa los lee (con la instrucción **B1 = INPUT(1, #1)**) y los convierte en código ASCII para posteriormente ser desplegados. Con estos elementos definidos dentro del programa es posible la adquisición de datos de los canales analógicos.

Cabe mencionar que este programa es sólo un ejemplo para la adquisición de datos, por lo tanto lo que se realice con ellos dependerá de la aplicación y de las necesidades del usuario.

**NOTA:** En algunas versiones de basic, como las de "Microsoft", la instrucción de **OPEN** no acepta el protocolo de ocho bits de datos y uno de paridad. La versión utilizada de **BASIC** para el programa anterior fue la 3.1 de **XEROX**.



Las entradas digitales son utilizadas como indicadores para probar el correcto funcionamiento de los actuadores y prevenir algún error o desperfecto interno en el sistema de adquisición y transmisión de datos. Este diagnóstico se realiza utilizando un conector DB25 preparado especialmente para ello, cuya característica principal radica en que cada una de las terminales correspondientes a las salidas digitales se conectan a cada una de las entradas digitales.

Para diagnosticar esto con un programa de computadora debe considerarse que las señales transmitidas a las terminales de los actuadores serán retransmitidas a través de las entradas digitales hacia la PC, por lo que la segunda palabra recibida dará el estado de cada uno de los dispositivos actuadores.

#### **4.7 Transmisión del sistema a la PC.**

Una vez establecido el contacto de la PC con el sistema, mediante el envío de la dirección y el comando, el sistema de adquisición y transmisión de datos envía dos palabras de ocho bits (Figura 4b), la primera es el dato muestreado en la entrada analógica seleccionada y la segunda es el dato existente en la entrada digital. Para el programa de la computadora del sistema de monitoreo de signos vitales el dato importante es el contenido en la primera palabra, el segundo es utilizado para verificar el estado en el que se encuentran los actuadores. Aunque es posible que el programa pueda omitir la lectura de la segunda palabra, para que el segundo dato no se quede en el "buffer" se lee para liberar el espacio. En general la forma en que se ocupen estos dos bytes depende de la aplicación.

El protocolo es el mismo que se utiliza en la recepción, esto es, un bit de inicio, ocho de datos, uno de paridad y uno de paro, la velocidad está regulada también por el microinterruptor SW2 de la figura 5. La única diferencia en la transmisión de datos radica en que estos bytes no se utilizan bits indicadores, por lo que la cantidad de bits totales de datos es de ocho.

#### 4.8 Especificaciones.

##### Generales:

Descripción: Sistema de adquisición y transmisión de datos.  
Comunicación: Serial - RS232  
Velocidad: Seleccionable entre 2400 y 4800 baudios  
Potencia requerida: 7.5 - 12 V a 100mA  
Peso: 250 gramos  
Tamaño: 10x10x3 cm  
Temperatura de Oper: 0 a 50°C  
Consumo de potencia: 1.2 Watts

##### Entradas Analógicas:

# de canales analógicos: 8, multiplexados  
Rango de voltaje: 0 a +5.0 V  
Resolución: 8 bits  
Precisión:  $\pm 1$  LSB(bit menos significativo)  
Tipo de conversión: Aproximaciones sucesivas.  
Voltaje de referencia: 5.1 V.

##### Salidas Digitales:

# de salidas digitales: 8.  
Rango de voltaje: 0 a +5.0 V  
Nivel lógico de 0: 1 V max  
Nivel lógico de 1: 4.5V a 5.5 V max  
Resolución: 1 bit por salida  
Precisión:  $\pm 1$  LSB(bit menos significativo)  
Voltaje de referencia: 5.1 V.

**Entradas Digitales:**

# de entradas digitales: 8.

Rango de voltaje: 0 a +5.0 V

Nivel lógico de 0: 1 V max

Nivel lógico de 1: 3.5V a 5.5 V max

Resolución: 1 bit por entrada .

Precisión:  $\pm 1$  LSB(bit menos significativo)

Voltaje de referencia: 5.1 V.



## **Capítulo 5**

---

# **DISEÑO DEL SOFTWARE**

## 5.1 Introducción

La compatibilidad en materia de software fué el diseño del sistema operativo MS-DOS que opera en computadoras de diversos fabricantes.

La principal característica de MS-DOS es que independientemente del tipo de microprocesador que posea, un programa podrá funcionar de igual forma en cualquier computador, pero esto perjudica a las computadoras más potentes, pues dicho programa no aprovechará todas las características de estas.

Las nuevas tendencias han obligado a incluir en nuestro vocabulario dos conceptos, ambiente gráfico y operativo. Cuando se tiene en la computadora una serie de elementos entre dispositivos, figuras y ventanas, que dan la oportunidad de ver lo que se puede hacer y la forma de hacerlo, se tiene un ambiente gráfico, cuando se sigue una serie de pasos en un pseudo inglés o en el mejor de los casos escoger con el teclado de un menú, las funciones que se quieren realizar, se tiene un ambiente operativo.

Las personas se comunican básicamente mediante un lenguaje verbal y un lenguaje visual. Ambos tipos de comunicación se complementan cuando existe transmisión de información, aunque hay diferencias tanto en los mensajes como en la recepción y procesamiento de esos mensajes. Para la comunicación verbal se emplea el lenguaje articulado, mientras que para comunicarse visualmente el hombre se basa en la información que recibe a través de los ojos y se denomina percepción visual.

Dentro de la percepción visual se pueden distinguir dos niveles: uno que comprende el reconocimiento de formas, colores y luces y otro que se añade al anterior permitiendo relacionar objetos y encontrar su significado.

Cuando se observa una imagen además del proceso fisiológico que hace que la retina del ojo capte los rayos de luz y transmita la información al cerebro, existe un proceso

psicológico que hace que el individuo sea conciente de lo que observa y analice la forma, color, movimiento y la relación con los elementos de la imagen.

El ser humano es capaz de absorber y extraer más información por la vista que por cualquier otro sentido. Somos mucho más rápidos viendo un objeto o su representación gráfica que "traduciendo" las palabras que lo describen para llegar a comprenderlo.

Todo ello para conformar un "paisaje" lo suficientemente tentador como para que nadie pueda resistirse a asomarse a estas ventanas que son la entrada a la informática.

El sistema fué elaborado en C++. ya que su rapidez de compilación es extraordinaria y la eficiencia del código que produce es altísima, además combina elementos de lenguaje de alto nivel con la funcionalidad del lenguaje ensamblador y, sobre todo tiene una característica muy especial y de gran importancia para el sistema de monitoreo de signos vitales que es la manipulación directa de bits, bytes y palabras; de gran importancia para establecer la comunicación entre la computadora y los nodos.

Este lenguaje de programación es un lenguaje estructurado tiene la capacidad de un lenguaje de seccionar y esconder de el resto del programa toda la información y las instrucciones necesarias para llevar a cabo determinada tarea, ya que cada rutina contiene sus propias variables locales (temporales), además hay que seguir una serie de reglas en cuanto la estructura de programar sistemas en C++, estas reglas son: incluir las librerías a utilizar al principio del programa, definiciones de variables, enseguida las variables globales, una función principal llamada main(), siendo la primera función que es llamada cuando comienza la ejecución del sistema y finalmente las funciones del sistema con sus respectivas variables locales.

Estas reglas se siguieron en la elaboración del sistema de monitoreo de signos vitales, pero en cuanto a la forma de programarlo, estructurarlo, depende de la forma de trabajar de cada persona.

El sistema, al inicio de este incluye las librerías de las que posteriormente hará uso, estas librerías se muestran en el segmento siguiente; enseguida está la sección de

declaraciones de las variables, estructuras de archivos a utilizar y la declaración de todas las subrutinas contenidas en el sistema. El orden de las rutinas empleadas es de acuerdo al orden de la pantalla de presentación, es decir, primero están las rutinas de configuración con todos sus módulos, luego las rutinas del módulo de pacientes y así hasta el último módulo que es el de salida del sistema y, solamente se encuentran al final del sistema las rutinas que se utilizan a lo largo de este, como el de sonidos, cambio de tonos de colores, rutinas de mouse, rutinas de elección de opciones, ya que se trata de modo gráfico y no de modo texto, y las formas de trabajo son diferentes.

#### Librerías contenidas en el sistema.

```
#include <dos.h>
#include <dir.h>
#include <float.h>
#include <process.h>
#include <alloc.h>
#include "math.h"
#include "conio.h"
#include "proces.h"
#include "mouse.h"
#include "object.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "string.h"
#include "graphics.h"
#include "ctype.h"
#include "time.h"
#include "serial.h"
#include "gtext.h"
#include <string.h>
#include <bios.h>

#define CR 13
#define BAUDRATE 4800
#define DATOS 14
#define ARCHIVO "DATAVAR.DAT"
#define DATACLAVE 2
#define NOMBREVARIA 15
#define ARCH "PACIENT.DAT"
```

```
#define LONGNOMBRE 40
#define LONGCAMA 3
#define ARCHI "DATACONF.DAT"
#define DATAOPCION 3
#define CO_PROCESSOR_MASK 0x0002
#define ESC 0x1b /* definicion de escape */

#define TRUE 1
#define FALSE 0
#define _MAXINC 32
```

Ahora bien, todo el sistema está desarrollado en rutinas: gráficas, presentación, para ordenar datos para manipulación, efectos de sonido, comunicación, etcétera, esto para facilitar actualizaciones, mejoras, adaptaciones para otras aplicaciones o simplemente para un mejor control de su funcionamiento, y sea fácilmente de identificar la sección o subrutina a modificar sin necesidad de perder tiempo localizando el área donde se encuentra y las rutinas que se modificarán con los cambios a realizar.

La sección anterior describe la estructuración del sistema en forma general, enseguida se describirá en forma particular: al inicio de cada rutina se declaran las variables locales a utilizar, enseguida la sección gráfica; en el segmento siguiente se muestra parte de la rutina de la pantalla de ingreso de pacientes con la imagen de una carpeta, si la rutina contiene manejo de archivos ya sean solo de lectura, escritura o ambas, o comunicación con los nodos se declaran las variables a utilizar para evitar confusión con las variables gráficas y variables locales propias de la rutina, siguiendo la sección donde existe la interacción con el usuario dependiendo de la función a realizar, pidiendo a este que proporcione datos para procesarlos y, finalmente mostrar los resultados de acuerdo con lo proporcionado anteriormente por el usuario y cerrar los archivos si es que la rutina contiene manejo de archivos o cerrar el puerto si es rutina de comunicación.

```
/* inicia subrutina de folder */
```

```
void Folder(void)
{
  closegraph();
  Initialize();
  cleardevice();
  hidemouse();
  int i, ii, x, y, c;

  setcolor(3);
  setrgbpalette(3,0,31,25);
  setcolor(5);
  setrgbpalette(5,3,0,31);

  setcolor(1);
  rectangle(0,0,639,479);
  setfillstyle(SOLID_FILL, BLUF);
  floodfill(1,1,1);

  setcolor(8);
  rectangle(11,71,589,429);
  setfillstyle(SOLID_FILL,8);
  floodfill(12,80,8);
  rectangle(15,430,605,440);
  setfillstyle(SOLID_FILL,8);
  floodfill(16,435,8);
  rectangle(590,81,608,435);
  setfillstyle(SOLID_FILL, 8);
  floodfill(592,85,8);
  rectangle(23,51,395,120);
  setfillstyle(SOLID_FILL, 8);
  floodfill(25,55,8);
  line(20,53,397,53);
  line(19,54,398,54);
  line(18,55,399,55);
  line(17,56,400,56);
  line(16,57,400,57);
  line(15,58,400,58);
  line(14,59,400,59);
  line(13,60,400,60);
  for (i = 0; i < i + +)
  {
    line(10,63 + i, 400 + i, 63 + i);
  }
  for (i = 0; i < i + +)
  {
    line(580,71 + i, 598 + i, 71 + i);
  }
}
```

```
line(609,85,609,448);
for (i = 0; i < 10; i++)
{
    line(11,430+i,610,430+i);
}
line(13,439,611,439);
for (i = 0; i < 10; i++)
{
    line(13+i,440+i,611,440+i);
}

setcolor(7);
line(10,70,10,430);
arc(30,70,90,180,20);

line(30,50,390,50);
arc(30,430,180,270,20);

line(30,450,590,450);

arc(390,60,0,90,10);
arc(410,60,180,270,10);

line(410,70,590,70);
arc(590,90,0,90,20);

arc(600,105,0,60,20);
line(620,105,620,445);

line(610,90,610,430);
arc(590,430,270,360,20);

arc(600,445,270,360,20);

arc(50,445,196,270,20);
line(50,465,600,465);

setcolor(5);
for (i = 0; i < 10; i++)
{
    line(32+i,451+i,618-i,451+i);
}
line(32,452,618,452);
line(33,453,617,453);
line(34,454,616,454);
line(35,455,615,455);
line(36,456,614,456);
line(37,457,613,457);
line(38,458,612,458);
```

```
line(37,459,613,459);  
line(38,460,612,460);  
line(39,461,611,461);  
line(41,462,609,462);  
line(46,464,607,464);
```

Sin embargo, existen otras rutinas que tienen una sola función: ordenar los datos contenidos en los archivos, apagar los actuadores, leer el estado de ellos, mostrar determinados datos, en las que solo existen las variables locales y la función a realizar se cierran, estas rutinas son llamadas por otras rutinas principales y se utilizan para mejorar la manipulación del sistema.

En los siguientes segmentos se presenta la rutina de operación de mouse y uno de los cuales es la rutina de comunicación (apagado de actuadores).

*/\* subrutinas especiales de manejo de iconos \*/*

```
void add_object(unsigned obdtype, char code, int left, int top,  
                int right, int bottom, void (*fcn) ())  
  
    {  
        if (objnum = MAX_OBJECTS)  
            return;  
        objlist[objnum].objtype = obdtype;  
        objlist[objnum].accesscode = code;  
        objlist[objnum].left = left;  
        objlist[objnum].top = top;  
        objlist[objnum].right = right;  
        objlist[objnum].bottom = bottom;  
        objlist[objnum].fcn = fcn;  
        objnum ++ ;  
    }
```



```
void anytoprocess(int c)
{
    int i,x,y;
    /* outtextxy(100,100,"ENTRE ANY TO PROCESS");*/
    if ( c 0)
    {
        for (i=0; i <objnum; i++)
        {
            if (objlist[i].accesscode == c)
            {
                if (objlist[i].objtype == ICON)
                    reverse_icon(&objlist[i]);

                (*objlist[i].fcn) ();
                if (objlist[i].objtype == ICON)
                    reverse_icon(&objlist[i]);
                return;
            }
        }
    }
    else
    if ( c 0)
    {
        getmousecoords(&x, &y);
        for (i=0; i <objnum; i++)
        {
            if (mouseinbx(objlist[i].left, objlist[i].top,
                objlist[i].right, objlist[i].bottom, x, y))
            {
                if ((objlist[i].objtype == ICON) || (objlist[i].objtype == USER))
                    reverse_icon(&objlist[i]);
                (*objlist[i].fcn) ();
                if (objlist[i].objtype == ICON)
                    reverse_icon(&objlist[i]);
                return;
            }
        }
    }
}

void reverse_icon(struct OBJECT *obj)
{
}

void Lee_Icono( int x, int y, int ancho, int largo, int color0, int color1,
```

```
int color2, int color3, int color4, int color5, char *nom_archi)
{
FILE *archivo;
int i,j;
int ancho2, largo2;
int pixel_icono;

if ((archivo = fopen(nom_archi, "r")) == NULL)
{
closegraph();
printf("No se encuentra el archivo: %s\n", nom_archi);

Pause();      exit(1);
}

fscanf(archivo, "%d %d", &ancho2, &largo2);
if (ancho2 != ancho || largo2 != largo)
{
closegraph();
printf("Incompatibilidad de archivo \n");
Pause();
exit(1);
}

for (j = 0; j < largo; j++)
{
for (i = 0; i < ancho; i++)
{
fscanf(archivo, "%s", &pixel_icono);
if (pixel_icono == 0)
{
putpixel(2*i + x, y + j, color0);
putpixel(2*i + 1 + x, y + j, color0);
}
/* else*/
if (pixel_icono == 1)
{
putpixel(2*i + x, y + j, color1);
putpixel(2*i + 1 + x, y + j, color1);
}
/* else*/
if (pixel_icono == 2)
{
putpixel(2*i + x, y + j, color2);
putpixel(2*i + 1 + x, y + j, color2);
}
/* else*/
if (pixel_icono == 3)
{
putpixel(2*i + x, y + j, color3);
putpixel(2*i + 1 + x, y + j, color3);
}
}
}
}
```

```

        if (pixel_licono == 4)
        {
            putpixel(2*i + x, y + j, color4);
            putpixel(2*i + 1 + x, y + j, color4);
        }
        /* else*/
        if (pixel_licono == 5)
        {
            putpixel(2*i + x, y + j, color5);
            putpixel(2*i + 1 + x, y + j, color5);
        }
    }
}
fclose(archivo);
}

/* terminan rutinas de mouse */

/* subrutina de cambio de texto */

void changetextstyle(int font, int direction, int charsize, int color)
{
    int ErrorCode;
    graphresult();
    settxtstyle(font, direction, charsize);
    ErrorCode = graphresult();
    if (ErrorCode != grOk) {
        closegraph();

        exit(1);
    }
}

/* .....*/

/* Subrutina de mouse */

int dedo[32] = {
    0xF3FF, 0xE1FF, 0xE1FF, 0xE1FF, 0xE049, 0xE000, 0X8000,
    0X0000, 0X0000, 0X07FC, 0X07FB, 0X9FF9, 0XBFF1, 0XC003, 0XE007,
    0X0C00, 0X1200, 0X1200, 0X1200, 0X1200, 0X13B6, 0X1249, 0X7249,
    0X9249, 0X9001, 0X9001, 0X8001, 0X4002, 0X4002, 0X2004, 0X1FF8,
};

```

```

int reley[32] = {
    0xPC3F, 0XF00F, 0XE247, 0XCCE73, 0X9E79, 0X0E79, 0X3E7C, 0X1E78,
    0X1F35, 0X3F9D, 0X9FC9, 0X9FF9, 0XCFF3, 0XE247, 0XF00F, 0XP0CF,
    0X0000, 0X03CD, 0X0CB0, 0X1108, 0X2104, 0X2104, 0X4102, 0XC103,
    0XE087, 0X4042, 0X2024, 0X2004, 0X1008, 0X0CB0, 0X03CD, 0X0000,
};

```

```

int mouseexists;
int _cx, _cy;
int _ciac;
/* char */ void * _cursor;
/* char */ void * _undecursor;
int _cursoron;
int _cminx, _cmaxx;
int _cminy, _cmaxy;
int _buttonpress;
int _rbuttonpress;
int _numlpress;
int _numrpress;
int _numlrelease;
int _numrrelease;
int _cdeltax, _cdeltay;

```

```

void mouse(int *m1, int *m2, int *m3, int *m4)

```

```

{
    union REGS inregs, outregs;
    inregs.ax = *m1;
    inregs.bx = *m2;
    inregs.cx = *m3;
    inregs.dx = *m4;
    int86(0x33, &inregs, &outregs);
    *m1 = outregs.ax;
    *m2 = outregs.bx;
    *m3 = outregs.cx;
    *m4 = outregs.dx;
}

```

```

int initmouse(void)

```

```

{
    int gmode;
    char far *memory = (char far *)0x004000049L;

    mouseexists = TRUE;
    if (resetmouse())
    {
        gmode = getgraphmode();
        if (gmode == HERCMONOHI)
        {

```

```
        *memory = 0x06;
        resetmouse();
    }
    Rango_Horizontal(5, getmaxx()-18);
    Rango_Vertical(0, getmaxy()-15);
    movemouse(0,0);
    showmouse(MANO);
    return(1);
}
else
{
    mouseexists = FALSE;
    _cx = _cy = 0;
    _cinc = 1; _cminx = 0;
    _cmaxx = 625;
    _cminy = 0;
    _cmaxy = getmaxy()-4;
    _cdeltax = _cdeltay = 0;
    _lbuttonpress = FALSE;
    _rbuttonpress = FALSE;
    _numlpress = 0;
    _numrpress = 0;
    _numlrelease = 0;
    _numrrelease = 0;
    _initcursor();
    _cursoron = FALSE;
    showmouse(MANO);
    return(0);
}
}

int resetmouse(void)
{
    int m1, m2, m3, m4;

    m1 = RESET_MOUSE;
    mouse(&m1, &m2, &m3, &m4);
    return(m1);
}

void movemouse(int x, int y)
{
    int m1, m2;

    if (!mouseexists)
    {
        hidemouse();
        _cx = x; _cy = y;
        showmouse(MANO);
    }
}
```

```
        _cdeltax = _cdeltay = 0;
        return;
    }
    m1 = SET_MOUSE_COORD;
    if (getmaxc() == 319)
        x *= 2;
    mouse(&m1, &m2, &x, &y);
}

void hidemouse(void)
{
    int m1, m2, m3, m4;

    if (!mouseexists)
        _togglecursor();
    else
    {
        m1 = HIDE_MOUSE;
        mouse(&m1, &m2, &m3, &m4);
    }
}

void showmouse(int tipo_cursor)
{
    int m1, m2, m3, m4;

    if (!mouseexists)
        _togglecursor();
    else
    {
        Haz_Cursor(5, 0, tipo_cursor);
        m1 = SHOW_MOUSE;
        mouse(&m1, &m2, &m3, &m4);
    }
}

void getmousecoords(int *x, int *y)
{
    int m1, m2;
    if (!mouseexists)
    {
        *x = _cx; *y = _cy;
        return;
    }
    m1 = GET_MOUSE_STATUS;
    mouse(&m1, &m2, x, y);
    if (getmaxc() == 319)
        (*x) /= 2;
}

```

```
void getmousemovement(int *x, int *y)
{
    int m1, m2;
    if (!mouseexists)
    {
        *x = _cdeltax; *y = _cdeltay;
        _cdeltax = _cdeltay = 0;
        return;
    }
    m1 = GET_MOUSE_MOVEMENT;
    mouse(&m1, &m2, x, y);
    if (getmaxx() == 319)
        (*x) /= 2;
}

int mousebuttonreleased( int whichbutton)
{
    if (!mouseexists)
        return(_simulatedbuttonreleased(whichbutton));
    return(testbutton(CHECK_BUTTON_RELEASE, whichbutton));
}

int mousebuttonpressed(int whichbutton)
{
    if (!mouseexists)
        return(_simulatedbuttonpressed(whichbutton));
    return(testbutton(CHECK_BUTTON_PRESS, whichbutton));
}

int testbutton(int testtype, int whichbutton)
{
    int m1, m2, m3, m4;
    m1 = testtype;
    if(whichbutton == LEFT_BUTTON || whichbutton == EITHER_BUTTON)
    {
        m2 = LEFT_BUTTON;
        mouse(&m1, &m2, &m3, &m4);
        if (m2)
            return(TRUE);
    }
    if(whichbutton == RIGHT_BUTTON || whichbutton == EITHER_BUTTON)
    {
        m1 = testtype;
        m2 = RIGHT_BUTTON;
        mouse(&m1, &m2, &m3, &m4);
        if (m2)
            return(TRUE);
    }
    return(FALSE);
}
```

```
}  
  
int _simulatedbuttonpressed( int whichbutton)  
{  
    if((whichbutton == LEFT_BUTTON || whichbutton == EITHER_BUTTON) && _numlpress)  
    {  
        _numlpress--;  
        return(TRUE);  
    }  
    else  
    if((whichbutton == RIGHT_BUTTON || whichbutton == EITHER_BUTTON) && _numrpress)  
    {  
        _numrpress--;  
        return(TRUE);  
    }  
    else  
    if(_getkbinteraction(whichbutton) 0)  
    {  
        if (whichbutton == LEFT_BUTTON || whichbutton == EITHER_BUTTON)  
            return(TRUE);  
        else  
            if (whichbutton == RIGHT_BUTTON || whichbutton == EITHER_BUTTON)  
                return(TRUE);  
    }  
}  
  
int _simulatedbuttonreleased( int whichbutton)  
{  
    if((whichbutton == LEFT_BUTTON || whichbutton == EITHER_BUTTON) && _numlrelease)  
    {  
        _numlrelease--;  
        return(TRUE);  
    }  
    else  
    if((whichbutton == RIGHT_BUTTON || whichbutton == EITHER_BUTTON) && _numrrelease)  
    {  
        _numrrelease--;  
        return(TRUE);  
    }  
    else  
    if(_getkbinteraction(whichbutton) 0)  
    {  
        if (whichbutton == LEFT_BUTTON || whichbutton == EITHER_BUTTON)  
            return(TRUE);  
        else  
            if (whichbutton == RIGHT_BUTTON || whichbutton == EITHER_BUTTON)  
                return(TRUE);  
    }  
    return(FALSE);  
}
```



```
    }  
  
int mouseinbow(int left, int top, int right, int bottom, int x, int y)  
{  
    return((x >= left && x <= right && y >= top && y <= bottom) ? 1 : 0);  
}  
  
int getinput(int whichbutton)  
{  
    int c, c2;  
    if(!mouseexists)  
    {  
        c = _getkbinteraction(whichbutton);  
        return(c);  
    }  
    if(kbhit())  
    {  
        c = getch();  
        if(c == 0)  
        {  
            c2 = getch();  
            c2 = c2 < 8;  
            c2 |= c;  
            return(c2);  
        }  
        else  
            return(c);  
    }  
    else  
    {  
        if(mousebuttonpressed(whichbutton))  
        {  
            while(!mousebuttonreleased(whichbutton));  
            return(-1);  
        }  
        else  
            if(mousebuttonreleased(whichbutton))  
                return(-1);  
            return(0);  
    }  
}  
  
int waitforinput(int whichbutton)  
{  
    int c;  
    while((c = getinput(whichbutton)) == 0);  
    return(c);  
}
```

```

int _getkbinteraction( int whichbutton)
{
    int c;
    if (kbhit())
    {
        c = _getkb();
        hidemouse();
        switch(c)
        {
            case 0x5200:
                _lbuttonpress = (_lbuttonpress) ? FALSE : TRUE;
                showmouse(MANO);
                if (whichbutton == LEFT_BUTTON || whichbutton == EITHER_BUTTON)
                    return(-1);
                if (_lbuttonpress)
                    _numlpress ++;
                else
                    _numlrelease ++;
                return(0);

            case 0x5300:
                showmouse(MANO);
                _rbuttonpress = (_rbuttonpress) ? FALSE : TRUE;
                if (whichbutton == LEFT_BUTTON,
                ↓| whichbutton == EITHER_BUTTON)
                    return(-1);
                if (_rbuttonpress)
                    _numrpress ++;
                else
                    _numrrelease ++;
                return(0);

            case 0x002B:
                _cinc = (_cinc == _MAXINC) ? _cinc + 6 : _MAXINC;
                break;

            case 0x002D:
                _cinc = (_cinc != 6) ? _cinc - 6 : 1;
                break;

            case 0x4800:
                if (( _cy == _cinc) & _cminy)
                    _cy = _cminy;
                    _cdeltay = _cinc;
                    break
        }
    }
}

```

```
case 0x5000 :  
  
    if((_cy += _cinc) == _cmay)   
        _cy = _cmay;  
        _cdelay += _cinc;  
        break;  
  
case 0x4B00 :  
  
    if((_cx -= _cinc) == _cminx)   
        _cx = _cminx;  
        _cdeltax += _cinc;  
        break;  
  
case 0x4D00 :  
  
    if((_cx += _cinc) == _cmaxx)   
        _cx = _cmaxx;  
        _cdeltax += _cinc;  
  
        break;  
  
case 0x4700 :  
  
    if((_cy -= _cinc) == _cminy)   
        _cy = _cminy;  
    if((_cx -= _cinc) == _cminx)   
        _cx = _cminx;  
        _cdeltax += _cinc;  
        _cdelay -= _cinc;  
        break;  
        case 0x4900 :  
    if((_cy -= _cinc) == _cminy)   
        _cy = _cminy;  
    if((_cx += _cinc) == _cmaxx)   
        _cx = _cmaxx;  
        _cdeltax += _cinc;  
        _cdelay -= _cinc;  
        break;  
  
case 0x4F00 :  
  
    if((_cy += _cinc) == _cmay)   
        _cy = _cmay;  
    if((_cx -= _cinc) == _cminx)   
        _cx = _cminx;  
        _cdeltax -= _cinc;  
        _cdelay += _cinc;  
        break;
```

```
case Oc5100:
    if ((_cy + = _cinc) == _cmaxy)
        _cy = _cmaxy;
    if ((_cx + = _cinc) == _cmaxx)
        _cx = _cmaxx;
    _cdelay -= _cinc;
    _cdelay += _cinc;
    break;

    default:
        showmouse(MANO);
        return(c);
}
showmouse(MANO);
}
return(0);
}

int _getkb(void)
{
    int ch1, ch2;
    ch1 = getch();
    if (ch1 == 0)
    {
        ch2 = getch();
        ch2 = ch2 < 8;
        ch2 |= ch1;
        return(ch2);
    }
    else
        return(ch1);
}

void _initcursor(void)
{
    _cursor = malloc(imagesize(0,0,CURSWIDTH,CURSHEIGHT));
    _undercursor = malloc(imagesize(0,0,CURSWIDTH,CURSHEIGHT));
    if (_cursor == NULL || _undercursor == NULL)
    {
        closegraph();
        printf("No hay memoria suficiente. \n");
        exit(1);
    }

    getimage(_cx, _cy, _cx + CURSWIDTH, _cy + CURSHEIGHT, _undercursor);
    setlinestyle(SOLID_LINE, 0, 0);
    setfillstyle(SOLID_FILL, BLACK);
    bar(_cx, _cy, _cx + CURSWIDTH, _cy + CURSHEIGHT);
}
```

```
    _drawcursor(_cx, _cy);

    getimage(_cx, _cy, _cx + CURSWIDTH, _cy + CURSHEIGHT, _cursor);
    putimage(_cx, _cy, _undercursor, COPY_PUT);
}

void _togglecursor(void)
{
    struct viewporttype vp;
    int oldx, oldy;

    getviewsettings(&vp);
    oldx = getx();
    oldy = gety();
    setviewport(0,0, getmaxx(), getmaxy(), 1);
    if (_cursoron)
    {
        putimage(_cx, _cy, _undercursor, COPY_PUT);
        _cursoron = FALSE;
    }
    else
    {
        getimage(_cx, _cy, _cx + CURSWIDTH, _cy + CURSHEIGHT, _undercursor);
        putimage(_cx, _cy, _cursor, COPY_PUT);
        _cursoron = TRUE;

        setviewport(vp.left, vp.top, vp.right, vp.bottom, 1);
        moveto(oldx, oldy);
    }
}

void _drawcursor( int x, int y)
{
    setcolor(getmaxcolor());
    line(x + 1, y + 1, x + CURSWIDTH - 1, y + CURSHEIGHT - 1);
    line(x + 2, y + 1, x + CURSWIDTH - 1, y + CURSHEIGHT - 2);
    line(x + 1, y + 2, x + CURSWIDTH - 2, y + CURSHEIGHT - 1);
    line(x + 2, y + 1, x + 2, y + 5);
    line(x + 1, y + 1, x + 1, y + 5);
    line(x + 1, y + 1, x + 5, y + 1);
    line(x + 1, y + 2, x + 5, y + 2);
}
```

```
void Haz_Cursor( int xhot, int yhot, int forma)
{
    union REGS inregs, outregs;

    inregs.x.ax = DEFINE_CURSOR;
    inregs.x.bx = xhot;
    inregs.x.cx = yhot;
    switch(forma)
    {
        case MANO :
            inregs.x.dx = FP_OFF(&dedo[0]);
            _ES = FP_SEG(&dedo[0]);
            break;

        case RELOJ :
            inregs.x.dx = FP_OFF(&reloj[0]);
            _ES = FP_SEG(&reloj[0]);
            break;
    }
    int86(MOUSE, &inregs, &outregs);
}

void Rango_Horizontal( int minx, int maxx)
{
    union REGS inregs, outregs;

    inregs.x.ax = RANGO_HORIZONTAL;
    inregs.x.cx = minx;
    inregs.x.dx = maxx;
    int86(MOUSE, &inregs, &outregs);
}

void Rango_Vertical(int miny, int maxy)
{
    union REGS inregs, outregs;
    inregs.x.ax = RANGO_VERTICAL;
    inregs.x.cx = miny;
    inregs.x.dx = maxy;
    int86(MOUSE, &inregs, &outregs);
}

int gprintfxy(int xloc, int yloc, char *fmt, ...)
{
    va_list argptr;
    char str[BUFSIZE];
    int cnt;
    struct fillsettings type oldfill;
    char userfillpattern[8];
```

```

va_start(argptr, fmt);

cnt = vsprintf(str, fmt, argptr);
if (str[0] == NULL) return (0);

getfillsettings(&oldfill);
if (oldfill.pattern == USER_FILL)
    getfillpattern(userfillpattern);
setfillstyle(SOLID_FILL, getbkcolor());
bar(xloc, yloc, xloc + textwidth(str), yloc + textheight("H")*5/4);
if (oldfill.pattern == USER_FILL)
    setfillpattern(userfillpattern, oldfill.color);
else
    setfillstyle(oldfill.pattern, oldfill.color);
outtextxy(xloc, yloc, str);
va_end(argptr);
return(cnt);
}

int gprintf(char *fmt, ...)
{
va_list argptr;
char str[BUFSIZE];
int cnt;
struct fillsettingstype oldfill;
char userfillpattern[8];
int xloc, yloc;

va_start(argptr, fmt);
cnt = vsprintf(str, fmt, argptr);
if (str[0] == NULL) return(0);
xloc = getx(); yloc = gety();
getfillsettings(&oldfill);
if (oldfill.pattern == USER_FILL)
    getfillpattern(userfillpattern);
setfillstyle(SOLID_FILL, getbkcolor());
bar(xloc, yloc, xloc + textwidth(str), yloc + textheight("H")*5/4);
if (oldfill.pattern == USER_FILL)
    setfillpattern(userfillpattern, oldfill.color);
else
    setfillstyle(oldfill.pattern, oldfill.color);
outtext(str);
va_end(argptr);
return(cnt);
}

int ggetche(void)
{
char ch;
ch = getch();
}

```

```
gprintf("%c", ch);
return(ch);
}

int gputch(int c)
{
char buffer[2];
sprintf(buffer, "%c", c);
gprintf(buffer);
return(c);
}

char *ggets(char *buffer)
{
int currlc, maxchars, oldcolor;
struct viewporttype view;
char ch, charbuff[3];

buffer[0] = '\0';
currlc = 0;
getviewsettings(&view);
maxchars = maxchars1;
if (maxchars == 0) return(NULL);
gprintfxy(getx(), gety(), ".");
while ((ch = getch()) != CR) {
if (ch == BS) {
if (currlc > 0) {
currlc--;
if (currlc == maxchars) {
oldcolor = getcolor();
setcolor(getbkcolor());
sprintf(charbuff, "%c", buffer[currlc]);
gprintfxy(getx()-textwidth(charbuff), gety(), "%c_", buffer[currlc]);
setcolor(oldcolor);
moveto(getx()-textwidth(charbuff), gety());
}
}
else {
if (currlc == maxchars) {
oldcolor = getcolor();
setcolor(getbkcolor());

gprintfxy(getx(), gety(), ".");
setcolor(oldcolor);
buffer[currlc] = ch;
gputch(ch);

currlc++;
}
}
}
}
}
```



```
        else
            putchar(0x07);
        }
        if (currloc == maachars)
            gprintby(getx(), gety(), " ");
    }
    if (currloc == maachars) {
        oldcolor = getcolor();
        setcolor(getbkcolor());
        gprintby(getx(), gety(), " ");
        setcolor(oldcolor);
    }
    buffer[currloc] = '\0';
    return(buffer);
}

int gscanf(char *fmt, ...)
{
    va_list argptr;
    char str[BUFSIZE];
    int cnt;

    va_start(argptr, fmt);
    ggets(str);
    cnt = vscanf(str, fmt, argptr);

    va_end(argptr);
    return(cnt);
}

int gscanfxy(int xloc, int yloc, char *fmt, ...)
{
    va_list argptr;
    char str[BUFSIZE];
    int cnt;
    int olds, oldy;

    olds = xloc;
    oldy = yloc;
    moveto(xloc, yloc);
    va_start(argptr, fmt);
    ggets(str);
    str1 = str;

    cnt = vscanf(str, fmt, argptr);
    va_end(argptr);
    moveto(olds, oldy);
    return(cnt);
}
```

```
/* apagado de actuadores */

void Actoff(void)
{
    int dirmodo, direct, e, edoact, i, lca;
    dirmodo = 0x00;
    direct = 0x28;

    open_port(PORT, 1024);
    set_port(BAUDRATE, EVEN_PARITY, 1);
    for (i = 0; i; ++i)
    {
        for (lca = 0; lca; ++lca)
        {
            outputb(0x3f8, dirmodo);
            delay(5);
            outputb(0x3f8, direct);
            delay(5);
            e = get_serial();
            delay(5);
            edoact = get_serial();
            delay(5);
            outputb(0x3f8, dirmodo);
            delay(5);
            outputb(0x3f8, direct);
            delay(5);
            e = get_serial();
            delay(5);
            edoact = get_serial();
            direct = direct + 1;
        }
        dirmodo = dirmodo + 1;
        direct = 0x28;
    }
}
```

El software es diseñado y elaborado para contemplar, además del monitoreo de signos vitales, el problema que representa el poco o nulo contacto que tienen los médicos y/o enfermeras con equipos de cómputo y otorgar facilidad de operación que en un momento dado, pueden hacer ineficaz el control de monitoreo por computadora del usuario.

Es diseñado en forma de entorno gráfico, permitiendo tener así en pantalla la mayor cantidad de información y que ésta sea mucho más fácil de entender, personalizando el entorno de acuerdo a gustos y necesidades.

A través de este software se monitorean los signos vitales de manera fácil y sencilla, ya que solo es necesario conocer unas cuantas teclas o en su caso el mouse para su uso; con obtención de resultados satisfactorios, en un tiempo infinitamente menor al tradicional.

En este capítulo se explica como utilizar el software así como las funciones de algunas teclas o mouse, ya que se asume que el usuario no está familiarizado con equipos de cómputo además se representan las diversas opciones que existen por medio de figuras (íconos) para identificar la opción elegida.

El sistema es desarrollado en lenguaje "C+ +" y está apoyado con íconos (pequeñas representaciones gráficas que representan la aplicación a ejecutar) para proporcionar un entorno gráfico, permitiendo tener en pantalla la mayor cantidad de información y que ésta sea más fácil de entender y manejar ya que el ser humano es capaz de absorber y extraer más información por el sentido de la vista que por cualquier otro, somos más rápidos observando un objeto o su representación gráfica que "traduciendo" las palabras que lo describen para llegar a comprenderlo; todo esto se realiza con el mouse o en su caso cada ícono tiene su respectivo nombre para identificar su función; su instalación se puede realizar en IBM-PC, XT, AT o compatibles, puede estar residente en disco duro que es lo más aconsejable o en floppys de preferencia de alta densidad, ya que el sistema en realidad no ocupa mucho espacio sino por la cantidad de pacientes que se pueden manejar.

El sistema se acerca al modo de médicos y enfermeras de gestionar la información y es capaz de adaptarse a fórmulas convencionales, tratando de acercarse al estilo de trabajo habitual de las personas que lo operarán.

El sistema está estructurado en rutinas que contienen las operaciones que solo son llamadas con poca frecuencia y que solo son utilizadas cuando el usuario las solicita y, en subrutinas que contienen las operaciones que pueden ser solicitadas en cualquier momento por el sistema sin importar la operación que esté ejecutando en ese momento (activación de alarmas de algún signo vital, por ejemplo) y que por su importancia se programan todas las subrutinas en una sola zona del sistema al final de este y así puedan ser requeridas en cualquier momento y en cualquier operación.

Los íconos permiten ejecutar tareas sin necesidad de moverse por menús desplegables.

Las microcomputadoras personales contienen varias teclas que realizan funciones especiales. Los teclados fabricados por las diversas compañías están dispuestos en formas diferentes y algunas veces asignan nombres diferentes a las teclas; sin embargo, todos ellos tienen esencialmente las mismas funciones disponibles.

Los nombres de las teclas que se utilizan corresponden al modelo IBM Pc o XT.

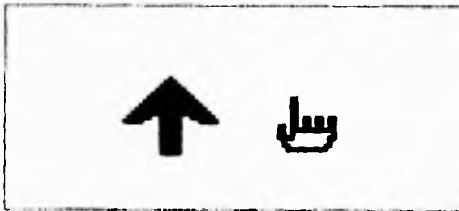
A continuación se describen las principales teclas que son utilizadas y su función.

Teclas de función:

**Enter:** Manda una línea escrita o una opción del menú a procesar.

- Backspace:** Mueve el cursor una posición hacia la izquierda a través del texto, borrándolo.
- Spacebar:** Inserta blancos al mismo tiempo que mueve el cursor hacia la derecha.
- Insert:** Actúa como un interruptor habilitando y deshabilitando la inserción de caracteres a la izquierda del cursor moviendo el resto del texto a la derecha. Cuando el modo de inserción está deshabilitado, al escribir, los caracteres se escriben sobre aquellos que se encuentren en el campo.
- Delete:** Borra el carácter donde se encuentra el cursor.
- Llaves de movimiento:** Conjunto de teclas que mueven un espacio en la dirección indicada por la flecha en la tecla.

El uso del mouse es muy sencillo solo basta moverlo para que el cursor se desplace en la dirección dada por el movimiento, dependiendo del tipo de mouse que se esté utilizando el botón de **Enter** generalmente es el de la izquierda, sin embargo algunos lo tienen localizado en el botón de la derecha, aunque si existen dudas puede determinarlo colocando el cursor del mouse en alguno de los íconos presionar cualquiera de los botones y observar cual es el que responde y ese será el botón de **Enter**.



El cursor estará representado por una mano o por una flecha cuando no está conectado, se desplaza en la pantalla a nuestra voluntad y se utilizará para seleccionar objetos de la pantalla.

Para tener acceso a las diversas opciones del sistema solo es necesario pulsar la tecla con la letra remarcada en el nombre de la opción o con el mouse ubicarse en la figura de la opción a elegir.

## **5.2 Estructura del Sistema de Monitoreo de Signos Vitales.**

La sección de Menu principal describe los diversos módulos que componen el sistema (Configuración, Menú de pacientes, Monitoreo, Respaldo de información, Conversión de datos y Salida).

El sistema se conforma de un módulo principal que a su vez está compuesto de seis módulos :

- Configuración : Icono de computadora y nodos
- Pacientes : Icono de agenda
- Monitoreo : Icono de graficación
- Respaldos : Icono de diskettes
- Conversión : Icono de impresora
- Salida : Icono de dirección salida

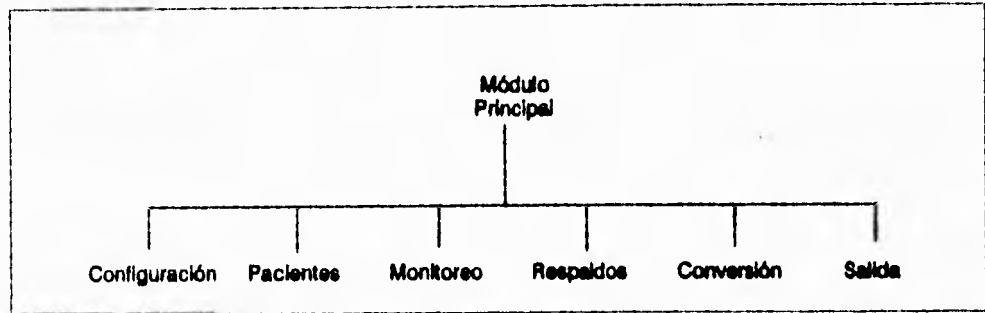
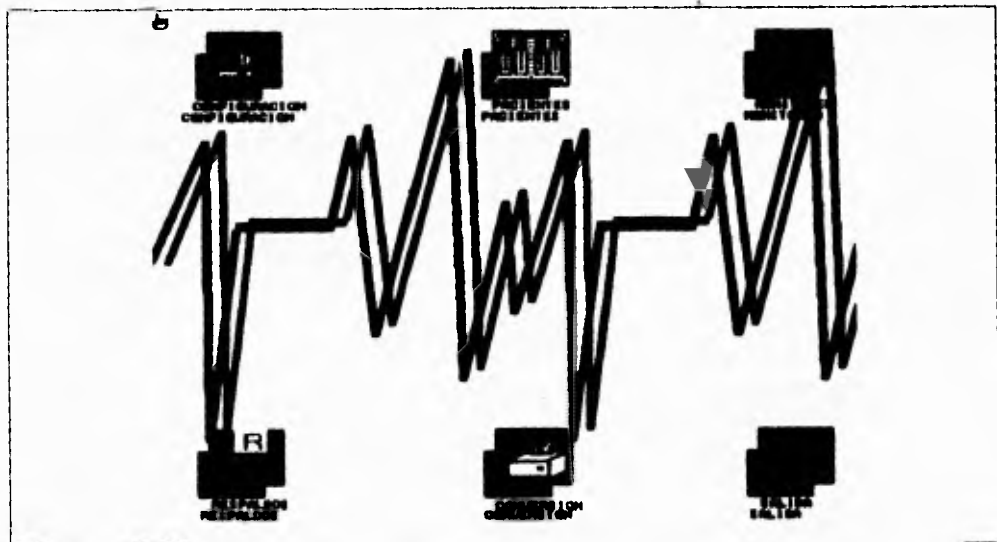


Figura 1. Diagrama General del sistema.



**Módulo de Configuración (Estructurado en 4 módulos)**

La sección de "Configuración" explica el manejo de los datos que se pueden modificar, el estado de los equipos que están conectados con la computadora para monitorear, intervalos de monitoreo y algunos parámetros que son importantes.

Este módulo contiene rutinas de elementos gráficos, manejo de archivos, rutinas de comunicación, rutinas de mouse, etc.

- Cambio de estados de actuadores
- Cambio de signos vitales
- Despliegue de nodos y signos vitales
- Cambio de tiempos de monitoreo

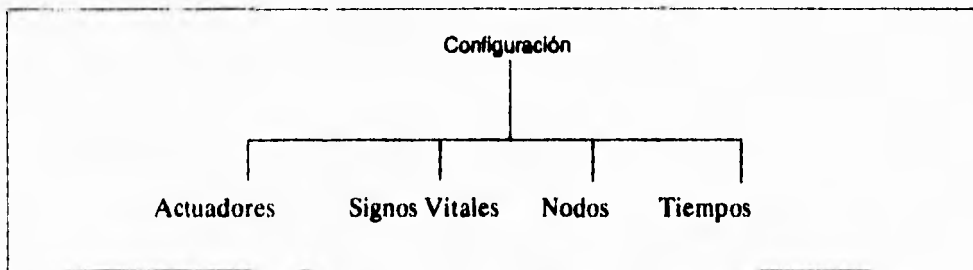


Figura 2. Diagrama del módulo de configuración.

**Configuración:**

En este menú se tiene la opción de configurar los actuadores que se desean encender o apagar (se recomienda tener todos los actuadores apagados) con solo teclear el número del actuator que se quiere modificar su estado actual una X indica que el actuator esta apagado y un símbolo √ si esta encendido; además el sistema indica con una X si el dispositivo de monitoreo está desconectado indicando cual es el dispositivo con problemas y, con un símbolo √ si el dispositivo está operando correctamente.



En esta sección se puede modificar el nombre de las variables a monitorear o cambiar el orden de monitoreo, teniendo en cuenta que la primera variable es la que se grafica en el menú de monitoreo o, añadir 3 variables más para monitorear teniendo un total de hasta 8 signos vitales. La temperatura se puede monitorear en grados Kelvin o Centígrados solo eligiendo la opción el sistema le pedirá que teclee una K para grados Kelvin o una C para grados Centígrados. Al final de la pantalla el sistema le indicará la capacidad del disco duro, memoria que dispone, si cuenta o no con coprocesador matematico (este dispositivo acelera las operaciones matematicas) y del espacio disponible en su disco es muy importante tener espacio suficiente para que posteriormente no se presenten problemas al querer respaldar información de los pacientes. Para pasar a la segunda pantalla de configuración solo tecle "S".

Direcciones:										
128	129	130	131	132	133	134	135	136	137	
1	2	3	4	5	6	7	8	9	10	
18	18	28	24	42	38	38	46	74	82	
11	19	27	28	42	31	38	47	79	82	
12	28	28	24	44	32	38	48	78	84	
13	21	28	27	48	32	41	48	77	88	
14	22	28	28	46	34	42	78	78	86	
15	23	31	28	47	38	42	71	78	87	
16	24	32	48	48	34	44	72	88	88	
17	25	22	41	48	37	48	72	81	88	

OPCION:	<input type="checkbox"/>
SALIDA	

Tamaño de memoria RAM mínima contenida:	640
Se existe Coprocesador Matematica	
Capacidad total de almacenamiento (Mb) :	128
Capacidad Libre de almacenamiento (Mb) :	3. 14816



**Módulo de Pacientes (Estructurado en 4 módulos)**

La sección de "Menu de Pacientes" describe los datos referentes al paciente como nombre, edad, sexo, número de equipo asignado y los niveles de alarma tanto máximos como mínimos propios del paciente en cuestión.

Este módulo contiene elementos gráficos, rutinas de manejo de archivos, rutinas de manejo de mouse, etc.

- Ingreso de pacientes
- Bajas de registros de pacientes
- Cambio de datos de pacientes
- Salida del módulo

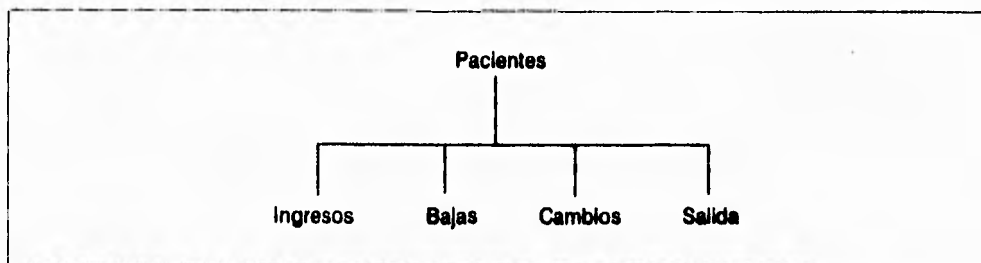
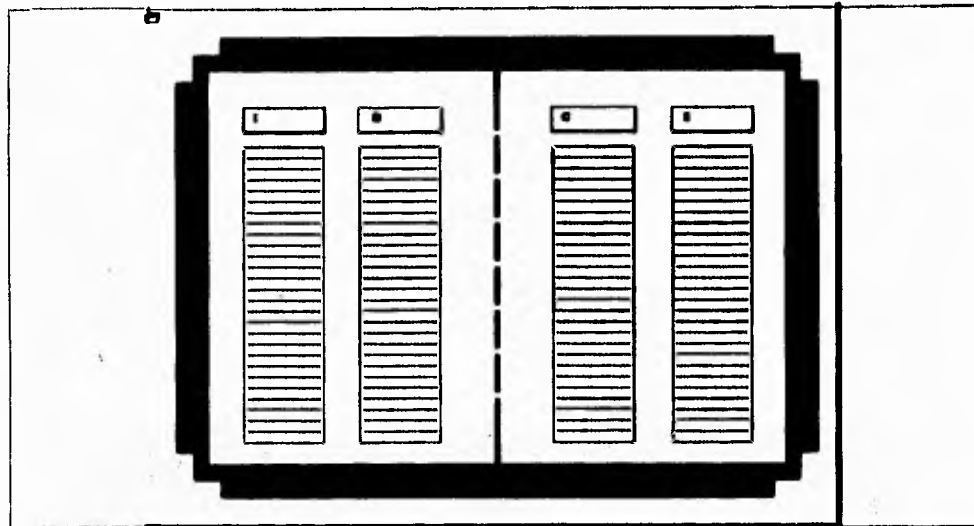


Figura 3. Diagrama del módulo de pacientes.

**Menú de Pacientes:** Este módulo se presenta en forma de carpeta en donde tenemos los siguientes submenús:

**Ingresos:**

En este módulo es donde se proporciona la información referente al paciente (nombre del paciente, edad, sexo, cama asignada) además de sus signos vitales que se van a monitorear con sus respectivos niveles de alarma tanto máximos como mínimos, solo es necesario escribir los niveles y el sistema automáticamente accesará al siguiente signo vital y así hasta el último signo. Para guardar la información de cada paciente se presiona la tecla "G" o con el mouse posicionarse en la figura del disco con la letra "G" y para salir de esta opción con la letra "S" o por medio del mouse en el ícono de Salida.

**Bajas:**

En este módulo se presenta la información del paciente que se eligió dar de baja para monitoreo si desea darlo de baja solo es necesario presionar la letra "B" o con el ícono del disco que contiene la letra "B" por medio del mouse; para salir de este menú con la letra "S" o con la figura de Salida. Si elige la opción de salida el registro no se cancelará.

**Cambios:**

En este menú se presentan los datos referentes del paciente que se van a modificar desde sus datos personales hasta los signos vitales a monitorear con sus respectivos niveles de alarma para guardar la información con la letra "C" o por medio del ícono del disco con la letra "C" o usando el mouse, la salida de este menú es igual a las anteriores. En la opción de salida los cambios no se registran es necesario para actualizar los datos elegir la opción "C".

The screenshot shows a software interface for patient data modification. At the top, there are four input fields: 'Nombre:', 'Edad:', 'Cana:', and 'Sexo:'. To the right of these fields is a floppy disk icon. Below the input fields are two tables. The first table is titled 'MODULO' and has 8 rows and 2 columns. The second table is titled 'ALARMA' and has 8 rows and 2 columns. The tables are currently empty.

**Módulo de Monitoreo (Estructurado en 4 módulos)**

La sección de "Monitoreo" explica el monitoreo de los signos vitales con las opciones de zoom, pausa, elegir y salir.

El módulo está formado por elementos gráficos, manejo de archivos, rutinas de comunicación, rutinas de mouse, etc.

- Elección de pacientes
- Amplificación de monitoreo de un paciente
- Pausa para análisis
- Salida del módulo

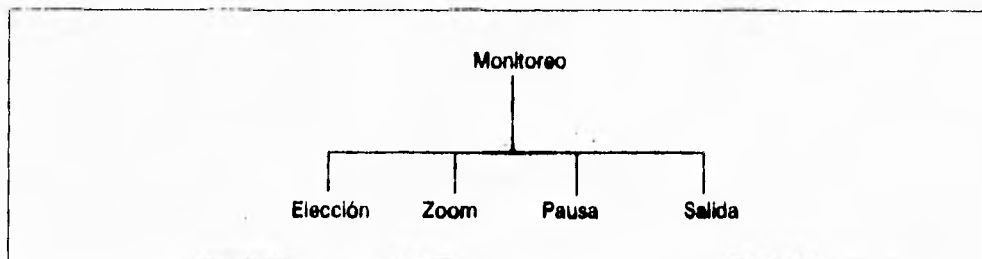


Figura 4. Diagrama del módulo de monitoreo.

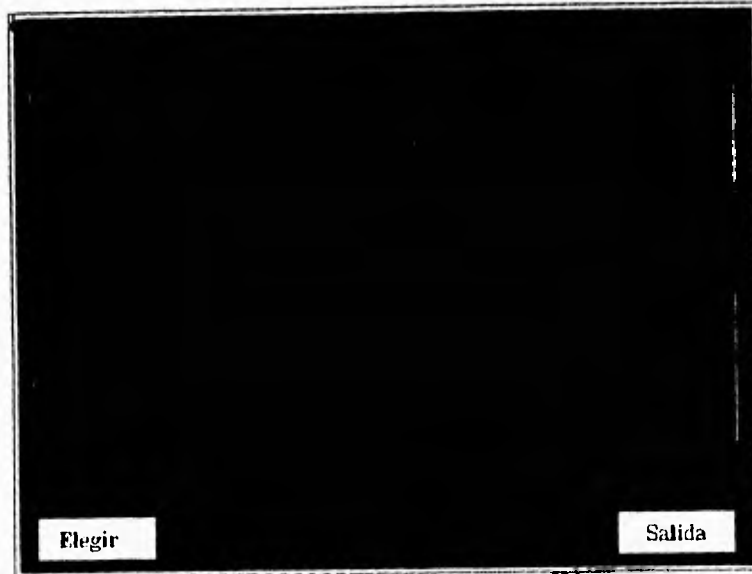
**Elegir:**

Con este módulo se eligen los 4 pacientes a monitorear solo proporcionando el número de la cama del paciente.



**Zoom:** Este módulo amplifica el monitoreo del paciente que se quiere monitorear, al elegir la opción el sistema le pedirá el número de la cama del paciente a monitorear. Para cambiar de paciente se elige nuevamente la opción de Elegir para regresar al menú anterior elegir la opción de Salida. Al regresar al menú principal es necesario Elegir nuevamente a los pacientes a monitorear para tener el monitoreo actualizado.

**Pausa:** Esta opción detiene el monitoreo para observar o analizar una gráfica o un valor de algún signo vital, para volver activar el monitoreo presionar cualquier tecla. Es necesario elegir nuevamente los pacientes a monitorear para tener la información actualizada.



**Salida:** Con esta opción abandona el módulo de monitoreo.

Para elegir las opciones del módulo de monitoreo solo es necesario presionar la tecla con la letra asignada a cada módulo o por medio del mouse, ubicarse en el cuadro que contiene el nombre de la opción a elegir.

Cuando se realiza el monitoreo de los pacientes y la información está fuera de los límites permitidos se encenderá el actuador correspondiente a ese signo vital y se activará una alarma sonora de la computadora y no se desactivará hasta que los signos vitales del paciente estén dentro de sus límites.



**Módulo de Respaldo (Estructurado en un módulo)**

La sección de "Respaldo" describe como guardar la información del paciente para su posterior análisis.

Conformado por elementos gráficos, manejo de archivos y de comunicación.

**Respaldos:** Este módulo almacena los datos de los pacientes con solo proporcionar el nombre del archivo donde se almacenarán los datos la cama del paciente y el tiempo de almacenamiento (máximo 60 seg.). También se indica el espacio disponible en su disco para almacenar datos, si no existe espacio suficiente no será posible el respaldo de los datos.

El diagrama muestra un formulario de respaldo de datos de un paciente, representado como una pantalla de software. El formulario tiene un fondo negro y un área de texto blanca. El texto dentro del formulario es el siguiente:

ARCHIVO : \_\_\_\_\_

CAMA : \_\_\_\_\_

TIEMPO (MAX. 60 SEG.) \_\_\_\_\_

Espacio Libre en Disco (Mb) : 9

**Módulo de Conversión (Estructurado en un módulo)**

La sección de "Conversion" explica como realizar la conversion de los datos guardados en C + + a Lotus o Excel para su posterior graficación y estudio.

Estructurado con elementos gráficos y manejo de archivos

**Conversion:** En este módulo se "convierte" el archivo creado en C + + en un archivo que puede ser importado por Lotus o Excel para graficar los datos respaldados en el menu anterior para su posterior análisis y de esta forma manipular los datos como mejor convenga para su estudio y presentación. Pueden ser importados en cualquier versión de los sistemas mencionados anteriormente, por ejemplo, en el caso de lotus solo es necesario presionar la tecla de comandos enseguida F (de file en su versión en inglés), después I (para importar el archivo), N (para importar los datos a graficar), y enseguida la ruta de acceso al archivo que se encontrará en el directorio donde se instaló el sistema de monitoreo de signos vitales seguido del nombre completo del archivo con todo y extensión donde se realizó la conversión. Si se escribe un archivo que no existe el sistema lo indicará en caso contrario mostrará el avance de la conversión de datos hasta completarla indicando cuando haya finalizado.

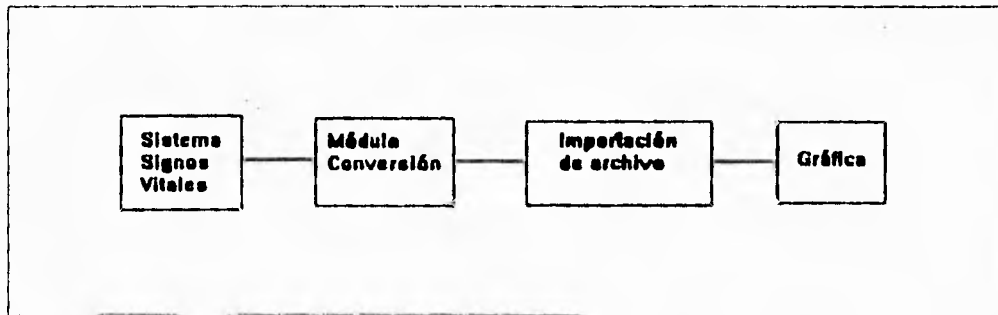
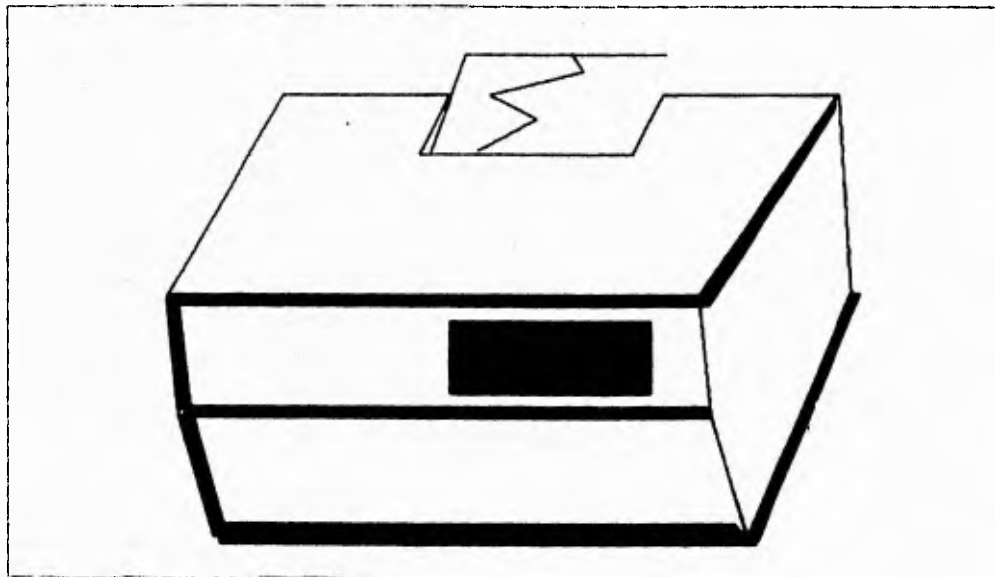
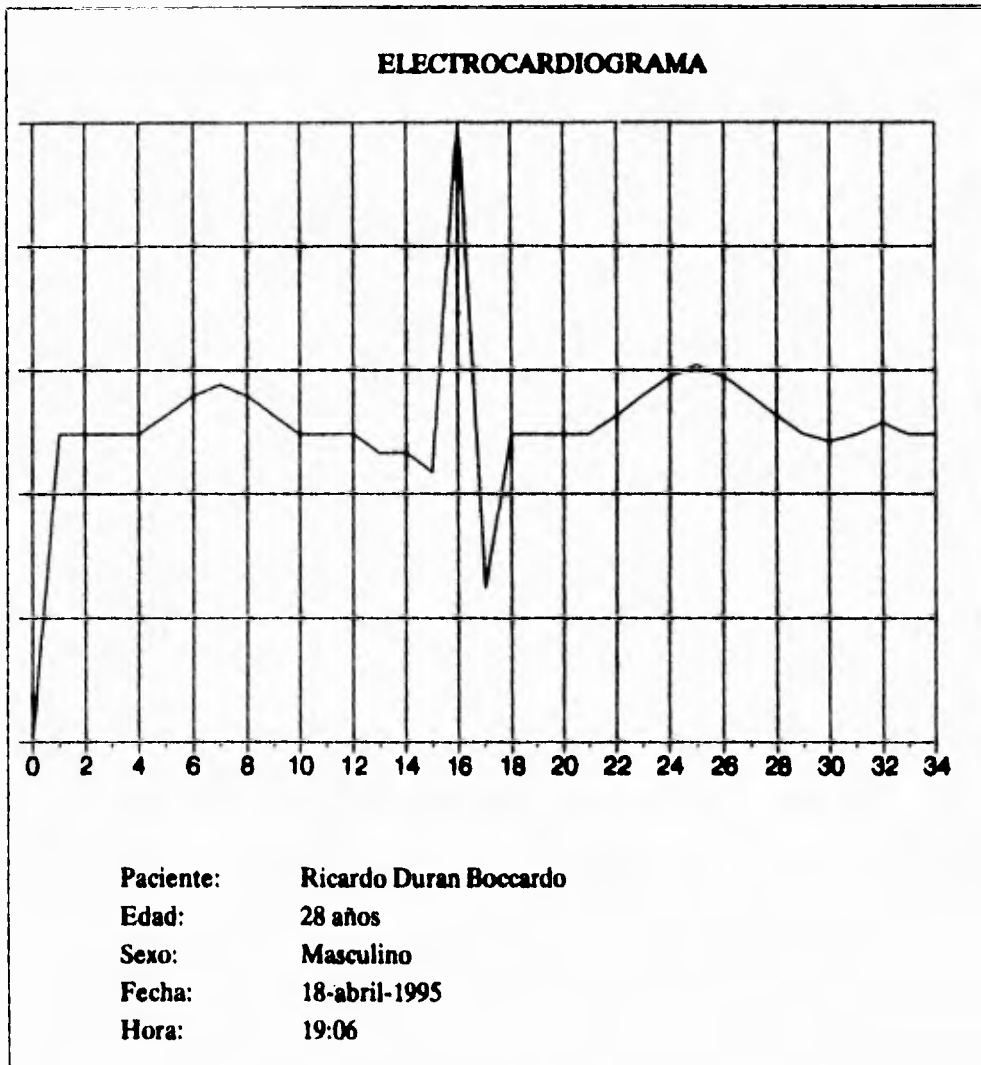


Figura 5. Generación de Archivo para importación a EXCEL.





Ejemplo de datos importados por lotus desde C + +.

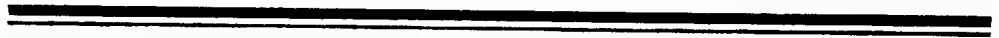
### **Módulo de Salida**

Conformado con elementos gráficos y rutinas de archivos (cerrar todos los archivos antes de abandonar el sistema).

**Salida:** Esta opción abandona el sistema por medio del ícono de Salida con el mouse o con la letra "S".

Para acceder al sistema de monitoreo de signos vitales escribir MSV en el subdirectorío donde esté instalado.

## **CONCLUSIONES**



Las necesidades de comunicación rápida y eficiente en todas las ramas de la industria ha llevado a un desarrollo acelerado de la tecnología en materia de comunicaciones. Con el fin de evitar diferencias entre los fabricantes se han establecido distintas normas para permitir un desarrollo mayor y objetivo. La correcta aplicación de estas normas nos guía con el fin de utilizar los nuevos equipos y la nueva tecnología, sin necesidad de estar dependiendo de un solo fabricante de equipo.

En la red de monitoreo de signos vitales se parte de la norma RS-232 para poder utilizar el sistema en cualquier tipo de PC y cubrir las necesidades de supervisión en un medio hospitalario.

El sistema de transmisión y recepción de datos, que es la parte central de la red, es muy confiable. Las pruebas realizadas en el laboratorio muestran que el tiempo de adquisición entre datos es de  $0.005491 \pm 5\%$  segundos; este tiempo no presenta variaciones si la lectura se realiza unicamente en uno o varios canales de un nodo, o diversos canales de varios nodos (hasta cinco). Las pruebas de velocidad se realizaron en una computadora con un microprocesador 80386 y un velocidad de 40 Mhz.

Las pruebas de distancia se realizaron con un conductor de una longitud mayor a los 80 metros de longitud. Se comprobó que la velocidad y la confiabilidad del sistema se mantiene sin afectar los datos, lo cual hace posible la implementación del sistema en este rango sin preocuparse por la distancia. Para distancias superiores es necesario realizar otras pruebas para determinar su confiabilidad utilizando por ejemplo el arreglo mostrado en el capítulo 4, aunque por las características de los elementos que componen el nodo no es recomendable. La impedancia por metro del conductor utilizado es de 0.07 ohms por metro.

Las pruebas realizadas para el funcionamiento del circuito se realizaron con una computadora personal con un microprocesador 386 de 40 Mhz de velocidad y 4 M de memoria RAM y con el programa en lenguaje C mostrado en el capítulo 4.

**PAGINACION VARIA**

**COMPLETA LA INFORMACION**



Las principales ventajas del sistema en red de monitoreo de signos vitales son las siguientes:

- 1) Fácil de implementar en todas las computadoras personales que cuenten con un puerto serial RS-232, recomendándose utilizar una computadora con un microprocesador 386 o superior dependiendo la aplicación, por razones de tiempo de respuesta y proceso.
- 2) Es posible utilizar cualquier tipo de equipo médico analógico para ser comunicado a cualquier PC, siempre y cuando cumpla con las características de acoplamiento vistas en el Capítulo que describe al "Sistema en red de adquisición y transmisión de datos" (en la parte correspondiente a especificaciones de entradas analógicas).
- 3) Bajo costo y fácil mantenimiento.
- 4) Configurando el software del sistema es posible utilizar el nodo en cualquier aplicación y no únicamente en el sistema de monitoreo de pacientes de este caso.

Las desventajas que presenta el sistema son:

- 1) Al no contar con un dispositivo propio de memoria, no permite la adquisición de datos si la computadora esta apagada o desconectada.
- 2) Para distancias superiores a los 100 m, no es recomendable su uso con la interfase RS232, por lo que se recomienda utilizar el arreglo de "Convertidor de RS232 a RS422" mostrado en el capítulo 4.

3) Solo es posible obtener a la vez un dato por nodo. Si se requiera demasiados nodos, el tiempo de respuesta es afectado, es por esto que se recomienda no utilizar más de 10 nodos.

Por ejemplo: si existen siete nodos y ocho canales por nodo, se tendrán que leer cincuenta y seis canales, por lo tanto, el tiempo de que tardará en realizar un ciclo completo de lectura será:

$$56 * 0.0549454 = 3.076 \text{ segundos}$$

De lo anterior concluimos que dependiendo de la aplicación, queda a criterio del usuario el número de nodos a ser utilizados con el fin de lograr un grado óptimo de operación en función del tiempo.

Las características de programación se diseñaron tomando en cuenta que la aplicación esta enfocada a personas que no estan familiarizadas con un ambiente de programación, lo cual hace necesario que el ambiente sea totalmente amigable y no sea necesario tener una gran experiencia en la utilización de equipos para su uso. Las ventajas de la programación son las siguientes:

- 1) Todo el sistema se puede almacenar en un disco de 3.5 de alta densidad (esto es debido a las características del compilador de C++ que genera un programa ejecutable), lo cual lo hace de fácil instalación y transportación.
- 2) Su diseño permite que con un mínimo de modificaciones en su estructura se pueda aplicar en cualquier área.
- 3) La comunicación entre el sistema y los instrumentos médicos se realiza con gran rapidez, por lo que la pérdida de datos es practicamente inexistente.
- 4) El sistema puede ser manejado por cualquier persona sin complicaciones, con poco o nulo conocimiento de equipos de computo.

5) De acuerdo a las pruebas realizadas los datos que son enviados a la computadora y los que son monitoreados son iguales con una diferencia despreciable del  $\pm 1\%$ , por lo que se considera que los datos no sufren variación desde el nodo hasta la pantalla de monitoreo del sistema.

## **APENDICE**



**Addressable Asynchronous  
Receiver/Transmitter  
CMOS**

The MC14469 receives one or two eleven-bit words in a serial data stream. One of the incoming words contains the address and when the address matches, the MC14469 then transmits information in two eleven-bit-word data streams. Each of the transmitted words contains eight data bits, an even parity bit, and start and stop bits.

The received word contains seven address bits with the address of the MC14469 set on seven pins. Thus 2<sup>7</sup> or 128 units can be interconnected in simplex or full duplex data transmission. In addition to the address received, seven command bits may be received for general-purpose data or control use.

The MC14469 finds application in transmitting data from remote A-to-D converters, remote MPUs, or remote digital transducers to the master computer or MPU.

- Supply Voltage Range: 4.5 V to 18 V
- Low Quiescent Current: 75  $\mu$ A Maximum @ 5 V, 25°C
- Guaranteed Data Rates to 4800 Baud @ 5 V, to 9600 Baud @ 12 V
- Receive — Serial to Parallel  
 Transmit — Parallel to Serial
- Transmit and Receive Simultaneously in Full Duplex
- Crystal or Resonator Operation for On-Chip Oscillator
- See Application Note AN-806A
- Chip Complexity: 1200 FETs or 300 Equivalent Gates

**MC14469**



P SUFFIX  
 PLASTIC DIP  
 CASE 711



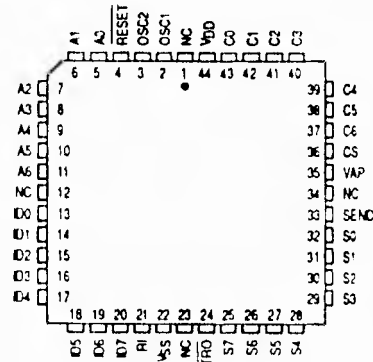
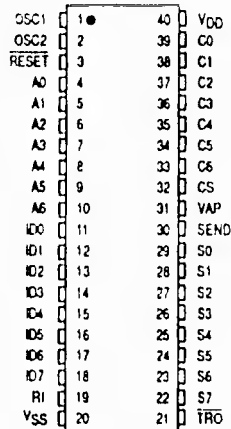
FN SUFFIX  
 PLCC  
 CASE 777

**ORDERING INFORMATION**

MC14469P Plastic DIP  
 MC14469FN PLCC

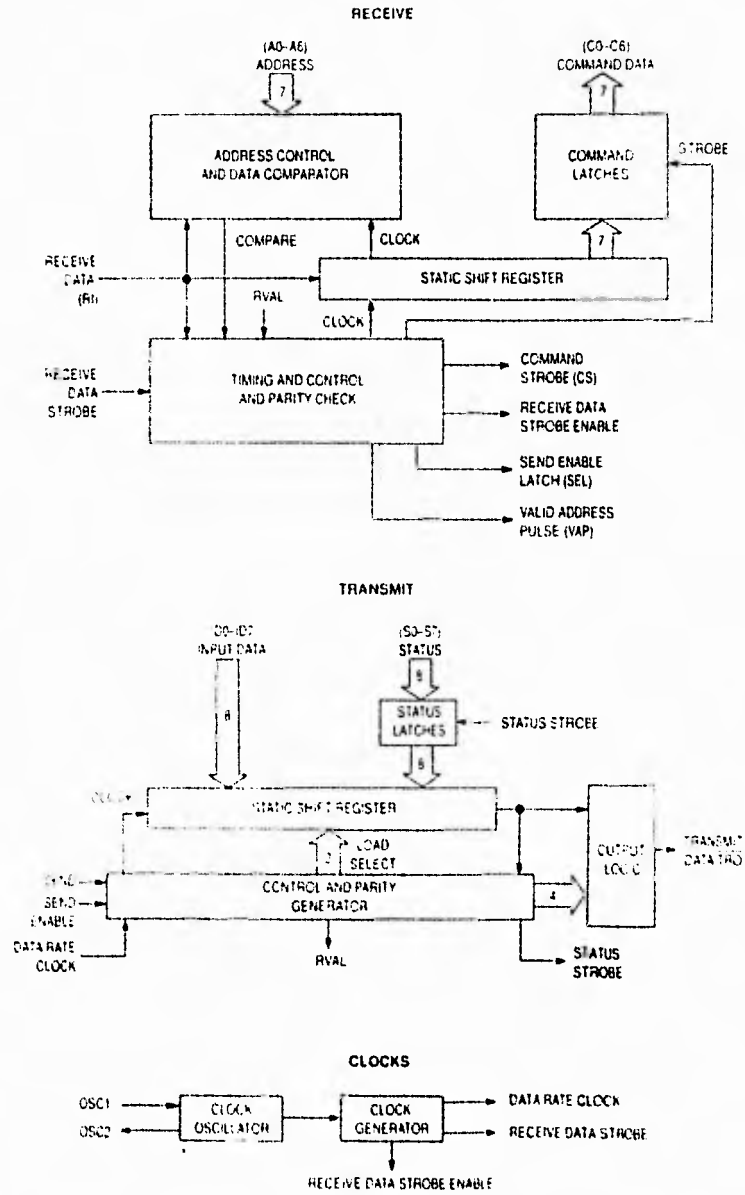


**PIN ASSIGNMENTS**



NC - NO CONNECTION

BLOCK DIAGRAM



# MC14469

## MAXIMUM RATINGS (Voltages referenced to VSS)

Parameter	Symbol	Value	Unit
DC Supply Voltage	V <sub>DD</sub>	-0.5 to +18	V
Input Voltage, All Inputs	V <sub>in</sub>	-0.5 to V <sub>DD</sub> +0.5	V
DC Current Drain per Pin	I	10	mA
Operating Temperature Range	T <sub>A</sub>	-40 to +85	°C
Storage Temperature Range	T <sub>stg</sub>	-65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V<sub>in</sub> and V<sub>out</sub> be constrained to the range V<sub>SS</sub> ≤ (V<sub>in</sub> or V<sub>out</sub>)

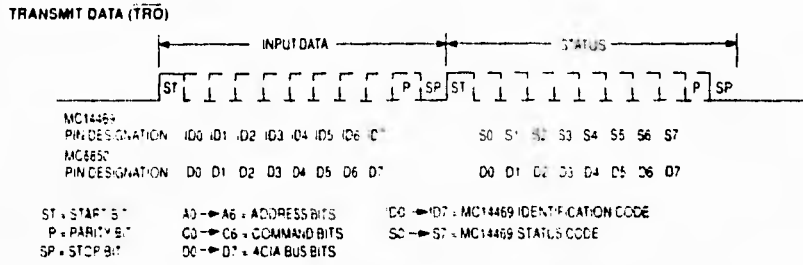
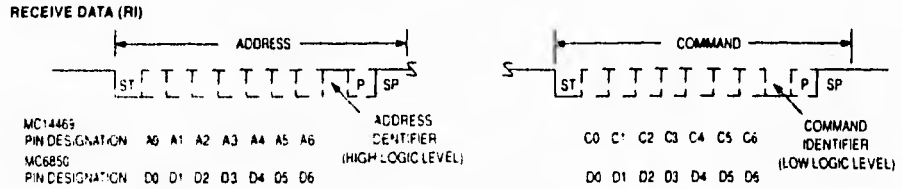
≤ V<sub>DD</sub>  
 Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V<sub>SS</sub> or V<sub>DD</sub>).

## ELECTRICAL CHARACTERISTICS (Voltages Referenced to VSS)

Characteristic	Symbol	V <sub>DD</sub> V	-40°C		25°C		+85°C		Unit	
			Min	Max	Min	Max	Min	Max		
Output Voltage V <sub>in</sub> = V <sub>DD</sub> or 0	"0" Level	VOL	5.0	—	0.05	—	0.05	—	0.05	V
			10	—	0.05	—	0.05	—	0.05	
			15	—	0.05	—	0.05	—	0.05	
	V <sub>in</sub> = 0 or V <sub>DD</sub>	VOH	5.0	4.95	—	4.95	—	4.95	—	V
			10	9.95	—	9.95	—	9.95	—	
			15	14.95	—	14.95	—	14.95	—	
Input Voltage, Except OSC1† (V <sub>O</sub> = 4.5 or 0.5 V) (V <sub>O</sub> = 9.0 or 1.0 V) (V <sub>O</sub> = 13.5 or 1.5 V)	"0" Level	V <sub>IL</sub>	5.0	—	1.5	—	1.5	—	1.5	V
			10	—	3.0	—	3.0	—	3.0	
			15	—	4.0	—	4.0	—	4.0	
	"1" Level	V <sub>IH</sub>	5.0	3.5	—	3.5	—	3.5	—	V
			10	7.0	—	7.0	—	7.0	—	
			15	11	—	11	—	11	—	
Output Drive Current (Except OSC2)	Source	I <sub>OH</sub>	5.0	-1.0	—	-0.8	—	-0.6	—	mA
			5.0	-0.2	—	-0.16	—	-0.12	—	
			10	-0.5	—	-0.4	—	-0.3	—	
			15	-1.4	—	-1.2	—	-1.0	—	
	Sink	I <sub>OL</sub>	5.0	0.52	—	0.44	—	0.36	—	mA
			10	1.3	—	1.1	—	0.9	—	
Output Drive Current (OSC2 Only)	Source	I <sub>OH</sub>	5.0	-0.19	—	-0.16	—	-0.13	—	mA
			5.0	-0.04	—	-0.035	—	-0.03	—	
			10	-0.09	—	-0.06	—	-0.06	—	
			15	-0.29	—	-0.27	—	-0.2	—	
	Sink	I <sub>OL</sub>	5.0	0.1	—	0.085	—	0.07	—	mA
			10	0.17	—	0.14	—	0.1	—	
OSC Frequency	f <sub>OSC</sub>	4.5	0	400	0	365	0	310	kHz	
		12	0	800	0	730	0	620		
Input Current	I <sub>in</sub>	15	—	+0.3	—	+0.3	—	±1.0	μA	
Pull-Up Current (A0-A6, ID0-ID7)	I <sub>UP</sub>	15	12	120	10	100	80	85	μA	
Input Capacitance (V <sub>in</sub> = 0)	C <sub>in</sub>	—	—	—	—	7.5	—	—	pF	
Quiescent Current (Per Package)	I <sub>DD</sub>	5.0	—	75	—	75	—	565	μA	
		10	—	150	—	150	—	1125		
		15	—	300	—	300	—	2250		
Supply Voltage	V <sub>DD</sub>	—	+4.5	+18	+4.5	+18	+4.5	+18	V	

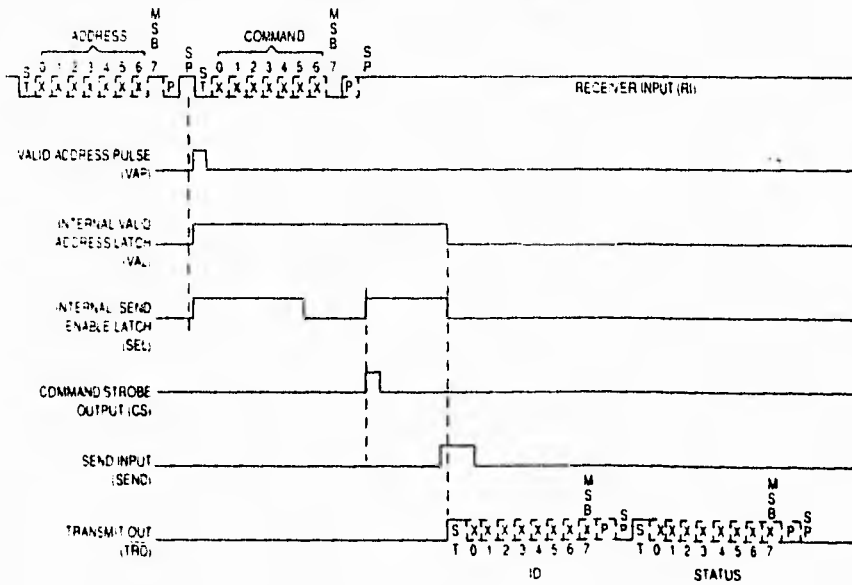
MOTOROLA CMOS APPLICATION-SPECIFIC DIGITAL-ANALOG INTEGRATED CIRCUITS

DATA FORMAT AND CORRESPONDING DATA POSITION AND PINS FOR MC14469 AND MC6850



6

TYPICAL RECEIVE-SEND CYCLE

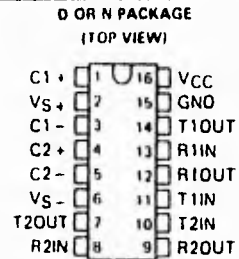




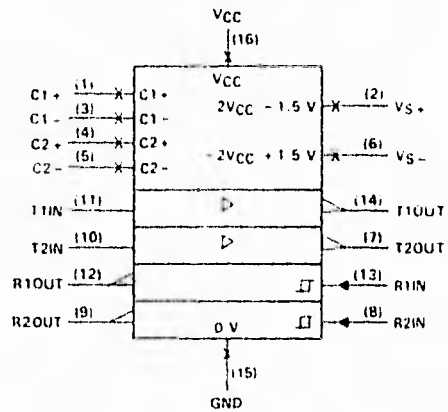
# MAX232 DUAL EIA-232 DRIVER/RECEIVER

D3120, FEBRUARY 1989 - REVISED JUNE 1989

- Operates with Single 5-V Power Supply
- LinBiCMOS™ Process Technology
- Two Drivers and Two Receivers
- ±30-V Input Levels
- Low Supply Current . . . 8 mA Typ
- Meets ANSI/EIA-232-D-1986 Specifications (Revision of EIA Standard RS-232-C)
- Designed to be Interchangeable with Maxim MAX232
- Applications
  - EIA-232 Interface
  - Battery-Powered Systems
  - Terminals
  - Modems
  - Computers



logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

## description

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

## absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Input supply voltage, VCC (see Note 1)	-0.3 V to 6 V
Positive output supply voltage, VS+	VCC - 0.3 V to 15 V
Negative output supply voltage, VS-	0.3 V to -15 V
Input voltage range: Driver	-0.3 V to VCC + 0.3 V
Receiver	±30 V
Output voltage range: T1OUT, T2OUT	VS- - 0.3 V to VS+ + 0.3 V
R1OUT, R2OUT	-0.3 V to VCC - 0.3 V
Short-circuit duration: VS+	30 s
VS-	30 s
T1OUT, T2OUT	unlimited
Operating free-air temperature range	0°C to 70°C
Storage temperature range	-65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C

NOTE 1: All voltage values are with respect to network ground terminal

LinASIC and LinBiCMOS are trademarks of Texas Instruments Incorporated.

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

**TEXAS  
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 1989 Texas Instruments Incorporated

242

**MAX232**  
**DUAL EIA-232 DRIVER/RECEIVER**

**recommended operating conditions**

	MIN	NOM	MAX	UNIT
Supply voltage, $V_{CC}$	4.5	5	5.5	V
High level input voltage, $V_{IH}$ (T1IN, T2IN)	2			V
Low level input voltage, $V_{IL}$ (T1IN, T2IN)			0.8	V
Receiver input voltage, R1IN, R2IN			$\pm 30$	V
Operating free air temperature, $T_A$	0		70	$^{\circ}\text{C}$

**electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (unless otherwise noted)**

PARAMETER		TEST CONDITIONS	MIN	TYP <sup>1</sup>	MAX	UNIT
$V_{OH}$ High level output voltage	T1OUT, T2OUT	$R_L = 3\text{ k}\Omega$ to GND	5	7		V
	R1OUT, R2OUT	$I_{OH} = -1\text{ mA}$	3.5			
$V_{OL}$ Low level output voltage <sup>2</sup>	T1OUT, T2OUT	$R_L = 3\text{ k}\Omega$ to GND		-7	-5	V
	R1OUT, R2OUT	$I_{OL} = 3.2\text{ mA}$			0.4	
$V_{I+}$ Receiver positive going input threshold voltage	R1IN, R2IN	$V_{CC} = 5\text{ V}$ , $T_A = 25^{\circ}\text{C}$		1.7	2.4	V
$V_{I-}$ Receiver negative going input threshold voltage	R1IN, R2IN	$V_{CC} = 5\text{ V}$ , $T_A = 25^{\circ}\text{C}$	0.8	1.2		V
$V_{Hys}$ Input hysteresis	R1IN, R2IN	$V_{CC} = 5\text{ V}$	0.2	0.5	1	k $\Omega$
$r_{i+}$ Receiver input resistance	R1IN, R2IN	$V_{CC} = 5\text{ V}$ , $T_A = 25^{\circ}\text{C}$	3	5	7	k $\Omega$
$r_o$ Output resistance	T1OUT, T2OUT	$V_{S+} = V_{S-} = 0$ , $V_O = \pm 2\text{ V}$	300			$\Omega$
$I_{OS}^3$ Short-circuit output current	T1OUT, T2OUT	$V_{CC} = 5.5\text{ V}$ , $V_O = 0$		$\pm 10$		mA
$I_{IS}$ Short-circuit input current	T1IN, T2IN	$V_I = 0$			200	$\mu\text{A}$
$I_{CC}$ Supply current		$V_{CC} = 5.5\text{ V}$ , All outputs open, $T_A = 25^{\circ}\text{C}$		8	10	mA

<sup>1</sup>All typical values are at  $V_{CC} = 5\text{ V}$ ,  $T_A = 25^{\circ}\text{C}$ .

<sup>2</sup>The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

<sup>3</sup>Not more than one output should be shorted at a time.

**switching characteristics,  $V_{CC} = 5\text{ V}$ ,  $T_A = 25^{\circ}\text{C}$**

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{PLH(R)}$ Receiver propagation delay time, low-to-high level output	See Figure 2		500		ns
$t_{PHL(R)}$ Receiver propagation delay time, high-to-low level output	See Figure 2		500		ns
SR Driver slew rate	$R_L = 3\text{ k}\Omega$ to 7 k $\Omega$ . See Figure 3			30	V/ $\mu\text{s}$
SR(tr) Driver transition region slew rate	See Figure 4		3		V/ $\mu\text{s}$

**MAX232**  
**DUAL EIA-232 DRIVER/RECEIVER**

**TYPICAL APPLICATION DATA**

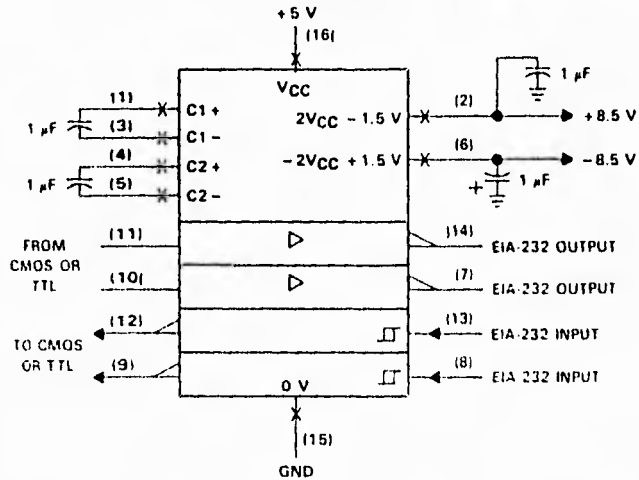
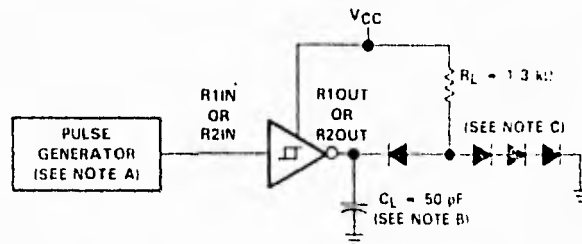
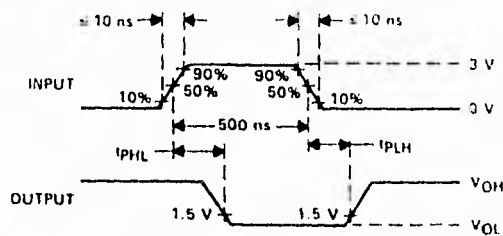


FIGURE 1. TYPICAL OPERATING CIRCUIT

**PARAMETER MEASUREMENT INFORMATION**



**TEST CIRCUIT**



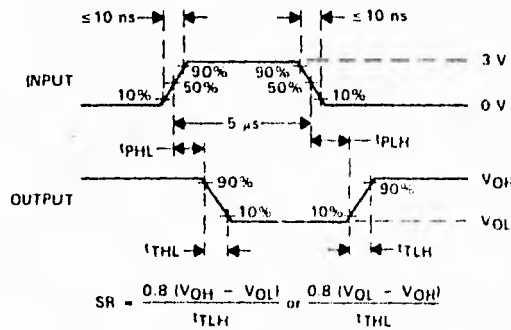
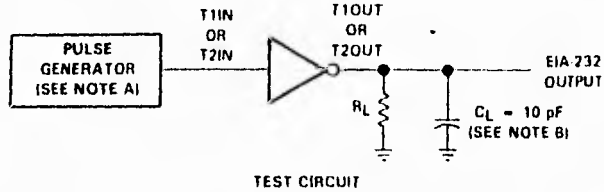
**WAVEFORMS**

- NOTES: A The pulse generator has the following characteristics:  $Z_{OUT} = 50 \Omega$  Dnt,  $C_{line} \leq 50$  pF.  
 B  $C_L$  includes probe and jig capacitance.  
 C All diodes are 1N3064 or equivalent.

FIGURE 2. RECEIVER TEST CIRCUIT AND WAVEFORMS FOR  $t_{PHL}$  AND  $t_{PLH}$  MEASUREMENT

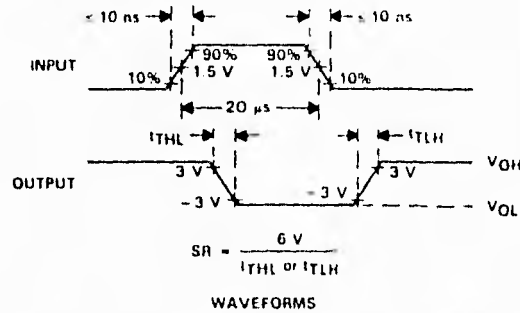
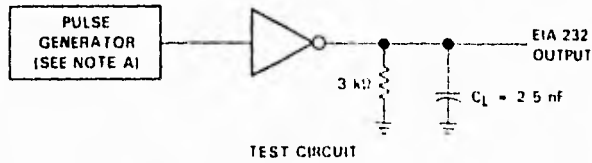
**MAX232**  
**DUAL EIA-232 DRIVER/RECEIVER**

**PARAMETER MEASUREMENT INFORMATION**



NOTE A: The pulse generator has the following characteristics:  $Z_{OUT} = 50 \Omega$ ; Duty Cycle  $\leq 50\%$   
 B:  $C_L$  includes probe and jig capacitance.

**FIGURE 3. DRIVER TEST CIRCUIT AND WAVEFORMS FOR  $t_{PHL}$  AND  $t_{PLH}$  MEASUREMENT (5- $\mu$ s INPUT)**



NOTE A: The pulse generator has the following characteristics:  $Z_{OUT} = 50 \Omega$ ; Duty Cycle  $\leq 50\%$

**FIGURE 4. TEST CIRCUIT AND WAVEFORMS FOR  $t_{THL}$  AND  $t_{TLH}$  MEASUREMENT (20- $\mu$ s INPUT)**



National  
Semiconductor  
Corporation

## ADC0808, ADC0809 8-Bit $\mu$ P Compatible A/D Converters with 8-Channel Multiplexer

### General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8 single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE® outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.)

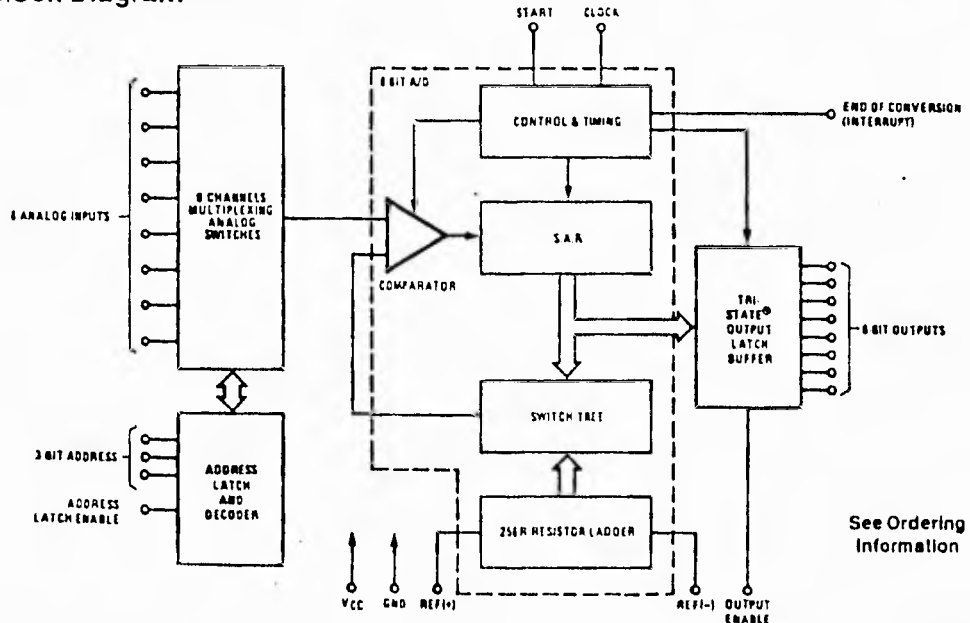
### Features

- Easy interface to all microprocessors
- Operates ratiometrically or with 5 V<sub>DC</sub> or analog span adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications
- Standard hermetic or molded 28-pin DIP package
- 28-pin molded chip carrier package

### Key Specifications

- |                          |                               |
|--------------------------|-------------------------------|
| ■ Resolution             | 8 Bits                        |
| ■ Total Unadjusted Error | $\pm 1/2$ LSB and $\pm 1$ LSB |
| ■ Single Supply          | 5 V <sub>DC</sub>             |
| ■ Low Power              | 15 mW                         |
| ■ Conversion Time        | 100 $\mu$ s                   |

### Block Diagram



See Ordering  
Information

**Absolute Maximum Ratings** (Notes 1 & 2)

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V <sub>CC</sub> ) (Note 3)	6.5V
Voltage at Any Pin	-0.3V to (V <sub>CC</sub> + 0.3V)
Except Control Inputs	
Voltage at Control Inputs	-0.3V to +15V
(START, DE, CLOCK, ALE, ADD A, ADD B, ADD C)	
Storage Temperature Range	-65°C to +150°C
Package Dissipation at T <sub>A</sub> = 25°C	875 mW
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (plastic)	260°C
Dual-In-Line Package (ceramic)	300°C
Molded Chip Carrier Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 11)	400V

**Operating Conditions** (Notes 1 & 2)

Temperature Range (Note 1)	T <sub>MIN</sub> ≤ T <sub>A</sub> ≤ T <sub>MAX</sub>
ADC0808CJ	-55°C ≤ T <sub>A</sub> ≤ +125°C
ADC0808CCJ, ADC0808CCN,	
ADC0809CCN	-40°C ≤ T <sub>A</sub> ≤ +85°C
ADC0808CCV, ADC0809CCV	-40°C ≤ T <sub>A</sub> ≤ +85°C
Range of V <sub>CC</sub> (Note 1)	4.5 V <sub>DC</sub> to 6.0 V <sub>DC</sub>

ESTR. TESTS IN DIRE  
SALIR DE LA BILIBIOTECA

**Electrical Characteristics**

Converter Specifications: V<sub>CC</sub> = 5 V<sub>DC</sub> = V<sub>REF+</sub>, V<sub>REF(-)</sub> = GND, T<sub>MIN</sub> ≤ T<sub>A</sub> ≤ T<sub>MAX</sub> and f<sub>CLK</sub> = 640 kHz unless otherwise stated.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	ADC0808 Total Unadjusted Error (Note 5)	25°C T <sub>MIN</sub> to T <sub>MAX</sub>			± 1/2 ± 3/4	LSB LSB
	ADC0809 Total Unadjusted Error (Note 5)	0°C to 70°C T <sub>MIN</sub> to T <sub>MAX</sub>			± 1 ± 1 1/4	LSB LSB
	Input Resistance	From Rel(+) to Rel(-)	1.0	2.5		kΩ
	Analog Input Voltage Range	(Note 4) V(+) or V(-)	GND - 0.10		V <sub>CC</sub> + 0.10	V <sub>DC</sub>
V <sub>REF(+)</sub>	Voltage, Top of Ladder	Measured at Rel(+)		V <sub>CC</sub>	V <sub>CC</sub> + 0.1	V
$\frac{V_{REF(+)} + V_{REF(-)}}{2}$	Voltage, Center of Ladder		V <sub>CC</sub> /2 - 0.1	V <sub>CC</sub> /2	V <sub>CC</sub> /2 + 0.1	V
V <sub>REF(-)</sub>	Voltage, Bottom of Ladder	Measured at Rel(-)	-0.1	0		V
I <sub>IN</sub>	Comparator Input Current	f <sub>C</sub> = 640 kHz, (Note 6)	-2	± 0.5	2	μA

**Electrical Characteristics**

Digital Levels and DC Specifications: ADC0808CJ 4.5V ≤ V<sub>CC</sub> ≤ 5.5V, -55°C ≤ T<sub>A</sub> ≤ +125°C unless otherwise noted  
ADC0808CCJ, ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, 4.75V ≤ V<sub>CC</sub> ≤ 5.25V, -40°C ≤ T<sub>A</sub> ≤ +85°C unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
<b>ANALOG MULTIPLEXER</b>						
I <sub>OFF(+)</sub>	OFF Channel Leakage Current	V <sub>CC</sub> = 5V, V <sub>IN</sub> = 5V, T <sub>A</sub> = 25°C T <sub>MIN</sub> to T <sub>MAX</sub>		10	200 1.0	nA μA
I <sub>OFF(-)</sub>	OFF Channel Leakage Current	V <sub>CC</sub> = 5V, V <sub>IN</sub> = 0, T <sub>A</sub> = 25°C T <sub>MIN</sub> to T <sub>MAX</sub>	-200 -1.0	-10		nA μA

**Electrical Characteristics** (Continued)

Digital Levels and DC Specifications: ADC0808CJ, 4.5V ≤ V<sub>CC</sub> ≤ 5.5V, -55°C ≤ T<sub>A</sub> ≤ +125°C unless otherwise noted  
 ADC0808CCJ, ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, 4.75V ≤ V<sub>CC</sub> ≤ 5.25V, -40°C ≤ T<sub>A</sub> ≤ +85°C unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
<b>CONTROL INPUTS</b>						
V <sub>IN(1)</sub>	Logical "1" Input Voltage		V <sub>CC</sub> - 1.5			V
V <sub>IN(0)</sub>	Logical "0" Input Voltage				1.5	V
I <sub>IN(1)</sub>	Logical "1" Input Current (The Control Inputs)	V <sub>IN</sub> = 15V			1.0	μA
I <sub>IN(0)</sub>	Logical "0" Input Current (The Control Inputs)	V <sub>IN</sub> = 0	-1.0			μA
I <sub>CC</sub>	Supply Current	f <sub>CLK</sub> = 640 kHz		0.3	3.0	mA
<b>DATA OUTPUTS AND EOC (INTERRUPT)</b>						
V <sub>OUT(1)</sub>	Logical "1" Output Voltage	I <sub>O</sub> = -360 μA	V <sub>CC</sub> - 0.4			V
V <sub>OUT(0)</sub>	Logical "0" Output Voltage	I <sub>O</sub> = 1.6 mA			0.45	V
V <sub>OUT(0)</sub>	Logical "0" Output Voltage EOC	I <sub>O</sub> = 1.2 mA			0.45	V
I <sub>OUT</sub>	TRI-STATE Output Current	V <sub>O</sub> = 5V V <sub>O</sub> = 0	-3		3	μA μA

**Electrical Characteristics**

Timing Specifications V<sub>CC</sub> = V<sub>REF(+)</sub> = 5V, V<sub>REF(-)</sub> = GND, t<sub>r</sub> = t<sub>f</sub> = 20 ns and T<sub>A</sub> = 25°C unless otherwise noted.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t <sub>WS</sub>	Minimum Start Pulse Width	(Figure 5)		100	200	ns
t <sub>WALE</sub>	Minimum ALE Pulse Width	(Figure 5)		100	200	ns
t <sub>S</sub>	Minimum Address Set-Up Time	(Figure 5)		25	50	ns
t <sub>H</sub>	Minimum Address Hold Time	(Figure 5)		25	50	ns
t <sub>D</sub>	Analog MUX Delay Time From ALE	R <sub>S</sub> = 0Ω (Figure 5)		1	2.5	μS
t <sub>H1</sub> , t <sub>H0</sub>	OE Control to Q Logic State	C <sub>L</sub> = 50 pF, R <sub>L</sub> = 10k (Figure 8)		125	250	ns
t <sub>TH</sub> , t <sub>OH</sub>	OE Control to Hi-Z	C <sub>L</sub> = 10 pF, R <sub>L</sub> = 10k (Figure 8)		125	250	ns
t <sub>C</sub>	Conversion Time	f <sub>C</sub> = 640 kHz, (Figure 5) (Note 7)	90	100	116	μS
f <sub>C</sub>	Clock Frequency		10	640	1280	kHz
t <sub>EOC</sub>	EOC Delay Time	(Figure 5)	0		8 + 2 μS	Clock Periods
C <sub>IN</sub>	Input Capacitance	At Control Inputs		10	15	pF
C <sub>OUT</sub>	TRI-STATE Output Capacitance	At TRI-STATE Outputs, (Note 12)		10	15	pF

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to GND, unless otherwise specified.

Note 3: A zener diode exists, internally, from V<sub>CC</sub> to GND and has a typical breakdown voltage of 7 V<sub>DC</sub>.

Note 4: Two on-chip diodes are tied to each analog input which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V<sub>CC</sub> supply. The spec allows 100 mV forward bias of either diode. This means that as long as the analog V<sub>IN</sub> does not exceed the supply voltage by more than 100 mV, the output code will be correct. To achieve an absolute 0V<sub>DC</sub> to 5V<sub>DC</sub> input voltage range will therefore require a minimum supply voltage of 4.900 V<sub>DC</sub> over temperature variations, initial tolerance and loading.

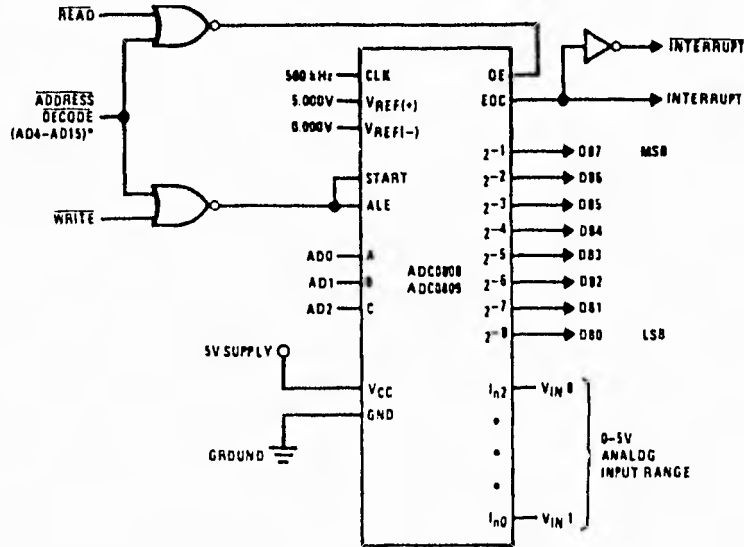
Note 5: Total unadjusted error includes offset, full-scale, linearity, and multiplexer errors. See Figure 3. None of these A/Ds requires a zero or full-scale adjust. However, if an all zero code is desired for an analog input other than 0.0V, or if a narrow full-scale span exists (for example 0.5V to 4.5V full-scale) the reference voltages can be adjusted to achieve this. See Figure 13.

Note 6: Comparator input current is a bias current into or out of the chopper stabilized comparator. The bias current varies directly with clock frequency and has little temperature dependence (Figure 6). See paragraph 4.0.

Note 7: The outputs of the data register are updated one clock cycle before the rising edge of EOC.

Note 8: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

### Typical Application



TL/H/5672-10

\* Address latches needed for 8085 and SC/MP interfacing the ADC0808 to a microprocessor

MICROPROCESSOR INTERFACE TABLE

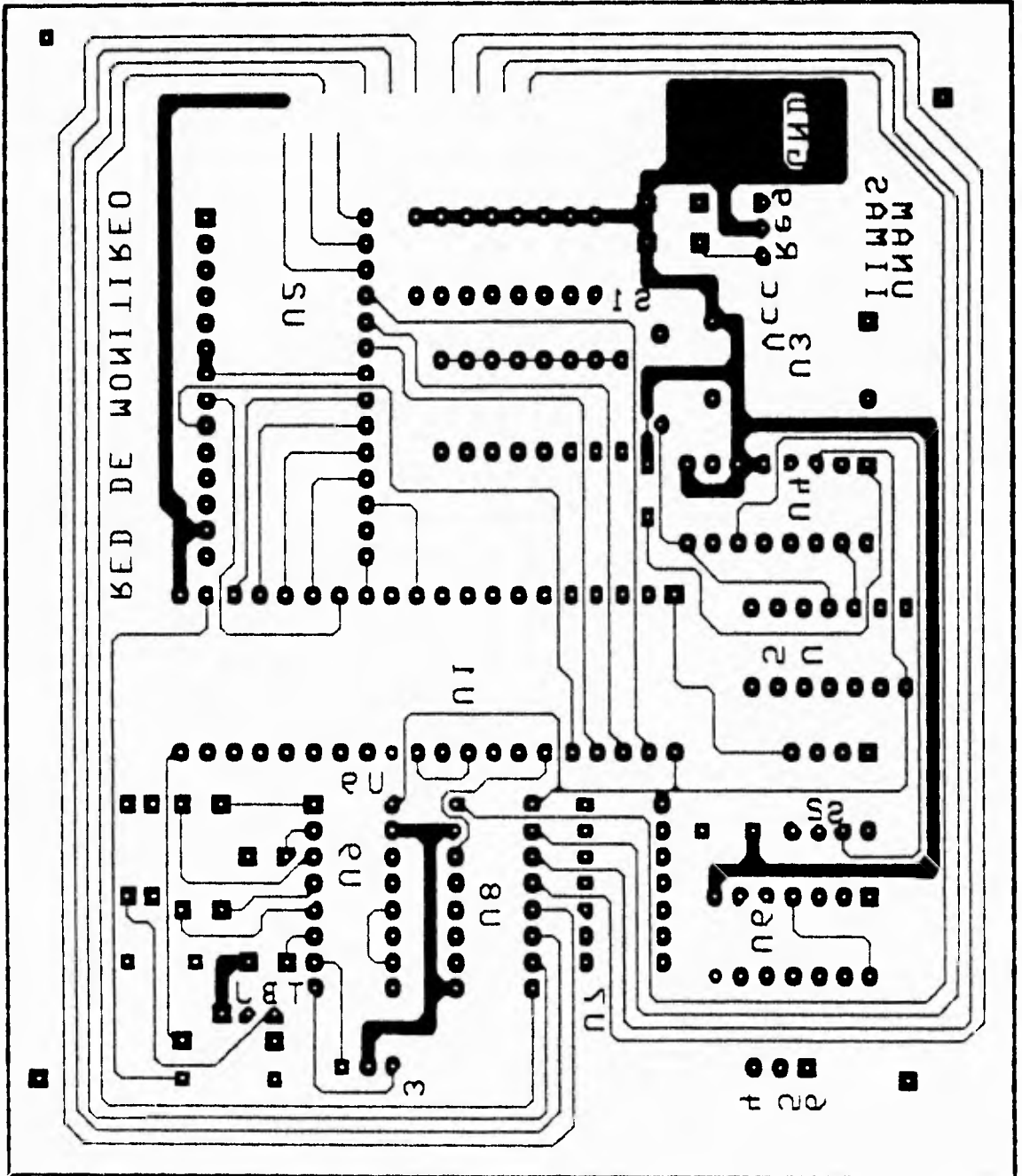
PROCESSOR	READ	WRITE	INTERRUPT (COMMENT)
8080	MEMR	MEMW	INTR (Thru RST Circuit)
8085	$\overline{RD}$	$\overline{WR}$	INTR (Thru RST Circuit)
Z-80	$\overline{RD}$	$\overline{WR}$	$\overline{INT}$ (Thru RST Circuit, Mode 0)
SC/MP	NRDS	NWDS	SA (Thru Sense A)
6800	$VMA \cdot \phi 2 \cdot R/W$	$VMA \cdot \phi \cdot R/W$	IRQA or IRQB (Thru PIA)

### Ordering Information

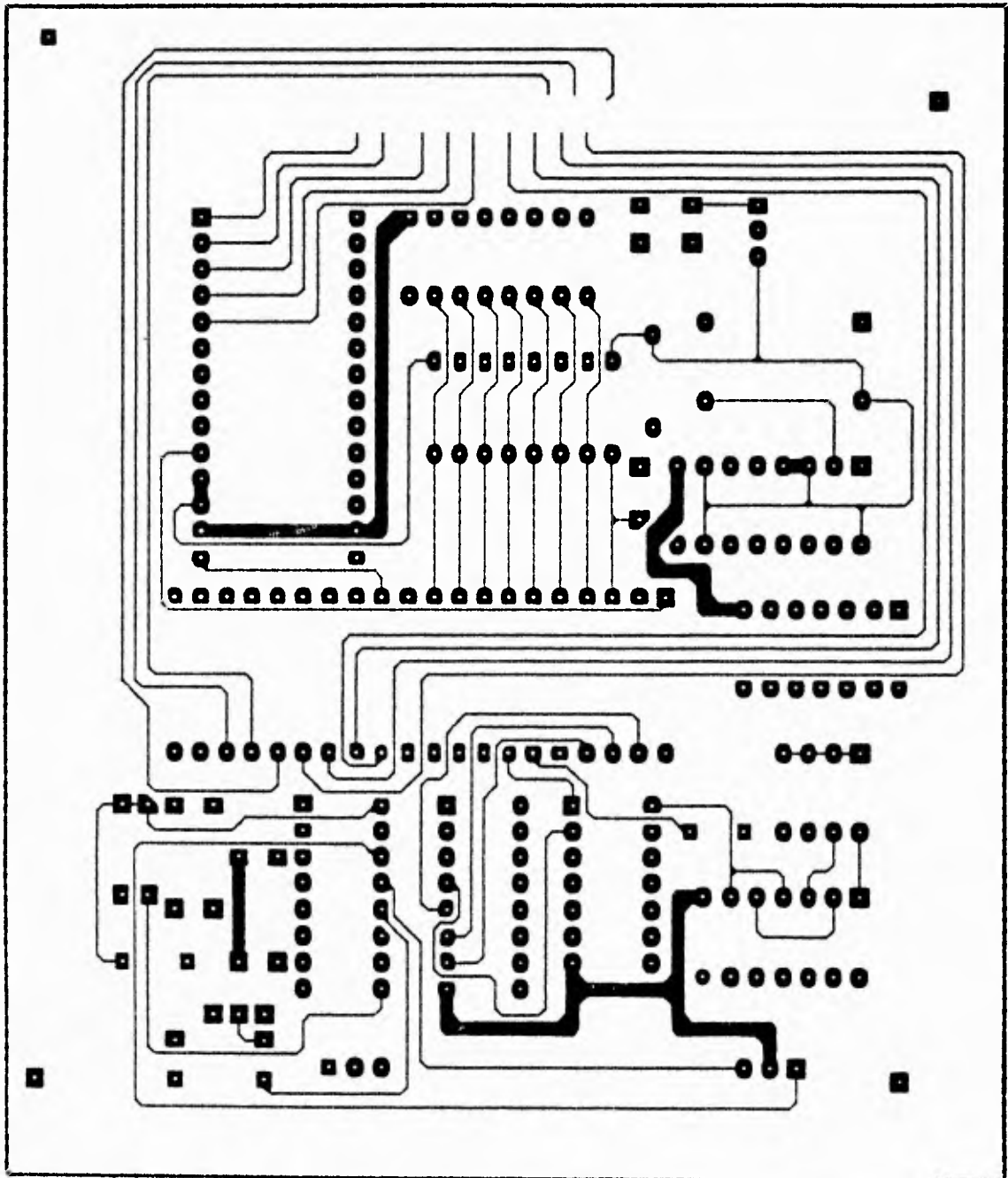
TEMPERATURE RANGE		- 40°C to + 85°C			- 55°C to + 125°C
Error	$\pm 1/2$ LSB Unadjusted	ADC0808CCN	ADC0808CCV	ADC0808CCJ	ADC0808CJ
	$\pm 1$ LSB Unadjusted	ADC0809CCN	ADC0809CCV		
Package Outline		N28A Molded DIP	V28A Molded Chip Carrier	J28A Ceramic DIP	J28A Ceramic DIP



2X artwork 18 Apr 95 20:53:38  
 newlay3  
 vi.4 r1 holes: 246 solder side  
 approximate size: 4.30 by 3.70 inches



2X artwork 18 Apr 95 21:11:00  
newlay2  
v1.4 r1 holes: 246 component side  
approximate size: 4.30 by 3.70 inches



## **GLOSARIO**

---

---

**ASCII:** Es un acrónimo de "American Standard Code for Information Interchange" (Código estándar norteamericano para intercambio de información). ASCII es un código digital adoptado como norma para facilitar el intercambio de datos entre varios tipos de equipos de procesamiento y comunicación de datos.

**BAUD:** En comunicación de datos, unidad de velocidad de la señalización. Es igual al número de eventos de señal por segundo y se utiliza como medida del flujo de datos en serie. En el caso binario, la razón de bauds es igual al número de bits por segundo. Diez caracteres por segundo con un formato de 11 bits tiene una velocidad de transferencia de 110 bauds.

**BIT:** Abreviatura de dígito binario, es el elemento básico de datos de las computadoras y comunicaciones digitales. Un bit puede tener un valor de cero o de uno.

**Bus:** Trayectoria sobre la cual se transfiere la información digital de una o varias fuentes a cualquiera de los destinos. Las fuentes y los destinos pueden encontrarse dentro o fuera de la computadora.

**Comunicación asíncrona de datos:** La forma más popular de intercambio de datos entre computadoras y terminales remotas. La transmisión asíncrona de datos es un método de transferir datos en el que la posición del carácter en el tiempo no resulta tan crítica cuando se conectan las líneas de comunicación. Cada carácter transferido es precedido por un bit de arranque y seguido por un bit de detención, lo que permite que varíe el intervalo entre los caracteres.

**Comunicación de datos:** La comunicación de datos es un término muy amplio que se refiere al intercambio de datos desde un punto a otro.

**Comunicación Síncrona de datos:** Método de transferir datos binarios entre el sistema de cómputo y un dispositivo periférico; los datos binarios son transferidos a una velocidad constante, estando sincronizados el transmisor y el receptor. Al principio cada mensaje o bloque de datos se envían unos caracteres de control para sincronizar el flujo.

---

**Comunicaciones BYSYNC:** Comunicaciones síncronas binarias. Es una forma de comunicación síncrona de datos en la cual se usan diversos caracteres especiales de control para sincronizar el receptor con el transmisor.

**Datos analógicos:** Representación eléctrica de la información en la cual los datos guardan una relación exactamente proporcional con la información real.

**Datos digitales:** Representación eléctrica de la información en la que ésta se codifica en secuencia de elementos binarios o bits. Todas las computadoras modernas utilizan datos digitales. Además, los sistemas telefónicos también recurren cada vez más a la conversión de la voz analógica en datos digitales para mejorar la eficiencia de la comunicación.

**Microprocesador:** Un microprocesador es un solo circuito integrado que tiene todos los elementos de control de una computadora. La adición de elementos de memoria para el almacenamiento de programas convierte al microprocesador en una microcomputadora.

**Multiplexado :** El multiplexado es una técnica que permite dividir un enlace de comunicaciones en dos o más canales.

**Periféricos:** Término general para designar diferentes tipos de equipo electrónico que operan en combinación con la computadora, pero que no forman parte física de ella. Los dispositivos periféricos se encargan típicamente de visualizar datos, almacenar datos de la computadora y regresárselos posteriormente cuando son requeridos, preparar los datos para uso humano, o adquirir datos de una fuente y convertirlos a una forma utilizable para la computadora.

**Registro de corrimiento:** Tipo de memoria serie en la que los datos son recorridos a través de la pastilla desde la entrada a la salida.

**SDLC:** El protocolo SDLC es una disciplina de líneas de comunicaciones asociada con la SNA (Arquitectura de Redes de Sistemas) de IBM que ofrece varias ventajas a los usuarios de redes de datos. El SDLC es el protocolo más reciente de IBM, e inicia, controla, verifica y termina los intercambios de información o líneas de comunicación. Está diseñado para operación dúplex en la que simultáneamente se envían y se reciben datos.

---

**UART ("Universal Asynchronous Receiver Transmitter"):** Receptor/Transmisor universal asíncrono. Circuito integrado utilizado para la comunicación asíncrona de datos.

**USART ("Universal Synchronous-Asynchronous Receiver/Transmitter"):** Receptor/Transmisor universal síncrono/asíncrono. Circuito integrado utilizado para la comunicación tanto síncrona como asíncrona de datos.

**USRT ("Universal Synchronous Receiver/transmitter"):** Receptor/Transmisor universal asíncrono. Circuito integrado utilizado para la comunicación asíncrona de datos.

#### **Terminos médicos.**

**Arritmia Cardíaca:** La característica fundamental del corazón sano al latir rítmicamente en condiciones normales, es decir, de una manera uniforme; esta uniformidad se refiere a la frecuencia de los latidos, la fuerza de los mismos o al intervalo entre latido y latido. Cuando esta uniformidad o ritmo de trabajo se altera transitoria o permanentemente se produce entonces una arritmia cardíaca.

**Arterias:** Conductos de paredes blandas y elásticas que tienen su origen en el corazón y subdividiéndose sucesivamente en troncos cada vez de menor calibre o dando ramas colaterales.

**Apnea:** Significa literalmente "Sin respiración".

**Aurícula:** Cada una de las dos cavidades de la parte superior del corazón, que recibe sangre de las venas.

**Auscultación:** Aplicar el oído o el estetoscopio a ciertos puntos del cuerpo humano para explorar los sonidos en las cavidades del tórax o del abdomen.

---

**Bioeléctricas:** Fuerzas eléctricas generadas por la despolarización y repolarización de las células.

**Catéter** (o sonda vesical): Tubito rígido (de metal o de vidrio) o flexible (de caucho).

**Célula:** Es la partícula más reducida de un organismo, representa el sustrato material mínimo en el que se encarna la propia vida.

**Ectopia:** Es un término científico de derivación griega que quiere decir "fuera de lugar", o sea, "sede anormal, no habitual", se refiere por lo tanto, a cualquier órgano que no ocupa su localización fisiológica normal por anomalía congénita o por haber emigrado de su sede habitual.

**Electrofisiología:** Tiene por objeto el estudio de las funciones de los seres orgánicos y sus acciones eléctricas.

**Embolia:** Es el estado patológico, creado por la oclusión de un vaso sanguíneo a consecuencia de un corpúsculo extraño intravascular.

**Endocardio:** En una sutil membrana que tapiza la superficie interna de las cuatro cavidades cardíacas (las dos aurículas y los dos ventrículos); constituye por lo tanto, la capa más interna de la pared cardíaca.

**Epicardio:** Es la capa más externa de las paredes del corazón.

**Hipotermia:** Es el descenso de la temperatura corporal por debajo de los 36 grados centígrados de la escala termométrica.

**Humoral:** En el cuerpo humano circularían cuatro humores principales o cardinales; la sangre, la bilis, la pituita y el agua; de estos la bilis se regiría por el hígado, la pituita por el cerebro y el agua por el bazo.

**Marcapaso:** Aparato eléctrico que provoca la contracción del corazón cuando ésta no puede realizarse normalmente.

---

**Metabolismo:** Conjunto de transformaciones materiales que se efectúa constantemente en las células del organismo vivo.

**Morfológicas:** Referente a la morfología:

**Morfología:** Parte de la biología que trata de la forma y la estructura de los seres orgánicos. Aspecto general del cuerpo humano.

**Miocardio:** Es el músculo cardíaco, es decir la capa, la capa de fibras musculares estiradas e involuntarias que constituyen casi todo el espesor de las paredes del corazón.

**Músculos:** Son órganos dotados de capacidad contráctil y, por lo tanto, capaces de asegurar la ejecución de todos los movimientos del cuerpo conjunto, o de una de sus partes.

**Papila:** Es un término descriptivo anatómico utilizado para designar los pequeños sobresalientes del cutis o de las mucosas.

**Patología:** Estudio de las enfermedades.

**Patológico:** Relativo a la patología.

**Septum:** Palabra latina que significa pared o tabique que separa dos cavidades contiguas.

**Sublingual:** Son dos glándulas situadas por debajo de la lengua y que segregan saliva.

**Trombosis:** Es el fenómeno patológico de la coagulación intravascular parcial de la sangre en un animal vivo, con formación de un pequeño coágulo llamado Trombo que provoca un trastorno más o menos, grave de la circulación sanguínea.

**Ventrículos:** Se denominan así a algunas cavidades internas de nuestro organismo.



## **BIBLIOGRAFIA**

---

---

- Camarena, G. & González, A. "Conceptos fundamentales de Redes de datos"  
Ed. Fundación Arturo Rosenbluck , México 1992
- Carlson, A. Bruce "Sistemas de Comunicación"  
Ed. Mc Graw Hill, México 1991
- De Silva, Clarence W. "Control and actuadors"  
Ed. Prentice Hall, USA 1989
- Mano, M. Morris "Ingenieria Computacional, Diseño de Hardware"  
Ed. Prentice Hall, México 1993
- Mark, Goodwin "Serial Communications in C and C + +"  
Ed. Mis Press
- Monk, Timothy S. "Serial Communications"  
Ed. Sams Publishing
- Morgan, L. Christopher "Introducción al Microprocesador 8086/8088"  
Ed. Mc Graw Hill, México 1989
- Seyer, Martin D. "RS-232 made esay: connecting computers, printers, terminals and modems"  
Ed. Prentice Hall, USA 1991
- Schild, Herbert "Manual de Referencia de Turbos C y C + +"  
Ed. Mc Graw Hill, México 1992
- Segature "Diccionario médico Taide"  
Ed. Taide, Barcelona 1976

- 
- Silverman, Mark E. "Manual Clínico de electrocardiografía"  
Ed. Mc Graw Hill, México 1985
- Stevens, Roger T. "C Graphics Handbook"  
Ed. Academic Press
- Suros, J "Semiología Médica y Técnica Exploratoria"  
Ed. Salvat Editores S.A., España 1973
- Tocci, Ronald J. "Sistemas Digitales, Principios y Aplicaciones"  
Ed. Prentice Hall, México 1991
- Weiskamp Keith "Power Graphics in turbo C"  
Ed. Wiley
- Williams, Arthur B. "Microprocesadores, Dispositivos perifericos, optoelectrónicos y de interfaz".  
Ed. Mc Graw Hill, USA 1992

**Manuales**

"CMOS Application-Specific Standard IC's"  
Ed. Motorola, USA 1993

"CMOS Logic data"  
Ed. Motorola, Usa 1988

"Fast and LS TTL data"  
Ed. Motorola 1989

**Hemerografia**

**"Build an Ultrasonic Ranger"**  
Computer Craft, June 1992  
PP 47 - 50, 81-83

**"Robotics, Control and Monitoring"**  
Computer Craft, August 1992  
PP 10-20, 80 - 83

**"Build an RS-232 to RS-422 Converter"**  
Computer Craft, May 1993  
PP 76-78

**"Networking Basics"**  
Computer Craft, April 1992  
PP 14-19