

01170

8
25

**DISEÑO DE UNA RED DE MONITOREO DE SIGNOS
VITALES**

ASESOR : M. en I. ESAU VICENTE VIVAS.

ALUMNO : ING. DONALDO ROJAS REYES

DEPFI JULIO DE 1996 UNAM

**TESIS CON
FALLA DE ORIGEN**

**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.

AGRADECIMIENTOS

A mi querida madre y a mi padre.

A mi esposa Rebeca Jiménez H.

A mis hijos Alan y Erick.

A mis hermanos.

A la dirección del M. en I. Esau Vicente Vivas.

A mis maestros.

A mis amigos.

RECONOCIMIENTOS

Se agradece la ayuda y cooperación de todas las personas que apoyaron el desarrollo de este trabajo, como fueron mis compañeros de generación y de aquellos que llegaron después, ya que con su respaldo pusieron un grano de arena para la culminación de esté.

Al M. en I. Esau Vicente Vivas, que con su ayuda invaluable se llegó a la feliz conclusión de este trabajo, por lo que le estoy muy agradecido.

Un reconocimiento a la asesoría de todos mis maestros y en particular al Dr. Francisco García Ugalde, por su gran apoyo moral, ya que siempre conté con su apoyo.

Un agradecimiento al Dr Boris Escalnte R.y al M. en I. Jose Cohen S. por sus correcciones y expresar sus puntos de vista en la elaboración de este trabajo

También quiero agradecer a todo el personal de la DEPMI su gran ayuda para la realización de mis estudios.

ING. DONALDO ROJAS REYES.

INDICE

CAPITULO 1	1
INTRODUCCION	
1.1.-ANTECEDENTES	1
1.2.-MODULO DE DETECCION DE SIGNOS VITALES (MDSV)	2
1.3.-MODULO DE ADQUISICION DIGITAL Y COMUNICACION DE SIGNOS VITALES (MADCSV)	4
1.4.-ENLACE DE COMUNICACIONES	4
1.5.-PUESTO CENTRAL DE ENFERMERAS (PCE)	5
CAPITULO 2	6
DISEÑO DEL MODULO DE DETECCION DE SIGNOS VITALES (MDSV)	
2.1.-SEÑAL DE ELECTROCARDIOGRAMA (ECG)	7
2.1.1.-Sensores y/o transductores empleados	10
2.1.2.-Detección de ECG	13
2.1.3.-Preamplificación	20
2.1.4.-Aislamiento	20
2.1.5.-Amplificación	24
2.1.6.-Acondicionamiento y/o Filtrado	26
2.2.-FRECUENCIA CARDIACA (FC)	28
2.2.1.-Sensores y/o transductores empleados	28
2.2.2.-Detección de FC	29
2.2.3.-Acondicionamiento y/o Filtrado	29
2.3.-FRECUENCIA RESPIRATORIA (FR)	31
2.3.1.-Sensores y/o transductores empleados	31
2.3.2.-Detección de FR	32
2.3.3.-Amplificación	33
2.3.4.-Acondicionamiento y/o Filtrado	34
2.4.-TEMPERATURA	36
2.4.1.-Sensores y/o transductores empleados	38
2.4.2.-Detección de temperatura	39
2.4.3.-Amplificación	40

CAPITULO 3	43
DISEÑO DEL MODULO DE ADQUISICION DIGITAL Y COMUNICACION DE SIGNOS VITALES (MADCSV)	
3.1.-ACOPLAMIENTO CON EL SISTEMA DE ADQUISICION DE DATOS	44
3.1.1.-Etapas Indicadoras del sistema	44
3.1.2.-Acoplamiento del MDSV con el MADCSV	47
3.2.-DESCRIPCION DEL MICROCONTROLADOR MC68HC11	48
3.3.-ARQUITECTURA DEL MADCSV	51
3.4.-MUESTREO DE SIGNOS VITALES	54
3.5.-DESPLIEGUE DE SIGNOS VITALES	60
3.6.-TRANSMISION DE DATOS	60
 CAPITULO 4	 63
INTERACCION ENTRE EL MODULO BASICO DE MONITOREO (MBM) Y EL PUESTO CENTRAL DE ENFERMERAS (PCE)	
4.1.-TRANSMISION SERIE DEL MBM	63
4.2.-ARQUITECTURA PARA INTERCONECTAR MBM'S EN RED	66
4.3.-PROTOCOLOS DE COMUNICACION Y PROGRAMACION PARA LA INTERACCION ENTRE EL MBM Y EL PCE	71
 CAPITULO 5	 76
PROGRAMACION INTERACTIVA ELABORADA PARA EL PCE	
5.1.-PROGRAMACION INTERACTIVA CON EL PERSONAL MEDICO	77
5.2.-PROGRAMACION PARA EL MONITOREO DE PACIENTES	79

CAPITULO 6	85
PRUEBAS REALIZADAS, CONCLUSIONES Y RECOMENDACIONES	
6.1.-PRUEBAS MODULARES	85
6.1.1.-Pruebas al MDSV	85
6.1.2.-Pruebas realizadas al MADCSV	86
6.1.3.-Pruebas a la programación del PCE	86
6.2.-PRUEBAS REALIZADAS CON SIMULADOR DE ECG	88
6.3.-PRUEBAS REALIZADAS CON PACIENTE	90
6.4.-CONCLUSIONES Y RECOMENDACIONES	92
REFERENCIAS	97
APENDICE A	99
APENDICE B	106

CAPITULO 1

INTRODUCCION

1.1.-ANTECEDENTES

La Ingeniería es una disciplina que a lo largo de la historia a hecho grandes aportaciones al desarrollo de la humanidad, logrando avances científicos y tecnológicos en las áreas de electrónica, computación, comunicaciones, etcétera. Pero a la vez también se ha logrado relacionar con otras áreas, como la medicina, así tenemos que la interacción de la Ingeniería y la Medicina han dado origen a una nueva rama del conocimiento conocida como Bioingeniería. Este campo ha proporcionado una serie de descubrimientos y adelantos benéficos para el hombre, pero además ha originado el nacimiento de una nueva rama que es la Instrumentación Médica la cual se encarga de diseñar y desarrollar todo tipo de instrumentos y aparatos que son de gran utilidad en hospitales, clínicas, ambulancias, etcétera. Tan grande ha sido su impulso y desarrollo tecnológico en los últimos años, que ha beneficiado a pacientes en todas partes del mundo por medio de equipos de alta tecnología, ejemplos de ello son la Resonancia Magnética Nuclear (RMN), la Tomografía Axial Computarizada (TAC), la Tomografía por Emisión de Positrón (TEP), la Cirugía por Rayo Láser, etcétera.

La infraestructura que poseen los países más desarrollados en cuanto ha diseño de equipo biomédico es sumamente fuerte, lo cual los ubica a la vanguardia tecnológica. En México el desarrollo tecnológico en Bioingeniería avanza lenta pero constantemente, varias instituciones han iniciado investigaciones que las ha llevado a desarrollar prototipos con tecnología propia eliminando parcialmente la dependencia extranjera. Esta dependencia tecnológica ha sido la principal preocupación de diversas instituciones (UAM-I, CEDAT, IPN, UNAM, ETC.) [1], [2] para realizar investigaciones y diseñar sistemas equivalentes a los equipos comerciales actuales, considerando que el costo de ellos es sumamente elevado. En estas instituciones se han realizado diversos diseños de equipo biomédico, de los cuales muchos proyectos terminados se han encaminado a la detección de determinados signos vitales. Por ejemplo algunos se enfocan a la frecuencia cardiaca, (UAM - IZTAPALAPA), otros detectan frecuencia respiratoria (CINVESTAN-IPN) y otros la temperatura (UAM - IZTAPALAPA), lo cual provoca que su campo de acción y comercialización se limite, originando además la necesidad de tener varios

equipos para obtener un registro general, aspecto que en cuestión de eficiencia, funcionamiento e interconectividad se ve fuertemente afectado.

Ante el panorama planteado se detecta la necesidad de un sistema capaz de detectar y amplificar signos vitales básicos como la señal del electrocardiograma (ECG), Frecuencia Cardíaca (FC), Frecuencia Respiratoria (FR) y Temperatura, que pueda operar como un equipo independiente, es decir como un Módulo Básico de Monitoreo (MBM), y que pudiera conectarse en red con otros equipos similares, generando salidas las cuales se reportaran a un puesto central de enfermeras (PCE). De esta problemática nace el proyecto de la presente tesis, es decir el "**Diseño de una Red de Monitoreo de Signos Vitales**".

Estos sistemas existen actualmente en hospitales de países desarrollados y en México sólo algunas instituciones disponen de este tipo de equipos, dado que su compra, instalación, puesta en marcha y mantenimiento implican costos excesivamente elevados, siendo entonces prohibitivo para la mayor parte de las instituciones médicas de países como el nuestro.

El equipo desarrollado para esta tesis fue diseñado para monitorear los signos vitales antes mencionados, previamente amplificados y filtrados por un módulo analógico. Adicionalmente, en un módulo digital las señales son muestreadas, procesadas y almacenadas temporalmente en RAM, para ser enviadas posteriormente al PCE para su despliegue.

El sistema está formado básicamente por cuatro etapas como se muestra en la figura 1.1, las cuales se describen a continuación.

1.2.-MODULO DE DETECCION DE SIGNOS VITALES (MDSV)

Este módulo está compuesto principalmente por sensores y/o transductores, amplificadores y filtros, encargados de detectar, amplificar y filtrar signos vitales del paciente. Este módulo representa una parte fundamental en la detección de signos vitales por lo que su diseño estuvo asesorado por personal médico especializado (cardiólogos, cirujanos, enfermeras, etcétera)[3]. Básicamente cuenta con etapas de selección de derivación [4], [5], preamplificación, aislamiento, amplificación y filtrado; el equipo fue diseñado para cumplir con las normas de construcción de equipo de monitoreo cardíaco AAMI [6]

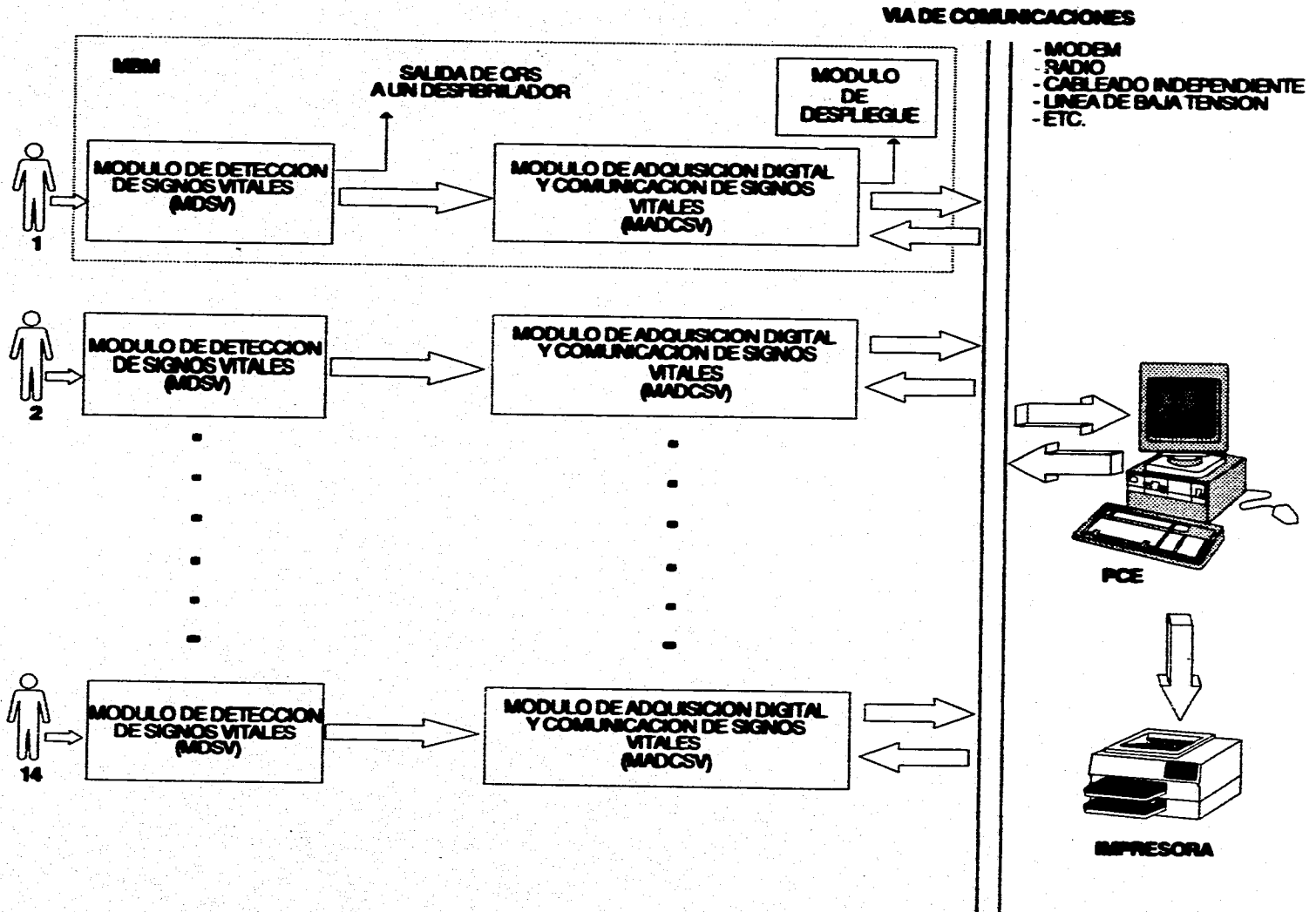


Figura 1.1 Diagrama de bloques de la "Red de Monitoreo de Signos Vitales"

(Association for the Advancement of Medical Instrumentation). En el capítulo 2 se describe el diseño del MDSV.

1.3.-MODULO DE ADQUISICION DIGITAL Y COMUNICACION DE SIGNOS VITALES (MADCSV)

El MADCSV es la parte digital del sistema y está constituido principalmente por el microcontrolador MC68HC11E9, del cual se hace uso extensivo de sus periféricos internos para reducir el número de componentes requeridos en el equipo. En este módulo se lleva a cabo el control local del sistema, es decir, se controla el convertidor analógico-digital (C/A/D) para muestrear los signos vitales; se direcciona una memoria RAM para el almacenamiento temporal de los mismos, de tal forma que funcione como una memoria de video y se pueda integrar en el futuro un sistema de despliegue adecuado, como un monitor convencional de TV o una pantalla de plasma; también se monitorean los puertos de entrada, para la detección de electrodos sueltos en el paciente; se controla el puerto serie para la recepción y transmisión de información entre el MADCSV (el cual forma parte del MBM) y el PCE en caso de encontrarse conectados en red. Además se efectúa el "procesamiento" de las señales utilizando las técnicas más precisas para el cálculo de la FC y FR. En el capítulo 3 se detalla la arquitectura diseñada para el MADCSV, así como su funcionamiento, mientras que en el capítulo 4 se presentan los algoritmos de programación para el control e interacción del sistema con el PCE.

1.4.-ENLACE DE COMUNICACIONES

Cuando el MBM (MDSV y MADCSV interconectados) se encuentra conectado en red es necesario un medio de comunicación con el PCE. Tomando en cuenta que éste no siempre se encuentra en el mismo piso o se encuentra alejado de los pacientes, existen diversos medios de comunicación para resolver el problema, como por ejemplo un sistema de radio, una línea telefónica, o bien, un cableado adicional para la conexión entre equipos; sin embargo uno de los objetivos de esta tesis es el bajo costo y la modularidad del equipo, por lo que ninguna de las soluciones anteriores resuelve el problema, por resultar demasiado costosas, por requerir permisos para la transmisión y otras por que hacen al sistema poco transportable. Debido a estos problemas se presenta como una opción el enlace de comunicaciones vía red eléctrica de baja tensión, ya que ésta se encuentra

instalada por todo el inmueble y sólo requiere de un módem especial de bajo costo y diseño versátil para la transmisión serie de datos al PCE. Debe subrayarse que el enlace de comunicaciones, así como el diseño de estos módems quedan fuera de esta tesis por lo que sólo se mencionará la mejor opción y se deja que la selección del enlace más adecuado se realice de acuerdo con las necesidades del usuario.

1.5.-PUESTO CENTRAL DE ENFERMERAS (PCE)

Para el monitoreo global (varios MBM) o individual de pacientes conectados a la red se utiliza una computadora central ubicada en el PCE. En la presente tesis se desarrollaron diversos programas para desplegar signos vitales de hasta 14 MBM'S; esto debido a que son los que se pueden visualizar en una pantalla. Para esta cantidad de equipos se grafican solamente 150 muestras de ECG por paciente, el número de muestras está determinado por la resolución del monitor (VGA) de la computadora central. La resolución es relativamente baja, sin embargo, el objetivo del monitoreo global es detectar la ausencia de ECG y el envío de alarmas al registrarse signos vitales anómalos en el paciente, los cuales en caso de presentarse se pueden monitorear de manera individual, utilizando para ello 250 muestras por segundo. Además de los programas para manejar las gráficas descritas anteriormente, se diseñaron otros programas en el PCE para interactuar con el personal médico, los cuales proporcionan información de manera amigable y de fácil acceso. Los programas utilizan menús, con los cuales se permite introducir datos de nuevos pacientes para su registro, fijar condiciones de alarma para todas las variables, monitorear pacientes de manera global (para detección de alarmas principalmente) o de manera individual (graficación de signos vitales) y permite también la graficación en papel de los signos vitales. En el capítulo 5 se presentan en detalle los algoritmos para la programación del PCE.

Por último, en el capítulo 6 se describen las pruebas realizadas al equipo de monitoreo de signos vitales, así como las conclusiones y recomendaciones para diseños posteriores.

CAPITULO 2

DISEÑO DEL MODULO DE DETECCION DE SIGNOS VITALES (MDSV)

Dada la particularidad del cuerpo humano es necesario elegir dispositivos adecuados que detecten los fenómenos o cambios que se desarrollan dentro del cuerpo. Estos dispositivos son comúnmente conocidos como sensores y/o transductores. Existen infinidad de éstos, los cuales pueden clasificarse por su forma de funcionamiento o por los materiales con los que son fabricados, sin embargo, no todos son adecuados para un proyecto de esta naturaleza dado que es importante la rapidez y la precisión de su respuesta. Para poder obtener cualquier tipo de información referente a algún fenómeno fisiológico es necesario conocerlo a detalle. Hay que saber qué procesos se desarrollan en el interior del cuerpo humano, qué comportamiento poseen y de qué manera se manifiestan. Tomando en consideración esto, tenemos que es posible obtener mediciones partiendo de las funciones y reacciones sistemáticas que se interrelacionan en zonas específicas del cuerpo. En lo anterior basa su desarrollo el monitoreo de signos vitales. Por lo tanto es necesario conocer como son y que caracteriza a las señales que originan los signos vitales.

En el campo clínico existe una clasificación que divide a los transductores en **invasivos y no invasivos**. Los transductores invasivos son denominados así debido a que para que éstos puedan detectar información es necesario que penetren la piel del paciente, siendo esta situación la que provoca que éstos no sean comúnmente utilizados debido a lo incómodo y traumático que resulta su empleo para las personas que se vayan a examinar. Los transductores no invasivos son aquellos que no requieren de ninguna incisión en la superficie de la piel pues con sólo colocarse en la superficie registran de manera indirecta [4] los eventos desarrollados dentro del organismo.

En la tabla número 1 se hace referencia a los órganos del cuerpo que son de interés para nosotros por las características del diseño realizado, anticipándonos un poco a lo que será el desarrollo de los siguientes capítulos.

Cuando hablamos de detección de signos vitales es posible definir que órganos y áreas del cuerpo humano nos proporcionan ésta información. Hay que analizar y

comprender que tipo de reacciones son originadas por estos órganos para saber de que manera es posible detectarlas. Este punto es muy importante pues a partir de éste hecho sabremos que tipo de sensores son los que se pueden utilizar. El cuerpo humano produce fenómenos fisiológicos los cuales podemos medir, analizar y representar. Pero estos procesos presentan una gran complejidad en sus orígenes, aspectos que son de gran importancia pero que su explicación cae fuera de este trabajo, debido a que sería necesario tener conocimientos más profundos de fisiología humana,

TABLA 1

SISTEMA	ÓRGANO	UNIDAD	INSTRUMENTO	REGISTRADOR
Cardíaco	Corazón	Ancho de pulso Forma de onda	Monitor de arritmia Desfibrilador Estetoscopio	Resucitador Aplicador Registro de pulsos
Pulmonar	Pulmones	Respiración Volumen periódico Resistencia torácica	Espirómetro Respirador Nebulizador	Bombas de acción Fluoroscopio
Circulatorio	Sangre	Presión Rango de flujo Contenido de oxígeno Contenido de CO Acidez.	Presión Flujo Temperatura Medidor PH	Coagulómetro Hemoglobiómetro Cromatógrafo Electropéresis
Tegumentario	Piel Orificios Cabello	Toque Acidez Peso Textura	Pletismocopia Termografía Reflexómetro Depilador	

2.1.- SEÑAL DE ELECTROCARDIOGRAMA (ECG)

Una de las señales más importantes que posee cualquier monitor de signos vitales es sin duda alguna la señal de ECG. Por lo que daremos algunas características básicas de esta señal. El corazón o músculo cardíaco tiene dos privilegios no compartidos por otras células excitables, el de asumir los procesos de activación miocárdica y su automatismo, que es su capacidad para engendrar latidos por si mismo. Así tenemos que la adecuada circulación de la sangre a través del sistema cardiovascular depende directamente de una adecuada secuencia de contracciones de las cámaras internas del corazón. En la figura 2.1 se muestra al músculo cardíaco. Estas contracciones son originadas por impulsos eléctricos los cuales son generados por el propio corazón, y es ésta actividad eléctrica la

que nos interesa detectar. Esto sería posible si introdujéramos electrodos que crucen las paredes torácicas lo cual implicaría la utilización de sensores invasivos. Dicha técnica sería sumamente costosa y peligrosa, razones por las cuales se utilizan preferentemente electrodos externos debido a la capacidad que tienen los tejidos que conforman el cuerpo humano para conducir los potenciales eléctricos; los cuales son captados y registrados de manera indirecta o derivada empleando métodos convencionales desde puntos situados en la vecindad de los órganos explorados.

Una de las características más importantes del corazón, es la de mantener la sangre en movimiento por todo el sistema cardiovascular. Su representación puede considerarse como si estuviese conformado por dos bombas que actúan mecánicamente, una completamente independiente de la otra, de tal forma que sus movimientos estén en fase con el llenado y la expulsión de fluido sanguíneo. La acción de las válvulas es puramente pasiva, es decir, la diferencia de presión a los lados de la válvula ocasiona que ésta se abra o cierre automáticamente.

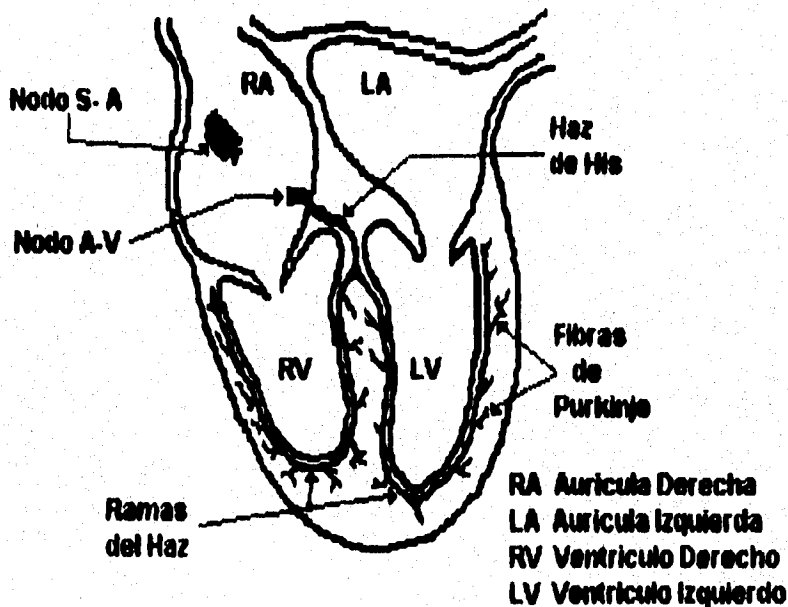
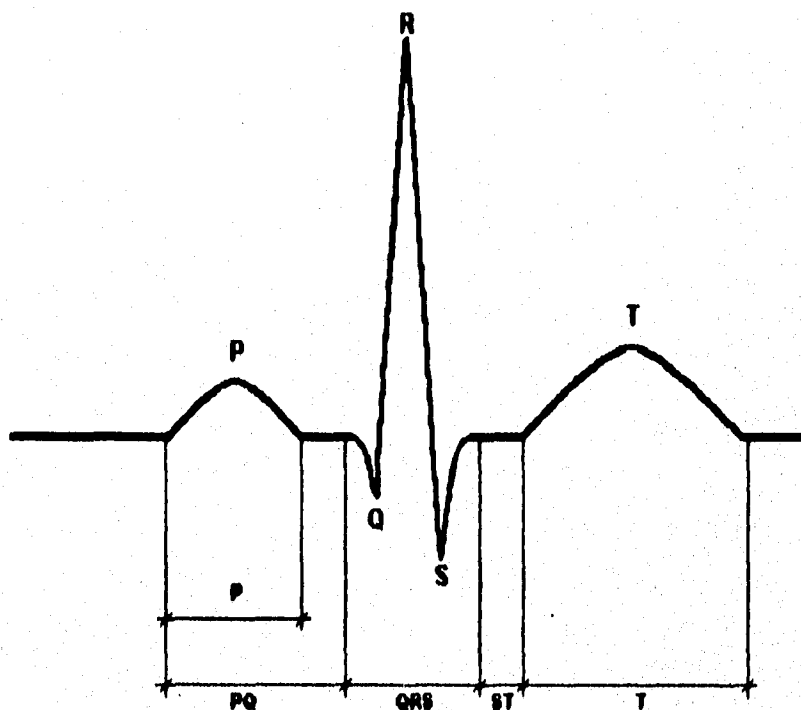


Figura 2.1.- Cámaras que conforman al corazón.

Como hemos visto, la actividad del corazón implica el desarrollo de procesos bioquímicos y musculares que generan potenciales eléctricos. Éstos eventos tienen una

representación gráfica conocida como electrocardiograma (ECG). La forma de onda del ECG consta de varias regiones, en las cuales se describe la actividad de un ciclo cardíaco. Estas regiones consideran a la forma de onda P,Q,R,S,T en su conjunto como un complejo. En la figura 2.2 podemos observar la forma de onda que posee la señal del corazón, así como el tiempo de duración que poseen cada uno de los segmentos que la conforman.



INTERVALO	AMPLITUD (mV)	DURACIÓN (S)
Onda P	0.1 a 0.3	0.09 a 0.11
Complejo QRS	0.5 a 1.5	0.06 a 0.10
Onda T	0.1 a 0.5	0.09 a 0.15
Segmento P-R	Menos de 0.1	0.04 a 0.08
Segmento ST	Menos de 0.1	0.12 a 0.15

Figura 2.2.- Forma de onda y segmentos característicos de la señal de ECG.

2.1.1.-Sensores y/o Transductores Empleados

Los potenciales eléctricos son generados por flujos de corrientes iónicas, los cuales requieren de una conversión a potenciales eléctricos antes de ser registrados por nuestra circuitería. El dispositivo que se encarga de convertir potenciales iónicos en potenciales eléctricos es el "electrodo".

Un electrodo es un elemento metálico que se pone en contacto con la piel, existiendo entre éstos dos un elemento conductor o gel para formar un electrolito. El potencial obtenido en el electrodo es generado por electrones que atraviesan el electrolito y entran al metal figura 2.3, iniciando una distribución de cargas la cual varía en función de la posición. Ésta distribución de cargas es similar a la que se da en un capacitor iniciando positivo sobre una superficie y negativa en la otra. La distribución de cargas origina un potencial eléctrico llamada potencial de la semicélula [5]. El proceso que tiene lugar en la interfase metal - tejido es de naturaleza electroquímica interviniendo fenómenos muy complejos tales como efectos de superficie, formación de dobles capas iónicas, polarizaciones, no linealidades, etcétera.

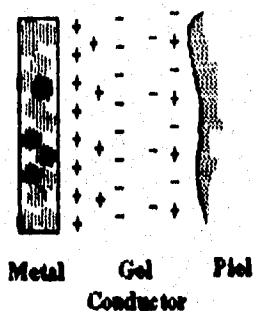
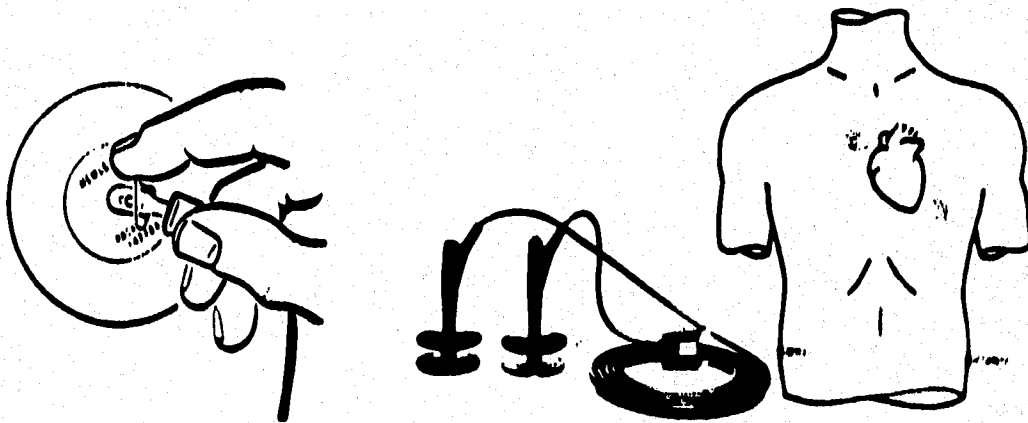


Figura 2.3.- Interfase Metal - Tejido.

Para realizar éstas mediciones es necesario que los electrodos sean lo más parecidos, de tal manera que no se generen voltajes producidos por las diferencias existentes entre éstos, además de que los materiales y formas de fabricación pueden provocar fluctuaciones que aparecerían como ruido en la señal detectada. Se ha encontrado que el electrodo que presenta mayor estabilidad es el conocido con el nombre de electrodo de plata-cloruro de plata.[5]



a) Bandas ,electrodos de placa,electrodos de copa (perillas precordiales), cable de paciente, electrodos tipo clip



b) Electrodo desechables, electrodos para desfibrilación

Figura 2.4.- Diferentes tipos de electrodos para electrocardiografia y monitoreo

Para la detección del ECG podemos mencionar que existen tres tipos principales de electrodos los cuales se pueden observar en la figura. 2.4. La primera es una placa de cloruro de plata con una terminal de conexión y un pivote, el cual se sujeta con una banda elástica con el objeto de mantener adherido firmemente éste a la piel (usualmente empleado en electrocardiografía). Estos disponen de una amplia superficie de contacto lo que añadido a la utilización de gel conductor provocan un buen contacto. Desafortunadamente, para periodos largos no son muy eficientes debido a que pueden llegar a perder contacto con la piel o secarse el gel, ocasionando interferencia, pérdida o saturación de la señal de ECG. Otro tipo de electrodo del cual disponemos es el llamado de copa, que se emplea comúnmente para la obtención de las derivaciones precordiales y el cual se coloca en el pecho. Este consiste de una media esfera de metal hueca ayudada de una esfera plástica la cual al ser presionada provoca una succión del dispositivo hacia la piel logrando adherirse a ésta. La desventaja de este electrodo es que únicamente la orilla o el aro de la copa es el que hace un buen contacto con la superficie de la piel lo cual puede dar origen a interferencia debido a la poca superficie que está en contacto con la piel. Por último tenemos al electrodo que es el más utilizado en la actualidad, conformado de tal manera que la placa metálica no está en contacto directo con la piel sino que usa gel conductor entre el metal sensor y la piel [7] el cual transmitirá la señal al electrodo. Este dispositivo está compuesto de un disco pequeño de plata dorada, que a su vez incorpora un botón donde se conecta el cable. Este tipo de electrodo, figura 2.5, agrega alrededor del disco de plata un material adhesivo que permite unir a éste con el pecho del paciente.

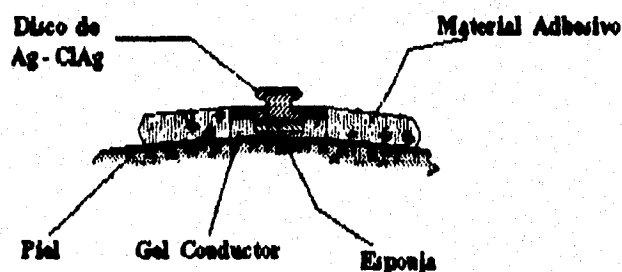


Figura 2.5.- Electrodo para electrocardiografía.

El gel y el disco se encuentran aislados del medio ambiente impidiendo de ésta forma que el gel se seque, evitando así que pierda contacto con la piel. Este tipo de

electrodos son de fabricación sencilla y aptos para periodos largos de monitoreo además de ser de bajo costo, siendo el único inconveniente el que son desechables.

2.1.2.- Detección de ECG

La detección de la señal de ECG se lleva a cabo de manera indirecta o derivada, a través de electrodos. El sistema está diseñado con la intención de detectar las doce señales correspondientes a diferentes posiciones y combinaciones que guardan los electrodos. Cada una de estas doce señales es conocida como derivación y es el resultado de la exploración indirecta del corazón desde distintos planos, obtenidas a partir de cinco electrodos. Las primeras tres derivaciones se encuentran clasificadas como bipolares, las siguientes tres son conocidas como unipolares o aumentadas mientras que las seis siguientes corresponden a las llamadas precordiales.

Las primeras derivaciones (que datan del siglo XIX) fueron descritas por Einthoven y se les conoce como derivaciones estándares o clásicas [8] Se basan en una concepción de bipolaridad (polo positivo menos polo negativo) de allí que se les conozca como bipolares. Para el registro de estas derivaciones clásicas es común utilizar las extremidades del cuerpo para colocar los electrodos así como también colocarlos en el pecho formando un triángulo.

Las siguientes tres derivaciones son conocidas como unipolares o aumentadas y son obtenidas a partir de un electrodo, que tiene como polo contrario (para establecer un dipolo) un potencial que no es exactamente igual a cero, pero que se aproxima mucho a esa magnitud, por lo que su fuerza es despreciable. Éstas reciben el nombre de aumentadas para indicar que los potenciales eléctricos, a causa de su pequeñez original, son ampliados para su mejor observación.

Finalmente, las derivaciones conocidas como precordiales son seis y se obtienen al desplazar un electrodo explorador en seis diferentes puntos del pecho. Estas derivaciones permiten el registro de potenciales que escapan a las seis derivaciones anteriormente citadas; abarcan el tórax, partiendo de su lado derecho y llegando hasta la línea axilar media, o sea, rodean al corazón a manera de un semicírculo.

Al igual que las aumentadas éstas resultan también ser unipolares, siendo uno de los polos el electrodo explorador del pecho y el otro un potencial despreciable generado por el promedio de los potenciales existentes en los miembros o extremidades.

Las primeras seis derivaciones electrocardiográficas se pueden obtener básicamente con tres electrodos y en el caso de las precordiales se utilizan cuatro. Pero el quinto electrodo que se mencionó en párrafos anteriores se encuentra referido a un circuito especial que se conoce como "circuito de pierna derecha" el cual se explicará más adelante. En la figura 2.6. se muestra la localización de los electrodos y las siglas que por estándar internacional tienen asignadas para su identificación.

La etapa de detección de la señal del corazón podemos resumirla en el diagrama de bloques mostrado en la figura 2.7. La primera parte del circuito consiste en tener una línea de entrada para cada electrodo (cable de paciente). Todos estos pasan primero por una etapa de protección (dicha etapa es para el resguardo del circuito de detección) debido a que en salas de cuidado intensivo el paciente puede requerir de la aplicación de voltajes de gran magnitud provenientes de un equipo desfibrilador (que en el peor de los casos puede ser de 5000 volts o un equivalente a 400 Joules) cuando entra en paro cardíaco.

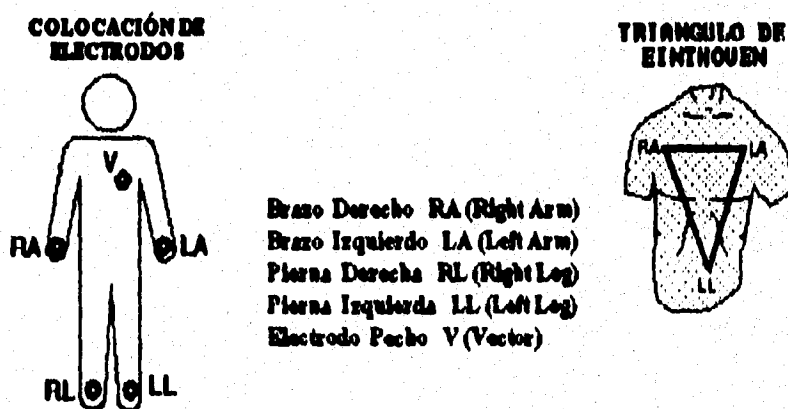


Figura 2.6.- Posición de los electrodos y triángulo de Einthoven.

Posteriormente la señal pasa por una fase de acoplamiento que también servirá de protección para el paciente. Esta etapa minimiza las señales parásitas y el ruido generado por las diferencias que puedan existir en los electrodos, tanto en el material que los constituyen como en su proceso de fabricación, estas señales parásitas suelen conocerse con el nombre de offset de los electrodos y corrientes de polarización, que son generadas en la interfase metal-electrolito.

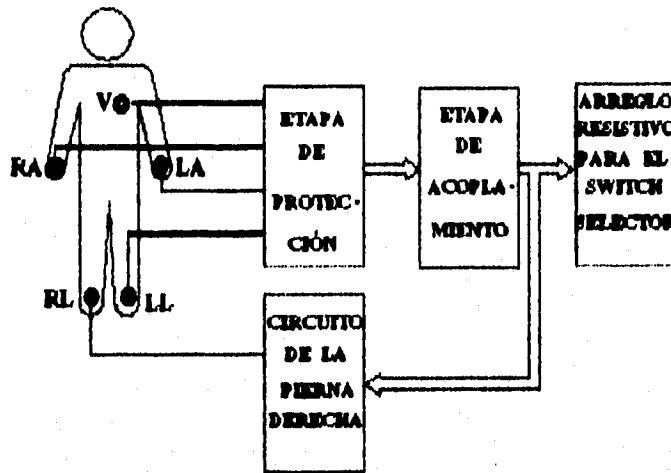


Figura 2.7.- Diagrama de bloques de la etapa de detección del ECG.

De acuerdo con estándares internacionales en el ámbito de la instrumentación médica (AAMI) se considera que el nivel seguro de corriente que puede llegar a admitir el músculo cardíaco, de manera directa sin provocar fibrilación, debe ser inferior a 10 microamperes [6]. Establecida esta condición tenemos que ninguna clase de instrumentación ni sistema de adquisición de información debe exceder estos parámetros. Esta situación nos hace tratar el manejo de corrientes con mucha delicadeza para mantenerlas dentro de un intervalo seguro. Este tipo de corrientes tiene sus orígenes por lo general cuando por alguna causa se sufren rompimientos accidentales de los aterrizajes de dispositivos, los cuales originan de manera directa o indirecta enlaces de corriente alterna que circulan rumbo al cuerpo del paciente.

Estos problemas pueden ser superados cuando se decide proteger al paciente mediante el uso de sondas o cables de alta resistencia y circuitos amplificadores que posean una alta impedancia de entrada. Como complemento a la etapa de detección de ECG se utiliza un circuito comúnmente empleado en electrocardiografía, el cual además de reforzar la seguridad del paciente permite reducir en gran medida el ruido, este arreglo se conoce como "circuito de la pierna derecha" [9].

"El circuito de pierna derecha" es un arreglo que fuerza la conexión de referencia localizada en la pierna derecha del paciente para que alcance un voltaje que sea igual a la suma de los voltajes proporcionados por los otros electrodos. De esta forma es posible llevar el voltaje en modo común a un valor sumamente bajo, además de reducir la interferencia de 60 Hz, logrando con esto que no se afecten las formas de onda de las señales de ECG.

Adicionalmente este circuito proporciona protección contra sobrecargas eléctricas ya que si algún voltaje de grandes proporciones llegase a aparecer entre el paciente y la tierra como resultado de una fuga eléctrica o alguna otra causa, el sumador se satura. De acuerdo con lo anterior se logra aislar al paciente del gabinete o de la tierra a la que se encuentre referido el amplificador de la señal de ECG y en casos prácticos se limita cualquier flujo de corriente que llegase a circular del sistema detector al paciente a menos de 10 microamperes.

Para detectar las 12 derivaciones definidas es necesario que éstas pasen por un arreglo de resistencias (Red Wilson) las cuales se encargan, junto con la posición de los electrodos, de dar la amplitud y forma característica al ECG

Las derivaciones se obtienen con la combinación de los electrodos fijados en las extremidades del cuerpo, las cuales pasan a través de unas resistencias (Red Wilson) siguiendo el modelo matemático establecido por las ecuaciones que a continuación se describen. Para obtener las primeras tres derivaciones se tiene:

- Primera Derivación (I) = LA - RA
- Segunda Derivación (II) = LL - RA
- Tercera Derivación (III) = LL - LA

Como se puede observar en la figura 2.8, cada par de electrodos van conectados a las entradas del amplificador a través de una resistencia de un valor de 5000 ohms para las tres primeras derivaciones.

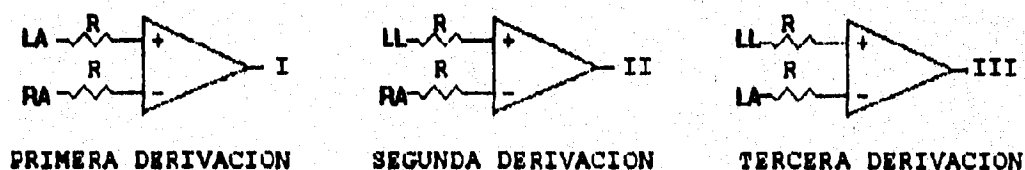


Figura 2.8. - Obtención de las derivaciones de Einthoven.

Las derivaciones unipolares (aumentadas y precordiales) se obtienen entre un electrodo y otro "indiferente", conocido como "terminal central" donde existe un promedio de las señales de los electrodos conectados a las extremidades con resistencia de igual valor.

Las mediciones hechas con respecto a ésta terminal llevan antepuesta la letra "V" de voltaje. La detección unipolar de LA con respecto al electrodo indiferente recibe el nombre de derivación VL, mientras que la de RA y LL son llamados VR y VF respectivamente.

Los potenciales de ECG obtenidos de cualquiera de los electrodos conectados a las extremidades con respecto a la terminal central resultan inconvenientemente pequeños. Para superar esta dificultad es necesario desconectar el electrodo del cual se desea obtener información de la red resistiva, logrando un incremento del 50% en el voltaje medido sin que se presente un cambio muy significativo en la forma de onda de la señal resultante. Las derivaciones unipolares obtenidas de esta forma son conocidas como derivaciones unipolares aumentadas, asignándoseles las siglas aVR, aVL y aVF, la "a" significa aumentadas. Cualquiera de éstas, es el producto de un electrodo con respecto al promedio de las señales proporcionadas por los otros dos. Representando de manera matemática las derivaciones aumentadas se obtiene lo siguiente:

$$aVR = RA - (LA + LL)/2$$

$$aVL = LA - (LL + RA)/2$$

$$aVF = LL - (LA + RA)/2$$

El incremento del 50% que se comentó se logra aumentando al doble el valor de las resistencias que conforman la terminal central las cuales tendrán ahora una magnitud de 10,000 ohms. A continuación se muestra la forma de obtener éstas derivaciones relacionando electrodos, resistencias y amplificador. En la figura 2.9 se observan las combinaciones de las diferentes extremidades para obtener una derivación dada.

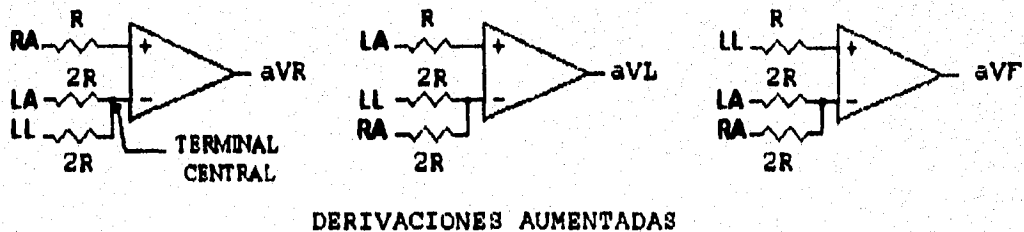


Figura 2.9.- Obtención de las derivaciones aumentadas.

Las derivaciones restantes se obtienen a partir de electrodos que están conectados en seis diferentes posiciones en el pecho, los cuales proporcionan información con respecto a una terminal central. Esta terminal, conocida también como Terminal Central de Wilson [10], es la misma que la mencionada para las derivaciones unipolares, donde los tres electrodos de las extremidades están conectados a la terminal mediante resistencias que ahora tendrán un valor de 15000 ohms. Los electrodos que se colocan en las seis diferentes posiciones del pecho son conocidos como precordiales unipolares y son reconocidos como V1 a V6. Si representamos como un modelo matemático el proceso de obtención de los precordiales este quedaría como se muestra a continuación:

$$V1 = V1 - (LA + RA + LL)/3$$

$$V2 = V2 - (LA + RA + LL)/3$$

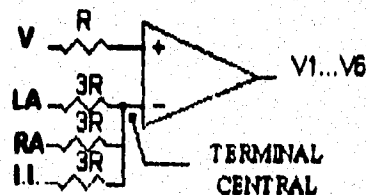
$$V3 = V3 - (LA + RA + LL)/3$$

$$V4 = V4 - (LA + RA + LL)/3$$

$$V5 = V5 - (LA + RA + LL)/3$$

$$V6 = V6 - (LA + RA + LL)/3$$

De acuerdo con lo anterior las entradas del amplificador serán las que se muestran en la figura 2.10. La entrada del amplificador recibirá las señales detectadas en el electrodo de exploración en las diferentes posiciones que se vaya colocando en el pecho. Estas detecciones se realizan una a la vez.



PRECORDIALES

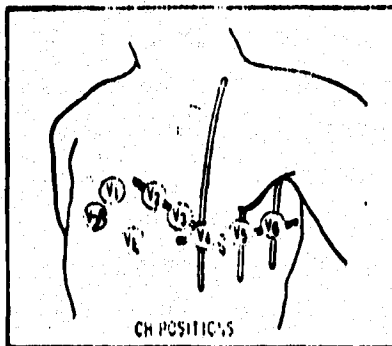
Figura 2.10. - Obtención de las derivaciones precordiales.

El registro de las doce derivaciones electrocardiográficas se obtiene una a la vez y se seleccionan a través de un circuito selector que permite observar cada derivación respetando una secuencia. Esta secuencia muestra primeramente las derivaciones bipolares (I, II, III) para después continuar con las derivaciones aumentadas (aVR, aVL, aVF) y por

ultimo las precordiales (V1, V2, V3, V4, V5, V6). Para el registro de las señales, el paciente es conectado al equipo mediante un cable blindado de cinco hilos donde cada hilo irá conectado a un electrodo. Por lo general estos hilos pueden ser identificados por un color que indica a que extremidad del cuerpo va conectado. Estos colores varían de acuerdo al código AAMI y al de IEC, en la figura 2.11 se muestran estos códigos así como las posiciones de los electrodos del cuerpo.

STANDAR

ROJO	AMARILLO	VERDE	NEGRO	BLANCO/ROJO	BLANCO/AMARILLO	BLANCO/VERDE	BLANCO/CAFE	BLANCO/NEGRO	BLANCO/VIOLETA	IEC
(Blanco)	(Negro)	(Rojo)	(Verde)	(Cafe/Rojo)	(Cafe/Amarillo)	(Cafe/Verde)	(Cafe/Azul)	(Cafe/Naranja)	(Cafe/Violeta)	AAMI
R (RA)	L(LA)	F(LL)	N(RI.)	C1(V1)	C2(V2)	C3(V3)	C4(V4)	C5(V5)	C6(V6)	
Brazo Derecho	Brazo Izquierdo	Pierna Izquierda	Pierna derecha	Precordial 1	Precordial 2	Precordial 3	Precordial 4	Precordial 5	Precordial 6	



- V1 : Cuarto espacio intercostal en el lado derecho del esternon
- V2 : Cuarto espacio intercostal en el lado izquierdo del esternon
- V3 : Punto medio entre V1 y V4
- V4 : Entre la quinta y sexta costilla en el centro de la mitad izquierda del pecho
- V5 : En el mismo nivel de V4 entre V4 y V6
- V6 : Al nivel de V4 en el lateral izquierdo

Figura 2.11.- códigos de colores y posiciones de los electrodos

Con el último electrodo identificado con color café se detectan las derivaciones precordiales, el cual recorre las distintas posiciones en el pecho. El electrodo que está colocado en la pierna derecha (identificado con,el color verde) se conecta al circuito de la pierna derecha. Los otros cuatro electrodos después de pasar por una etapa de acoplamiento y otra de resistencias se conectan a las terminales de entrada de un circuito que permite seleccionar las distintas derivaciones. El circuito selector no sólo permite elegir las derivaciones que deseamos observar, sino que también es capaz de realizar las combinaciones de resistencias necesarias para poder detectar la derivaciones aumentadas y las derivaciones precordiales. Esto se logra utilizando las ecuaciones previamente establecidas en los párrafos anteriores.

En la figura 2.12 muestra el diagrama electrónico del circuito de detección de ECG.

2.1.3.-Preamplificación

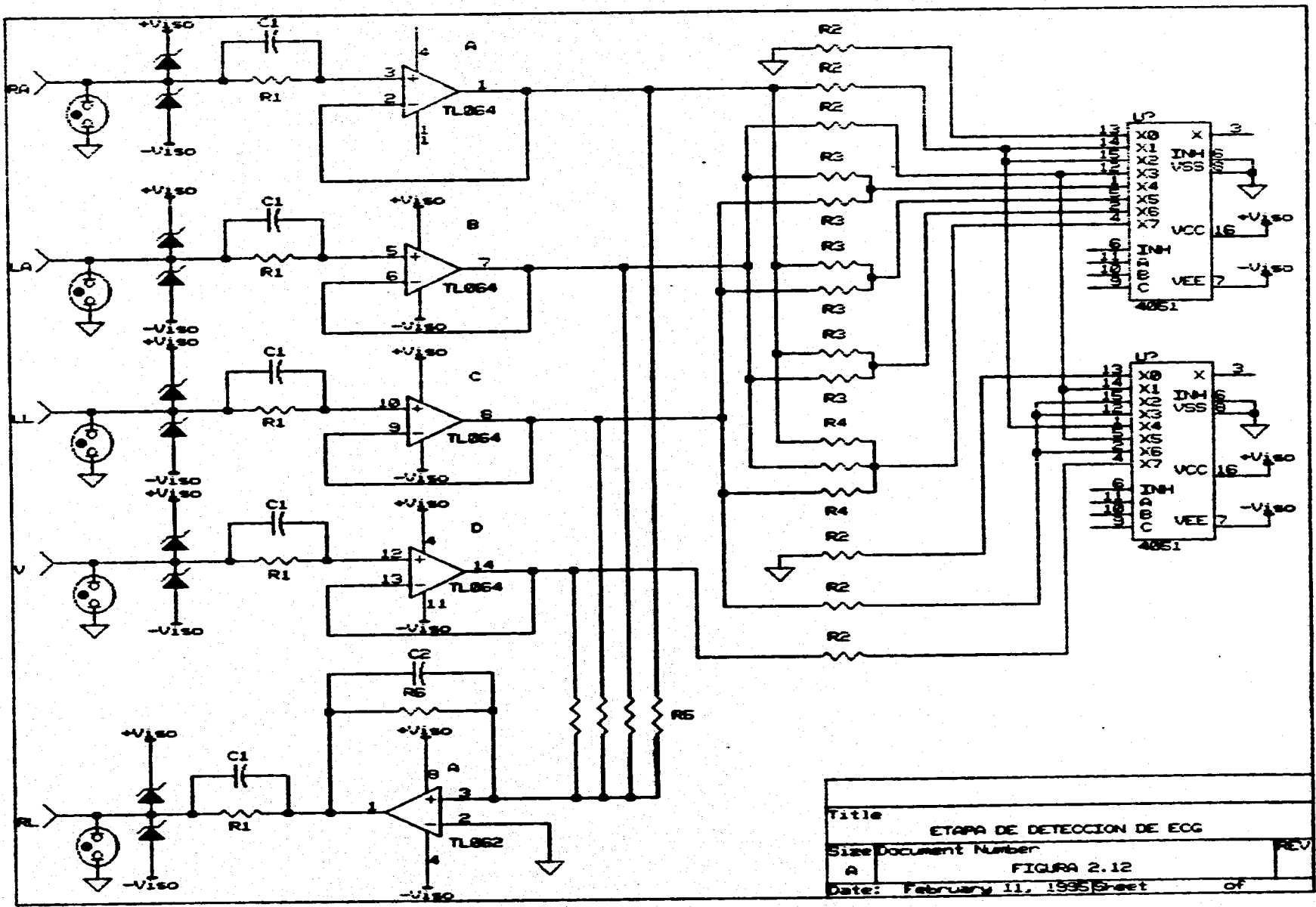
La etapa de preamplificación debe amplificar la señal ECG sin producir distorsión y al mismo tiempo debe reducir la interferencia que pudiese generarse sin que éste se sature. Para la señal de ECG se requiere que el preamplificador tenga ciertas características especiales las cuales se dan dentro de las especificaciones técnicas del equipo.

Esta etapa está compuesta por la sección de detección, la que aunada al amplificador de instrumentación, al arreglo resistivo del selector de derivaciones y al circuito de pierna derecha conforman la etapa preamplificadora. El amplificador de instrumentación proporciona a la salida una señal de ECG cuya amplitud ha sido amplificada (ganancia del preamplificador de 22) para someterla a un procesamiento posterior.

Cabe mencionar que para formar el amplificador de instrumentación de la etapa preamplificadora, se utilizó parte del amplificador de aislamiento seleccionado para la electrónica diseñada del ECG, el cual se verá en la siguiente sección.

2.1.4.-Aislamiento

Una de las consideraciones más importantes para el diseño de un equipo médico es la protección del paciente, por lo que al realizar un registro de la señal de ECG el paciente deberá estar aislado del equipo. Motivo por el cual es necesario tener una etapa de aislamiento que comúnmente se encuentra entre el preamplificador y el amplificador.



21

Title		ETAPA DE DETECCION DE ECG	
Size Document Number		REV	
A	FIGURA 2.12		
Date: February 11, 1995		Sheet	of

En este trabajo sólo fue necesario realizar una etapa de aislamiento para la señal de ECG debido a que los transductores que se utilizaron para la detección de los otros signos vitales se podían obtener de manera no invasiva.

Existen tres formas de efectuar el aislamiento: con transformadores, con aislamiento óptico y por aislamiento capacitivo [10], [11]. Para ésta etapa los equipos comerciales emplean generalmente el aislamiento con transformador. La técnica, consiste en aislar la etapa que va conectada directamente al paciente (preamplificador), así como su fuente de voltaje, generando con ello que toda la parte a la cual esté conectada el paciente se encuentre "flotando" con respecto a la tierra general.

En la figura 2.13 se muestra un diagrama de bloques de lo que se considera un estándar en los equipos para conformar la etapa de aislamiento. Cabe mencionar que debido a su importancia ésta suele ser contemplada como toda una tarjeta de circuitos, contribuyendo a que aumente de manera importante el volumen de los equipos.

Podemos observar mediante la línea punteada la separación que existe entre la entrada y la salida el cual rompe la continuidad óhmica entre ellos. El amplificador de entrada forma parte de un amplificador de instrumentación (etapa preamplificadora) el cual tiene una ganancia baja (22) para evitar que se sature.

Puesto que un transformador es un dispositivo para AC (corriente alterna) es necesario que se module la señal de ECG, de tal manera que se transmita de un punto a otro mediante una portadora.

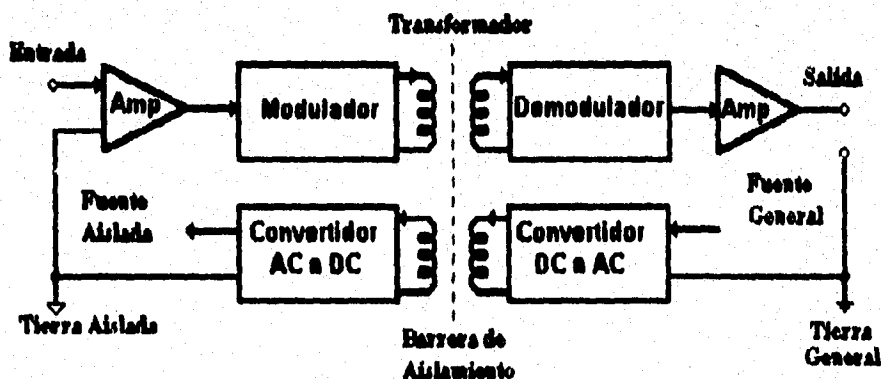


Figura 2.13.- Esquema general de aislamiento.

La señal de ECG se modula y se envía al primario del transformador para que posteriormente sea obtenida en el secundario donde se demodula y se filtra con el objeto de recobrar la señal original. Para obtener el voltaje de alimentación de la parte aislada es necesario que ésta se encuentre aislada magnéticamente de la fuente general conectada a la línea de voltaje comercial. Esto se logra mediante el empleo de convertidores de CD a CD. El voltaje de la fuente general es aplicado a un oscilador de onda cuadrada cuya frecuencia es de 1 a 100 khz el cual excita al primario del transformador de aislamiento. En el secundario del transformador se obtiene una señal cuadrada totalmente aislada de la original, la cual es rectificadora y filtrada de acuerdo a los voltajes de salida deseados, generando de ésta manera un voltaje de referencia (tierra) totalmente independiente a la línea de voltaje.

Ahora bien, un circuito con las características antes mencionadas es fabricado por Analog Devices con número 284J; Una de sus aplicaciones recomendadas es precisamente el registro de señales de ECG, el cual se muestra en la figura 2.14.

Este dispositivo tiene aisladas la señal de entrada de ECG, así como la fuente de alimentación para la misma. El diagrama electrónico del aislamiento del ECG se muestra en la figura 2.15.

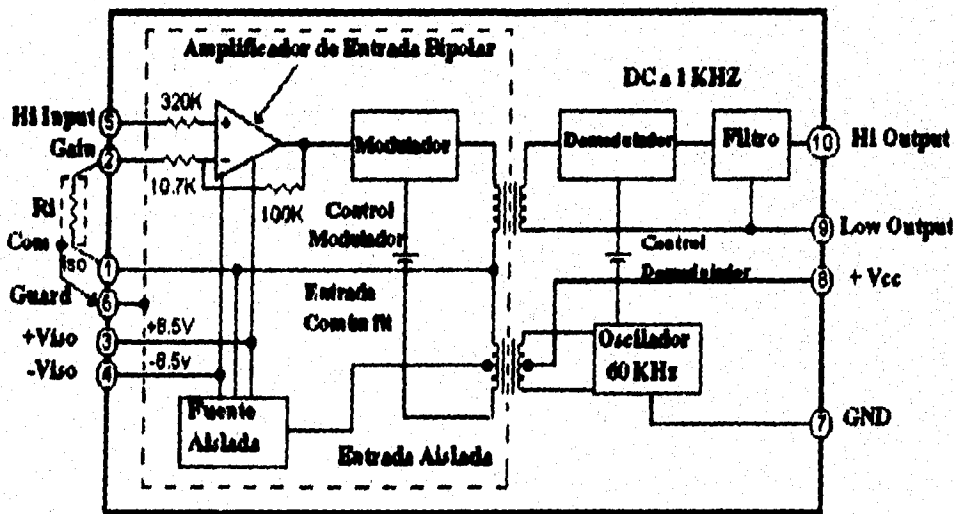


Figura 2.14.- Esquema del circuito 284J.

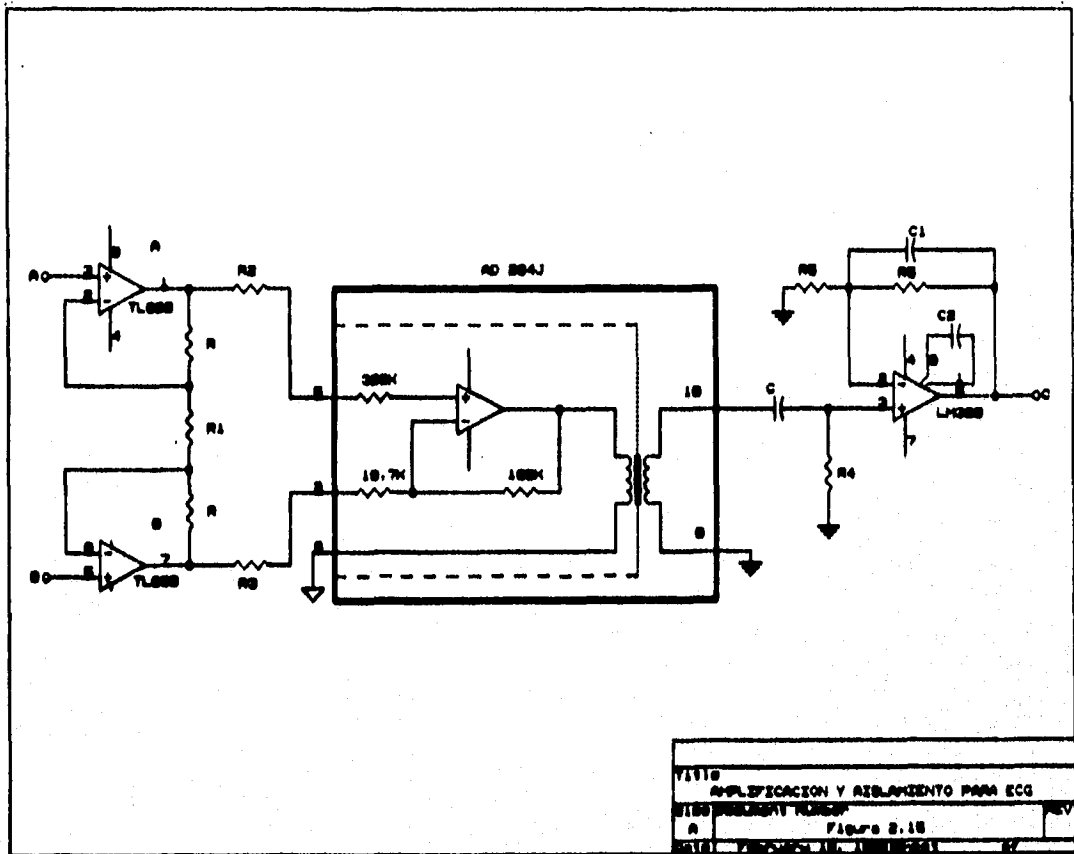


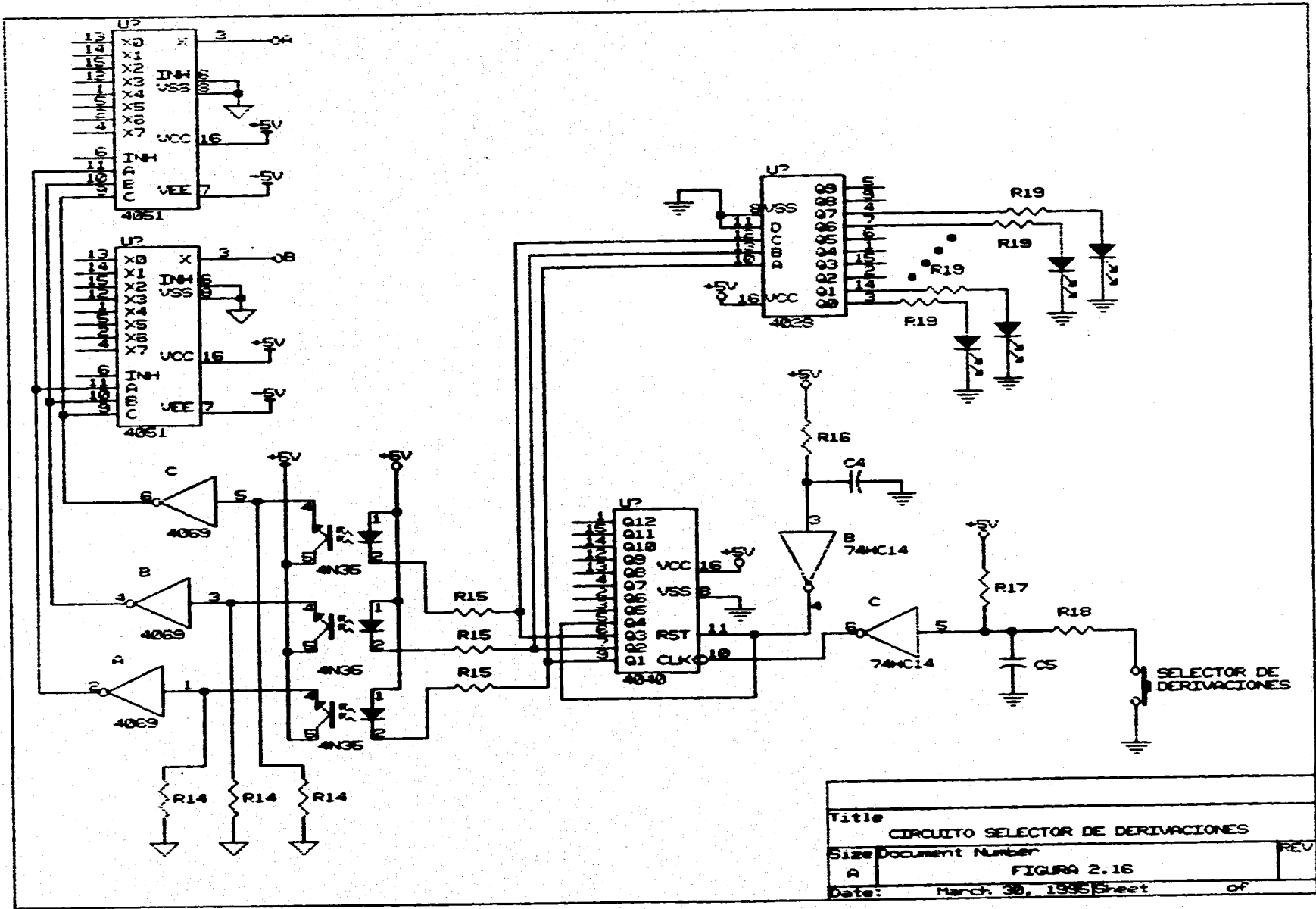
Figura 2.15.- Diagrama electrónico de aislamiento del ECG.

Otra parte donde se empleó aislamiento fue en el selector de derivaciones, en donde se utilizaron conmutadores analógicos para realizar los cambios de derivaciones, logrando con ello un cambio en la combinación de los electrodos. El tipo de aislamiento que se empleó aquí fue el óptico, el cual basa su funcionamiento en un optoacoplador. En la figura 2.16 se muestra el circuito del selector de derivaciones.

2.1.5.-Amplificación

Una vez que la señal de ECG ha sido preamplificada y ha pasado por el circuito de aislamiento (284J), pasa por un filtro paso altas RC cuya frecuencia de corte es de 0.48 Hz para ayudar a eliminar la interferencia producida principalmente por el movimiento del cable de paciente, así como el tremor muscular; posteriormente pasa por un amplificador no inversor cuya ganancia es de 68, el cual se muestra en la figura 2.17.

Cabe mencionar que la ganancia total de la electrónica diseñada para el ECG es de aproximadamente 1500, esto es, 22 del preamplificador y 68 del amplificador.



Title		
CIRCUITO SELECTOR DE DERIVACIONES		
Size Document Number		REV
A		FIGURA 2.16
Date:	March 30, 1995	Sheet of

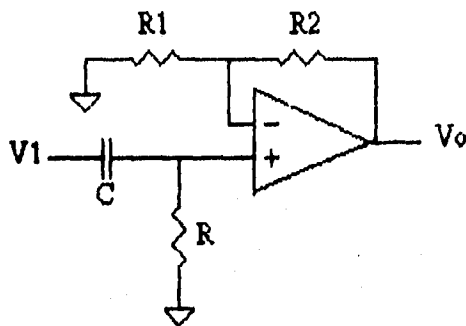


Figura 2.17.- Etapa Amplificadora.

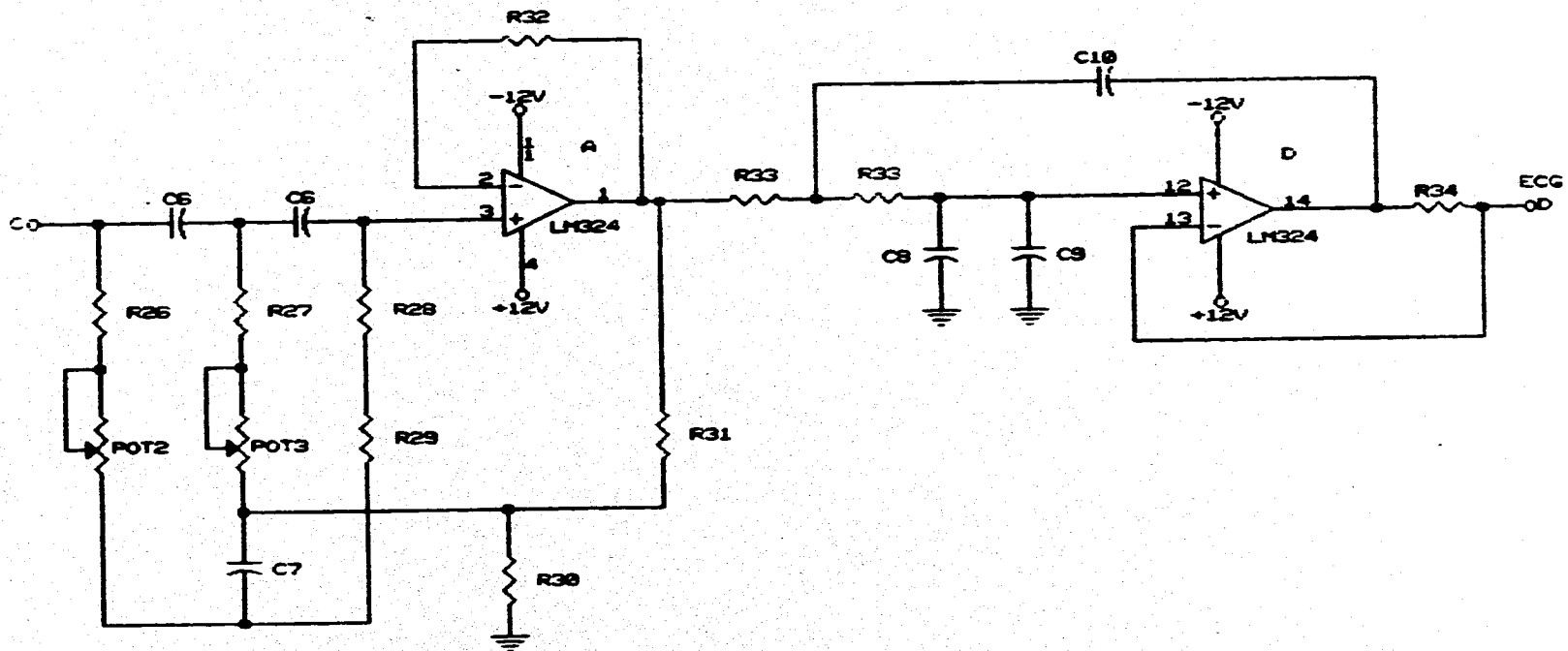
A partir de esta parte la señal de ECG se encuentra lista para recibir el acondicionamiento y/o filtrado, el cual producirá una señal "limpia" de toda clase de interferencia.

2.1.6.-Acondicionamiento y/o Filtrado

Es muy común que en áreas donde existe una gran cantidad de equipos trabajando (equipos de RX, electrocirugía, diatermias, ultrasonidos, etcétera) éstos generen un gran número de señales que se encuentran dispersas en el aire, las cuales pueden producir interferencia en la señal de ECG. Por esta razón es necesaria una etapa de acondicionamiento y/o filtrado que genere una señal libre de interferencias, de tal manera que no lleguen a afectar la buena visualización de los rasgos significativos de la señal de ECG (complejo PQRST).

Todo lo anterior implica la necesidad de diseñar y construir una etapa que elimine las señales conocidas como "parásitas", las cuales llegan a presentarse o mezclarse con la señal de ECG. Lo anterior se logra empleando filtros pasa banda con anchos de 0.05 a 100 Hz., para el caso de diagnóstico y de 0.5 a 40 Hz., para monitoreo. Puesto que el diseño se encuentra enfocado al área de monitoreo, se empleó el filtro de las características mencionadas con el que se eliminaron las señales producidas principalmente por movimientos del paciente, de los electrodos, así como de la interferencia mencionada en los párrafos anteriores. El diagrama electrónico de este filtro se muestra en la figura 2.18

ETAPA DE FILTRADO



Title	
ACONDICIONAMIENTO Y/O FILTRADO DE ECG	
Size Document Number	
A	FIGURA 2.18
Date:	March 30, 1995 Sheet of

Otra de las consideraciones que resulta de gran importancia en el monitoreo de pacientes es la interferencia ocasionada por la línea de tensión (60 Hz.), por lo que es necesario que se cuente con un filtro diseñado exclusivamente para eliminar o atenuar esta señal. Este filtro es del tipo Notch 60 Hz con una Q alta en configuración doble T [12]. El cual se observa en la figura 2.18.

De todo lo mencionado en las secciones anteriores se puede formar un diagrama de bloques general para la señal ECG el cual se muestra en la figura 2.19, donde se observan las diferentes etapas que conforman esta sección.

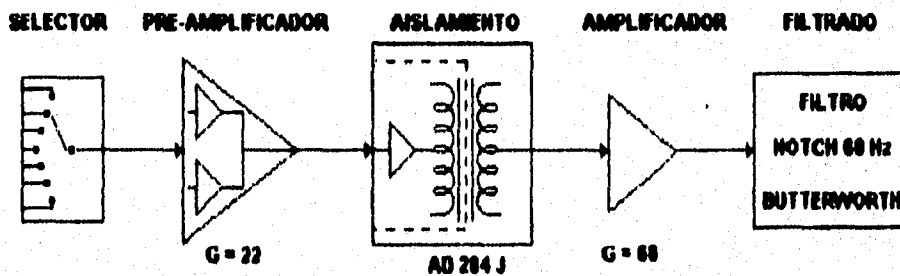


Figura 2.19.- Diagrama de bloques de la etapa de ECG.

2.2.- FRECUENCIA CARDIACA (FC)

La Frecuencia Cardíaca (FC) es una variable que indica el ritmo al cual trabaja el corazón, esto es, la FC define el número de impulsos sinusuales que se presentan en un lapso de un minuto.[13]

2.2.1.- Sensores y/o transductores empleados

Los sensores y/o transductores que se emplean generalmente para la obtención de la FC dependen del método de detección empleado. Una de las técnicas usadas es la "Plestimografía" que consiste en pasar un haz de luz Infrarrojo por un dedo o a través de

el lóbulo del oído, el cual detecta una mayor o menor concentración de sangre en la región, siendo esto un indicativo de que tan rápido bombea el corazón. Otro sensor empleado es el que se utiliza para obtener la señal de ECG (electrodos de ECG), del cual se extrae el complejo QRS, éste método es el que se utilizó en el desarrollo de esta tesis.

2.2.2.- Detección de FC

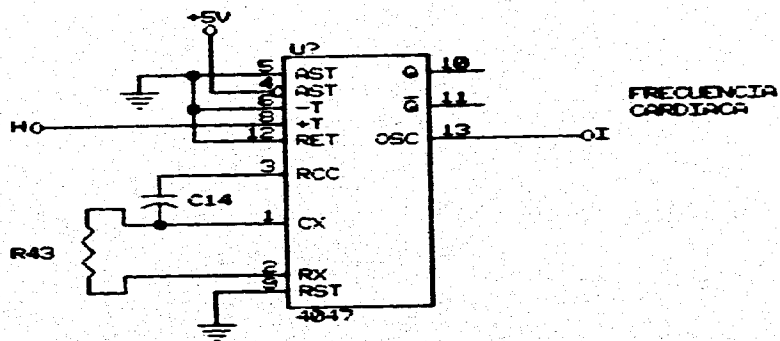
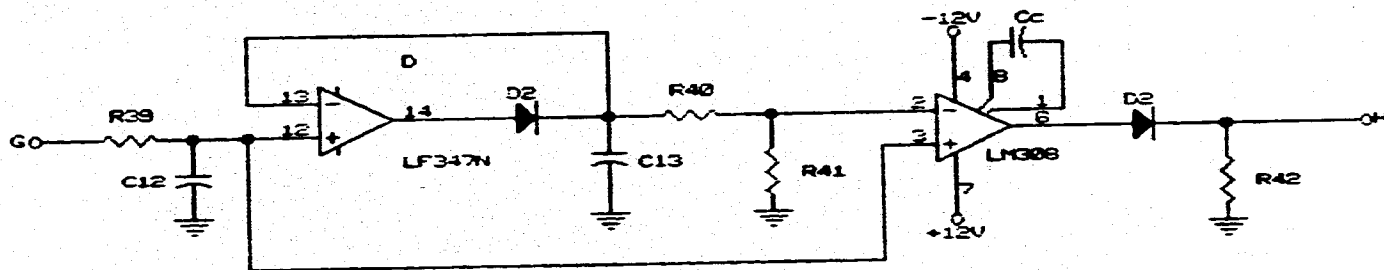
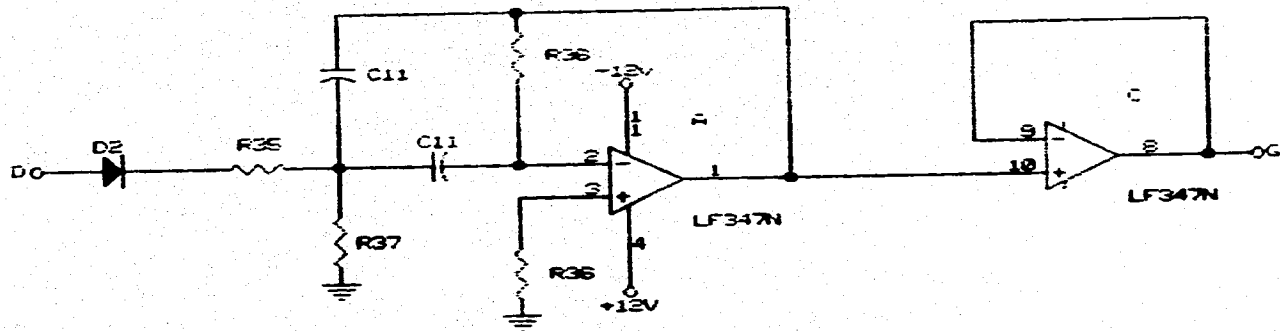
La FC "es el único signo vital que se obtiene sin la necesidad de un sensor o detector". Esto es debido a que el registro de la frecuencia cardiaca se lleva a cabo extrayendo el complejo QRS de la señal de ECG, la cual para los objetivos que pretendemos alcanzar, debe de haber sido sometida a un proceso de filtrado para proporcionar una señal limpia de cualquier clase de interferencia, por lo que no hay propiamente detección, sino un acondicionamiento.

Considerando que el complejo QRS es la parte más representativa del ECG, se puede obtener un registro del ritmo del corazón al detectar las amplitudes de voltaje que presentan cada uno éstos en las distintas derivaciones. Esto es válido si observamos que cada ciclo cardiaco posee un sólo complejo QRS cuya amplitud varía de acuerdo a la derivación que se registra en ese momento. Sin embargo, también las amplitudes de las ondas P y T, aumentan o disminuyen de manera proporcional por lo cual siempre se observará que el complejo QRS será el de mayor amplitud.

2.2.3.-Acondicionamiento y/o Filtrado

Como se mencionó en las secciones anteriores la FC se obtiene a partir de la señal de ECG, de la cual se extrae el complejo QRS que es el segmento donde se encuentra la mayor cantidad de información del ECG.

La salida que proporciona el filtro pasa banda de 0.5 a 40 Hz, de la figura 2.18, pasa por un diodo el cual elimina las partes negativas de la señal de ECG, para después pasar directamente a un filtro paso de banda de segundo orden de retroalimentación múltiple el cual se muestra en la figura. 2.20. El filtro fué diseñado con una frecuencia de corte igual a 17 Hz. y con un factor de calidad Q igual a 5 [14]



Title	
CIRCUITO DE DETECCION DE FRECUENCIA CARDIACA	
Size	Document Number
A	FIGURA 2.20
Date:	March 30, 1995 Sheet of

La función principal de este filtro es la de eliminar los segmentos P y T que son las formas de onda que anteceden y preceden respectivamente al complejo QRS, de tal manera que a la salida del filtro se tenga una señal cuya forma de onda sea la de un pico la cual saldrá defasada e invertida. Posteriormente esta señal pasa por un inversor de ganancia unitaria así como por un filtro paso bajas localizado a 7.23 Hz. para finalizar con un detector de pico, obteniendo así un pulso cada vez que aparece un complejo de QRS. El diagrama electrónico para la obtención de la FC se muestra en la figura. 2.20.

2.3.- FRECUENCIA RESPIRATORIA (FR)

La respiración es un proceso durante el cual nuestro organismo asimila el oxígeno del ambiente atmosférico para después desprender bióxido de carbono. La Frecuencia Respiratoria (FR) viene definida como el número de ciclos respiratorios ocurridos en un minuto.

El proceso de respiración consta de las siguientes etapas:

1. Se inhala con destino a los pulmones una cierta cantidad de aire atmosférico (Inhalación).
2. Una vez en los pulmones, la sangre absorbe el oxígeno y en los pulmones queda desprendido el bióxido de carbono que estaba contenido en la sangre.
3. Existe una corriente sanguínea continua que atraviesa los pulmones, procedente de la mitad derecha del corazón.
4. El aire con bióxido de carbono se expulsa de los pulmones (Exhalación).
5. La sangre enriquecida con el aporte de oxígeno pero privada de bióxido de carbono se envía por la mitad izquierda del corazón a las células del cuerpo liberando oxígeno e incorporando a su vez bióxido de carbono.

2.3.1.-Sensores y/o transductor empleados

Una forma de detectar la respiración consiste en utilizar la señal que se genera por el movimiento del tórax, lo cual implica utilizar los mismos electrodos que se usan para la detección del ECG. Otra manera sería detectar los cambios de temperatura que se originan cuando existen flujos de aire en las cavidades nasales. Esta última opción se eligió en el desarrollo de la presente tesis.

Al elegir la opción de detectar los cambios de temperatura que se dan al respirar, se buscó un dispositivo que pudiese sensar los cambios de temperatura ocurridos al

respirar con las fosas nasales, implicando con ello que el sensor fuese de un tamaño razonablemente pequeño para ser colocado en alguna de estas cavidades. El sensor utilizado para la presente tesis es el LM335, el cual es de alta precisión y de gran rapidez de respuesta, que basa su funcionamiento en la unión p-n de un diodo el cual opera como un diodo zener de 2 terminales con un voltaje de rompimiento directamente proporcional a la temperatura absoluta. Este dispositivo proporciona una salida lineal con un valor de 10 mV/°K y posee un rango de temperatura que va desde -40° C hasta 100° C

2.3.2.- Detección de FR

El criterio para la detección de la FR no está regido por la temperatura que se registra en las fosas nasales, sino por la variación que pueda presentar en ésta, esto es, que pueda detectar en que momento existe un flujo de aire caliente (exhalación) y en que otro existe un flujo de aire frío (inhalación). El número de veces que se repita dicho procedimiento y la continuidad con la que se presente, será la información útil para cuantificar la respiración.

La forma de onda que se obtiene de la detección es muy parecida a una senoidal donde el semiciclo positivo corresponde a la exhalación y el semiciclo negativo a la inhalación. La señal de FR se presenta de esa forma debido a que el sensor no cambia de manera inmediata sino que se calienta o se enfría paulatinamente. En la figura 2.21 se muestra la forma de onda de la frecuencia respiratoria.

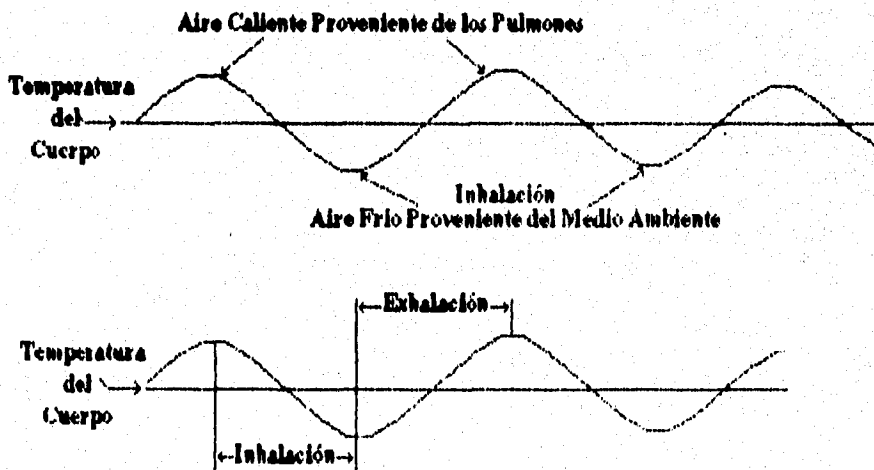


Figura 2.21.- Representación gráfica del proceso de respiración.

En la figura 2.22 se muestra el circuito que permite detectar la señal de la frecuencia respiratoria, el cual tiene 2 sensores LM335 conectados a la entrada del amplificador diferencial. Uno de ellos va conectado a la entrada negativa (colocado en la fosa nasal) y el otro a la entrada positiva el cual servirá de referencia, evitando que se sature el amplificador.

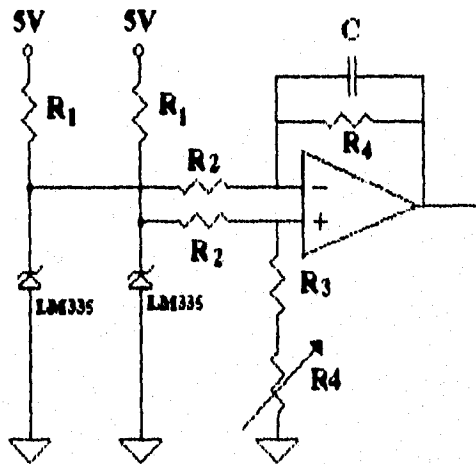


Figura 2.22- Amplificador diferencial para frecuencia respiratoria.

A la salida del amplificador se obtiene una señal senoidal de aproximadamente 10 mV de pico a pico. Cada semiciclo de esta señal representa una acción, inhalación en el semiciclo negativo y exhalación durante el semiciclo positivo.

2.3.3.-Amplificación

Como la señal de salida del circuito de detección (figura 2.22) resultó ser de un voltaje sumamente pequeño (10 mV de pico a pico) se optó por darle una mayor amplitud. Su salida se conectó a un circuito amplificador inversor, mostrado en la figura 2.23 con una ganancia de 390, logrando amplificarlo y ponerlo en su fase original, obteniendo una amplitud de 3.9 volts de pico a pico la cual hace más fácil su detección y manejo.

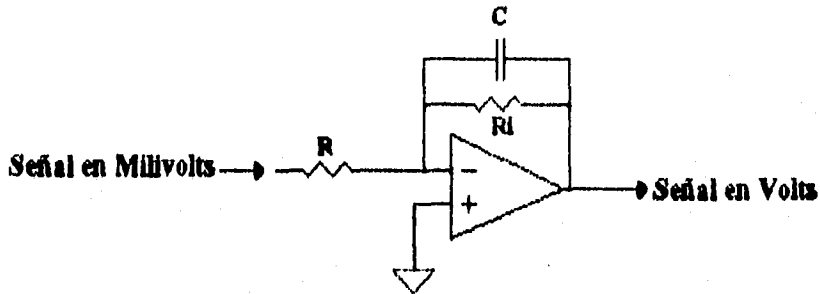


Figura 2.23.- Amplificador de la señal de frecuencia respiratoria.

2.3.4.-Acondicionamiento y/o Filtrado

La señal de salida que se obtuvo del circuito sensor es muy parecida a una senoidal la cual al ser amplificada (figura 2.23) presentó ciertas variaciones e irregularidades en su forma provocadas por el ruido, por esta razón se decidió filtrar la señal usando un filtro paso bajas RC que se muestra en la figura 2.24. A la salida de esta etapa se obtuvo una señal más pura en su forma senoidal, aspecto que como se verá más adelante resulta importante al momento de detectar los cambios que se presenten durante la inhalación y la exhalación.

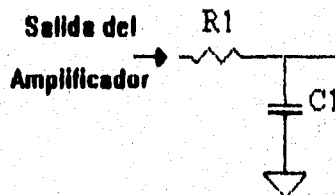


Figura 2.24.- Filtro paso bajas para frecuencia respiratoria.

Debido a que el circuito que se utiliza es un sensor de temperatura que se encuentra colocado en una fosa nasal, éste detectará la temperatura que el cuerpo posee en esa zona. Por lo tanto, además de registrar los cambios de temperatura que existan en los momentos en los que se está inhalando y exhalando también observaremos la temperatura que posee el paciente, por esta razón fué necesario fijar la señal de FR a una referencia.

Dada ésta situación se utilizó un arreglo que permitió sujetar la señal de salida a un punto de referencia específico. Para lograr lo anterior se realizó el arreglo mostrado en la figura 2.25. El cual permite acoplar la señal de FR para después poderla "sujetar" [15] a un voltaje de referencia que en éste caso es de cero volts.

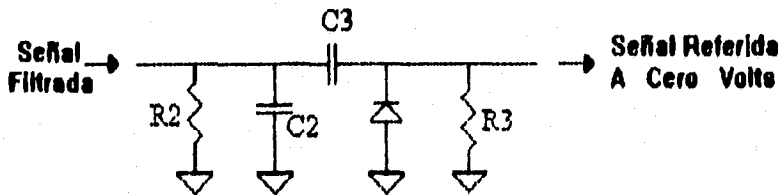


Figura 2.25.- Arreglo para Sujetar la Señal a Cero Volts.

La razón por la que se referencia la señal a cero volts es para utilizar el comparador de cruce por cero que aparece en la figura. 2.26. Obteniendo a la salida del comparador unos pulsos cuadrados que representan tanto el semiciclo positivo como el semiciclo negativo de la señal de FR.

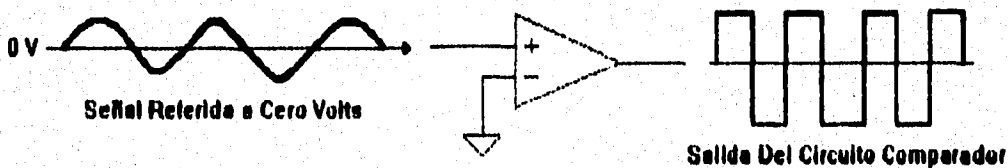


Figura 2.26.- Circuito comparador.

Si se observa detenidamente la señal cuadrada que proporciona el detector de cruce por cero, se verá que dicha señal presenta ciertas variaciones en los anchos de cada pulso. Éstas variaciones se originan por los cambios que existen en el ciclo respiratorio. Un ejemplo de éstos cambios es cuando, por alguna causa, se inhala más aire del que se exhala. Éstas variaciones representan un problema porque se requieren pulsos uniformes y cuya duración no "varie", este problema se resuelve con un circuito monoestable.

El diagrama de bloques de la figura 2.27 muestra en su totalidad el proceso de detección de Frecuencia Respiratoria, su correspondiente diagrama electrónico se muestra en la figura 2.28.

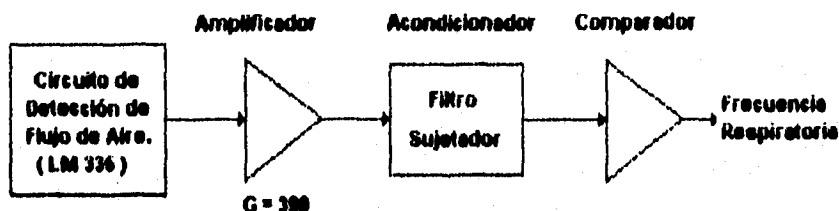
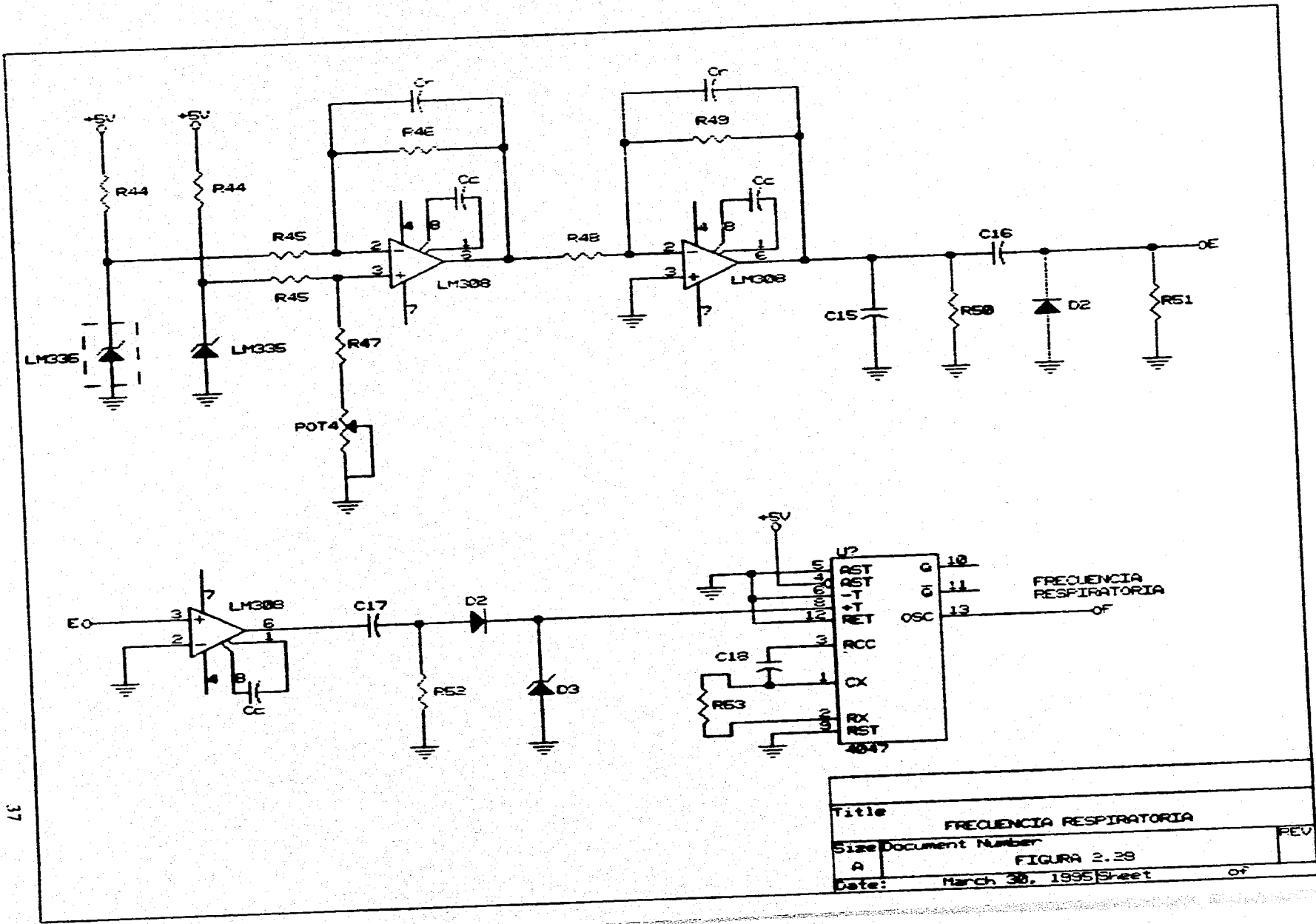


Figura 2.27.- Diagrama de bloques para la detección de la FR.

2.4.- TEMPERATURA

La Temperatura es una representación de la energía cinética de la materia que conforma a todo nuestro organismo, que al ser un sistema muy dinámico no presenta una distribución de temperatura uniforme, debido a las diferencias de riego sanguíneo entre las diversas zonas del cuerpo. [8] El sistema circulatorio aparte de su función como sistema de comunicación bioquímica tiene un papel importante en la termoregulación, aumentando o disminuyendo el caudal sanguíneo por las zonas calientes o frías del cuerpo.

Las variaciones de temperatura por lo general se originan cuando existe una penetración de microorganismos (bacterias, virus) y, de manera ocasional, sustancias nocivas las cuales originan tal situación. El cuerpo humano mantiene un nivel estable de temperatura el cual puede llegar a variar cuando nuestro cuerpo presenta alguna anomalía en su funcionamiento. Una de estas anomalías puede ser la fiebre la cual puede elevar la temperatura a más de 38° C (las temperaturas comprendidas entre los 37.3° C y los 38° C son consideradas como meras alteraciones de la temperatura normal de un ser humano). Para determinar la temperatura corporal se emplea un termómetro o un dispositivo sensor que puede ser colocado en el recto, la vagina, axila, o la lengua.[8] Éstos lugares son los más apropiados para la toma de lecturas ya que aquí se mantienen niveles casi constantes de temperatura.



Title		FRECUENCIA RESPIRATORIA	
Size	Document Number	FEV	
A	FIGURA 2.29		
Date:	March 30, 1995	Sheet	of

La temperatura corporal normal (aproximadamente 37 °C) se sitúa sólo 4 o 5 °C por debajo de la temperatura corporal más alta posible (42° a 43° C) [8]. Por ello un aumento sustancial de la generación de calor puede dar lugar con facilidad a una elevación peligrosa de temperatura, a menos que ésta sea contrarrestada con una pérdida radical de calor. Una persona que goza de buena salud comienza a sudar cuando la temperatura de la piel en el tronco llega aproximadamente a 34° C. Una elevación repentina de la misma se inicia casi siempre con escalofríos. Al principio del proceso la piel advierte el frío debido a la contracción de los capilares. Los escalofríos son los procedimientos a los que recurre el cuerpo para aumentar su producción de calor. Todos estos procesos que se llevan cabo en el interior de nuestro organismo se reflejan de manera palpable en la piel, las cuales sufren variaciones causadas principalmente por aspectos externos tales como el medio ambiente. Es por esta razón que las lecturas se toman en las regiones mencionadas anteriormente pues en dichas zonas la temperatura se mantiene aproximadamente constante . [8]

2.4.1.- Sensores y/o transductor empleados

Como se mencionó en párrafos anteriores, es importante conocer cuáles son las zonas o regiones del cuerpo que mantienen una temperatura estable o casi constante, de tal manera que obtengamos un registro verídico del mismo. Por ésta razón el área que nos ofrece un nivel de temperatura estable y además de fácil acceso, es la boca. Se tienen dos opciones (sensores y/o transductores) para la detección de ésta. La primera detecta las variaciones de temperatura por medio de un termistor lo cual implica calibrar éste mediante un circuito que compense la falta de linealidad propia del dispositivo, originándose una pérdida de exactitud en la obtención de resultados optándose entonces por otro método. La segunda opción utiliza el circuito integrado LM3911 para detectar la temperatura el cual incluye un sensor de temperatura, una referencia de voltaje y un amplificador operacional. El alto grado de linealidad es la particularidad esencial para la selección de este tipo de transductor, el cual es fácil de calibrar usando sólo dos puntos fijos de temperatura.

El LM3911 proporciona un voltaje de salida directamente proporcional a la temperatura en grados Kelvin siendo su salida del orden de 10mV/°K. Utilizando elementos externos es fácil calibrarlo en grados Centígrados o Fahrenheit. Una ventaja muy importante de este dispositivo es que no requiere muchos componentes para obtener valores adecuados. Interiormente cuenta con un amplificador operacional con una compensación en frecuencia y tiene un intervalo de operación que va desde -25 °C hasta 85°C.

2.4.2.- Detección de temperatura

La detección de temperatura se basa primordialmente en la conversión del voltaje proporcional a la temperatura en °K registrada por el sensor, a uno que corresponda a la temperatura pero ahora en °C. Como se explicó en el párrafo anterior, el sensor es un circuito integrado LM3911 el cual muestra un alto grado de linealidad entre la temperatura sensada (Temperatura del cuerpo en este caso) y el voltaje de salida del circuito; por lo que no es necesario implantar arreglos que compensen ésta falta de proporcionalidad entre la señal de entrada y la de salida. La linealidad de la que hablamos, temperatura/voltaje, se muestra en la figura 2.29.

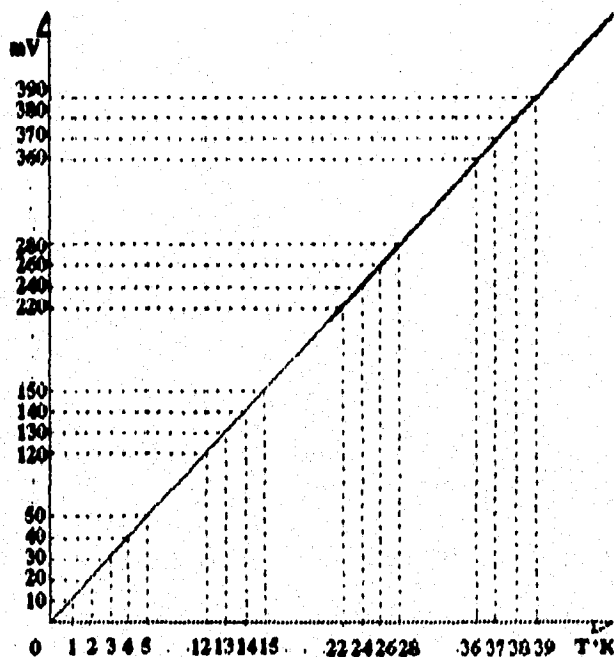


Figura 2.29.- Linealidad del LM3911.

Esta etapa consta de dos módulos, como se muestra en el diagrama a bloques de la figura 2.30, el primero corresponde a la electrónica necesaria para la operación del sensor por lo que podríamos llamarla etapa de detección, y el segundo, el proceso de conversión de °K/mV a °C/mV.

La variable a sensor, es la temperatura y la salida es un voltaje de CD que es proporcional a la temperatura sensada; se puede hacer una analogía temperatura-voltaje para implantar una etapa de conversión. Ésta analogía queda expresada como:

$$V_{Tc} = V_{Tk} - C$$

Donde:

V_{Tk} = es el voltaje correspondiente a la temperatura obtenida a la salida del sensor.

C = es un valor de voltaje en mV que es necesario restar al obtenido por el sensor para que la salida sea un valor de voltaje proporcional a la temperatura en °C registrada por el elemento.

T_c = es el voltaje obtenido a la salida de la etapa de conversión correspondiente a la temperatura en °C.

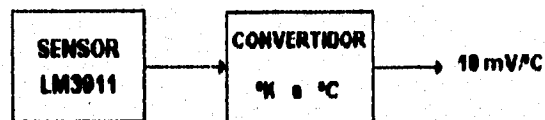


Figura 2.30.- Diagrama de bloques de la etapa de detección de temperatura.

En esta etapa no deben existir pérdidas en la linealidad T°C/mV por lo que los valores de los componentes deben ser lo más exactos posibles, así como el voltaje de conversión.

Es necesario recordar que el sensor es un CI con encapsulado epóxico el cual se cubre con una resina para evitar el contacto directo con el cuerpo humano. Este encapsulado evita el contacto entre la piel y los componentes internos del circuito.

2.4.3.-Amplificación

En el análisis de la figura 2.30, se tenía por ejemplo que 25 °C equivalían a 250 mV, el cual es un valor pequeño, por lo que es necesario amplificarlo hasta tenerlo del orden de los volts. Ello se realiza con un amplificador operacional al cual se le dio una ganancia de 10, de tal manera que:

$$25 \text{ } ^\circ\text{C} \text{ equivale a } 250 \text{ mV } (10) = 2.5 \text{ volts}$$

Los valores normales entre los cuales trabajará este sensor son la temperatura ambiente y la temperatura máxima del cuerpo humano (42 °C a 43 °C), por lo tanto no se corre el peligro de que rebase niveles de voltaje soportables por circuitos digitales (5.5 V) ni problemas de saturación de los componentes empleados.

En resumen, la etapa de temperatura consta de tres etapas la cual se muestra en la figura 2.31.

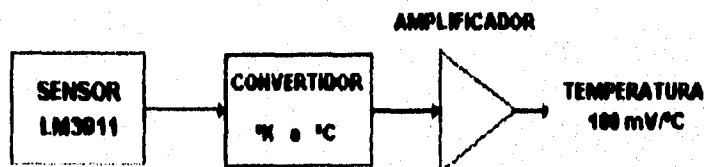
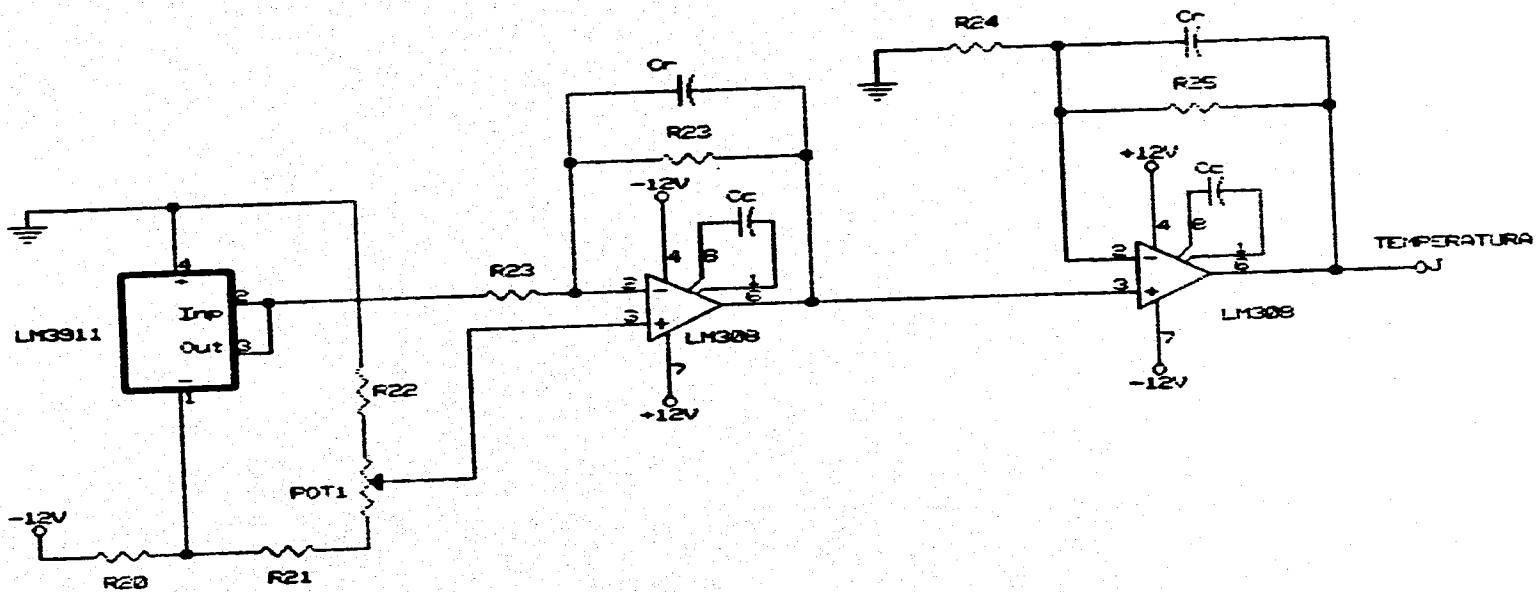


Figura 2.31.- Diagrama de Bloques del Circuito Detector de Temperatura.

El circuito completo para detección y amplificación de temperatura se muestra en la figura 2.32.

Esta parte no necesitó de una etapa de acondicionamiento y/o filtrado, ya que el ruido externo no afectó la señal de temperatura, por lo que solo fue necesario utilizar un circuito recortador de voltaje para la protección de la etapa digital.



12

Title		ETAPA DE TEMPERATURA	
Size	Document Number	REV	
A	FIGURA 2.32		
Date:	March 30, 1995	Sheet	of

CAPITULO 3

DISEÑO DEL MODULO DE ADQUISICION DIGITAL Y COMUNICACION DE SIGNOS VITALES (MADCSV)

Como se ha mencionado, una de las características fundamentales del sistema diseñado es la modularidad, es decir el equipo se compone del MDSV y el MADCSV los cuales en conjunto con una etapa de despliegue (para nuestro caso se utiliza el PCE como etapa de despliegue) forman el MBM los cuales son utilizados de manera individual para formar parte de otros sistemas.

El diseño del MADCSV debía ser lo más compacto posible, pero también debía ser capaz de realizar el procesamiento necesario para que el MBM funcione de manera individual, o en red, en caso de ser necesario. Debido a estas características de funcionamiento es necesaria la utilización de un microcontrolador, para el diseño del MADCSV.

Ahora bien, ¿porqué elegir un microcontrolador o un microprocesador para el diseño del MADCSV? Esto es debido a la imposibilidad de tomar datos digitales y realizarles un proceso, además de que los datos digitales son más inmunes al ruido durante transmisiones a distancia, lo cual es atractivo al operar en red. Otro motivo es también que en los últimos años ha tomado gran auge, en todas las áreas, el diseño de equipos basados en microprocesadores, tanto, que es difícil encontrar entre ellos alguno que no contenga un dispositivo programable para resolver las tareas de control.

Es por esta situación que para la aplicación desarrollada se recurrió a la utilización del microcontrolador MC68HC11 de Motorola [16], el cual de acuerdo a su arquitectura interna de 16 bits, sus módulos integrados (8 canales de conversión A/D, puerto serie, temporizador, 512 bytes de memoria RAM, 512 bytes de memoria EEPROM y 4 puertos de 8 bits cada uno) además de su precio accesible y su facilidad de programación, constituye una de las opciones actuales más eficientes y baratas para automatizar procesos.

También en este capítulo se describen algunos análisis realizados para el despliegue local, utilizando un monitor convencional de TV, así como la posible utilización de una

pantalla de cristal líquido; pero cabe mencionar que por el momento el despliegue se realiza en una computadora central.

3.1.-ACOPLAMIENTO CON EL SISTEMA DE ADQUISICION DE DATOS

Hasta este momento se han obtenido señales representativas de cada uno de los fenómenos fisiológicos, es decir, la parte correspondiente al MDSV se ha cubierto completamente; sin embargo en algunos casos la señal de ECG y Temperatura podrían ser difíciles de observar, dado que son señales de una frecuencia muy baja. En el caso de la temperatura, ésta se presenta como una línea recta de DC que a simple vista se puede perder por lo que es más importante el valor de la temperatura que su forma de onda en sí. En casos como la FC y la FR por sí solas no representan más que pulsos, las cuales no son señales representativas de un fenómeno. Para que todas estas señales adquieran un carácter más comprensible es necesario que se les "procese" y se obtengan así valores más entendibles. Por esta razón se hace necesario que el MDSV se acople con otro encargado procesar todas las señales (MADCSV) obtenidas. Este acoplamiento involucra generar algunas señales auditivas y de alarma, así como la parte de acoplamiento propiamente dicho con el MADCSV.

3.1.1.-Etapas Indicadoras del Sistema

Tener señales libres de ruido e interferencia, de fácil y clara visualización no es suficiente en un equipo de monitoreo de signos vitales. No siempre se puede tener fija la vista en una pantalla o prestar atención personal en cada momento a un paciente, es necesario que se cuente con algunos indicadores de tipo luminoso y auditivo que ayuden a complementar la supervisión del estado de un paciente.

En el equipo desarrollado para la presente tesis, se usa uno de estos indicadores luminosos para indicar la presencia de latidos del corazón. El indicador señala problemas con el paciente encendiendo un led de QRS cada que haya un pulso de ECG. Así mismo esta señal activa una bocina que emite un sonido cuando se presenta alguna situación de alarma.

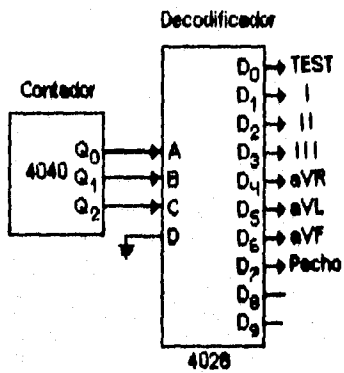
Otro led se enciende cuando alguno de los cables que se encuentran conectados al paciente se suelta o no hace buen contacto. Este indicador es uno de los de mayor importancia [17], pues alerta al personal médico cuando uno de los cables del paciente

pierde contacto con los electrodos sujetos a éste. Cuando uno de los electrodos se suelta hay un incremento en la resistencia de entrada del amplificador provocando que entre saturación y se tenga a la salida un voltaje igual a alguno de los voltajes de polarización empleados en el preamplificador.

Otro indicador es el que está relacionado con el selector de derivaciones el cual está constituido básicamente por un contador y un decodificador que selecciona las derivaciones deseadas. Cabe mencionar que las salidas de los contadores van a unos optoacopladores, los que a su vez van conectados a unos conmutadores analógicos para seleccionar las resistencias de la Red Wilson , logrando con esto que sólo se tome la derivación deseada de acuerdo a las expresiones matemáticas antes mencionadas, las cuales se observan en la figura 3.1.

El complejo de QRS además de servir como un indicador, también se utiliza para disparar un oscilador, el cual activa una bocina para producir una señal auditiva. La etapa consta de un circuito LM555 configurado como generador de onda cuadrada, de tal manera que cuando se produce un latido del corazón el comparador de frecuencia cardíaca produce un pulso que activa al oscilador durante el tiempo que dure el comparador saturado, obteniendo a la salida un grupo de pulsos cuadrados capaces de hacer funcionar la bocina. Esta parte se muestra en la figura 3.2.

Aparte de estos indicadores el sistema cuenta con alarmas para la FC, (bradicardia, y taquicardia), FR y temperatura las cuales se programan en el MADCSV.



CMA	D ₀ D ₁ D ₂ D ₃ D ₄ D ₅ D ₆ D ₇	DERIVACIÓN
000	1 0 0 0 0 0 0 0	INICIO
001	0 1 0 0 0 0 0 0	I
010	0 0 1 0 0 0 0 0	II
011	0 0 0 1 0 0 0 0	III
100	0 0 0 0 1 0 0 0	aVR
101	0 0 0 0 0 1 0 0	aVL
110	0 0 0 0 0 0 1 0	aVF
111	0 0 0 0 0 0 0 1	Precordial

Figura 3.1.- Circuito decodificador de derivaciones.

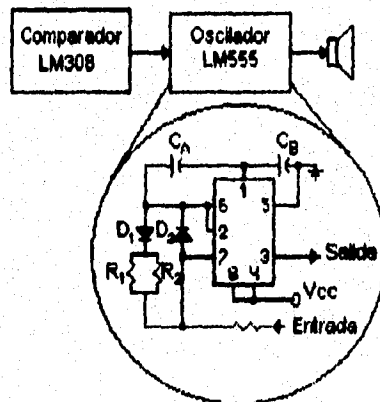


Figura 3.2.- Etapa auditiva.

3.1.2.-Acoplamiento del MDSV con el MADCSV

Para que el MDSV se complemente con el MADCSV debe existir una etapa intermedia que se encargue de acoplar las salidas del módulo de detección con las entradas del módulo de adquisición. Las entradas del MADCSV son las entradas de un CA/D (convertidor analógico a digital) el cual se encuentra dentro del 68HC11E9. Sus límites de operación aceptan señales dentro de un intervalo comprendido entre 0 y 5.5 volts, por lo que la etapa de acoplamiento debe adecuar las señales a estos rangos establecidos.

El acoplamiento de las señales de Temperatura y Cable Suelto fueron los más sencillos, debido a que son señales de DC; sólo fue necesario cuidar que no excedieran los valores anteriormente citados, para lo cual se utilizaron recortadores de voltaje.

Como la señal de ECG que proporciona el MDSV puede ser positiva o negativa dependiendo de la derivación seleccionada, fue necesario atenuarla y agregarle un nivel de offset (centrado a 2.5 volts) de tal manera que no pueda dañar al convertidor A/D del 68HC11E9.

Las señales de FC y FR casi no tienen problemas por que sólo son pulsos cuadrados los cuales no tienen duración constante. Por ejemplo el complejo QRS no siempre tiene el mismo ancho dada la diferencia de amplitud que existe entre las distintas derivaciones detectadas; para la respiración tenemos que en el momento en que la señal senoidal cruza por cero puede variar en su duración por lo que el ancho del pulso a la salida del comparador no siempre es constante.

Resumiendo, tenemos que para registrar la señal de FC y FR contamos con señales o pulsos cuadrados de duración variable que afectan al sistema de adquisición de datos, siendo necesario una etapa que proporcione como salida un pulso de duración constante independiente del tiempo que dure la señal de entrada.

Para lograr lo anterior se utilizó un multivibrador monoestable, el cual se activa con el disparo de subida del pulso de entrada; obteniéndose a la salida un pulso cuadrado de 5V de amplitud y una duración de 25 mseg, asegurando de esta manera que el circuito siempre proporcione una salida estable y constante, figura 3.3.

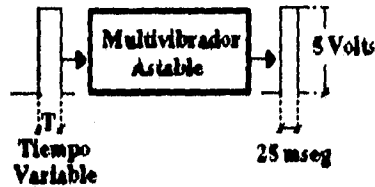


Figura 3.3.- Acoplamiento para señales de frecuencia.

3.2.-DESCRIPCION DEL MICROCONTROLADOR MC68HC11

Para elaborar la arquitectura del MADCSV en forma compacta, se optó por utilizar un microcontrolador de la familia de Motorola, del cual se hace uso exhaustivo de sus periféricos internos, para realizar las funciones de control del sistema y las tareas de muestreo, almacenamiento y transmisión de datos por puerto serie. A continuación se presenta una descripción general de este microcontrolador.

La unidad microcontroladora MCU (microcontroller unit) MC68HC11E9, es un sistema de alta integración y sofisticados periféricos internos, de alta velocidad de procesamiento y bajo consumo de energía. Sus características internas lo hacen ideal para el proyecto, como se puede observar en el diagrama de bloques de la figura 3.4; cuenta además con las siguientes características:

- Sistema de contador libre de 16 bits con 4 niveles de preescalamiento.
- Modos de espera para el ahorro de energía.
- Interfaz de comunicación serie.
- 5 puertos de 8 bits.
- 2 registros acumuladores de 8 bits y uno doble de 16 bits.
- 2 registros indexados de 16 bits.
- 8 kbytes de memoria ROM.
- 512 bytes de memoria RAM.
- 512 bytes de memoria EEPROM.
- Convertidor analógico/digital de 8 canales de 8 bits.
- 18 fuentes de interrupción

Modos de operación

El MCU utiliza dos pines (MOD A y MOD B) [15] para seleccionar uno de cuatro modos de operación, dos de ellos llamados básicos (*simple y expandido*) y dos especiales (*bootstrap y prueba*).

Simple .- en este modo de operación, el MCU hace uso de sus recursos internos y no tiene *buses* externos, por lo que se aprovechan al máximo sus puertos de entrada-salida; sin embargo se tiene la restricción de la memoria interna (512 bytes).

Multiplexado-expandido.- en este modo el MCU puede direccionar hasta 64 Kbytes para memoria externa o puertos adicionales. La parte alta del *bus* de direcciones son las salidas del puerto B, la parte baja del *bus* se encuentra multiplexada con el *bus* de datos y son tomadas del puerto C, la línea "AS" se utiliza como línea de control para el demultiplexaje de datos y direcciones; en este modo se reduce el número de puertos de entrada-salida, se cuenta además con líneas de control para el acceso a memoria o periféricos externos (AS, R/W y E).

Modo "bootstrap".- éste es un modo de operación muy versátil que combina los modos anteriores, aunque cambia la posición de los vectores de interrupción; puede ser usado para programar la EEPROM interna, para probar el programa residente en ROM y puede ser cambiado a otros modos bajo el control del programa.

Modo prueba.- éste solo se utiliza para calibrar los sistemas que componen al MCU y solamente es usado por el fabricante.

Puertos de entrada - salida

Puerto A .- está relacionado con el temporizador, puede ser configurado en todos los modos de operación con tres entradas para captura de datos, cuatro salidas para funciones de comparación y una entrada (PA1) para el acumulador de pulsos.

Puerto B .- en modo *simple chip*, todas las terminales son usados como salidas de propósito general, en modo multiplexado expandido se utiliza para el direccionamiento externo.

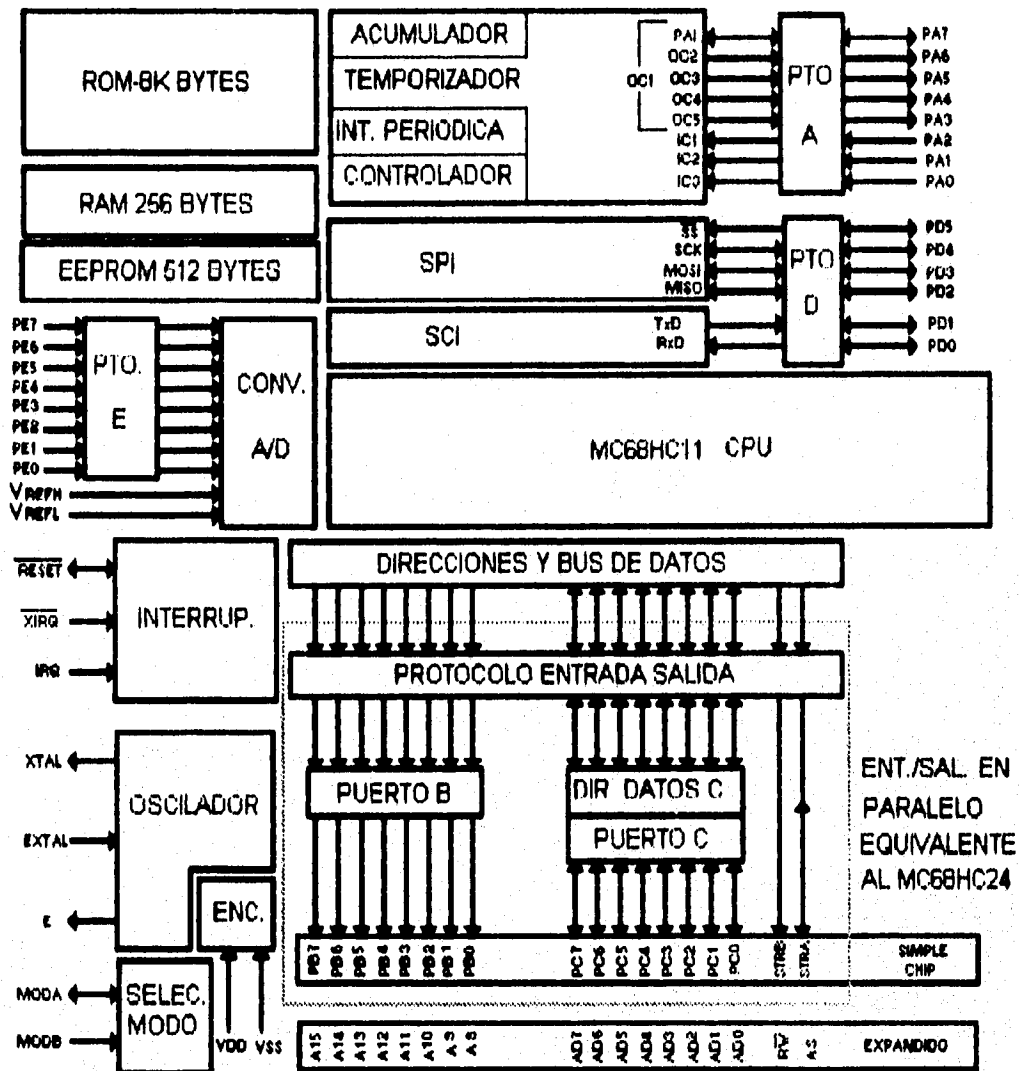


Figura 3.4 Estructura interna del MCU 68HC11.

Puerto C .- en modo *simple chip* se utiliza como un puerto de propósito general de entrada/salida, en modo *expandido* está multiplexado con el *bus* de datos y la parte baja de las direcciones, sus entradas pueden tener un registro *latch* controlado por la línea de control "STRA".

Puerto D .- es un puerto bidireccional de propósito general y puede ser configurado para obtener dos subsistemas de comunicación serie, sincrónico (PD2 - PD5) y asíncrónico (PD0 - PD1) de baudaje programable.

Puerto E .- puede ser usado como puerto de entrada de propósito general o como entradas para el convertidor analógico/digital, hasta 8 canales en los modelos PLCC (Plastic Leads Chip Carrier).

A lo largo del escrito se describirán con más detalle algunos de los periféricos empleados de este MCU.

3.3.-ARQUITECTURA DEL MADCSV

Una de las tareas del MADCSV es el almacenamiento de muestras de la señal de ECG para su posterior reconstrucción, por tal motivo se requiere del uso de una memoria RAM externa, ya que la RAM interna del MCU no es lo suficientemente grande para almacenar las muestras. Para el uso de memoria externa es necesario utilizar al MCU en modo *multiplexado expandido*, con el cual se tiene la posibilidad de direccionar más memoria o puertos de entrada/salida. Este punto se explicará más detalladamente en el punto 3.4

Como se muestra en el diagrama de bloques de la figura 3.5, el sistema cuenta con memoria RAM, EPROM (para el almacenamiento del programa de aplicación), un decodificador, un *latch*, y el MCU MC68HC11E9, por lo que se tiene un sistema digital compacto

En modo *multiplexado expandido* como ya se ha mencionado el *bus* de datos se encuentra multiplexado con la parte baja de las direcciones, para realizar el demultiplexaje se utiliza un circuito tipo *latch* cuya habilitación proviene de la línea de control del MCU "AS" [16], figura 3.5. La decodificación utiliza medio ciclo de la señal de reloj "E" para el *bus* de datos y el otro medio ciclo junto con la señal de "AS" para el *bus* de direcciones. Teniendo demultiplexados ambos *buses* se puede plantear el mapa de memoria para la arquitectura. El mapa se realizó dividiendo el espacio en bloques de 4 Kbytes. En la figura 3.6 se presenta este mapa, la parte sombreada corresponde a las localidades ocupadas por la memoria, registros internos y vectores de interrupción, el resto del espacio puede ser utilizado para direccionar más memoria o puertos de entrada - salida. En la parte alta se encuentran direccionados los vectores de interrupción y el vector de reset. Este último de

particular interés para inicializar el sistema, se encuentra en la dirección \$FFFE, en donde se debe guardar la dirección de inicio del programa de aplicación; este vector de interrupción, al igual que el programa de aplicación se grabaron en una EPROM externa de 4 Kbytes ubicada en la parte alta del mapa de memoria. Para su direccionamiento se utiliza la parte alta de las direcciones, es decir, las líneas de dirección A₁₂ - A₁₅ se conectan a una compuerta NAND, cuya salida se conecta al "OE" de la EPROM; con este arreglo se asegura que la memoria sólo se activa cuando se presenta una dirección comprendida entre \$F000 y \$FFFF. La RAM se localiza en el primer bloque de memoria, para su direccionamiento se utiliza un circuito decodificador (74HC138) para poder direccionar más memoria en caso de ser necesario; para ello se utilizan las líneas de dirección A₁₁ - A₁₅ y la señal de control "E", para direccionar la RAM en el espacio que va de la \$0800 a la \$0FFF (2 Kbytes); como se observa en la figura 3.6.

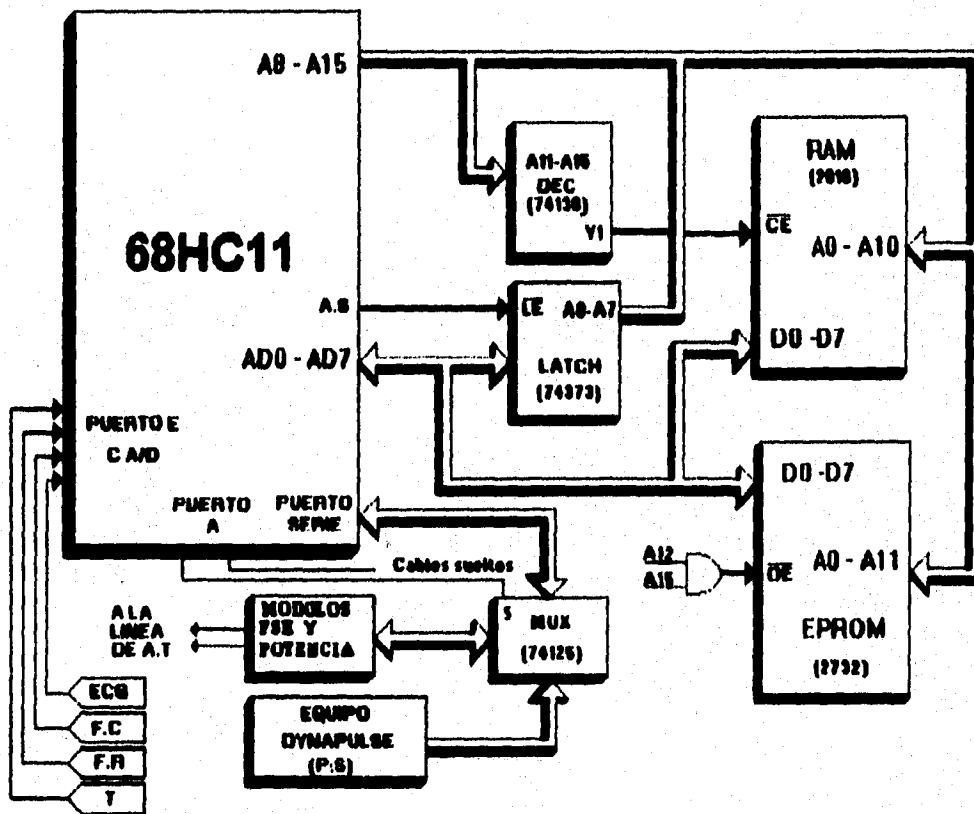


Figura 3.5 Diagrama de bloques de la arquitectura diseñada.

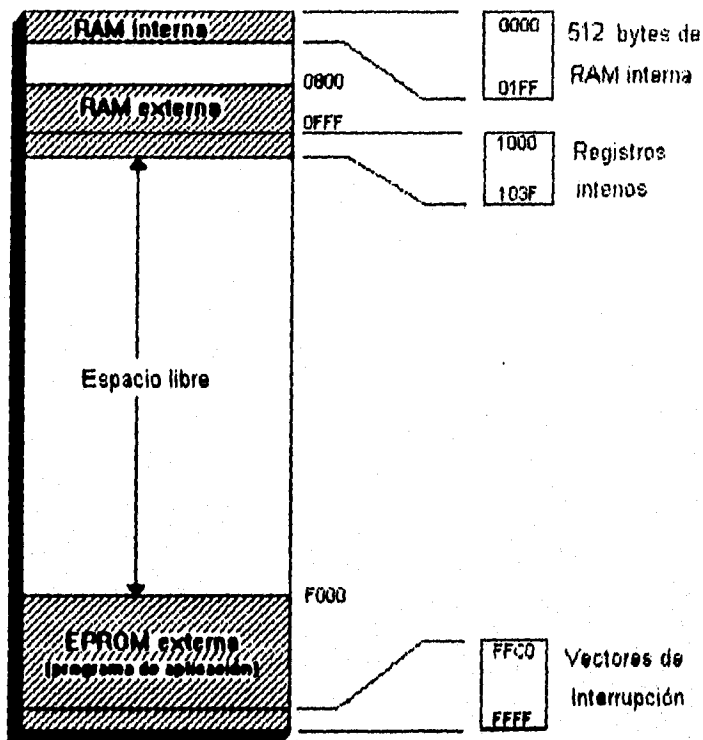


Figura 3.6 Mapa de memoria del MADCSV.

En el diagrama electrónico del MADCSV de la figura 3.7 se muestran los siguientes aspectos:

- 1.-La circuitería necesaria para el funcionamiento del MCU en modo *expandido*.
- 2.-El circuito de reset, que pone un nivel bajo en el pin de "RESET" para inicializar al MCU, eliminando los rebotes.
- 3.-El circuito del cristal de 8 Mhz para proporcionarle al MCU el reloj, con el cual se genera la señal de reloj "E" de 2 Mhz.
- 4.-El circuito de decodificación de memorias, descrito anteriormente.
- 5.-El circuito de acondicionamiento entre los niveles TTL del MCU y los niveles RS232 de la computadora, formado por el circuito MAX232, con el que se realiza la comunicación serial entre el MADCSV (MBM en conjunto) y el PCE.

También se muestran los niveles de referencia para el convertidor A/D (VRH y VRL), los cuales fueron fijados para tener una ventana de conversión de 0 a 5 Volts, que es el intervalo de variación de signos vitales proporcionadas por el módulo analógico ya

descrito. Las señales previamente amplificadas y filtradas de ECG, FC, FR y T se conectan a los primeros cuatro canales del convertidor A/D del puerto E. Por el puerto A se introduce la señal de alarma de cables caídos en el paciente. El puerto serie del MCU (PD0, PD1) se utiliza para la comunicación con el PCE y al mismo tiempo para recibir la señal de PS proveniente del equipo comercial DYNAPULSE (del cual se darán posteriormente sus características), por ello es necesario multiplexarlo por medio del circuito 74HC125, para el cual la señal de control de tercer estado se toma del puerto A.

Con este diseño se logra que el MADCSV sea muy compacto y funcional, donde la programación del MCU es una parte muy importante para su funcionamiento, ya que por medio de ella se realiza el control y procesamiento de las señales.

En los siguientes capítulos se detallarán los programas elaborados para el MADCSV.

En la siguiente lista se enumeran algunas características del MADCSV.

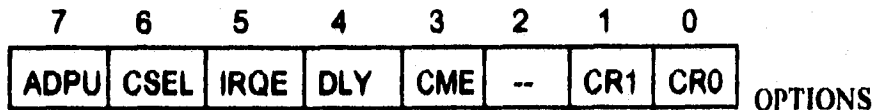
- 1.- Intervalo de conversión de señales de 0 a 5 V.
- 2.- Tiempo de conversión 16 microsegundos.
- 3.- Memoria RAM de 2 Kbytes.
- 4.- Memoria EPROM de 4 Kbytes.
- 5.- Velocidad de transmisión de 9600 bauds.

3.4.-MUESTREO DE SIGNOS VITALES

La función primordial del MADCSV es el muestreo y el tratamiento de los datos de la señal de ECG y de los demás signos vitales, el muestreo se seleccionó de tal manera que fuera posible reconstruir la señal de ECG lo más exacto posible para ello se recurrió al teorema de muestreo[18],[19], con el cual se calculó la frecuencia mínima para reconstruir la señal. Sin embargo, el período de muestreo para una señal de ECG se encuentra plenamente definido en la bibliografía[19],[20], es decir un período de muestreo de 256 Hz es suficiente para la reconstrucción de la señal con un error relativo de 1.6% a una frecuencia máxima de 250 pulsos por minuto (ppm), como se verá en el presente capítulo.

Como se mencionó en el apartado anterior, a través del convertidor A/D integrado en el MCU se muestrean 4 canales para adquirir muestras de ECG, FC, FR y T. El convertidor de 8 canales de 8 bits y retención de muestra, utiliza la técnica de conversión

de aproximaciones sucesivas con redistribución de cargas [16], este tipo de convertidor es el más popular en aplicaciones biomédicas ya que combina velocidad y simplicidad de conversión, tiene un error de $\pm 1/2$ LSB, con una velocidad de conversión de 16 microseg. Para su utilización es necesario configurar los siguientes registros: **OPTIONS** Y **ADCTL** (registro de control del convertidor A/D).



ADPU .- A/D powerup, proporciona la energía necesaria para activar el convertidor.

0 = desactivado

1 = activado

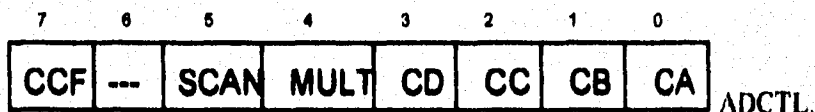
CSEL .- clockselect, selecciona el oscilador de referencia para la conversión.

0 = 2 MHz

1 = 750 kHz

Los bits restantes no son usados en el control del convertidor A/D.

Para trabajarlo a su máxima capacidad se selecciona el oscilador de 2 MHz y se activa la energía para su funcionamiento, por lo cual se configura este registro con un **\$80**.



CCF .- bandera de conversión completa, indica la terminación de una conversión.

0 = realizando conversión.

1 = conversión completa.

SCAN .- control de conversión continua, selecciona el tipo de conversión a realizar.

0 = conversión continua.

1 = una sola conversión.

MULT .- control de canal múltiple, selecciona la conversión de uno o cuatro canales.

0 = 1 canal.

1 = 4 canales.

CD, CC, CB, CA .- seleccionan el canal a convertir.

Cada vez que se requieran los datos de una conversión se debe escribir este registro, después de 16 microsegundos los resultados de la conversión de los cuatro canales se tendrán disponibles en los registros ADR1-ADR4 respectivamente, se debe realizar una conversión de 4 canales a la vez cada 3 milisegundos, seleccionando los primeros cuatro canales, por lo que se debe configurar este registro con un \$10.

En la figura 2.2 se muestra un periodo típico de una señal de ECG, la cual puede variar dependiendo de las características de salud y edad de la persona, pero que normalmente presenta las características mostradas. De esta señal el equipo desarrollado permite visualizar 4 periodos a la vez; si se toma una muestra cada 2.4 milisegundos (256 muestras/periodo) se tiene un periodo de muestreo de 427 Hz, valor que se encuentra por encima de la frecuencia de muestreo utilizada entre otros por L. Edenbrandt et al. [20] y Wendel C. Ocasio et al. [21] (256 Hz) y que además cumple con las normas AAMI para monitoreo cardiaco [6]. La frecuencia de muestreo usada se puede variar para tener diferentes resoluciones, generalmente se usa una frecuencia de 256 Hz con la que se tiene un error teórico de 1.6 %, como muestra la ecuación 3.1. Esta frecuencia es un valor adecuado para un sistema digital, ya que generalmente la señal se despliega a 256 Hz, sin embargo, para facilitar el procesamiento de las señales se utilizó una frecuencia de 333 Hz, es decir, se toma una muestra cada 3 milisegundos, a esta frecuencia se adquieren las señales vitales del paciente.

$$\text{error} = \frac{\text{Frec.max}}{\text{Frec.muestreo}} \times 100 = \frac{250\text{ppm}}{256\text{m/seg}} = 1.6\% \quad (3.1)$$

Para la señal ECG se toma una muestra cada 3 milisegundos y se almacena en RAM hasta alcanzar un total de 1024 muestras (el equivalente a 3 segundos aproximadamente de la señal), de donde son leídos los datos por el sistema de despliegue para su posterior reconstrucción. En el equipo construido la muestra se envía al PCE para su graficación si éste así lo requiere. En el caso de la temperatura el valor obtenido con el convertidor A/D se multiplica por un factor, para transformar los datos a grados centígrados y su valor se almacena en RAM para ser enviado al PCE cuando éste lo solicite. El factor de conversión está determinado por la ventana de conversión y la resolución del convertidor (8 bits), por lo que se pueden obtener 256 posibles valores provenientes del convertidor y cada valor corresponde a un voltaje de $5 \text{ V}/256 = 0.0195312 \text{ V}$, siendo éste el factor de conversión para la temperatura.

Las señales de FC y FR están formadas por pulsos, a los cuales se les tiene que calcular la frecuencia en pulsos por minuto (ppm) ; para calcularla se necesita conocer la duración de la parte baja del pulso ya que la parte alta es de duración constante. Bajo estas condiciones el MCU calcula la frecuencia en pulsos por minuto de acuerdo al diagrama de flujo de la figura 3.8, utilizando la técnica de "beat a beat", (latido a latido) es decir se calcula la frecuencia en cada periodo de la señal; posteriormente se utiliza la técnica de promediación con la cual se calcula la frecuencia tomando el promedio de varios periodos de la señal.

El MCU toma la muestra y evalúa si se trata de la parte alta o baja del pulso, mientras sea la parte baja incrementa un contador, si la muestra corresponde a la parte alta se realiza el cálculo de la frecuencia, multiplicando el valor del contador por el periodo de muestreo y se le suma el tiempo del pulso alto, teniendo así el periodo total de la señal. A este valor se le calcula la frecuencia en ppm. y su valor es almacenado en RAM. En las siguientes muestras de la parte alta se inicializa el contador para calcular la frecuencia del siguiente pulso. De acuerdo a los estándares de las normas AAMI, todo equipo para monitoreo cardiaco debe ser capaz de detectar frecuencias comprendidas de 20 a 250 ppm., para la frecuencia cardiaca, y de 8 a 150 ppm. para frecuencia respiratoria, ambas con un error máximo de ± 5 ppm. En nuestro caso el cálculo de las frecuencias se realiza dividiendo una constante por el número del contador, como se observa en la ecuación 3.2. Realizar esta división de 16 bits en el MCU involucra un error inherente en el cálculo de las frecuencias de ± 3 ppm., error que es aceptable de acuerdo a las normas AAMI.

$$FC = \frac{60}{\# \text{ contador} \cdot 3 \text{ mseg}} = \frac{20000}{\# \text{ contador}} \quad (3.2)$$

El valor de las frecuencias utilizando una frecuencia de muestreo de 256 Hz, tiene un error de 1.6 % cuando se detecta una frecuencia de 250 pulsos por minuto (ppm), como se explicó anteriormente, lo que corresponde a un error de ± 4 ppm. El error es aceptable de acuerdo a normas AAMI, sin embargo, como estamos utilizando una frecuencia de muestreo de 333 Hz este error se reduce a ± 3 ppm en la detección de frecuencias de 250 ppm, reduciéndose el error en la detección de frecuencias más bajas llegando a ser de ± 0.25 ppm en frecuencias de 20 ppm.

Para que el sistema funcione en tiempo real el MCU debe realizar las tareas de muestreo, procesamiento y transmisión serie, además de la detección de electrodos suel-

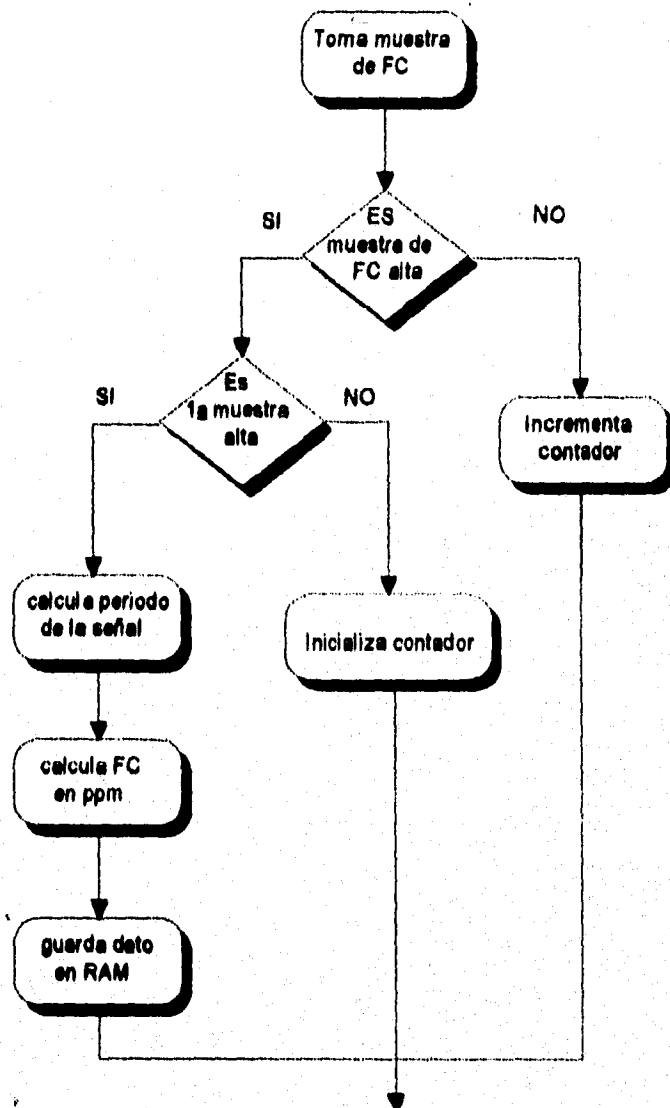


Figura 3.8 Diagrama de flujo para el cálculo de frecuencia cardíaca y respiratoria.

tos en el paciente (leyendo el estado del puerto A) en un período menor a 3 milisegundos, período que es suficiente si el MCU opera a 2 Mhz. Para lograr que el MCU realice las rutinas en un período de 3 milisegundos se tomaron en cuenta los ciclos de máquina de cada instrucción, de tal manera que a cada rutina se le añaden retardos para ajustar el tiempo de ejecución de la rutina a 3 milisegundos.

Debe subrayarse que tanto el MBM como el PCE son de tipo pasivo, por lo cual sólo monitorean y grafican los signos vitales. Por esta razón no se aplican algoritmos

digitales para el análisis o reconocimiento de señales, eventualmente este tipo de registro dinámico [21] se podría integrar en el futuro.

3.5.-DESPLIEGUE DE SIGNOS VITALES

El despliegue local de signos vitales podría realizarse mediante equipos especiales como osciloscopios con memoria o polígrafos, pero resultaría un equipo sumamente costoso; por esta razón se presenta el análisis de algunas opciones para el despliegue local, las cuales se encuentran en proceso de estudio para seleccionar la más adecuada, por el momento el despliegue se realiza mediante una computadora que es PCE.

Una de las opciones es la utilización de un monitor convencional de TV modificado [22], en donde se puede reproducir la señal de ECG en el tubo de rayos catódicos (TRC) controlando las placas deflectoras X-Y, en cuyo caso la señal de ECG sería la señal de contraste para el barrido vertical, para ello esta señal debe ser compatible con la norma NTSC (National Television System Comités) [23], en la cual se tienen 15750 Hz de barrido horizontal y 60 Hz de barrido vertical. Para visualizar los 4 periodos de ECG se tendrían que agrupar $(4)(256 \text{ muestras/periodo}) = 1024$ muestras que deben ser procesadas en 1/60 de segundo es decir 61440 muestras por segundo, frecuencia muy elevada en comparación con la frecuencia de muestreo del MCU, bajo estas circunstancias es necesario diseñar un autómata de despliegue que lea los datos de la RAM a la frecuencia televisiva de 61440 muestras por segundo y enviarlas a un convertidor digital / analógico para formar la señal de contraste. Para la visualización de los valores de FC, FR, y temperatura, y demás parámetros es necesaria la utilización de un generador de caracteres; con todas estas características el diseño del sistema se vuelve más robusto y más caro, sin embargo se estudia además la posibilidad de incorporar una pantalla de cristal líquido activo controlado por puerto serie, esto haría al sistema muy compacto y ligero, pero el costo también es elevado, por ello se encuentra en estudio la opción para el despliegue local.

3.6.-TRANSMISION DE DATOS

Como se ha mencionado en capítulos anteriores, el enlace de comunicaciones o la transmisión de datos del MBM hacia el PCE puede ser vía módem, radio o línea

telefónica, la transmisión de datos vía línea de voltaje es otra opción, la cual se detalla a continuación.

Para la transmisión de datos a través de la línea de voltaje, es necesario utilizar algún tipo de modulación, ya que los datos en formato RS-232 no pueden ser transmitidos directamente, debido a las bajas frecuencias que utiliza; es por esto que el diseño del módulo se basa en la técnica de modulación digital de onda continua denominada llaveo o manipulación por corrimiento de frecuencia FSK (frequency - shift - keying) [24]. En esta técnica la frecuencia de la señal moduladora o frecuencia de transmisión se desplaza entre dos frecuencias a medida que se transmite un "1" o un "0" lógicos.

En la figura 3.9 se muestra el diagrama de bloques para el módulo FSK el cual está formado por los siguientes bloques:

1.- El transmisor formado principalmente por un circuito integrado que genera ondas senoidales, del cual se obtienen dos frecuencias moduladoras correspondientes a un "1" y "0" lógicos. Las frecuencias se obtienen utilizando al generador como un modulador de FM, para ello se utiliza un circuito que acondiciona los voltajes RS-232 y los hace compatibles con el generador, teniendo un voltaje para el "1" lógico con su correspondiente frecuencia a la salida del generador y otro para un "0" lógico con su correspondiente frecuencia; ambas frecuencias moduladoras pasan a través de un amplificador de potencia para ser transmitidas por la línea de tensión.

2.- Etapa de acoplamiento con la línea, formada principalmente por capacitores. Para tener un acoplamiento con pérdidas mínimas los capacitores se seleccionan de tal manera que presenten una alta impedancia a la energía de 60 Hz y un equivalente serie resistivo de bajo valor para el rango de frecuencias moduladoras utilizadas por el transmisor, las cuales deben estar dentro del ancho de banda de la línea de tensión (30 - 500 KHz) [24]. Además, los capacitores deben soportar un voltaje mayor al voltaje pico a pico de operación de la línea. Para la selección de los capacitores se utiliza la ecuación que representa la impedancia de un capacitor, con la cual lo seleccionamos de acuerdo a la frecuencia de operación del transmisor y a la impedancia de acoplamiento.

$$Z_c = \frac{1}{2\pi f c}$$

f = frecuencia de operación

c = capacitancia

Z_c = impedancia del capacitor

3.- El receptor, formado principalmente por un filtro paso de banda, sintonizado a las frecuencias de operación del transmisor, con el cual se elimina una gran parte de las interferencias de baja frecuencia presentes en la línea. Las señales filtradas pasan a un circuito demodulador de FM formado por un PLL. (phase - locked loop), el cual opera con una excelente linealidad entre las frecuencias de entrada y el voltaje de salida, teniendo así dos niveles de voltaje correspondientes a las dos frecuencias de operación del transmisor. Finalmente estos voltajes pasan a un comparador de voltaje para acondicionar la salida del PLL a un formato RS-232.

Por el momento el diseño de este módulo se encuentra en proceso de pruebas y validación.

CAPITULO 4

INTERACCION ENTRE EL MODULO BASICO DE MONITOREO (MBM) Y EL PUESTO CENTRAL DE ENFERMERAS (PCE)

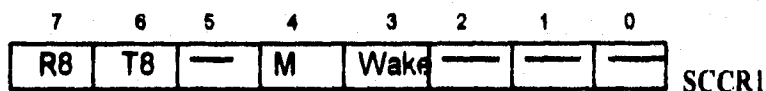
Una de las características del MBM es su posible uso en red, es decir, tiene la posibilidad de monitorear de manera local y al mismo tiempo estar conectado a la computadora central del PCE, o simplemente estar conectado al PCE. Como el PCE puede monitorear hasta 14 MBM's es necesario utilizar algún identificador para cada uno de ellos, así como un protocolo para realizar la transferencia de datos [25]. El identificador o clave de cada MBM se fija mediante *software*, de esta manera cuando se encuentren conectados más de un MBM al PCE y éste solicite datos de alguno de ellos todos los MBM conectados recibirán la clave pero solo responderá uno de ellos. En cuanto a la transmisión de información entre ambos módulos es necesario establecer un protocolo de comunicación para tener una sincronía entre ellos, por lo que fue necesario elaborar algoritmos de tiempo real para ambos módulos. Para ello se utilizó un sistema de comunicación serie eficiente como es el caso del SCI "(serial communication interface)" integrado en el MCU. El medio de comunicación entre ambos sistemas puede ser la conexión directa o enlazados mediante el módem FSK, en ambos casos realizando una comunicación half dúplex. En este capítulo se describe el sistema de transmisión serie de datos del MBM hacia el PCE, así como el protocolo establecido para realizar el intercambio de información.

4.1- TRANSMISION SERIE DEL MBM

La interacción entre el MBM y el PCE se realiza mediante la transmisión de datos a través del puerto serie del MCU, el cual es un sistema asíncrono full - dúplex tipo UART (universal asynchronous receiver / transmitter) [26], que permite una excelente comunicación con dispositivos periféricos, en nuestro caso se realizó una comunicación tipo half-dúplex con el PCE, debido a las características de la programación. El SCI puede trabajar a diferentes velocidades de transmisión, las cuales dependen del cristal utilizado en la arquitectura y de la programación. Para la transmisión utiliza el estándar NRZ (no retorno a cero) con formato de 1 bit de inicio, 8 o 9 bits de datos y 1 bit de paro.

Para la correcta utilización del SCI es necesario configurar 5 registros del MCU: SCCR1 (registro de control 1 del SCI), SCCR2 (registro de control 2 del SCI), BAUD (registro de control de baudaje), SCSR (registro de status del SCI) y SCDR (registro de datos del SCI); los tres primeros se utilizan para inicializar el puerto serie y los dos últimos para la transmisión y recepción.

En la parte de inicialización se fija el formato de los datos a transmitir, con el fin de hacerlos compatibles con el formato que espera recibir la computadora del PCE, para ello se utiliza el registro SCCR1 en el cual se tienen los siguientes bits.



R8 - activa la recepción del bit 8 en caso de formato con 9 bits de datos

T8 - activa la transmisión del bit 8

M - selecciona el formato de los datos

0 = 1 bit de inicio, 8 bits de datos, 1 bit de paro

1 = 1 bit de inicio, 9 bits de datos, 1 bit de paro

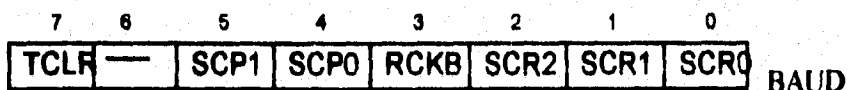
Wake - selecciona el método de activación de línea

0 = línea desocupada (idle line)

1 = marca de dirección (detecta el bit 8 o 9 para activar la línea)

Para el presente trabajo se seleccionó el formato de 8 bits de datos y el método de línea desocupada, es decir la línea permanece en estado alto mientras no haya datos a transmitir, por lo que se configura a este registro con un \$00.

Para fijar el baudaje al cual se transmitirá, se debe configurar el registro BAUD, en el cual se tienen los siguientes bits:



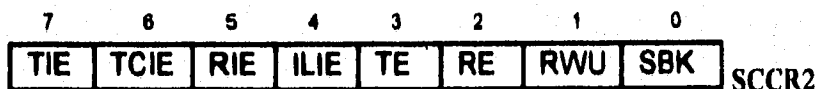
SCP0 - SCP1 - seleccionan el factor para preescalar la frecuencia del cristal

SCR0 - SCR2 - seleccionan el baudaje deseado

Los bits restantes sólo se usan para modo *prueba*. Utilizando un baudaje de 9600 bauds este registro se debe configurar con un \$30. Para utilizar baudajes diferentes sólo

hay que cambiar la configuración de este registro, teniendo en cuenta que para transmisiones en tiempo real se requiere de una velocidad de transmisión de 7500 bauds (250 muestras / seg / canal , enviando 10 bits / muestra) [19].

Para que el SCI funcione como un sistema full-dúplex se utiliza el registro SCCR2 en el cual se tienen los siguientes bits:



TE - habilita transmisión

0 = deshabilitada

1 = habilitada

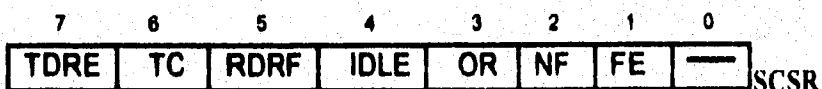
RE - habilita recepción

0 = deshabilitada

1 = habilitada

En este registro únicamente se habilitan los bits de transmisión y recepción escribiendo un \$0C, los bits restantes se utilizan para interrupciones en transmisión y recepción.

En la transmisión de datos se usan dos registros, el SCDR para escribir el dato a transmitir y el SCSR en el que se leerá el estado del puerto, en este último registro se tienen los siguientes bits:



TDRE - registro de transmisión de datos lleno

0 = no lleno

1 = lleno

TC - transmisión completa

1 = indica que ha terminado la transmisión

RDRF - registro de recepción de datos lleno

0 = no lleno

1 = lleno

En la recepción primero se lee el registro SCSR para averiguar si el bit 5 se encuentra activado, lo que indica que llegó un dato, posteriormente se lee el dato del registro SCDR. El resto de los bits se utilizan para saber si ocurrió alguna sobreescritura o algún problema con los datos.

En el apéndice A se muestra el listado en ensamblador de la programación realizada.

4.2.- ARQUITECTURA PARA INTERCONECTAR MBM'S EN RED (RMSV)

Una Red de Monitoreo de Signos Vitales (RMSV) es una línea de comunicación que permite conectar y mover información entre varios equipos de monitoreo de pacientes (MBM, PCE, impresoras ,etc.).

La RMSV de la compañía Marquette electronics de la figura. 4.1 se utiliza generalmente una LAN (local Area Network) como estructura a seguir y dependiendo de la distancia entre equipos las denomina TRAM-NET Y ETHERNET.

TRAM-NET.- Es una pequeña red de área local en la cual todos los dispositivos o equipos se encuentran ubicados en un solo lugar (salon). Esta red ofrece una gran flexibilidad, y opera a una alta velocidad (921.6 kbits/seg.), además de usar una topología en estrella.

ETHERNET. Es una LAN para comunicación de salón a salón o a través de las diferentes unidades de un Hospital. Esta red ofrece una alta velocidad de comunicación de 10 megabits/seg. Utiliza una topología bus y puede comunicar a muchos equipos. En las figuras 4.2, 4.3 podemos observar las diferentes topologías usadas por la Cia. Marquette Electronics, y en la figura 4.4 se muestra la arquitectura de un paquete de datos [27].

Para el caso de la presente tesis se utilizó una RMSV (LAN) usando una topología estrella, teniendo como método de acceso el sondeo por llamado de lista [25].

marquette

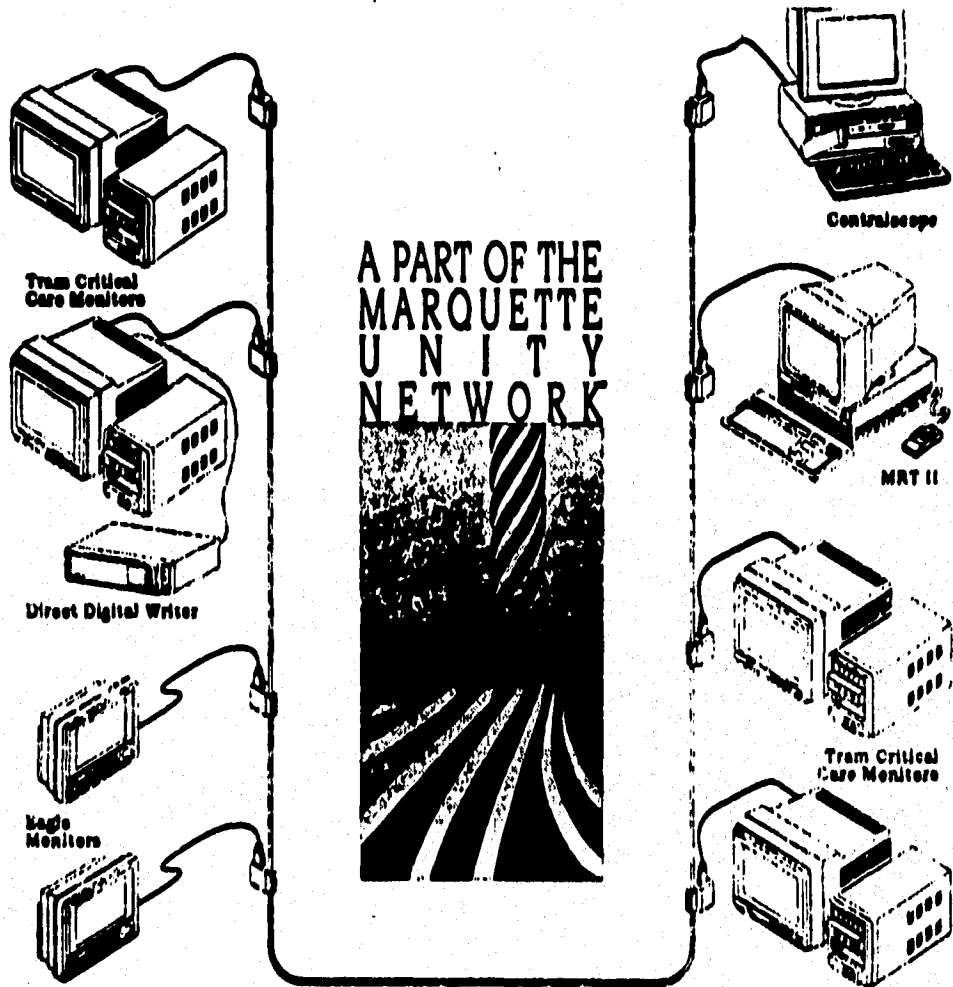


Figura 4.1.- RMSV de la Cia Marquette Electronics

Marquette Unity Network

Bus Topology

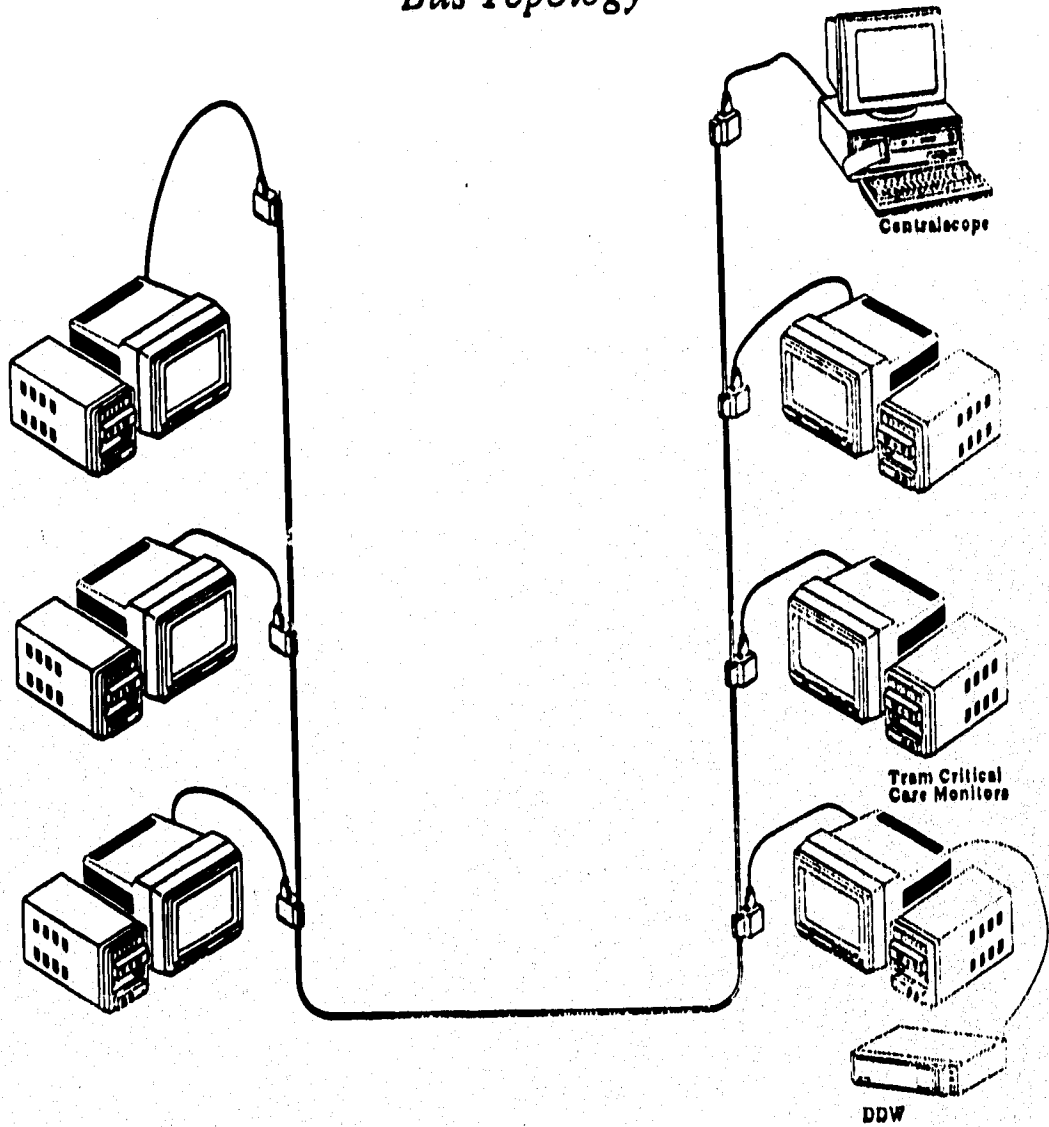


Figura 4.2.- Topologia Bus

Marquette Unity Network

Physical Star Topology

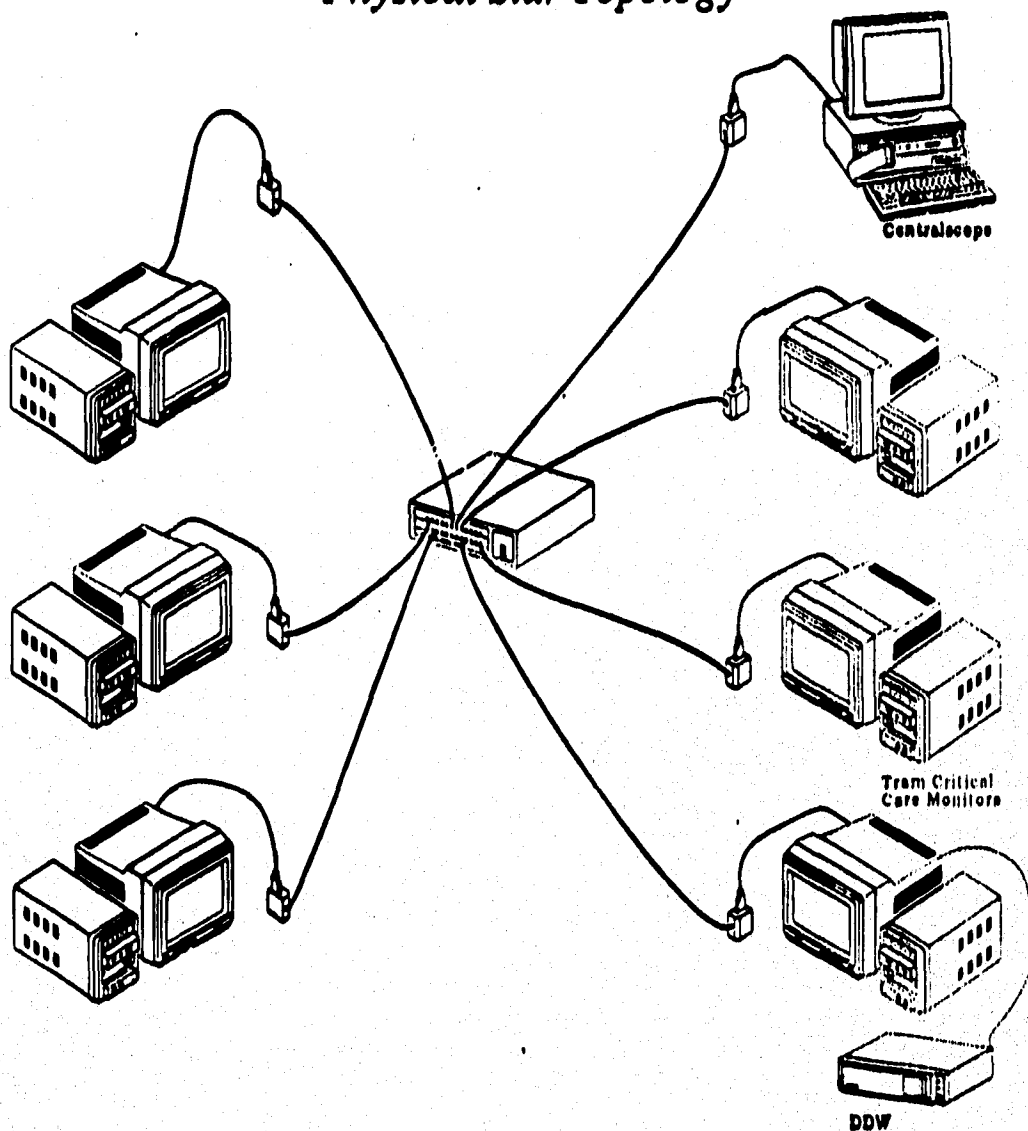


Figura 4.3.- Topologia Estrella

Marquette Unity Network

Ethernet Frame

IEEE 802.3 Data Packet Architecture

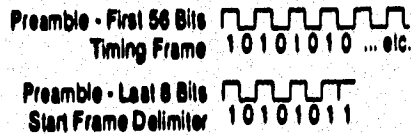
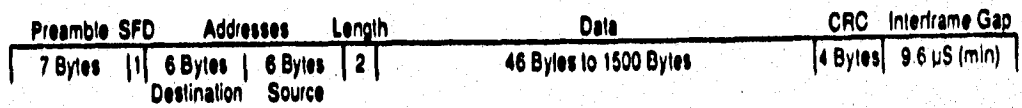


Figura 4.4.- Arquitectura de un paquete de datos (ETHERNET)

4.3.-PROTOCOLOS DE COMUNICACIÓN Y PROGRAMACIÓN PARA LA INTERACCION ENTRE MBM Y PCE

Dado que el sistema de comunicación serie es asíncrono y el proyecto requiere de una cierta sincronía para establecer la comunicación dúplex entre el MBM y el PCE, fue necesario programar ambos sistemas con un protocolo para realizar el intercambio de información en tiempo real. Este protocolo se diseñó particularmente para el proyecto, sin embargo, la modularidad del programa permite modificarlo para adecuarlo a otras necesidades.

El MBM realiza las tareas de conversión, detección, procesamiento y despliegue local, y al mismo tiempo está pendiente de la comunicación con el PCE. En la figura 4.5 se muestra el diagrama de flujo de la programación del MBM, en el cual se incluye el protocolo.

El protocolo se inicia cuando el PCE envía, vía puerto serie, la clave del o de los MBM's que desea supervisar. El programa del MBM detecta la clave que le llega y la compara con la clave que tiene asignada, si no son iguales continúa con el monitoreo y procesamiento de una muestra de signos vitales, generando un retardo 1 para esperar a que pasen 3 milisegundos, pero si corresponde a su clave, envía una señal de reconocimiento (ACK), la cual le indica al PCE que éste MBM se encuentra activo, posteriormente el PCE le envía el identificador para inicio de transmisión y el MBM envía un reconocimiento de dato recibido. En caso de que el MBM reciba un dato que no es inicio de transmisión, éste continúa con el procesamiento de otra muestra y genera un retardo 2 para ajustar esta rutina a 3 milisegundos; al recibir el inicio de transmisión el MBM se encuentra listo para enviar datos al PCE, como éste tiene que atender a todos los MBM conectados a la red, el tiempo transcurrido entre el envío de cada petición de dato puede ser mayor a 3 milisegundos, por lo que el programa del MBM tiene que continuar con su labor de muestreo y procesamiento, como se muestra en el diagrama de flujo de la figura 4.5.

En la misma figura se observa que para la transmisión de datos el PCE envía un "CHK" y el MBM un dato de ECG, continuando con este proceso hasta completar una pantalla, es decir la visualización de 4 periodos de ECG en un monitor VGA, lo que corresponde a enviar un dato cada 4 milisegundos (el equivalente a 256 muestras por segundo) por lo que el MBM envía una muestra al PCE por cada 2 muestras de ECG que procesa. Una vez concluido el envío de datos de ECG, el PCE envía un identificador

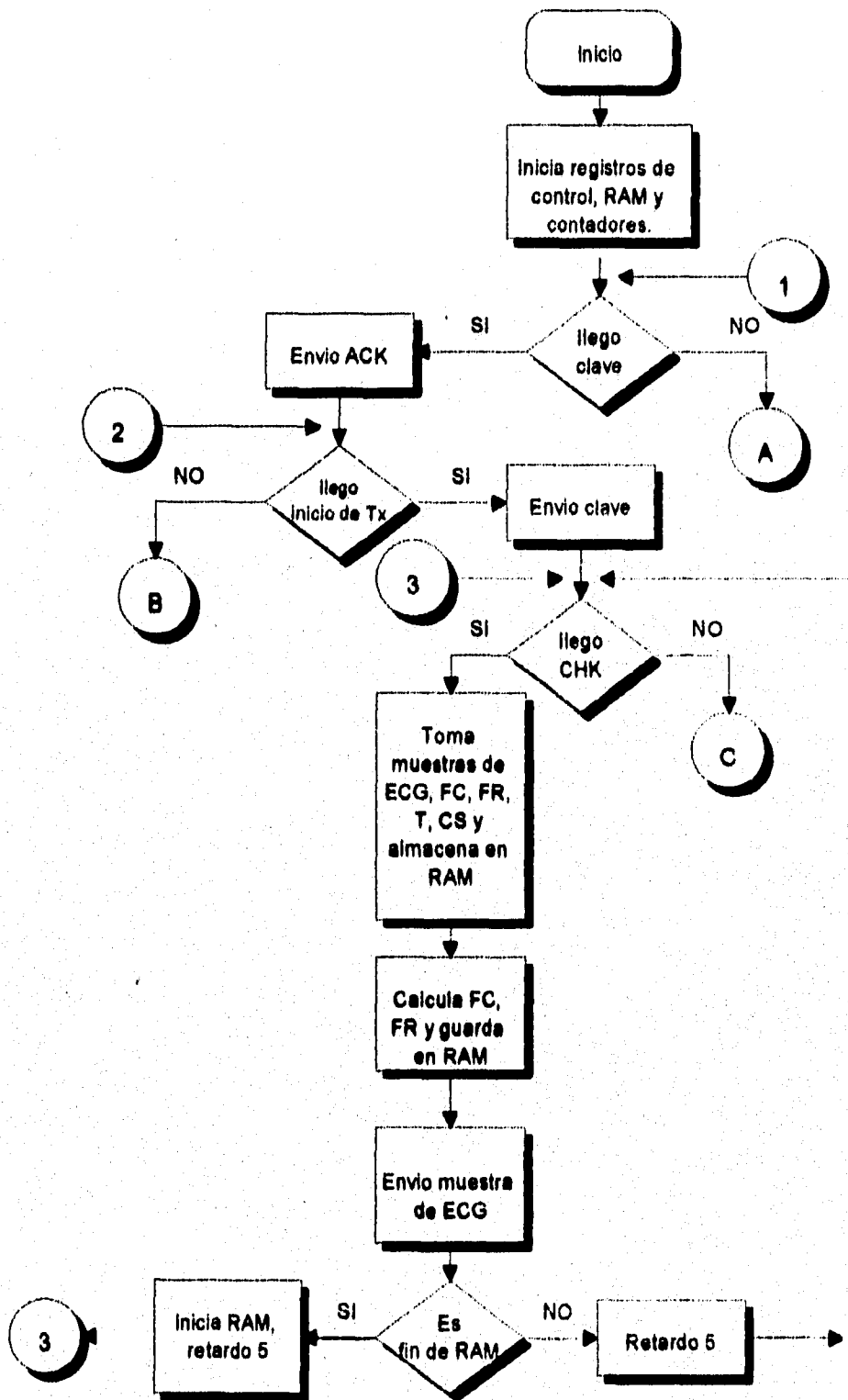


Figura 4.5 Diagrama de flujo de la programación del MBM

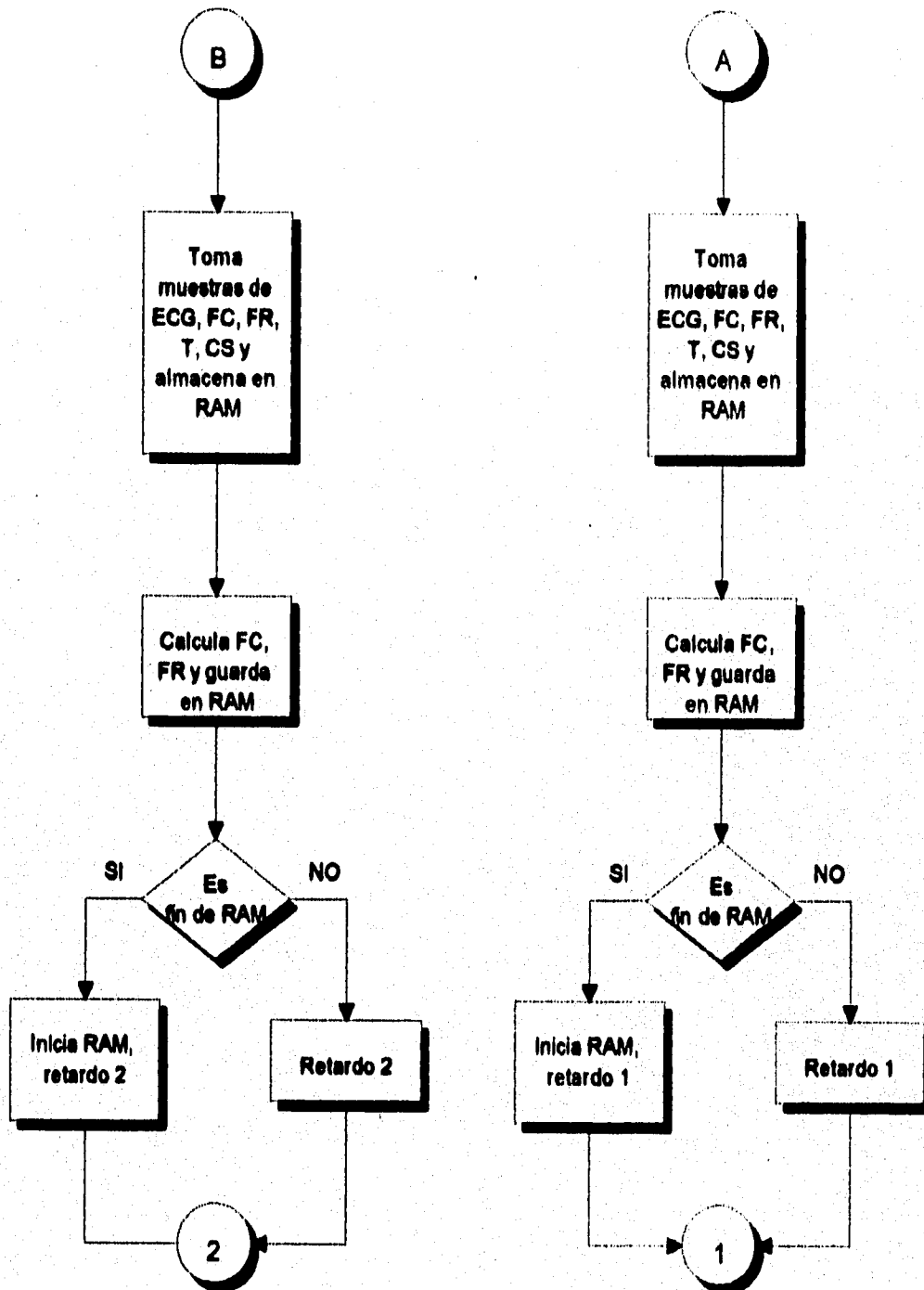


Figura 4.5.-Continuación.

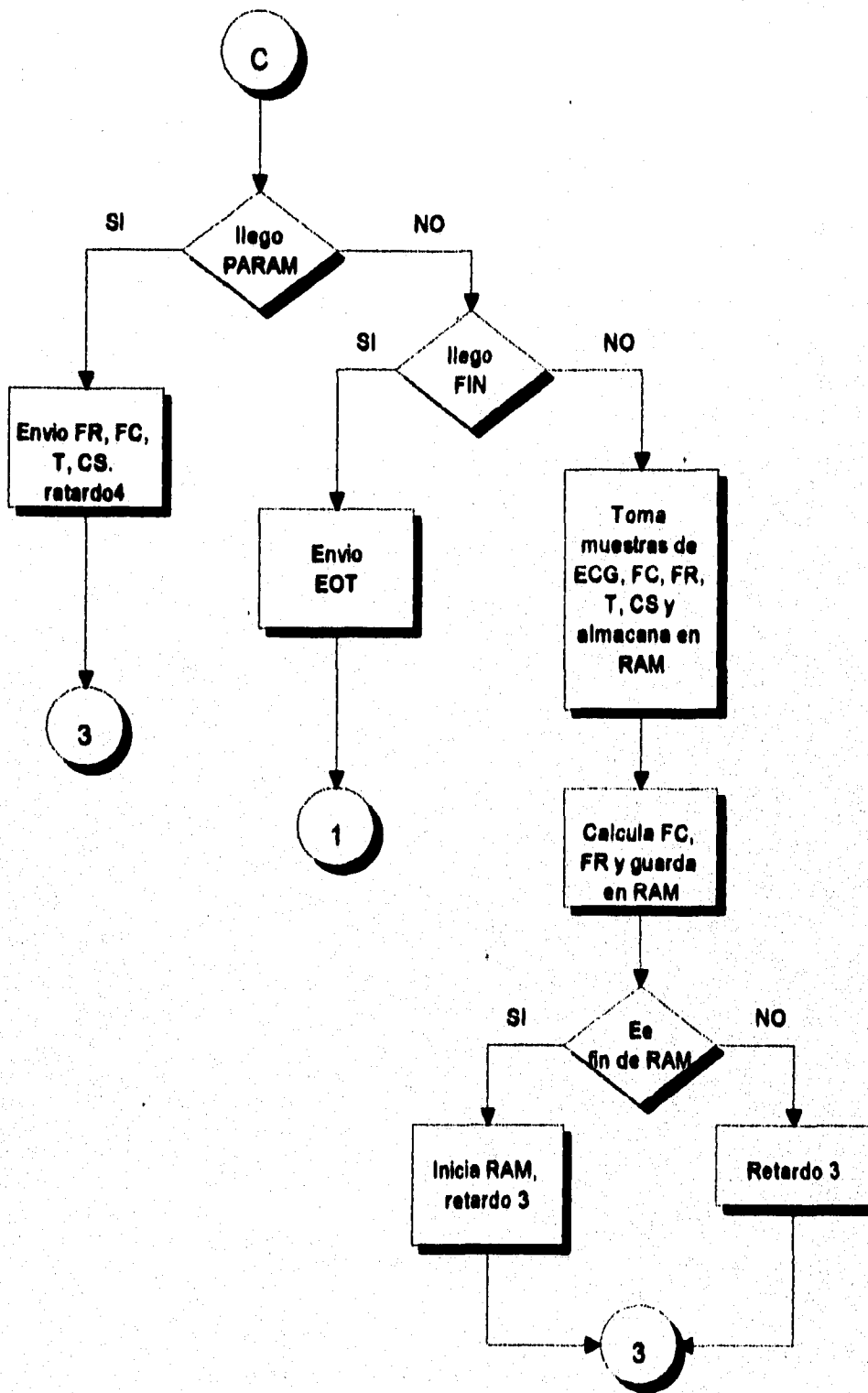


Figura 4.5.- Continuación.

(PARAM) para que le sean enviados los datos de las otras variables, al recibir este identificador el MBM lee de la RAM el dato de cada variable, enviándolo al PCE, continuando con este proceso mientras dure el monitoreo. Cuando termina se envía el identificador de fin de transmisión; al recibirlo el MBM, continua monitoreando y procesando los signos vitales, pero se encuentra en espera del protocolo para interactuar nuevamente con el PCE.

La programación del MBM se realizó en lenguaje ensamblador, su diseño es modular para permitir la inserción de nuevas rutinas que permitan realizar el despliegue local. En el apéndice A se incluye un listado completo del programa utilizado para el MBM, el cual se realizó siguiendo el diagrama de flujo de la figura 4.5.

CAPITULO 5

PROGRAMACION INTERACTIVA ELABORADA PARA EL PCE

El uso de una computadora para el despliegue y adquisición de datos de signos vitales en unidades de cuidados intensivos o en situaciones clinicas similares, es una ayuda muy significativa, ya que con ella se cumplen satisfactoriamente los requerimientos de las normas industriales de fabricación de monitores de signos vitales [6]; algunos de los requerimientos para el despliegue son los siguientes:

- Los métodos de despliegue deben ser capaces de presentar datos válidos en tiempo real.
- Deben tener la capacidad de retener la imagen para que pueda ser visualizada.
- Debe contar con una sección de presentación de alarmas.
- Deben desplegar a la vez caracteres alfanuméricos y gráficas.
- Deben tener la posibilidad de introducir fácilmente datos que interaccionen con el despliegue.

Tomando en cuenta estas características se diseñó un programa de supervisión y administración de la red para el PCE. El programa realiza una serie de tareas en las que se incluyen: la interacción con el personal médico, manejo de la información de los pacientes proporcionada por el usuario y la interacción con cada MBM conectado a la red. Estas tareas son transparentes para el usuario, por ello la programación se elaboró con base en menús, en los que el usuario solamente debe elegir la opción que desee realizar y proporcionar la información que se le pide. El programa se elaboró de manera modular para facilitar su mantenimiento y poder así integrar nuevas rutinas que mejoren el sistema; fue escrito totalmente en lenguaje C, ya que éste cuenta con las características necesarias para hacer el programa funcional y transportable [28].

Como se ha mencionado el programa se divide en dos partes fundamentales, interacción con el personal medico e interacción con cada MBM, la primera parte se encarga de administrar una pequeña base de datos para almacenar información general de los pacientes, así como de regular las condiciones de encendido de alarmas, esta parte del programa se desarrolló con el auxilio de la utileria de C denominada "Vlib" [29]. La

programación para la interacción con cada MBM incluye el protocolo de comunicaciones descrito en el capítulo anterior, incluye también la parte de ayudas gráficas, con las que se gráfica la señal de ECG y los demás signos vitales, cuenta con la activación de alarmas visuales y auditivas para la detección de signos vitales anómalos en el paciente. En los siguientes párrafos se describen detalladamente ambas secciones del programa, mientras que en el apéndice B se muestra un listado completo de la programación elaborada.

5.1.-PROGRAMACION INTERACTIVA CON EL PERSONAL MEDICO

La programación modular permite agregar rutinas para modificar el funcionamiento básico del sistema, por ello en esta parte de la programación se elaboraron varias rutinas que permiten trabajar con la base de datos e interactuar con el personal médico, estas rutinas son las opciones del menú principal, como podemos observar en el diagrama de flujo de la figura 5.1, las cuales se describen a continuación.

Alta de un paciente.- esta rutina actualiza los datos de un nuevo paciente, al seleccionarla aparece en pantalla un formato que el usuario debe llenar, en él se pide número de cama, nombre del paciente, dirección, edad y peso, los cuales pueden ser cambiados en cualquier momento. Se programaron también dos teclas especiales: <ESC> que se utiliza en todos los casos para cancelar una opción y regresar al menú principal, y <F2> con la que se guardan los datos en disco duro.

Parámetros del paciente.- esta rutina es la encargada de actualizar los niveles de activación de todas las alarmas, al seleccionar esta opción, se despliega en pantalla un formato en el cual el usuario deberá escribir el número de cama, los niveles altos y bajos de activación de alarmas para FC, FR, PS y T, en este punto a diferencia de algunos equipos nacionales y otros comerciales que solamente tienen niveles de alarma determinados, se cuenta con la opción de poder elegir cualesquiera niveles, para activar las alarmas de todas las variables, cumpliendo así con los requerimientos para el sistema de alarmas especificados en la norma 3.2.8 de la (AAMI). Esta opción cuenta también con las teclas de <ESC> y <F2> para retornar al menú principal y guardar la información respectivamente.

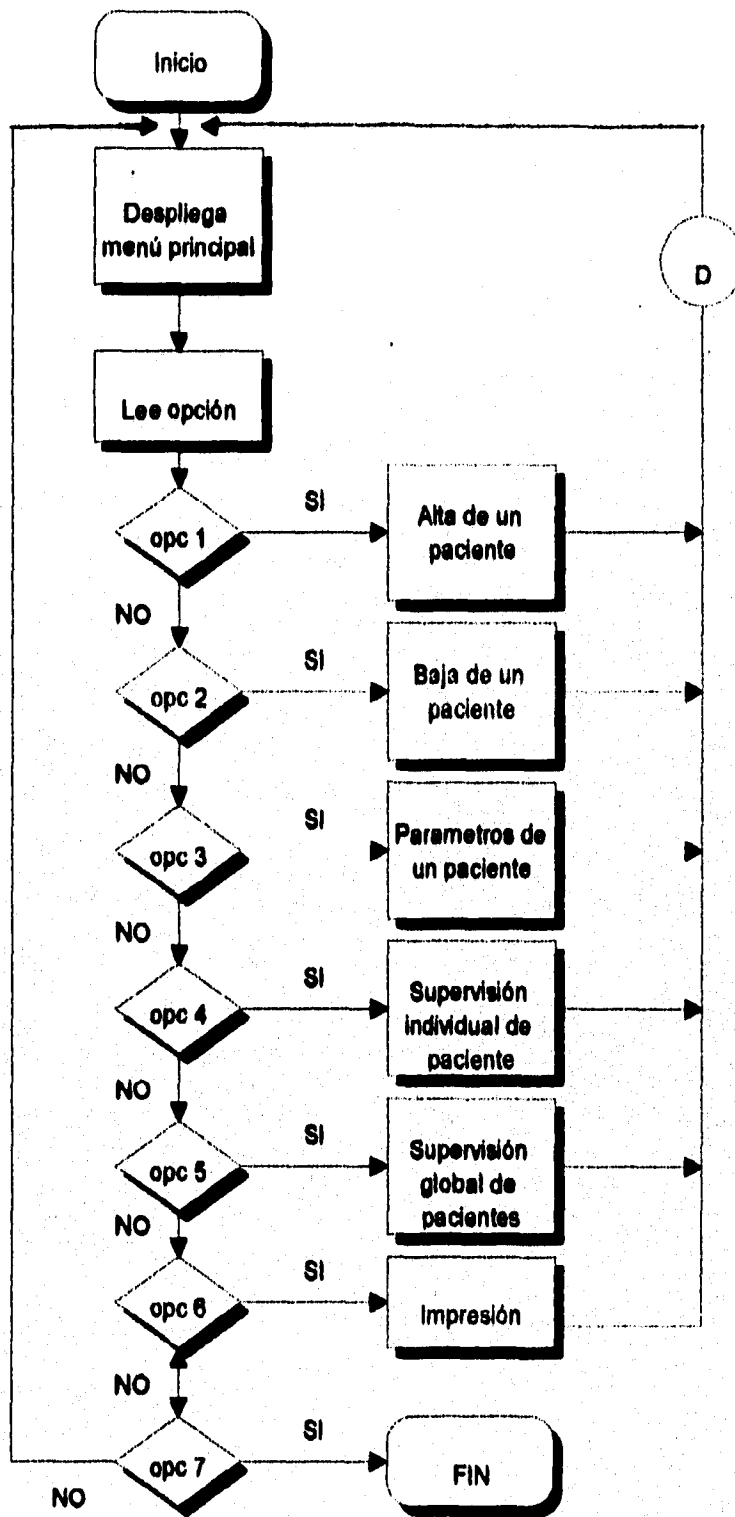


Figura 5.1 Diagrama de flujo para la programación interactiva con el personal medico

Baja de un paciente.- cuando se desea retirar un paciente de la base de datos, esta rutina es la encargada de eliminar todos sus datos incluyendo los niveles de activación de alarmas. Al seleccionar la opción se despliega una ventana preguntando por el número de cama a dar de baja, posteriormente se despliegan en pantalla los datos del paciente seleccionado, esto con el fin de verificar que sea el paciente que se desea dar de baja, finalmente se ratifica la opción seleccionada quedando limpio su registro y retornando el programa al menú principal.

Supervisión individual.- esta rutina se encarga de desplegar en forma gráfica los signos vitales del paciente, dibujando la señal ECG y mostrando los valores numéricos de FC, FR, PS y T, indicando también condiciones anómalas en los signos vitales. En la siguiente sección se discutirán los algoritmos empleados para la graficación de los signos vitales.

Supervisión global.- al igual que en la rutina anterior al ejecutar esta opción se grafican los datos de todos los MBM conectados al PCE, en esta opción se gráfica el ECG y se detectan anomalías en signos vitales para la activación de alarmas. Estos algoritmos se discuten en la siguiente sección.

Impresión.- esta rutina se encarga de adquirir un bloque de datos, es decir 4 periodos de la señal de ECG, con sus respectivos datos de FC, FR, PS y T, y los imprime junto con los datos del paciente, la rutina permite elegir dos tipos de impresora, matriz de puntos y láser. Esta opción no se utiliza para tener registro del monitoreo en tiempo real, ya que la velocidad de impresión es de aproximadamente 3 minutos por gráfica, solamente se incluye para tener un registro de un determinado momento del monitoreo.

5.2.-PROGRAMACION PARA EL MONITOREO DE PACIENTES

Esta parte del programa está formada por tres rutinas gráficas, dos de ellas se utilizan para el monitoreo de signos vitales y una para la impresión de la señal en papel. Como se está graficando en tiempo real los procedimientos no pueden ser muy largos, se deben ejecutar en el menor tiempo posible, por ello se utilizan utilerías gráficas de turbo C llamadas "graphics.h". Las rutinas de supervisión individual y global involucran procedimientos gráficos semejantes, por lo que detallando el algoritmo para supervisión

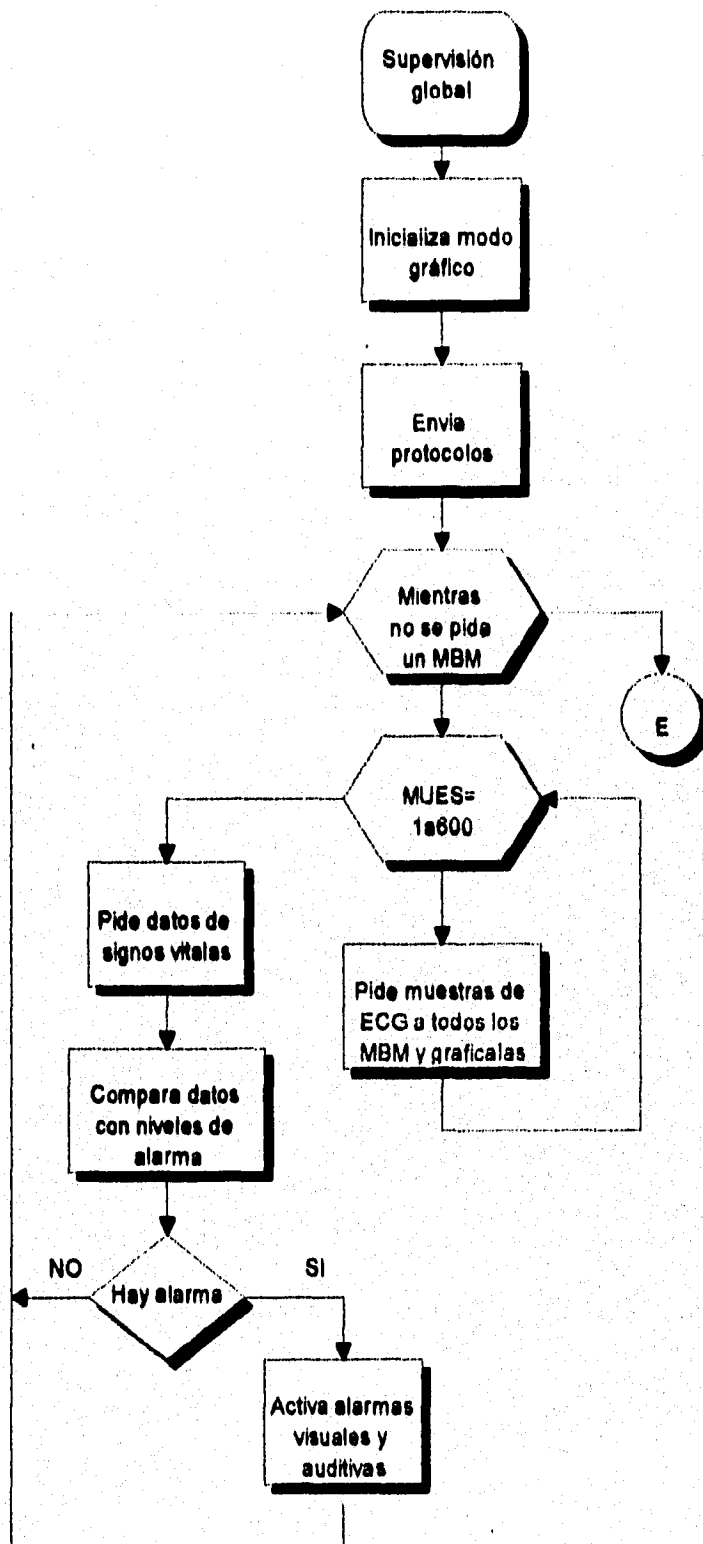


Figura 5.2 Diagrama de flujo para la programación de monitoreo de pacientes

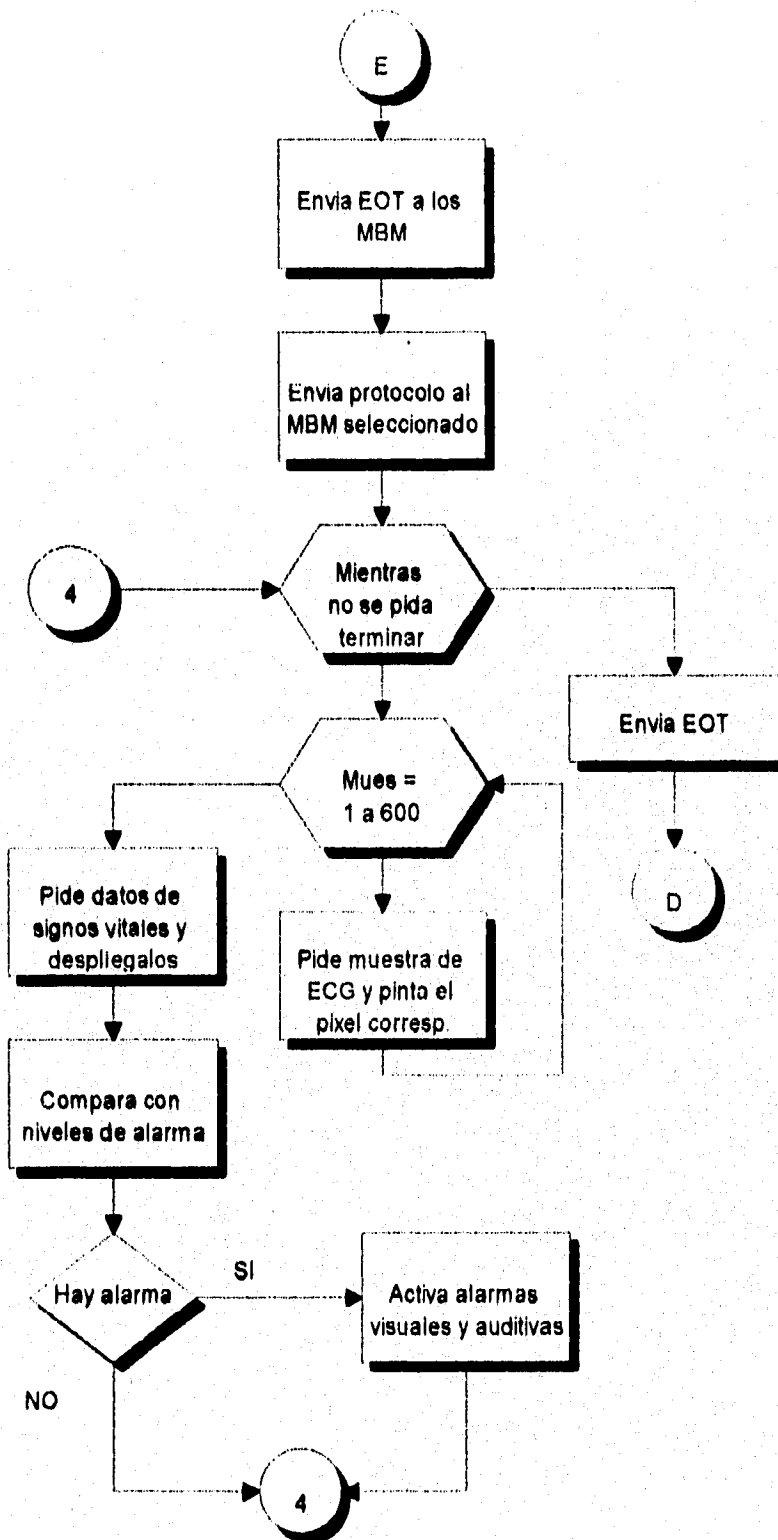


Figura 5.2 Continuación.

global se entenderá el individual. Estas rutinas cuentan con varios procedimientos, como se observa en el diagrama de flujo de la figura 5.2, los cuales se describen a continuación.

Inicializa.- en esta parte se inicializan todas las variables involucradas así como el modo gráfico de C para poder ejecutar las rutinas gráficas.

Protocolos.- para iniciar la identificación de los MBM conectados a la red se envía el protocolo de comunicaciones, dejando a cada MBM conectado listo para el intercambio de información.

Estando desplegada la pantalla de supervisión global y mientras no se pida supervisar un MBM en particular, se ejecutarán los siguientes procedimientos:

Muestras.- una vez identificados los MBM activos se procede a pedir muestras de ECG en orden ascendente, es decir, cada MBM es identificado con un número del 1 al 14, siendo en ese orden la petición de muestras de ECG, se pide una muestra y se pinta el pixel correspondiente de cada MBM, continuando con este proceso hasta completar 150 muestras, que es la resolución que tenemos en el monitor VGA para cada MBM, lo que representa una baja resolución para el monitoreo de ECG, sin embargo el objetivo de este modo de supervisión es el detectar signos vitales anómalos para encender las alarmas y pasar entonces a supervisar de manera individual.

Parámetros.- al completarse la graficación de las 150 muestras de ECG se ejecuta este procedimiento, en el cual se piden los datos de FC, FR, PS y T, los cuales serán comparados con los niveles de alarma proporcionados por el personal médico en la opción de parámetros del paciente.

Alarma.- en el caso de que la comparación reporte signos vitales anómalos, se ejecuta este procedimiento, el cual activa las alarmas visuales y auditivas, las visuales constituyen botones gráficos a los que se añade color y van acompañadas por tonos audibles de corta duración.

Si se pide monitorear algún MBM se ejecutarán los siguientes procedimientos, los cuales corresponden a la supervisión individual.

Termina.- este procedimiento envía el identificador de fin de transmisión a todos los MBM activos, dejándolos listos para iniciar un nuevo protocolo.

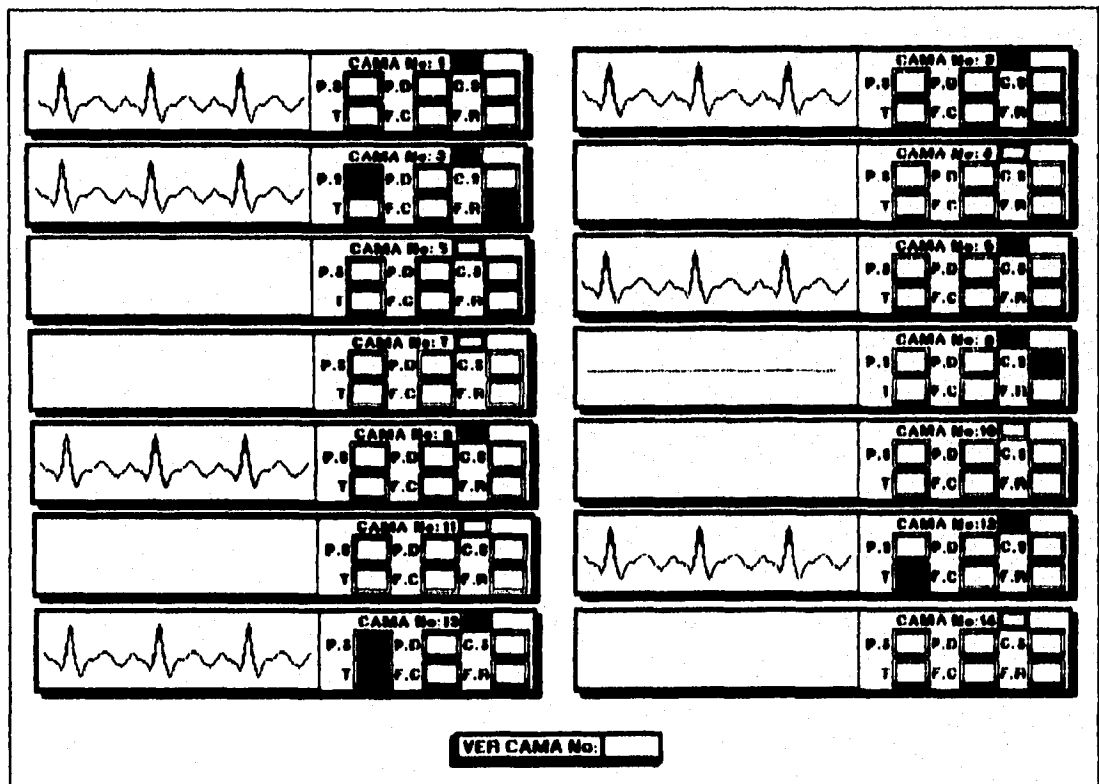
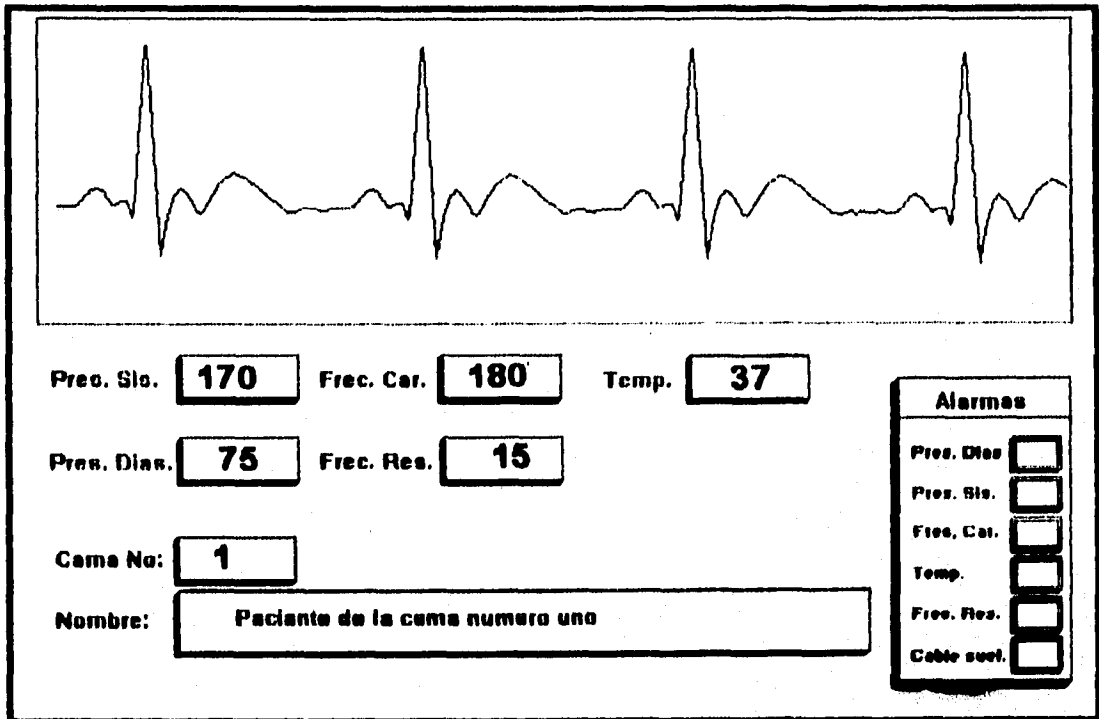


Figura 5.3 Pantallas para monitoreo individual y global.

Protocolo.- esta rutina envía el protocolo solamente al MBM seleccionado, es decir solamente habrá intercambio de información con él.

Muestra_indiv.- este procedimiento, solicita una muestra de ECG al MBM seleccionado y gráfica su pixel correspondiente, continuando con este proceso hasta completar 600 muestras (determinado por la resolución del monitor), lo que representa una frecuencia de muestreo de 250 Hz para una visualización de 4 periodos de ECG, tomando como base la señal mostrada en la figura 2.2; es decir, se tiene máxima resolución para el monitoreo de ECG.

Parámetros indiv.- al terminar de graficar las 600 muestras de ECG se ejecuta este procedimiento, el cual pide los datos de los otros signos vitales, los despliega en pantalla y realiza la comparación con los niveles de activación de alarmas, los que en caso de ser superiores a los previamente establecidos, activaran la rutina de encendido de alarmas descrita anteriormente.

Termina.- al ejecutarse este procedimiento se envía el comando de fin de transmisión al MBM y regresa el programa al menú principal.

El programa diseñado puede correr en cualquier sistema con 640 Kbytes de memoria, puerto serie, disco duro y monitor VGA, pudiendo correr con algunas modificaciones en un monitor EGA o CGA.

En la figura 5.3 se muestran las pantallas para el modo de supervisión individual y global.

CAPITULO 6

PRUEBAS REALIZADAS, CONCLUSIONES Y RECOMENDACIONES

Como se ha mencionado en los capítulos anteriores el equipo fue diseñado de manera modular, cada módulo fue probado de manera individual, esto es, primero se probó el MDSV, luego el MADCSV para que posteriormente de manera conjunta formaran el MBM. En el presente capítulo se describirán las pruebas de funcionamiento realizadas al MBM (MDSV Y MADCSV) y al PCE, así como las pruebas realizadas a todos los módulos trabajando en conjunto a excepción del enlace de comunicaciones.

Las pruebas realizadas se dividen en tres partes:

- 1.- Pruebas modulares, en las cuales se probó cada módulo simulando las señales a procesar.
- 2.-Pruebas con simulador, en las cuales se probó el equipo de manera conjunta utilizando como fuente de señal un simulador de ECG, el cual es utilizado para la calibración de equipo médico.
- 3.-Pruebas con paciente, en las cuales se cambia el simulador por un paciente voluntario, el cual fue supervisado por personal calificado en el área médica.

6.1.-PRUEBAS MODULARES

Este tipo de pruebas se llevaron a cabo de acuerdo al avance del proyecto, probando primeramente el MDSV, para luego seguir con el MADCSV y por último el PCE.

6.1.1.-Pruebas al MDSV

Las pruebas que se realizaron en esta etapa fueron iniciales, ya que se necesitaba del MADCSV y una etapa de despliegue para poder verificar su correcto funcionamiento. Por lo que lo único que se hizo fue suministrar a este módulo las variables necesarias para tener valores aproximados en cuanto a FC, FR, y temperatura.

Para la señal de ECG se utilizó un simulador de ECG el cual se visualizó en un osciloscopio de memoria , ya que como es una señal de una frecuencia muy baja no se alcanza a visualizar con un osciloscopio convencional. Cabe mencionar que en esta parte se diseñó un circuito de refresco de memoria el cual proporcionó una señal con las mismas características que la que mostraba el osciloscopio de memoria. Se debe subrayar que tanto la FC, como la FR no se podían probar a un 100 %, esto debido a que como estos signos vitales son valores numéricos, se necesitaba de un "frecuencímetro" especial , ya que lo que nos proporciona esta etapa son pulsos que siguen una relación con los valores antes mencionados. Para el caso de la temperatura no hubo tanto problema ya que se normalizó la temperatura a su equivalente en voltaje, por lo que con ayuda de un voltmetro se pudo verificar su valor.

6.1.2.-Pruebas realizadas al MADCSV

Estas pruebas fueron realizadas para probar la arquitectura digital diseñada. Como una primera aproximación debido a que el MDSV no había sido probado completamente, se simularon las señales vitales a procesar como se observa en el diagrama de la figura 6.1.

Se elaboró un circuito que genera señales de baja frecuencia, cubriendo el rango de la señal de ECG (0.5 - 40 Hz) y que entrega pulsos a la misma frecuencia (30 -240 pulsos por minuto) simulando la señal de FC y FR, además de proporcionar un voltaje correspondiente a la temperatura (15 - 45 °C), con este circuito se simularon la mayor parte de las características del MDSV. Las señales generadas se conectan al convertidor A/D para ser procesadas y enviadas por el puerto serie hacia el PCE para su graficación. Con el correcto funcionamiento del *software* desarrollado se obtuvo un buen funcionamiento de la arquitectura diseñada.

Cabe mencionar que una vez que se probó esta parte, se interconectó el MDSV y el MADCSV para que pudieran interactuar con el PCE como etapa de despliegue (simulando el MBM) y así corroborar el buen funcionamiento que tuvo con el circuito generador de señales que se diseñó

6.1.3.-Pruebas a la programación del PCE

El objetivo de estas pruebas fue verificar el buen funcionamiento de las rutinas en ensamblador para el MBM (o mejor dicho del MADCSV), las rutinas de graficación de

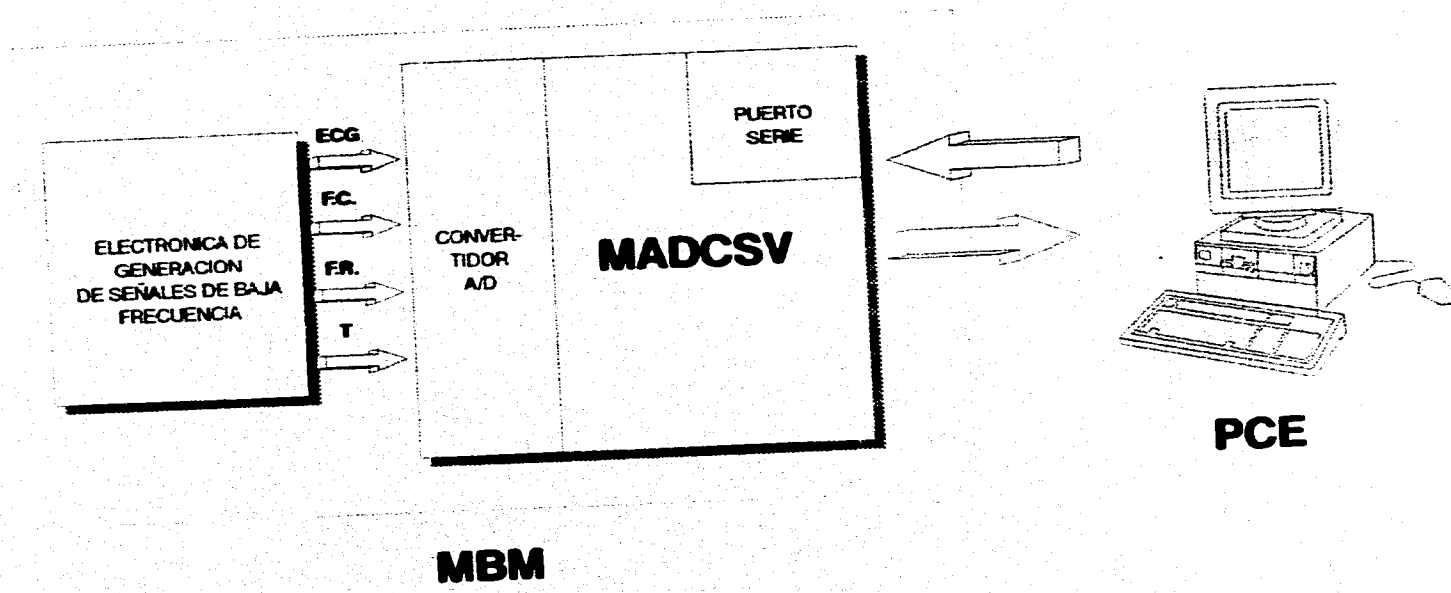


Figura 6.1.- Diagrama de bloques para las pruebas realizadas al MBM

ECG, como la obtención y presentación de los valores numéricos de los signos vitales del paciente así como la correcta activación de las alarmas . Para las rutinas de graficación se utilizaron archivos de datos para simular la señal de ECG y los valores de los signos vitales, la graficación de estos datos se realizó de manera adecuada en el modo de supervisión individual, para el modo de supervisión global se utilizó el mismo archivo de datos para los 14 MBM, en ambos casos la graficación se obtuvo de una manera nítida y sin problemas.

En cuanto a la programación en ensamblador del MBM, ésta se realizó probando por separado el buen funcionamiento de cada subrutina de procesamiento, utilizando para ello la ayuda de un simulador del microcontrolador llamado "AVSIM11" , con el cual podemos representar en la computadora todas las características del MCU y así evaluar los programas elaborados. Con el simulador se verificó el buen funcionamiento de las rutinas de conversión y almacenamiento de datos, el cálculo de FC y FR, el monitoreo de puertos, transmisión y recepción de datos a través del puerto serie, todo realizado en tiempo real.

6.2.- PRUEBAS REALIZADAS CON SIMULADOR DE ECG

El objetivo de este tipo de pruebas es realizar la integración de todos los módulos, es decir el MBM (MDSV y MADCSV interconectados) y PCE, además de realizar la calibración adecuada del MBM para la correcta obtención de los valores numéricos de FC y FR. Para lograrlo se utilizó como fuente de señal un simulador de ECG, el cual cuenta con todas las derivaciones posibles [4] para la generación de la señal ECG y frecuencias de 30, 60, 120 y 240 ppm., con lo que se cubre el rango de frecuencias para monitoreo cardíaco; este simulador es utilizado en hospitales para la calibración de equipo de monitoreo cardíaco, por lo que se utilizó como referencia para la calibración del equipo diseñado. Las pruebas se realizaron con la configuración mostrada en el diagrama de la figura 6.2, se utilizaron los programas del MBM y el PCE probados anteriormente. Las pruebas de funcionamiento continuo proporcionaron los siguientes resultados:

- La visualización de la señal de ECG fue satisfactoria, es decir se presenta una señal con buena nitidez comparándola con la señal original mostrada en el osciloscopio.
- Los valores numéricos de FC, FR y T son aceptables en el rango de 30 a 120 ppm; en el

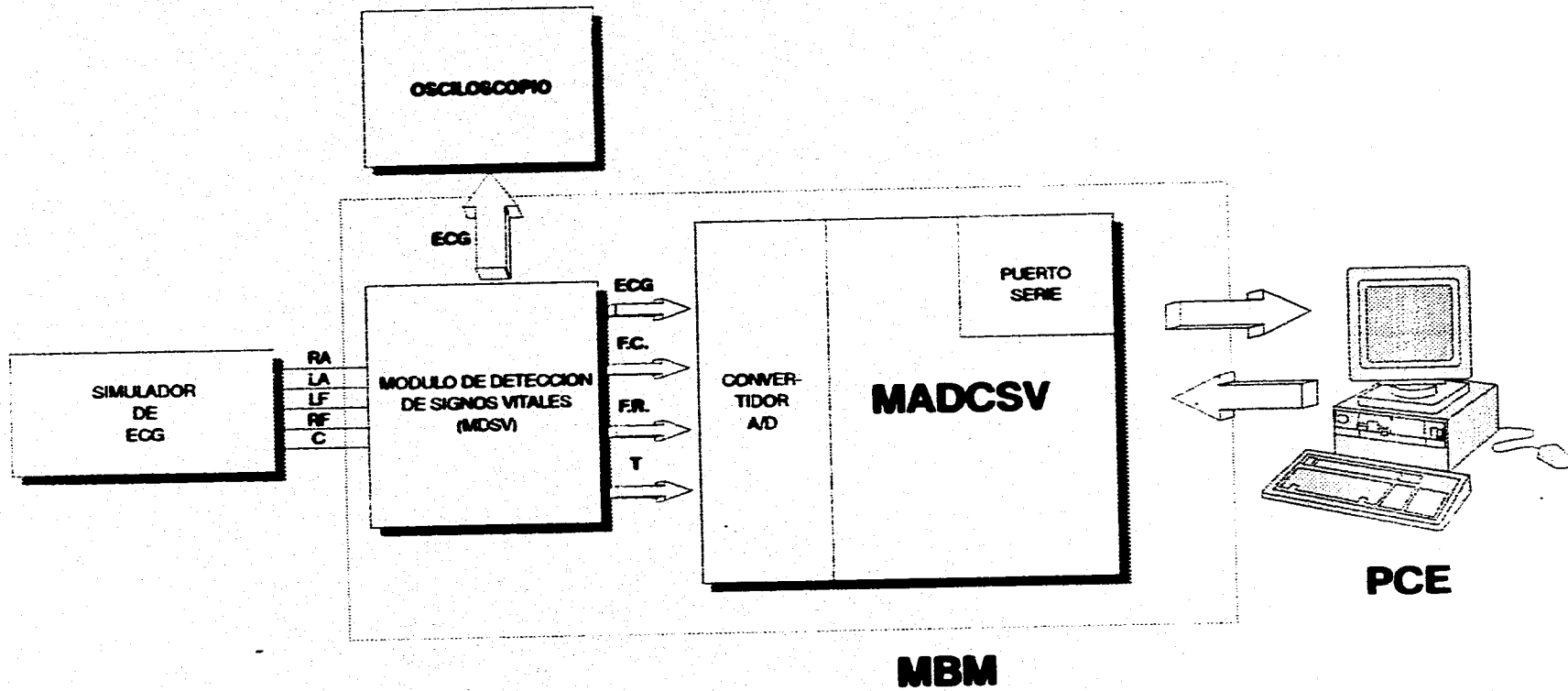


Figura 6.2.- Diagrama de bloques para las pruebas realizadas con simulador

TABLA 2

simulador	MBM	error
FC en ppm	FC en ppm	en ppm
30	30	0
60	60	0
120	121	1
240	238	2

rango de 240 ppm se tiene un error de 2 ppm, el cual de acuerdo a las normas AAMI es aceptable y coincide con el error teórico esperado, en la tabla 2 se muestran estas características de funcionamiento.

-La activación de alarmas se realizó de manera adecuada al sobrepasar los límites establecidos.

En la figura 6.3 se muestra una impresión de la señal ECG y los datos de los signos vitales obtenidos al conectar el simulador, esta gráfica muestra el buen funcionamiento del software diseñado para la impresión y graficación de la señal.

6.3.- PRUEBAS REALIZADAS CON PACIENTE

Este tipo de pruebas requiere de un control adecuado del equipo ya que al encontrarse conectado directamente al cuerpo puede resultar "peligroso" para el paciente si el equipo no está debidamente aislado. Estas pruebas se realizaron bajo la supervisión de personal especializado, los cuales realizaron la validación del equipo y con la ayuda de un voluntario se hicieron las pruebas de monitoreo obteniendo los siguientes resultados:

-El monitoreo de la señal de ECG es nítido pero presenta desplazamientos de la señal a medida que se mueve el paciente, esto es debido a que el módulo amplificador no cuenta con circuitos de corrección de línea basal

-Los valores de los signos vitales los podemos considerar correctos, ya que el equipo se probó y calibró con el simulador de ECG. El valor de FR se encuentra dentro del rango normal, comparado con un equipo comercial calibrado, con lo que aseguramos que se trata de un valor correcto de FR. La temperatura desplegada se comparó con un

termómetro digital, con lo cual podemos asegurar su correcto funcionamiento. Los sensores del equipo diseñado y los de equipos comerciales se colocaron al mismo paciente para realizar una comparación de ambos equipos. En la tabla 3 se muestran los resultados obtenidos en esta prueba.



Pres. Sist. : 1 Frec. Car. : 61 Temp. : 36.7

Pres. Diast. : 1 Frec. Resp. : 7

Canal No. : 1

Nombre :

Figura 6.3.-Impresión de ECG y signos vitales utilizando simulador

TABLA 3 .- Comparación de parámetros entre el MBM y un equipo comercial Air Shields Modelo System 6S.

MBM				Equipo comercial		
No lectura	FC	FR	Temp.	FC	FR	Temp
	(ppm)	(ppm)	(°C)	(ppm)	(ppm)	(°C)
1	58	14	36.4	58	13	36.5
2	60	13	36.6	60	13	36.6
3	60	15	36.1	60	15	36.0
4	65	15	36.3	65	15	36.5
5	71	17	37.0	70	16	36.9
6	78	18	36.6	78	18	36.6
7	70	16	36.4	71	15	36.4
8	68	14	36.7	68	14	36.6
9	60	13	36.3	60	13	36.3
10	60	12	36.4	60	12	36.1

En la figura 6.4 se muestra la impresión de siete derivaciones de ECG y los signos vitales del paciente, mientras que en la figura 6.5 se muestran algunas impresiones de otros equipos diseñados en el país. Es importante mencionar que para una validación clínica, el equipo debe ser probado y validado por personal médico especializado.

6.4.-CONCLUSIONES Y RECOMENDACIONES.

De las pruebas y las comparaciones a las que fue sometido el equipo diseñado con un equipo comercial, se pudo observar que los valores que proporcionó eran similares, los cuales se pueden observar en la tabla 3. Cabe mencionar que estas pruebas se llevaron durante largos periodos de monitoreo, quedando comprobado que éste puede soportar largas jornadas de trabajo, manteniendo un comportamiento estable. Además, durante las pruebas realizadas con paciente, uno de ellos presentó un rasgo anormal en el registro obtenido entre la finalización de la onda T y el inicio de la onda P del siguiente complejo, aspecto que nos preocupó, pero que se pudo aclarar cuando el paciente se hizo un examen médico, lográndose detectar un problema de grasa en el corazón, logrando con ello incrementar la confianza en el equipo desarrollado.

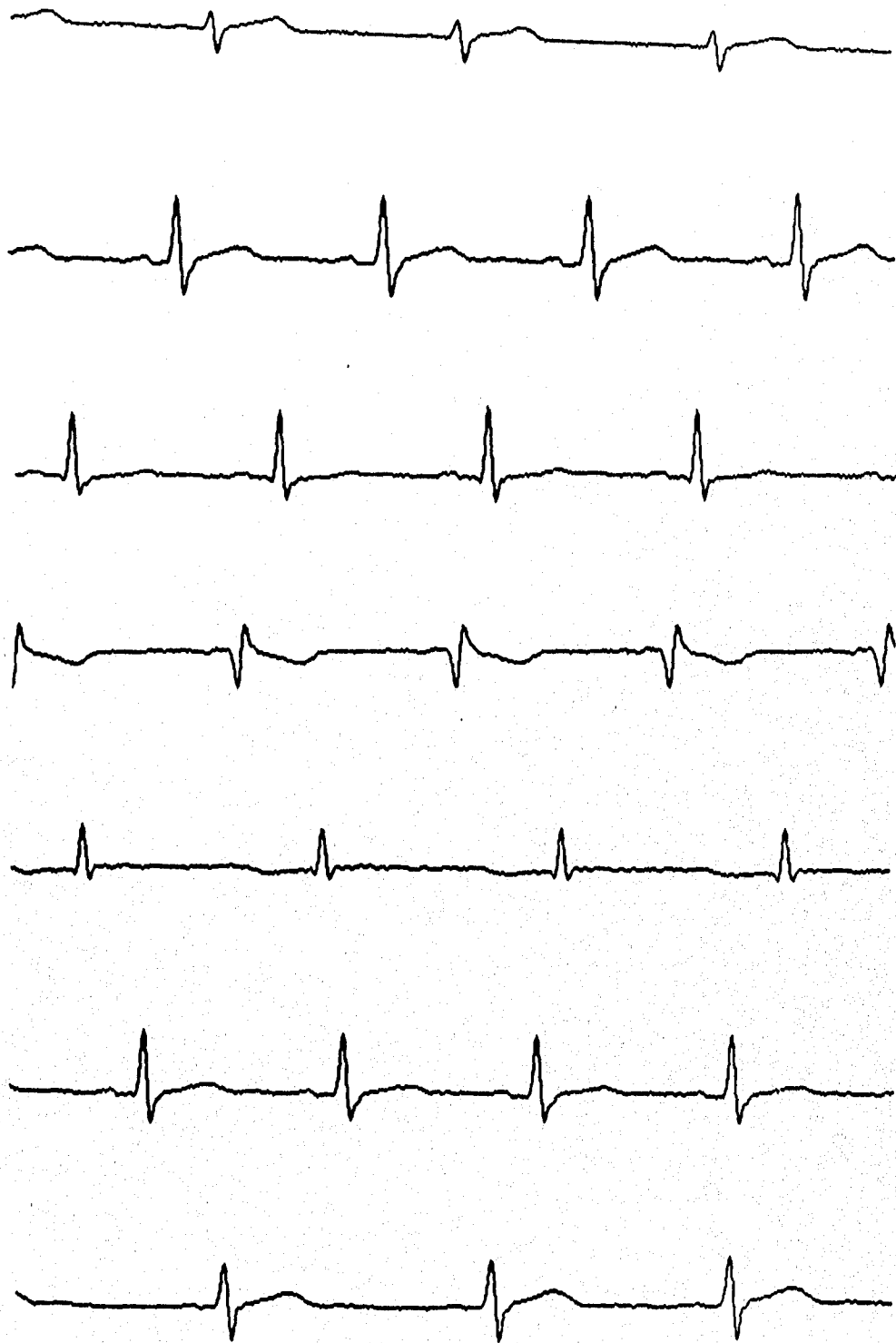
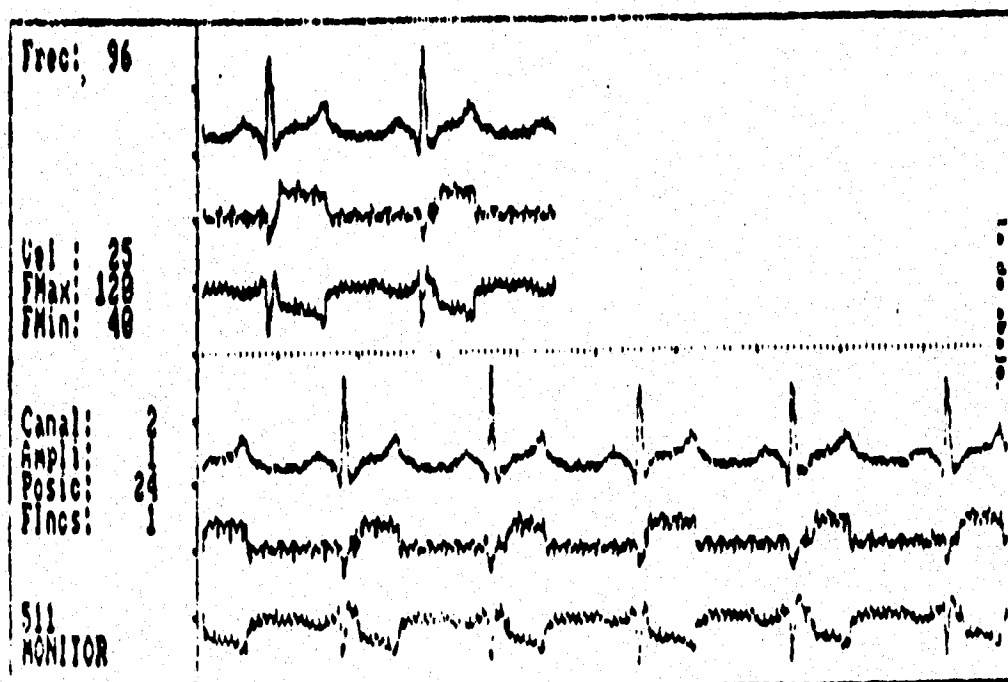
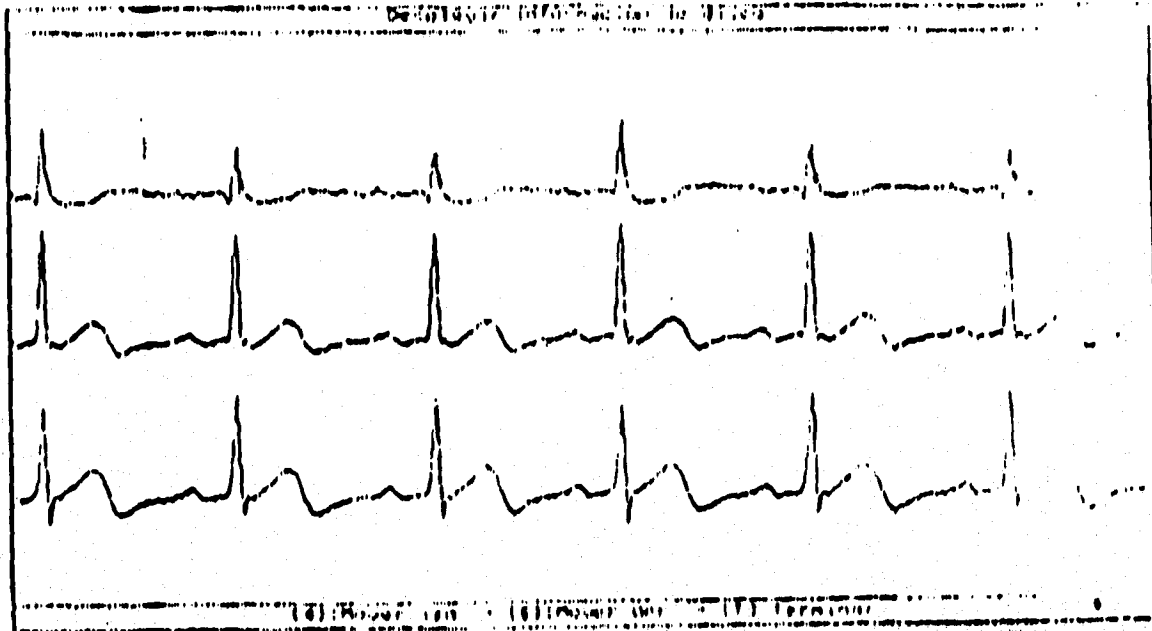


Figura 6.4.- Impresión de ECG y signos vitales de un paciente



F1)Sube F2)Baja F3)Congela F4)Guarda F5)Ampli F6)Reduce F7)Vel
 F8)Up-PgDn)NivelDc -+)Incremento 1..3)Canal (End)

Figura 6.5.-Impresiones de ECG de otros equipos diseñados en el país

Como todo equipo desarrollado este es susceptible de mejorarse, por lo que el sistema fue diseñado de tal manera que pudiera aceptar otras señales biomédicas, como pudieran ser la Presión Sanguínea (PS), la Saturación de Oxígeno (SaO), etcétera. Para el caso de la presión sanguínea el equipo fue pensado de tal manera que pudiera comunicarse con el a través de puerto serie, valiéndose para ello de un módulo de presión sanguínea (DYNAPULSE) el cual proporciona la información o datos vía RS-232. Este módulo es fabricado por Pulse Metric INC.

Para finalizar se puede decir que :

- El sistema proporciona el monitoreo de signos vitales de ECG, FC, FR y temperatura.
- Los valores de los signos vitales son muy precisos y el despliegue tiene una nitidez comparable a la de sistemas comerciales
- Permite la impresión de cualquiera de las derivaciones a través del PCE.
- Es modular por que permite incluir nuevas señales biomédicas.
- El sistema puede trabajar de manera individual o de manera global al estar conectado en red.
- Cumple con las normas AAMI para el monitoreo de pacientes
- El costo del equipo es sumamente bajo hasta en un 70% con respecto a los equipos comerciales.
- El diseño es reducido y compacto.
- Se utilizó tecnología de vanguardia para su desarrollo.
- Se presenta como una posible opción la comunicación de datos de los MBM's hacia el PCE a través de la línea eléctrica de baja tensión.

En la siguiente tabla se listan las especificaciones del equipo diseñado.

ESPECIFICACIONES GENERALES

Características	Especificaciones
<u>Impedancia de entrada</u> modo común modo diferencial	mayor a 5×10^{10} mayor a 10^8
<u>ECC</u> respuesta en frecuencia ganancia calibración aislamiento	0.5 a 40 Hz 1500 sin calibración mayor a 120 dB
<u>Frecuencia cardíaca</u> intervalo precisión límite de alarmas	20 a 250 PPM $\pm 1\%$ ó 3 PPM fijado por el usuario en cualquier rango
<u>Frecuencia respiratoria</u> intervalo precisión límite de alarmas	8 a 150 PPM $\pm 2\%$ ó 3 PPM fijado por el usuario en cualquier rango
<u>Temperatura</u> rango precisión límite de alarmas	15 a 45°C $\pm 0.2^\circ\text{C}$ fijado por el usuario en cualquier rango
<u>Comunicación</u> Longitud de palabra modo de operación velocidad de transmisión tipo de comunicación interface	8 bits de datos, 1 bit de inicio, 1 bit de stop, sin paridad full duplex 9600 bauds asíncrona RS-232
<u>Sistema digital</u> frecuencia de muestreo resolución del convertidor velocidad de conversión intervalo de conversión memoria RAM memoria EPROM	250 Hz 8 bits 16 microsegundos 0 a 5 volts 2 Kbytes 4 Kbytes

REFERENCIAS

- 1.-Revista Mexicana de ingeniería Biomédica "VII y VIII Congresos de Ingeniería Biomédica", Volúmenes 5 y 6; 1984, 1985.
- 2.-Centro de Desarrollo de Aplicaciones Tecnológicas (CEDAT) "Catalogo de Servicios y Productos", Secretaria de Salud, México 1992.
- 3.-Departamento de cardiología del Centro Medico la Raza (Hospital de Especialidades), Unidad de Cardiología y Unidad de Terapia Intensiva de Adultos del Hospital Central Militar.
- 4.-Walter Welkowitz "Biomedical Instruments Theory and Desing", Editorial John Wiley & Sons,, New York 1975.
- 5.-Richard Aston "Principles of Biomedical Instrumentation and Measurement", Editorial Merrill Publishing Company, Boston 1990
- 6.-Anonymous "American National Standar for Cardiac Monitors, Heart Rate Monitors, Heart Rate Meters and Alarms", ANSI-AAMI, American National Standar 1983.
- 7.-Leslie Cronwell, Erick A. Pfeiffer "Biomedical Instrumentation and Measurement", Editorial Prentice Hall, New York 1983.
- 8.-Chris Barnard "Enciclopedia Médica" Editorial Nauta, Sudáfrica 1991.
- 9.- Bruce C. Winter, John G. Webster "Driven - Right - Leg Circuit Design", IEEE Transactions on Biomedical Engineering, Vol. BME-30 No 1 January 1983.
- 10.-Enciclopedy of Biomedical Instrumentation, New York 1982.
- 11.-Marvin Weiss "Biomedical Instrumentation", Editorial Chilton Book, Baltimore 1979.
- 12.-Eugene R. Hnatek "Aplications of Linear Integrated Circuits", Editorial Wiley Interscience Publications, Chicago 1975.
- 13.-Como interpretar un ECG (Tribuna Medica)
- 14.-Nitish V. Thakor, John G. Webster and Willis J. Tompkins "Stimation of QRS complex Power Spectra for Design of QRS Filter", IEEE Transactionn on Biomedical Engineering, Vol BME-31, November 1984.
- 15.-Robert Boylestad "Electrónica, Teoria de Circuitos", Editorial Prentice Hall, México 1990.
- 16.-Motorola Semiconductors "Reference Manual MC68HC11", Motorola Corp. 1991.
- 17.-John Webster "Medical Instrumentation, Aplication and Design", Editorial Houshton Mifflin Corp.,Boston 1978.
- 18.-Robert J.Mayhon, "Discrete Time and Continuos Time Linear Systems" Editorial Adisson Wesley, 1984.

- 19.-Willis J. Tompkins & Webster "Design of Microcomputer - Based Medical Instrumentation" Editorial Prentice Hall, 1981.
- 20.-L. Edenbrandt, et al. "Reconstruction of Electrocardiogram Durin Heart Surgery", Computers and Biomedical Research, Vol. 25, No 6, pp538-546 December 1992.
- 21.-Wendell C. Ocasio et al. "Bpshape WK4: A Computer Program That Implements a Physiological Model for Analyzing the Shape of Blood Pressure Wavwforms" Computer Methods and Programs in Biomedicine Vol. 39, pp169 - 194, April 1993.
- 22.-Tovar O. et. al. "Monitor simulator de ECG para enseñanza", Rev. Mex Ing. Biomédica, pp. 79 - 85, Vol. 9, 1990.
- 23.-Schuartz Misha "Information, Transmission, Modulation and Noise",MC Graw Hill, 1990.
- 24.-Gerardo A. Paulin, Edgar C.M. "Comunicación por medio de las líneas de alta tensión", Rev. Ing. ENEP Aragón UNAM, pp. 19 - 24, Feb. 1986.
- 25.-Nestor Gonzales S. " Comunicaciones y Redes de Procesamiento de Datos" edit. Mc Graw Hill, 1990.
- 26.-Kruglinski David, "Guía de las comunicaciones del IBM PC", Mc Graw Hill, 1985.
- 27.-Marquett Electronics. " Course of training of Marquette Unity Network"
- 28.-Herbert Schildt, "Turbo C/C++, The Complete Reference", Mc Graw Hill, 1992.
- 29.-Pathfinder Associates Inc. "vlib Manual", 1991.

APENDICE A

LISTADO DE PROGRAMACION DEL MBM

* ASIGNACION DE CONSTANTES

ORG \$F000

OPTIONS	EQU \$1039	ACTIVO C A/D
SCCR1	EQU \$102C	CONTROLA LONGITUD DE DATOS
BAUD	EQU \$102B	CONTROLA BAUDAJE
SCCR2	EQU \$102D	CONTROLA TX Y RX
SCSR	EQU \$102E	REGISTRO DE STATUS
SCDR	EQU \$102F	REGISTRO DE DATOS
ADCTL	EQU \$1030	CONTROL DEL C A/D
ADR1	EQU \$1031	REGISTROS DE
ADR2	EQU \$1032	DATOS DEL
ADR3	EQU \$1033	CONVERTIDOR
ADR4	EQU \$1034	ANALOGICO-DIGITAL
PORTA	EQU \$1000	PUERTO A
CONFC	EQU \$0C10	CONTADOR DE F.C.
CONFR	EQU \$0C12	CONTADOR DE F.R.
CONTA	EQU \$0C14	
CONTB	EQU \$0C15	
CONTECG	EQU \$0C16	CONTADOR DE ECG
DTEMP	EQU \$0C18	DATO DE TEMPERATURA
DFC	EQU \$0C1A	DATO DE F.C
DFR	EQU \$0C1C	DATO DE F.R
DCS	EQU \$0C1E	DATO DE C.S
DPSS	EQU \$0C20	DATO DE P.S.SISTOLICA
DPSD	EQU \$0C22	DATO DE P.S.DIASTOLICA

* INICIALIZACION

LDS #\$0FFE	inicializo el stack
LDAB #\$90	activo potencia para el
STAB OPTIONS	C A/D
LDAB #\$00	fijo longitud de palabra
STAB SCCR1	en 8 bits
LDAB #\$30	fijo baudaje en 9600 bauds
STAB BAUD	
LDAB #\$0C	habilito Tx y Rx
STAB SCCR2	

LDY # \$0000	inicio cont. de F.C
STY CONTFC	guardo cont. en dir. 0
LDX # \$0000	Inicio cont. de F.R
STX CONTFR	guardo cont. en dir. 2
LDAA # \$00	inicio contadores de
STAA CONTA	F.C y F.R
LDAB # \$00	
STAB CONTB	
LDX # \$0800	inicio cont. de
STX CONTECG	muestras de ECG
LDAB # \$B4	inicio los datos
STAB DPSS	simulados de presión
LDAB # \$3C	sistolica y diastolica
STAB DPSD	

*** PROGRAMA PRINCIPAL**

*** PREGUNTA POR CLAVE (CHK)**

CHK: LDAB SCSR	pregunta si llego algo
ANDB # \$20	
CMPB # \$20	
BNE CNCHK	
LDAB SCDR	si llego comparalo con CHK
CMPB # \$1F	
BNE VAR	si no es chk ve a VAR
JSR TMGETC	si es chk toma muestra
LDX CONTECG	
LDAB \$00,X	
STAB SCDR TX	muestra
JSR TXC	
JSR F.C	calcula F.C
JSR F.R	calcula F.R
LDX CONTECG	cargo cont. de ECG
INX	incrementa dir. de datos
CPX # \$0C00	si no llego al final
BNE SAL	ve a retardo
LDX # \$0800	si llego al final
SAL: JMP RET	inicia contador de muestras
CNCHK: JMP NCHK	

*** SECCION DE ENVIO DE VARIABLES**

VAR: CMPB # \$26	envio variables?
BNE BEND	no, ve a endtx

LDAB DTEMP	si, envio todas las
STAB SCDR	variables
JSR TXC	
JSR RX	
FREC: LDAB SCDR	
CMPB #27	
BNE FREC	
LDAA \$OC1A	
CMPA #300	
BNE N.F	
LDAB \$OC1B	frecuencia cardiaca
STAB SCDR	
JSR TXC	
JSR RX	
FRER: LDAB SCDR	
CMPB #28	
BNE FRER	
LDAA \$OC1C	
CMPA #300	
BNE N.F.R	
LDAB \$OC1D	frecuencia respiratoria
STAB SCDR	
JSR TXC	
JSR RX	
CASU: LDAB SCDR	
CMPB #29	
BNE CASU	
LDAB DCS	cable suelto
STAB SCDR	
JSR TXC	
JSR RX	
PRES: LDAB SCDR	
CMPB #30	
BNE PRES	
LDAB DPSS	presion sanguinea sist.
STAB SCDR	
JSR TXC	
JSR RX	
PRED: LDAB SCDR	
CMPB #31	
BNE PRED	
LDAB DPSD	presion sanguinea dist.
STAB SCDR	
JSR TXC	
JMP CHK	ve a CHK
BEND JMP ENDTX	
N.F: LDAA #300	
STAA SCDR	
JSR TXC	
JMP FRER	

N.F.R LDAA #00
STAA SCDR
JSR TXC
JMP CASU

* SECCION DE FIN DE TRANSMISION

ENDTX: CMPB #04 si no es fin de tx
 BNE NCHK ve a nchk
 JMP CHK

* SECCION DE NO ENVIO DE MUESTRA

NCHK: JSR TMGETC ve a tomar una muestra
 JSR F.C calcula F.C
 JSR F.R calcula F.R
 LDX CONTECG
 INX si dir. de los datos no
 CPX #0C00 llego al final
 BNE RNCHK ve a retardo
 LDX #0800 si llego al final
 JMP RNCHK inicia cont. de muestras

* RETARDOS

RET: STX CONTECG genera un retardo
 LDX #0262 y va a preguntar
OCHO: DEX por chk
 BNE OCHO
 JMP CHK

RNCHK: STX CONTECG genera un retardo
 LDX #03CE y va a preguntar
SIETE: DEX por chk
 BNE SIETE
 JMP CHK

* SUBROUTINAS

*RECEPCION

RX: LDAB SCSR pregunta si llego dato
 ANDB #20
 CMPB #20 por el puerto serie
 BNE RX
 RTS

* TRANSMISION

TXC: LDAB SCSR envia un dato por el
 ANDB #80
 CMPB #80 puerto serie
 BNE TXC
 RTS

*TOMA MUESTRA Y GUARDA

TMGETC: LDAB #10 activo 4 canales de
 STAB ADCTL conversion
TC: LDAB ADCTL pregunto si termino
 ANDB #80 la conversion
 CMPB #80
 BNE TC
 LDX CONTECG guardo dato de ECG en la
 LDAB ADR1 direccion de CONTECG
 STAB \$00,X
 LDAB ADR2 guardo dato de temperatura
 STAB DTEMP
 LDAB PORTA leo el puerto a y guardo
 STAB DCS el dato de cable suelto
 RTS

* FRECUENCIA CARDIACA

F.C: LDAB ADR3 cargo el dato de F.C
 CMPB #80 si el dato es bajo
 BLS NOFCHI ve a nofchi
 LDAA CONTA si es alto cargo el C.A
 INCA para calcular o inic.

CMPA # \$01	el contador de F.C
BNE NOA	ve a noa
STAA CONTA	calcula la frecuencia
LDX CONTFC	cardiaca en pulsos por min
LDAB # \$07	"parte alta del pulso"
ABX	
LDD # \$4E20	
IDIV	
STX DFC	
JMP TER	
NOFCHI: LDY CONTFC	cargo el CONTFC, lo
INY	incrementa y la guarda
LDAA # \$00	
STY CONTFC	
STAA CONTA	
LDAB # \$0B	retardo
CUA: DECB	
BNE CUA	
JMP TER	
NOA: STAA CONTA	inicia contador de
LDY # \$0000	frecuencia cardiaca
STY CONTFC	
LDAB # \$0A	retardo
TRES: DECB	
BNE TRES	
TER: RTS	

 * FRECUENCIA RESPIRATORIA

F.R: LDAA ADR4	
CMPA # \$80	
BLS NOFRHI	
LDAB CONTB	
INCB	
CMPB # \$01	
BNE NOB	
STAB CONTB	
LDX CONTFR	
LDAB # \$07	20 miliseg.
ABX	
LDD # \$4E20	
IDIV	
STX DFR	
JMP FIN	
NOFRHI: LDX CONTFR	
INX	
LDAB # \$00	
STX CONTFR	

STAB CONTB
LDAB #S0B
DOS: DECB
BNE DOS
JMP FIN
NOB: STAB CONTB
LDX #S0000
STX CONFR
LDAB #S0A
UNO: DECB
BNE UNO
FIN: RTS
END

□

APENDICE B

PROGRAMA DE INTERACCIÓN CON EL PERSONAL MÉDICO

```
#include <vlib.h>                /* Archivos de cabecera */
#include <vlibmenu.h>
#include <stdio.h>
#include <conio.h>
#include <vlibkeys.h>
#include <vlibform.h>
#include <string.h>
#include <vlibmous.h>
#include <stdlib.h>

#define size 14                  /* Declaración de constantes */
#define si 1
#define no 0

int com,men,i;                  /* Declaración de variables */
int esckeys[]={ ESC,K_F2,0};
char nom[35],dom[50];
int edad,peso,cama,pcama,fca,fcf,fra,frb,psa,psb;
int ta,tb;

/* Declaración de funciones */

void alta(void),baja(void),guarda(void),limpia(void),signos(void),global(void);
void param(void),guarpar(void),abreread(void),abrewrte(void),tolread(void);
void tolwrite(void),cara(void),imprime(void),pausa(void);

/* Estructura del menu principal */

V_MENU menu[]=
{
{"1.-DAR DE ALTA UN PACIENTE",NULL,1},
{"",NULL,-2},
{"2.-FIJAR INDICES DEL PACIENTE",NULL,3},
{"",NULL,-4},
{"3.-DAR DE BAJA UN PACIENTE",NULL,5},
{"",NULL,-8},
{"4.-MOSTRAR SIGNOS DE UN PACIENTE",NULL,7},
{"",NULL,-8},
{"5.-SUPERVISION GLOBAL DE PACIENTES",NULL,9},
{"",NULL,-10},
{"6.-IMPRESION DE SIGNOS DE UN PACIENTE",NULL,11},
{"",NULL,-12},
```

```

{"7.-SALIR",NULL,13},
(NULL)
};

```

/* Estructura de formato de altas */

```

struct v_form_def form[]=
{
  8,25,VF_INT,2,"CAMA No",&cama,
  10,25,VF_STRING,sizeof(nom),"NOMBRE",nom,
  12,25,VF_STRING,sizeof(dom),"DOMICILIO",dom,
  14,25,VF_INT,3,"EDAD",&edad,
  16,25,VF_INT,3,"PESO",&peso,
  0,0,VF_EOL
};
struct v_form_def form1[]=
{
  8,25,VF_INT,2,"CAMA No",&pcama,
  10,25,VF_INT,3,"FREC.CAR.ALTA",&fca,
  10,50,VF_INT,3,"FREC.CAR.BAJA",&fcb,
  12,25,VF_INT,3,"FREC.RESP.ALTA",&fra,
  12,50,VF_INT,3,"FREC.RESP.BAJA",&frb,
  14,25,VF_INT,3,"PRES.SANG.ALTA",&psa,
  14,50,VF_INT,3,"PRES.SANG.BAJA",&psb,
  16,25,VF_INT,3,"TEMP.ALTA",&ta,
  16,50,VF_INT,3,"TEMP.BAJA",&tb,
  0,0,VF_EOL
};

```

/* Estructura para guardar los datos en un archivo */

```

struct pac {
  int gcama;
  char gnom[35];
  char gdom[50];
  int gedad;
  int gpeso;
  }gpac_pac[size],spac[size];
struct tol {
  int gpcama;
  int gfca;
  int gfcb;
  int gfra;
  int gfrb;
  int gpsa;
  int gpsb;
  int gta;
  int gtb;
}

```

```

    }gtol_tol[size];
struct dat{
    int camapac;
    int impre;
    }gdat_dat;

```

```

/* Estructura para seleccionar numero de cama */

```

```

V_MULTISELECT list[]=

```

```

{
    1,"CAMA No: 1",
    0,"CAMA No: 2",
    0,"CAMA No: 3",
    0,"CAMA No: 4",
    0,"CAMA No: 5",
    0,"CAMA No: 6",
    0,"CAMA No: 7",
    0,"CAMA No: 8",
    0,"CAMA No: 9",
    0,"CAMA No:10",
    0,"CAMA No:11",
    0,"CAMA No:12",
    0,"CAMA No:13",
    0,"CAMA No:14",
    NULL,
};

```

```

V_MULTISELECT list1[]=

```

```

{
    1,"EPSON (DOT-MATRIX)",
    0,"HP LASERJET III",
    NULL,
};

```

```

/* programa principal */

```

```

main()

```

```

{
/*cara();*/
vinit();          /* inicializo vlib */

```

```

/* Fijo los colores para el menu, formas y cajas de dialogo */

```

```

vSetColor(0,
    V_LIGHT_GREEN|V_BLUE_BG,
    V_BLUE|V_GREEN_BG,
    V_BRIGHT_WHITE|V_BLUE_BG);
vSetColor(V_MENU_GRP,
    V_RED_BG|V_WHITE,

```

```

    V_GREEN_BG|V_BLUE,
    V_RED_BG|V_BRIGHT_WHITE);
vSetColor(V_FORM_GRP,
    V_BLUE_BG|V_WHITE,
    V_GREEN_BG|V_BLUE,
    V_BLUE_BG|V_BRIGHT_WHITE);
vSetColor(V_POPUP_GRP,
    V_RED_BG|V_WHITE,
    V_GREEN_BG|V_RED,
    V_RED_BG|V_BRIGHT_WHITE);
cara(); /* Despliega caratula */
pausa();
men=1;
while(men==1)
{
    clrscr();
    textbackground(BLUE);
    clrscr();
    vBorder(); /* dibuja borde */
    vWriteStr(2,30,V_BRIGHT,"OPCIONES DE TRABAJO"); /* escribe letrero */
    vFillAttr(2,28,2,50,V_REVERSE);
    vFillAttr(3,29,3,50,V_BLACK);
    vFillAttr(2,50,3,50,V_BLACK);
    com=vPopMenuT(7,20,menu,1,"SELECCIONE OPCION"); /* despliega menu */
    switch(com)
    {
        case 1:
            clrscr();
            alta(); /* alta de un paciente */
            men=1;
            break;
        case 3:
            clrscr();
            param(); /* fija parametros de alarmas */
            men=1;
            break;
        case 5:
            clrscr();
            baja(); /* baja de un paciente */
            men=1;
            break;
        case 7:
            clrscr();
            signos(); /* signos de un paciente */
            men=1;
            break;
        case 9:
            clrscr();
            global(); /* supervision global de pacientes */
            break;
    }
}

```

```

        case 11:
            clrscr();
            imprime();
            men=1;
            break;
        case 13: clrscr();
            men=0;
            break;
        default: return(FALSE);clrscr();
    }
    clrscr();
}
clrscr();

} /* fin del main */

```

/* Funcion para dar de alta un paciente */

```

void alta(void)
{
    int dat,mas;
    mas=1;
    while(mas==1)
    {
        vBorder();
        vWriteStr(2,30,V_BRIGHT,"ALTA DE UN PACIENTE");
        vFillAttr(2,28,2,50,V_REVERSE);
        vFillAttr(3,29,3,50,V_BLACK);
        vFillAttr(2,50,3,50,V_BLACK);
        dat=vForm(form,esckey,"ESC = menu ,F2 = salvar"); /* despliega forma */

        switch(dat)
        {
            case 0: mas=0;
                break;
            case 1: clrscr();
                guarda();
                vClear();
                vFormClear(form); /* limpia forma */
                mas=1;
                break;
            default: return(FALSE);
        }
    }
    vFormClear(form);
}

```

```
/* Funcion para almacenar los datos en un archivo */
```

```
void guarda (void)
```

```
{  
    abread();  
  
    gpac_pac[cama-1].gcama=cama;  
    strcpy(gpac_pac[cama-1].gnom,nom);  
    strcpy(gpac_pac[cama-1].gdom,dom);  
    gpac_pac[cama-1].gedad=edad;  
    gpac_pac[cama-1].gpeso=peso;  
  
    abwrite();  
}
```

```
void abread(void)
```

```
{  
    int i;  
    FILE *fp;  
    if((fp=fopen("internos","r"))==NULL) /* abre un archivo */  
    {  
        printf("no puedo abrir el archivo");  
        return;  
    }  
    for(i=0;i<size;i++)  
    {  
        fread(&gpac_pac[i],sizeof(struct pac),1,fp); /* lee los datos */  
    }  
    fclose(fp); /* cierra el archivo */  
}
```

```
void abwrite(void)
```

```
{  
    int i;  
    FILE *fp;  
    if((fp=fopen("internos","w"))==NULL)  
    {  
        puts("no puedo abrir el archivo\n");  
        return;  
    }  
  
    for(i=0;i<size;i++)  
    {  
        fwrite(&gpac_pac[i],sizeof(struct pac),1,fp);  
    }  
    fclose(fp);  
}
```

/* Función para introducir niveles de alarma */

```
void param(void)
{
    int par,mparam;
    mparam=1;
    while(mparam==1)
    {
        vBorder();
        vWriteStr(2,30,V_BRIGHT,"PARAMETROS DE PACIENTE");
        vFillAttr(2,28,2,56,V_REVERSE);
        vFillAttr(3,29,3,56,V_BLACK);
        vFillAttr(2,58,3,56,V_BLACK);
        par=vForm(form1,esckey,"ESC = menu ,F2 = salvar"); /* despliega forma1 */
        switch(par)
        {
            case 0: mparam=0;
                    break;
            case 1: clrscr();
                    guarpar();
                    vClear();
                    vFormClear(form1); /* limpia forma */
                    mparam=1;
                    break;
            default: return(FALSE);
        }
    }
    vFormClear(form);
}

void guarpar(void)
{
    tolread();

    gtoi_tol[pcama-1].gpcama=pcama; /* guardo parametros */
    gtoi_tol[pcama-1].gfca=fca;
    gtoi_tol[pcama-1].gfcf=fcb;
    gtoi_tol[pcama-1].gfra=fra;
    gtoi_tol[pcama-1].gfrb=frb;
    gtoi_tol[pcama-1].gpsa=psa;
    gtoi_tol[pcama-1].gpsb=psb;
    gtoi_tol[pcama-1].gta=ta;
    gtoi_tol[pcama-1].gtb=tb;

    tolwrite();
}
```

```

void tolread(void)
{
    int i;
    FILE *fc;
    if((fc=fopen("tol_par","r"))==NULL) /* abre un archivo */
    {
        printf("no puedo abrir el archivo");
        return;
    }
    for(i=0;i<size;i++)
    {
        fread(&gtol_tol[i],sizeof(struct tol),1,fc); /* lee los datos */
    }
    fclose(fc); /* cierra el archivo */
}

```

```

void tolwrite(void)
{
    int i;
    FILE *fc;
    if((fc=fopen("tol_par","w"))==NULL)
    {
        puts("no puedo abrir el archivo\n");
        return;
    }

    for(i=0;i<size;i++)
    {
        fwrite(&gtol_tol[i],sizeof(struct tol),1,fc);
    }
    fclose(fc);
}

```

/* Funcion para dar de baja un paciente */

```

void baja (void)
{
    int cual,wb,i,seguro,noseg;
    noseg=1;
    while(noseg==1)
    {
        vBorder();
        vWriteStr(2,30,V_BRIGHT,"BAJA DE UN PACIENTE");
        vFillAttr(2,28,2,50,V_REVERSE);
        vFillAttr(3,29,3,50,V_BLACK);
        vFillAttr(2,50,3,50,V_BLACK);
        cual=vPopMultiselect(6,28,list,"Seleccione No. de cama",6); /* despliega caja de dialogo */
    }
}

```



```

if(cual==0){noseg=1;break;}
wb=vWindowAdd(4,10,16,70); /* crea una ventana */
vWindowGoto(wb);
vBorder2(); /* dibuja borde de la nueva ventana */
vFillAttr(1,1,11,59,V_LIGHT_GREY);

abreread();

vPut(2,4,V_REVERSE,"CAMA No:%d\n",gpac_pac[cual-1].gcama); /* despliega datos */
vPut(4,4,V_REVERSE,"NOMBRE:%s\n",gpac_pac[cual-1].gnom);
vPut(6,4,V_REVERSE,"DOMICILIO:%s\n",gpac_pac[cual-1].gdom);
vPut(8,4,V_REVERSE,"EDAD:%d\n",gpac_pac[cual-1].gedad);
vPut(10,4,V_REVERSE,"PESO:%d\n",gpac_pac[cual-1].gpseso);
seguro=vPopYesNo(18,24,"Esta usted seguro?",1);
switch(seguro)
{
    case 0: nose=0;
            vWindowDelete(wb);
            break;
    case 1: nose=1;
            vClearAll();
            vWindowDelete(wb);
            break;
    case 2: limpia();
            nose=0;
            vWindowDelete(wb);
            break;
}
}

/* Funcion que borra del archivo al paciente */

void limpia(void)
{
int cual,i;
char bnom[2],bdom[2];
gpac_pac[cual-1].gcama=0;
strcpy(gpac_pac[cual-1].gnom,bnom);
strcpy(gpac_pac[cual-1].gdom,bdom);
gpac_pac[cual-1].gedad=0;
gpac_pac[cual-1].gpseso=0;

abrewrite();
tolread();
gtol_tol[cual-1].gpcama=0; /* guardo parametros */
gtol_tol[cual-1].gfca=0;
gtol_tol[cual-1].gfcb=0;
gtol_tol[cual-1].gfra=0;
gtol_tol[cual-1].gfrb=0;
gtol_tol[cual-1].gpsa=0;

```

```

        gtol_tol[cual-1].gpsb=0;
        gtol_tol[cual-1].gta=0;
        gtol_tol[cual-1].gtb=0;
    tolwrite();
}

/* Funcion que muestra los signos de un paciente */

void signos(void)
{
    int sistem;
    sistem=system("mbmrtc.exe"); /* sale al sistema operativo y ejecuta */
    if(sistem===-1){ printf("error");}
}

```

```

/* Funcion que muestra los signos de los pacientes */

void global(void)
{
    int sistem1;
    sistem1=system("sg.exe");
    if(sistem1===-1){printf("error");}
}

```

```

/* Función que despliega caratula */

void cara(void)
{
    int sistem2;
    sistem2=system("porta.exe");
    if(sistem2===-1){printf("error");}
}

```

```

/* Función que imprime los datos del paciente */

void imprime(void)
{
    int cualimp,cualcama,i,sistem3;
    FILE *fd;
    vBorder();
    vWriteStr(2,30,V_BRIGHT,"IMPRESION DE SIGNOS");
    vFillAttr(2,28,2,50,V_REVERSE);
    vFillAttr(3,29,3,50,V_BLACK);
    vFillAttr(2,50,3,50,V_BLACK);
    cualimp=vPopMultiselect(10,25,list1,"Seleccione tipo de impresora",2);
    cualcama=vPopMultiselect(10,25,list,"Seleccione cama a imprimir",6);
}

```

```

gdat_dat.impre=cualimp;
gdat_dat.camapac=cualcama;
if((fd=fopen("gdat_dat","w"))==NULL)
{
puts("no puedo abrir el archivo\n");
return;
}

fwrite(&gdat_dat,sizeof(struct tol),1,fd);
fclose(fd);

```

```

sistem3=system("impre.dat.exe");
if(sistem3==-1){printf("error");}
}
void pausa(void)

```

```

{
int c;
c=getch();
if(0==c)
{
c=getch();
}
}

```

□

Programa para graficación de signos vitales

```

#include <stdio.h>           /* Archivos de cabecera */
#include <dos.h>
#include <graphics.h>
#include <conio.h>
#include "c:\entc\grafpcx.c"
#include "c:\entc\gprintf.c"

#define size 14             /* Declaración de constantes */
#define LONG 800
#define INI_T 0x10
#define EOT 0x04
#define CLAVE1 0x01
#define CLAVE2 0x02
#define CLAVE3 0x03
#define CLAVE4 0x14
#define CLAVE5 0x05
#define CLAVE6 0x06
#define CLAVE7 0x07
#define CLAVE8 0x08

```

```
#define CLAVE9 0x09
#define CLAVE10 0x0a
#define CLAVE11 0x0b
#define CLAVE12 0x0c
#define CLAVE13 0x0d
#define CLAVE14 0x0e
```

```
float tem,btem;          /* Declaración de variables */
```

```
unsigned short int frecc,freccs,freccr,press,pressd,cs,CLAVE,CHK;
unsigned short int bfrecc,bfreccr,bpress,bpressd;
```

```
int graphdr,graphmo,i,nucam,ca,clavec;
int xb1,xb2,x1a,x2a,x3a,x4a,x5a,x6a,x7a,xb;
int yb1,yb2,yb3,yb4,yb5,yb6,yb7,y1a,y2a,y3a,
    y4a,y5a,y6a,y7a,yb;
```

```
unsigned short int ecgp[LONG],ecgb[LONG],ecgp1[LONG],ecgb1[LONG];
unsigned short int ecgp2[LONG],ecgb2[LONG],ecgp3[LONG],ecgb3[LONG];
unsigned short int ecgp4[LONG],ecgb4[LONG],ecgp5[LONG],ecgb5[LONG];
unsigned short int ecgp6[LONG],ecgb6[LONG],ecgp7[LONG],ecgb7[LONG];
unsigned short int ecgp8[LONG],ecgb8[LONG],ecgp9[LONG],ecgb9[LONG];
unsigned short int ecgp10[LONG],ecgb10[LONG],ecgp11[LONG],ecgb11[LONG];
unsigned short int ecgp12[LONG],ecgb12[LONG],ecgp13[LONG],ecgb13[LONG];
unsigned short int ecgp14[LONG],ecgb14[LONG];
```

```
void abrear(void);
void inicio(void);
void proto(void);
void mues();
void param(void),mparam(void),alarma(void);
void termina(void);
void pausa(void);
void indivi(void),protoin(void),paramin(void);
```

```
FILE *fp;
FILE *fc;
```

```
/* Declaración de estructuras */
```

```
struct pac{
    int gcama;
    char gnom[35];
    char gdom[50];
    int gedad;
    int gpeso;
    }gpac_pac[size];
```

```
struct tol{
    int gpcama;
    int gfca;
```

```

int gfc;
int gfr;
int gfb;
int gpsa; /* presion sistolica */
int gpsb; /* presion diastolica */
int gta;
int gtb;
}gtol_tol[size];

/* Programa principal */

main()
{
    abrear(); /* abre archivo de pacientes y parametros */

    graphdr=DETECT; /* inicializa modo grafico*/
    initgraph(&graphdr,&graphmo,"c:\\tc");

    pcx("sg.pcx"); /* dibujo de fondo */

    inicio(); /* inicializa puerto serie */
    xb1=12;xb2=333;
    yb1=102;yb2=159;yb3=216;yb4=273;yb5=330;yb6=387;yb7=445;
    while(!kbhit()) /*mientras no se pida cama pide muestra y grafica*/
    {
        for(i=0;i<LONG;i++)
        {
            if(gpac_pac[0].gcama==1)
            {
                outp(0x3f8,0x1f); /* envia chk */
                check_rd_full(); /* checa si el registro de llegada */
                ecgp1[i]=inp(0x3f8); /* tiene algo y lee el valor */
                putpixel(xb1+i/4,yb1-(ecgp1[i]/6),BLACK);
                putpixel(xb1+i/4,yb1-(ecgp1[i]/6),GREEN);
                ecgb1[i]=ecgp1[i];
            }
            if(gpac_pac[1].gcama==2)
            {
                outp(0x3f8,0x2f);
                check_rd_full();
                ecgp2[i]=inp(0x3f8);
                putpixel(xb2+i/4,yb1-(ecgp2[i]/6),BLACK);
                putpixel(xb2+i/4,yb1-(ecgp2[i]/6),GREEN);
                ecgb2[i]=ecgp2[i];
            }
            if(gpac_pac[2].gcama==3)
            {
                outp(0x3f8,0x3f);
                check_rd_full();
                ecgp3[i]=inp(0x3f8);
                putpixel(xb1+i/4,yb2-(ecgp3[i]/6),BLACK);
                putpixel(xb1+i/4,yb2-(ecgp3[i]/6),GREEN);
            }
        }
    }
}

```

```

    ecgb3[i]=ecgp3[i];
}
if(gpac_pac[3].gcama==4)
{
    outp(0x3f8,0x4f);
    check_rd_full();
    ecgp4[i]=inp(0x3f8);
    putpixel(xb2+i/4,yb2-(ecgb4[i]/6),BLACK);
    putpixel(xb2+i/4,yb2-(ecgp4[i]/6),GREEN);
    ecgb4[i]=ecgp4[i];
}
if(gpac_pac[4].gcama==5)
{
    outp(0x3f8,0x5f);
    check_rd_full();
    ecgp5[i]=inp(0x3f8);
    putpixel(xb1+i/4,yb3-(ecgb5[i]/6),BLACK);
    putpixel(xb1+i/4,yb3-(ecgp5[i]/6),GREEN);
    ecgb5[i]=ecgp5[i];
}
if(gpac_pac[5].gcama==6)
{
    outp(0x3f8,0x6f);
    check_rd_full();
    ecgp6[i]=inp(0x3f8);
    putpixel(xb2+i/4,yb3-(ecgb6[i]/6),BLACK);
    putpixel(xb2+i/4,yb3-(ecgp6[i]/6),GREEN);
    ecgb6[i]=ecgp6[i];
}
if(gpac_pac[6].gcama==7)
{
    outp(0x3f8,0x7f);
    check_rd_full();
    ecgp7[i]=inp(0x3f8);
    putpixel(xb1+i/4,yb4-(ecgb7[i]/6),BLACK);
    putpixel(xb1+i/4,yb4-(ecgp7[i]/6),GREEN);
    ecgb7[i]=ecgp7[i];
}
if(gpac_pac[7].gcama==8)
{
    outp(0x3f8,0x8f);
    check_rd_full();
    ecgp8[i]=inp(0x3f8);
    putpixel(xb2+i/4,yb4-(ecgb8[i]/6),BLACK);
    putpixel(xb2+i/4,yb4-(ecgp8[i]/6),GREEN);
    ecgb8[i]=ecgp8[i];
}
if(gpac_pac[8].gcama==9)
{
    outp(0x3f8,0x9f);

```

```

    check_rd_full();
    ecgp9[i]=inp(0x3f8);
    putpixel(xb1+i/4,yb5-(ecgb9[i]/6),BLACK);
    putpixel(xb1+i/4,yb5-(ecgp9[i]/6),GREEN);
    ecgb9[i]=ecgp9[i];
}

if(gpac_pac[9].gcama==10)
{
    outp(0x3f8,0xaf);
    check_rd_full();
    ecgp10[i]=inp(0x3f8);
    putpixel(xb2+i/4,yb5-(ecgb10[i]/6),BLACK);
    putpixel(xb2+i/4,yb5-(ecgp10[i]/6),GREEN);
    ecgb10[i]=ecgp10[i];
}

if(gpac_pac[10].gcama==11)
{
    outp(0x3f8,0xbf);
    check_rd_full();
    ecgp11[i]=inp(0x3f8);
    putpixel(xb1+i/4,yb6-(ecgb11[i]/6),BLACK);
    putpixel(xb1+i/4,yb6-(ecgp11[i]/6),GREEN);
    ecgb11[i]=ecgp11[i];
}

if(gpac_pac[11].gcama==12)
{
    outp(0x3f8,0xcf);
    check_rd_full();
    ecgp12[i]=inp(0x3f8);
    putpixel(xb2+i/4,yb6-(ecgb12[i]/6),BLACK);
    putpixel(xb2+i/4,yb6-(ecgp12[i]/6),GREEN);
    ecgb12[i]=ecgp12[i];
}

if(gpac_pac[12].gcama==13)
{
    outp(0x3f8,0xdf);
    check_rd_full();
    ecgp13[i]=inp(0x3f8);
    putpixel(xb1+i/4,yb7-(ecgb13[i]/6),BLACK);
    putpixel(xb1+i/4,yb7-(ecgp13[i]/6),GREEN);
    ecgb13[i]=ecgp13[i];
}

if(gpac_pac[13].gcama==14)
{
    outp(0x3f8,0xef);
    check_rd_full();
    ecgp14[i]=inp(0x3f8);

```

```

putpixel(xb2+i/4,yb7-(ecgb14[i]/8),BLACK);
putpixel(xb2+i/4,yb7-(ecgp14[i]/8),GREEN);
ecgb14[i]=ecgp14[i];
}

}
param(); /* activa alarmas */
}

termina(); /* envia el comando de fin de transmision */

pausa();
scanf("%d",&ca); /* lee el numero de cama y lo imprime */
setcolor(BLACK);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
gprintf(338,462,"%d",ca);

switch(ca) /* envia la clave de la cama deseada */
{
case 1: CLAVE=CLAVE1;
clavec=1;
CHK=0x1f;
indivi(); /* grafica un paciente */
break;
case 2: CLAVE=CLAVE2;
clavec=2;
CHK=0x2f;
indivi();
break;
case 3: CLAVE=CLAVE3;
clavec=3;
CHK=0x3f;
indivi();
break;
case 4: CLAVE=CLAVE4;
clavec=4;
CHK=0x4f;
indivi();
break;
case 5: CLAVE=CLAVE5;
clavec=5;
CHK=0x5f;
indivi();
break;
case 6: CLAVE=CLAVE6;
clavec=6;
CHK=0x6f;
indivi();
break;
}

```



```
case 7: CLAVE=CLAVE7;
        clavec=7;
        CHK=0x7f;
        indivi();
        break;
case 8: CLAVE=CLAVE8;
        clavec=8;
        CHK=0x8f;
        indivi();
        break;
case 9: CLAVE=CLAVE9;
        clavec=9;
        CHK=0x9f;
        indivi();
        break;
case 10: CLAVE=CLAVE10;
        clavec=10;
        CHK=0xaf;
        indivi();
        break;
case 11: CLAVE=CLAVE11;
        clavec=11;
        CHK=0xbf;
        indivi();
        break;
case 12: CLAVE=CLAVE12;
        clavec=12;
        CHK=0xcf;
        indivi();
        break;
case 13: CLAVE=CLAVE13;
        clavec=13;
        CHK=0xdf;
        indivi();
        break;
case 14: CLAVE=CLAVE14;
        clavec=14;
        CHK=0xef;
        indivi();
        break;
default : break;
}
```

```
closegraph(); /* cierra el modo grafico */
```

```
} /* fin del main */
```

/*----- Funcion que abre el archivo de datos de pacientes ----- */

```
void abrear(void)  
{  
    if((fp=fopen("internos","r"))==NULL)  
    {  
        printf("no puedo abrir el archivo");  
        return;  
    }  
    for(i=0;i<size;i++)  
    {  
        fread(&gpac_pac[i],sizeof(struct pac),1,fp);  
    }  
    fclose(fp);  
    if((fc=fopen("tol_par","r"))==NULL)  
    {  
        printf("no puedo abrir el archivo");  
        return;  
    }  
    for(j=0;j<size;j++)  
    {  
        fread(&gtol_tol[j],sizeof(struct tol),1,fc);  
    }  
    fclose(fc);  
}
```

/* ----- funcion para inicializar el puerto serie ----- */

```
void inicio(void)  
{  
    outp(0x3fb,0x80); /*direcciona line_control para baudaje*/  
    outp(0x3f8,0x0c); /*LSB para 9600 Bauds*/  
    outp(0x3f9,0x00); /*MSB para 9600 Bauds*/  
    outp(0x3fb,0x03); /*inicializa el line_control en paridad non,  
        1 bit stop, 8 bits de datos*/  
    outp(0x3fc,0x0b); /*inicializa control de modem para interrupciones*/  
    outp(0x3f9,0x0f); /*habilita todas las clases de interrupciones*/  
}
```

/* ----- Funcion que pide parametros de todas las camas ----- */

```
void param(void)  
{  
  
    if(gpac_pac[0].gcama==1)
```

```

(
  mparam();
  nucam=0;
  x2a=205;y2a=80; x3a=245;y3a=80; x4a=284;y4a=80;
  x5a=205;y5a=95; x6a=245;y6a=95; x7a=284;y7a=95;
  alarma();
)
if(gpac_pac[1].gcama==2)
(
  mparam();
  nucam=1;
  x2a=525;y2a=80; x3a=565;y3a=80; x4a=605;y4a=80;
  x5a=525;y5a=95; x6a=565;y6a=95; x7a=605;y7a=95;
  alarma();
)
if(gpac_pac[2].gcama==3)
(
  mparam();
  nucam=2;
  x2a=205;y2a=135; x3a=245;y3a=135; x4a=284;y4a=135;
  x5a=205;y5a=155; x6a=245;y6a=155; x7a=284;y7a=155;
  alarma();
)
if(gpac_pac[3].gcama==4)
(
  mparam();
  nucam=3;
  x2a=525;y2a=135; x3a=565;y3a=135; x4a=605;y4a=135;
  x5a=525;y5a=155; x6a=565;y6a=155; x7a=605;y7a=155;
  alarma();
)
if(gpac_pac[4].gcama==5)
(
  mparam();
  nucam=4;
  x2a=205;y2a=195; x3a=245;y3a=195; x4a=284;y4a=195;
  x5a=205;y5a=210; x6a=245;y6a=210; x7a=284;y7a=210;
  alarma();
)
if(gpac_pac[5].gcama==6)
(
  mparam();
  nucam=5;
  x2a=525;y2a=195; x3a=565;y3a=195; x4a=605;y4a=195;
  x5a=525;y5a=210; x6a=565;y6a=210; x7a=605;y7a=210;
  alarma();
)
if(gpac_pac[6].gcama==7)

```

```

{
    mparam();
    nucam=6;
    x2a=205;y2a=250; x3a=245;y3a=250; x4a=284;y4a=250;
    x5a=205;y5a=288; x6a=245;y6a=288; x7a=284;y7a=288;
    alarma();
}
if(gpac_pac[7].gcama==8)
{
    mparam();
    nucam=7;
    x2a=525;y2a=250; x3a=565;y3a=250; x4a=605;y4a=250;
    x5a=525;y5a=288; x6a=565;y6a=288; x7a=605;y7a=288;
    alarma();
}
if(gpac_pac[8].gcama==9)
{
    mparam();
    nucam=8;
    x2a=205;y2a=305; x3a=245;y3a=305; x4a=284;y4a=305;
    x5a=205;y5a=325; x6a=245;y6a=325; x7a=284;y7a=325;
    alarma();
}
if(gpac_pac[9].gcama==10)
{
    mparam();
    nucam=9;
    x2a=525;y2a=305; x3a=565;y3a=305; x4a=605;y4a=305;
    x5a=525;y5a=325; x6a=565;y6a=325; x7a=605;y7a=325;
    alarma();
}
if(gpac_pac[10].gcama==11)
{
    mparam();
    nucam=10;
    x2a=205;y2a=365; x3a=245;y3a=365; x4a=284;y4a=365;
    x5a=205;y5a=380; x6a=245;y6a=380; x7a=284;y7a=380;
    alarma();
}
if(gpac_pac[11].gcama==12)
{
    mparam();
    nucam=11;
    x2a=525;y2a=365; x3a=565;y3a=365; x4a=605;y4a=365;
    x5a=525;y5a=380; x6a=565;y6a=380; x7a=605;y7a=380;
    alarma();
}

```

```

if(gpac_pac[12].gcama==13)
{
    mparam();
    nucam=12;
    x2a=205;y2a=420; x3a=245;y3a=420; x4a=284;y4a=420;
    x5a=205;y5a=440; x6a=245;y6a=440; x7a=284;y7a=440;
    alarma();
}
if(gpac_pac[13].gcama==14)
{
    mparam();
    nucam=13;
    x2a=525;y2a=420; x3a=565;y3a=420; x4a=605;y4a=420;
    x5a=525;y5a=440; x6a=565;y6a=440; x7a=605;y7a=440;
    alarma();
}
}

```

```

void mparam(void)
{
    unsigned short int tempe, frecc1;
    outp(0x3f8, 0x26);
    check_rd_full();
    tempe=inp(0x3f8);
    tem=(tempe*0.195312);
    outp(0x3f8, 0x27);
    check_rd_full();
    frecc1=inp(0x3f8);
    frecc=frecc1+frecc1/10;
    outp(0x3f8, 0x28);
    check_rd_full();
    frecr=inp(0x3f8);
    outp(0x3f8, 0x29);
    check_rd_full();
    cs=inp(0x3f8);

    outp(0x3f8, 0x30);
    check_rd_full();
    press=inp(0x3f8);

    outp(0x3f8, 0x31);
    check_rd_full();
    presd=inp(0x3f8);
}

```

```
/* --- funcion que envia un EOT para indicar que termino la adquisicion ---*/
```

```
void termina(void)
{
  for(i=0;i<10;i++)

  {
    outp(0x3f8,EOT);
  }
}
```

```
/* ----- funcion que despliega un solo paciente ----- */
```

```
void indivi(void)
{
  int i;
  int x2a,x3a,x5a,x6a,x7a,y2a,y3a,y5a,y6a,y7a;
  pcx("individ.pcx");
  setcolor(LIGHTGRAY);
  settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
  gprintf(110,375,"%d",ca);
  gprintf(110,420,"%s",gpac_pac[clavec-1].gnom);
  xb=20;
  yb=210;
  ecgb[0]=0;ecgp[0]=0;
  while(!kbhit())
  {
    for(i=1;i<600;i++)
    {
      outp(0x3f8,CHK);
      check_rd_full();
      ecgp[i]=inp(0x3f8)/2;
      if(ecgp[i]>0xc8){
        ecgp[i]=0xc8;}
      setlinestyle(SOLID_LINE,0,NORM_WIDTH);
      setcolor(BLACK);
      line(xb+i,yb-ecgb[i],xb+i-1,yb-ecgb[i-1]);
      setcolor(GREEN);
      line(xb+i,yb-ecgp[i],xb+i-1,yb-ecgp[i-1]);
      delay(4);
    }
    paramin();
    for(i=1;i<LONG;i++){
      ecgb[i]=ecgp[i]; }

  }
  termina();
  pausa();
}
```

```
/* ----- Funcion que pide parametros ----- */
```

```
void paramin(void)
```

```
{  
  unsigned short int tempe;  
  outp(0x3f8,0x26);  
  check_rd_full();  
  tempe=inp(0x3f8);  
  tem=(tempe*0.195312);  
  outp(0x3f8,0x27);  
  check_rd_full();  
  freccs=inp(0x3f8);  
  frecc=freccs+freccs/10;  
  outp(0x3f8,0x28);  
  check_rd_full();  
  frecr=inp(0x3f8);  
  outp(0x3f8,0x29);  
  check_rd_full();  
  cs=inp(0x3f8);  
  outp(0x3f8,0x30);  
  check_rd_full();  
  press=inp(0x3f8);  
  outp(0x3f8,0x31);  
  check_rd_full();  
  presd=inp(0x3f8);  
  setcolor(LIGHTGRAY);  
  settextstyle(DEFAULT_FONT,HORIZ_DIR,2);  
  gprintf(268,308,"%d",bfrecr);  
  gprintf(268,252,"%d",bfrecc);  
  gprintf(117,252,"%d",bpress);  
  gprintf(117,308,"%d",bpresd);  
  gprintf(415,252,"%-2.1f",btem);  
  setcolor(BLACK);  
  settextstyle(DEFAULT_FONT,HORIZ_DIR,2);  
  gprintf(268,308,"%d",frecr);  
  gprintf(268,252,"%d",frecc);  
  gprintf(117,252,"%d",press);  
  gprintf(117,308,"%d",presd);  
  gprintf(415,252,"%-2.1f",tem);  
  nucam=clavec-1;  
  bfrecc=frecc;bpress=press;btem=tem;  
  bfrecr=frecr,bpresd=presd;  
  x2a=600;y2a=325;x3a=600;y3a=300;x5a=600;y5a=380;  
  x6a=600;y6a=355;x7a=600;y7a=410;  
  alarma();  
}
```

```
/* ---- funcion que detecta si el registro de llegada tiene algo ---- */
```

```
check_rd_full()
{
  for(;;)
  {
    if( inp(0x3fd) & 0x01 == 0x01) return;
    if(kbhit()) return;
  }
}
```

```
/* ----- funcion que iniciliza el !kbhit() ----- */
```

```
void pausa(void)
{
  int c;
  c=getch();
  if(0==c)
  {
    c=getch();
  }
}
```

```
/* ----- funcion que detecta datos anormales y enciende alarmas ----- */
```

```
void alarma(void)
{
  if(frecc>gtol_tol[nucam].gfca || frecc<gtol_tol[nucam].gfc)
  {
    setfillstyle(SOLID_FILL,RED);
    floodfill(x6a,y6a,BLACK);
    sound(300);delay(100);nosound();
  }
  else setfillstyle(SOLID_FILL,LIGHTGRAY);
    floodfill(x6a,y6a,BLACK);
  if(freccr>gtol_tol[nucam].gfra || freccr<gtol_tol[nucam].gfr)
  {
    setfillstyle(SOLID_FILL,RED);
    floodfill(x7a,y7a,BLACK);
    sound(300);delay(100);nosound();
  }
  else setfillstyle(SOLID_FILL,LIGHTGRAY);
    floodfill(x7a,y7a,BLACK);
  if(press>gtol_tol[nucam].gpsa)
  {
    setfillstyle(SOLID_FILL,RED);
    floodfill(x2a,y2a,BLACK);
  }
}
```



```

    sound(300);delay(100);nosound();
  }
  else setfillstyle(SOLID_FILL,LIGHTGRAY);
    floodfill(x2a,y2a,BLACK);
  if(presd<gtol_tol[nucam] gpsb)

  {
    setfillstyle(SOLID_FILL,RED);
    floodfill(x3a,y3a,BLACK);
    sound(300);delay(100);nosound();
  }

  else setfillstyle(SOLID_FILL,LIGHTGRAY);
    floodfill(x3a,y3a,BLACK);
  if(tem>gtol_tol[nucam].gta || tem<gtol_tol[nucam].gtb)
  { setfillstyle(SOLID_FILL,RED);
    floodfill(x5a,y5a,BLACK);

    sound(300);delay(100);nosound();
  }
  else setfillstyle(SOLID_FILL,LIGHTGRAY);
    floodfill(x5a,y5a,BLACK);
  }
}

```

□

Programa para la impresión de signos vitales

/* Archivos de cabecera */

```

#include <stdio.h>
#include <graphics.h>
#include "c:\envtc\grafpcx.c"
#include "c:\envtc\gprintf.c"
#define size 14 /* Declaración de constantes */
#define ACKN 0x06
#define EOT 0x04
#define LONG 600
#define INI_T 0x10
#define CLAVE1 0x01
#define CLAVE2 0x02
#define CLAVE3 0x03
#define CLAVE4 0x14
#define CLAVE5 0x05
#define CLAVE6 0x16
#define CLAVE7 0x07
#define CLAVE8 0x08
#define CLAVE9 0x09
#define CLAVE10 0x0a
#define CLAVE11 0x0b
#define CLAVE12 0x0c
#define CLAVE13 0x0d
#define CLAVE14 0x0e

```

```

float tem,temper;          /* Declaración de variables */
int graphdr,graphmo,i,cama,tipoimp,xb,yb,fracc,frccs,frcr,press,presd,cs;
unsigned short int ACK,CLAVE,CHK,ack,clav;
unsigned short int ecgp[LONG],ecgb[LONG];

```

```

int format(double posicion)
{
    int width=6;
    if(posicion<1000.0) width--;
    if(posicion<100.0) width--;
    if(posicion<10.0) width--;
    return(width);
}

```

/*Declaración de funciones */

```

void mues(void),epson(void),laser(void);
void inicio(void);
void param(void);
void proto(void);

```

```

FILE*fd;
FILE*fp;
struct dat{
    int camapac;
    int impre;
    }gdat_dat;

```

/* Declaración de estructuras */

```

struct pac{
    int gcama;
    char gnom[35];
    char gdom[50];
    int gedad;
    int gpeso;
    }gpac_pac[size];

```

/* Programa principal */

```

main()
{
    graphdr=DETECT;
    initgraph(&graphdr,&graphmo,"c:\\tc");

    if((fd=fopen("gdat_dat","r"))==NULL)
    {
        puts("no puedo abrir el archivo\n");
        return;
    }

    fread(&gdat_dat,sizeof(struct dat),1,fd);
    fclose(fd);
    if((fp=fopen("internos","r"))==NULL)
    {

```

```

printf("no puedo abrir el archivo");
return;
}
for(i=0;i<size;i++)
{
fread(&gpac_pac[i],sizeof(struct pac),1,fp);
}
fclose(fp);

inicio();
cama=gdat_dat.camapac;
switch(cama)
{
case 0: break;
case 1: CLAVE=CLAVE1;CHK=0x1f;mues();
tipoimp=gdat_dat.impre;
switch(tipoimp)
{
case 0: break;
case 1: Epson();
break;
case 2: laser();
break;
}
break;
case 2: CLAVE=CLAVE2;CHK=0x2f;mues();
tipoimp=gdat_dat.impre;
switch(tipoimp)
{
case 0: break;
case 1: Epson();
break;
case 2: laser();
break;
}
break;
case 3: CLAVE=CLAVE3;CHK=0x3f;mues();
tipoimp=gdat_dat.impre;
switch(tipoimp)
{
case 0: break;
case 1: Epson();
break;
case 2: laser();
break;
}
break;
case 4: CLAVE=CLAVE4;CHK=0x4f;mues();
tipoimp=gdat_dat.impre;

```

```

switch(tipoimp)
(
    case 0: break;
    case 1: Epson();
            break;
    case 2: Laser();
            break;
)
break;
case 5: CLAVE=CLAVE5;CHK=0x5f;mues();
      tipoimp=gdat_dat.impre;
      switch(tipoimp)
      (
          case 0: break;
          case 1: Epson();
                  break;
          case 2: Laser();
                  break;
      )
break;
case 6: CLAVE=CLAVE6;CHK=0x6f;mues();
      tipoimp=gdat_dat.impre;
      switch(tipoimp)
      (
          case 0: break;
          case 1: Epson();
                  break;
          case 2: Laser();
                  break;
      )
break;
case 7: CLAVE=CLAVE7;CHK=0x7f;mues();
      tipoimp=gdat_dat.impre;
      switch(tipoimp)
      (
          case 0: break;
          case 1: Epson();
                  break;
          case 2: Laser();
                  break;
      )
break;
case 8: CLAVE=CLAVE8;CHK=0x8f;mues();
      tipoimp=gdat_dat.impre;
      switch(tipoimp)
      (
          case 0: break;
          case 1: Epson();
                  break;
          case 2: Laser();

```

```

        break;
    }
    break;
case 9: CLAVE=CLAVE9;CHK=0x9f;mues();
    tipoimp=gdat_dat.impre;
    switch(tipoimp)
    {
        case 0: break;
        case 1: Epson();
            break;
        case 2: laser();
            break;
    }
    break;
case 10: CLAVE=CLAVE10;CHK=0xaf;mues();
    tipoimp=gdat_dat.impre;
    switch(tipoimp)
    {
        case 0: break;
        case 1: Epson();
            break;
        case 2: laser();
            break;
    }
    break;
case 11: CLAVE=CLAVE11;CHK=0xbf;mues();
    tipoimp=gdat_dat.impre;
    switch(tipoimp)
    {
        case 0: break;
        case 1: Epson();
            break;
        case 2: laser();
            break;
    }
    break;
case 12: CLAVE=CLAVE12;CHK=0xcf;mues();
    tipoimp=gdat_dat.impre;
    switch(tipoimp)
    {
        case 0: break;
        case 1: Epson();
            break;
        case 2: laser();
            break;
    }
    break;
case 13: CLAVE=CLAVE13;CHK=0xdf;mues();
    tipoimp=gdat_dat.impre;
    switch(tipoimp)

```

```

        {
            case 0: break;
            case 1: Epson();
                    break;
            case 2: Laser();
                    break;
        }
    break;
case 14: CLAVE=CLAVE14;CHK=0xef;mues();
        tipoimp=gdat_dat.impre;
        switch(tipoimp)
        {
            case 0: break;
            case 1: Epson();
                    break;
            case 2: Laser();
                    break;
        }
    break;
}

closegraph();
}

void inicio(void)
{
/*funcion para inicializar el puerto serie*/
outp(0x3fb,0x80);/*direcciona line_control para baudaje*/
outp(0x3f8,0x0c);/*LSB para 9600 Bauds*/
outp(0x3f9,0x00);/*MSB para 9600 Bauds*/
outp(0x3fb,0x03);/*inicializa el line_control en paridad non,
                1 bit stop, 8 bits de datos*/
outp(0x3fc,0x0b);/*inicializa contol de modem para interrupciones*/
outp(0x3f9,0x0f);/*habilita todas las clases de interrupciones*/
}

/* Función que solicita signos vitales y los grafica */
void mues(void)
{
    int i,xb,yb;
    xb=21;yb=208;

    setcolor(WHITE);
    setlinestyle(SOLID_LINE,0,NORM_WIDTH);
    rectangle(0,0,639,479);
    rectangle(20,10,622,210);
    for(i=0;i<LONG;i++)

```

```

{
  outp(0x3f8,CHK);
  check_rd_full();
  ecgp[i]=inp(0x3f8)/2;
  if(ecgp[i]>0xc8)
  {
    ecgp[i]=0xc8;
  }
  setcolor(WHITE);
  line(xb+i,yb-ecgp[i],xb+i-1,yb-ecgp[i-1]);
}
param();
for(i=0;i<10;i++){
  outp(0x3f8,EOT); }
}

check_rd_full()
{
  /*funcion que detecta si el registro de llegada tiene algo*/
  for(;;){
    if( inp(0x3fd) & 0x01 == 0x01) return;
    if(kbhit()) return;
  }
}

void param(void)
{
  unsigned short int tempe;
  outp(0x3f8,0x26);
  check_rd_full();
  tempe=inp(0x3f8);
  tem=(tempe*0.195312);
  outp(0x3f8,0x27);
  check_rd_full();
  freccs=inp(0x3f8);
  frecc=freccs+freccs/10;
  outp(0x3f8,0x28);
  check_rd_full();
  frecr=inp(0x3f8);
  outp(0x3f8,0x29);
  check_rd_full();
  outp(0x3f8,0x30);
  check_rd_full();
  press=inp(0x3f8);
  outp(0x3f8,0x31);
  check_rd_full();
  presd=inp(0x3f8);

  setcolor(WHITE);
  settxtstyle(DEFAULT_FONT,HORIZ_DIR,1);
}

```

```

gprintf(260,308,"Frec. Resp. : %d",frecr);
gprintf(260,252,"Frec. Car. : %d",frecc);
gprintf(100,252,"Pres. Sist. : %d",press);
gprintf(100,308,"Pres. Diast. : %d",presd);
gprintf(420,252,"Temp. : %-2.1f",tem);
gprintf(100,375,"Cama No.: %d",gpac_pac[cama-1].gcama);
gprintf(100,420,"Nombre : %s",gpac_pac[cama-1].gnom);
}

```

/* Función que imprime signos vitales en una impresora de matriz de puntos */

```
void epson(void)
```

```

{
    char m;
    int i,j,k,Msb,LSb,
        MaxX=getmaxx(),
        MaxY=getmaxy();

    setviewport(0,0,MaxX,MaxY,0);

    fprintf(stdprn,"x1B%x",31);

    Lsb=MaxY & 0x00FF;
    Msb=MaxY>>8;
    for(j=0;j<=MaxX;j+=8)
    {
        fprintf(stdprn,"x1B*%c%c%c",0,LSb,MSb);
        for(i=MaxY;i>=0;i--)
        {
            m=0;
            for(k=0;k<8;k++)
            {
                m<<=1;
                if(getpixel(j+k,i))m++;
            }
            fprintf(stdprn,"%c",m);
        }
        fprintf(stdprn,"xODx0A");
    }
}

```

/* Función que imprime signos vitales en una impresora laser */

```
void laser(void)
```

```

{
    char m,resolucion[3];
    int i,j,k,p,q,xasp,yasp,
        MaxX=getmaxx()+1,
        MaxY=getmaxy()+1;

```



```

static char graph_end[]="\x1B*rB";
static char graph_ini[]="\x1B*E\x1B&11H\x1B&10\x1B*p0X\x1Bp0Y\x1B*t";
double xprint, yprint,prstep,AspR;

getaspectratio(&xasp,&yasp);
AspR=(double)xasp/(double)yasp;

setviewport(0,0,MaxX,MaxY,0);

xprint=690.0,
yprint=500.0,
strcpy(resolucion,"100");
prstep=7.2/AspR;

fprintf(stdprn,"%s%sR",graph_ini,resolucion);

for(j=0;j<=MaxY;j++)
{
    fprintf(stdprn,"\x1B&a%-.1fh%-.1fv",format(xprint),xprint,
        format(yprint),yprint);
    yprint +=prstep;
    fprintf(stdprn,"\x1B*r1A\x1B*b%dW",MaxX/8);

    for(i=0;i<MaxX/8;i++)
    {
        m=0;
        for(k=0;k<8;k++)
        {
            m<=1;
            if(getpixel(i*8+k,j))m++;
        }

        fprintf(stdprn,"%c",m);
    }

    fprintf(stdprn,"%s",graph_end);
}
fprintf(stdprn,"\x0C\x1B&10\x1B(8U\x1B(sp10h12v&b3T\x1B&11H");
}

```

□