

61  
2es.



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

FACULTAD DE INGENIERIA

SISTEMA DE ADQUISICION Y CONTROL DE  
PROPOSITO GENERAL

**T E S I S**

QUE PARA OBTENER EL TITULO DE:

**INGENIERO MECANICO ELECTRICISTA**  
( AREA ELECTRONICA )

P R E S E N T A :

**LUIS BERNARDO FIERRO CERVANTES**

DIRECTOR: M. EN I. ROBERTO QUAAS WEPEN.



MEXICO, D. F.

1998

TESIS CON  
FALLA DE ORIGEN

257362



## **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 está 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.

## DEDICATORIAS

*A mi madre con amor, Irma Cervantes, por ser el mayor apoyo y el mejor ejemplo que he tenido*

*A mis hermanas, Heana y Brenda, con cariño*

*A mi pequeñita, Jimena, por subirse al camión 165 Côte des Neiges*

## AGRADECIMIENTOS

Mi más profundo reconocimiento y gratitud a mi amigo y director de tesis, M en I. Roberto Quaas Weppen, jefe de la Coordinación de Instrumentación del Centro Nacional de Prevención de Desastres CENAPRED, por su excelente guía a lo largo del desarrollo de este proyecto, por sus consejos juiciosos y por la disponibilidad de la que ha hecho prueba.

Quisiera agradecer en especial a mi compañero y amigo Ing. Miguel Angel Franco Sánchez, investigador e ingeniero en instrumentación sísmica del CENAPRED, por sus comentarios y la gran ayuda que me brindó durante la realización del SAC.

Agradezco igualmente a todo el personal de la Coordinación de Instrumentación del CENAPRED.

Gracias a todos mis profesores y a la Universidad Nacional Autónoma de México por haber contribuido en mi formación profesional.

Finalmente, quisiera muy especialmente expresar mi más profunda gratitud a mi madre por su constante apoyo y aliento a lo largo de mis estudios.

# ÍNDICE

<b>I INTRODUCCIÓN</b> .....	1
<b>II DESCRIPCIÓN DEL SISTEMA DE ADQUISICIÓN Y CONTROL DE PROPÓSITO GENERAL SAC</b> .....	4
2.1 Características Generales .....	4
2.2 Sistema de control y procesamiento .....	6
Descripción general .....	6
Microcontrolador .....	8
Mapa de memoria del sistema y decodificación .....	14
Memorias .....	17
Puertos de entrada .....	17
Puertos de salida .....	20
Puertos de comunicación .....	20
Puerto especial .....	22
Puerto de expansión .....	22
Convertidor analógico-digital .....	22
Temporizador .....	23
Interrupciones .....	23
Reloj fechador de tiempo real .....	24
Circuito supervisor .....	25
Alimentación del sistema .....	25
Batería de respaldo .....	25
Distribución de componentes y lista de material .....	26
2.3 Módulo acondicionador de señales y puerto de detección .....	29
Puerto de detección .....	29
Acondicionador de señales .....	29
2.4 Módulo de despliegue y teclado .....	33
Módulo de despliegue .....	33
Teclado .....	33
2.5 Puertos de entrada y de salida .....	36
Puerto de entrada 1 .....	36
Puerto de entrada 2 .....	36
Puerto de salida 1 .....	38
Puerto de salida 2 con interfaz de potencia .....	38
2.6 Unidad de almacenamiento de datos UAD-PCMCIA .....	39
<b>III SISTEMA OPERATIVO DEL SAC</b> .....	43
3.1 Descripción .....	43
3.2 Rutina I de inicialización .....	45
3.3 Rutina II de recepción del programa .....	47
3.4 Rutina III de ejecución del programa recibido .....	49
3.5 Rutina IV de diagnóstico .....	51

<b>IV OPERACIÓN Y MANEJO DEL SAC</b> .....	58
4.1 Módulo de despliegue y teclado .....	58
Módulo de despliegue y <i>leds</i> indicadores .....	58
Teclado .....	59
4.2 Modos de operación .....	61
Modo de espera .....	61
Modo de recepción .....	61
Modo de ejecución .....	61
Modo de diagnóstico .....	61
4.3 Manejo del SAC .....	62
Puesta en operación .....	62
Realización de un diagnóstico .....	62
Paro o interrupción de la ejecución de un programa .....	63
4.4 Menús disponibles .....	63
<b>V GENERACIÓN DE PROGRAMAS DE APLICACIÓN DESDE UNA PC</b> .....	66
5.1 Introducción .....	66
5.2 Carga del programa de aplicación en el SAC .....	67
Programación con conexión al SAC (" <i>in-line</i> ") .....	67
Programación sin conexión al SAC (" <i>off-line</i> ") .....	67
5.3 Lenguaje de programación LENSAC .....	69
5.3.1 Formato de las instrucciones y sentencias de control del LENSAC .....	69
5.3.2 Parámetros asociados a las instrucciones y sentencias de control .....	70
5.3.3 Puertos, variables y operadores .....	71
5.3.4 Lista de instrucciones y sentencias de control .....	73
5.4 Programa PROSAC para Windows .....	79
5.4.1 Descripción .....	79
5.4.2 Estructura del programa PROSAC .....	80
5.4.3 Editor e intérprete .....	80
5.4.4 Compilador cruzado para MC68HC11 .....	83
5.4.5 Ensamblador .....	84
5.4.6 Pantallas de PROSAC .....	85
5.4.7 Botones o iconos .....	87
5.4.8 Ventana de edición .....	90
5.4.9 Menú Archivo .....	91
5.4.10 Menú Editar .....	92
5.4.11 Menú Compilar .....	93
5.4.12 Menú Transferir .....	94
5.4.13 Requerimientos y limitaciones de PROSAC .....	95
<b>VI INTEGRACIÓN E INSTALACIÓN DEL SAC</b> .....	97
6.1 Diagrama de interconexión .....	97
6.2 Configuración del <i>Hardware</i> .....	102
6.3 Puertos y conectores no accesibles al usuario .....	103

6.4 Puertos y conectores accesibles al usuario .....	104
6.5 Procedimiento de instalación .....	104
6.6 Cables y conectores .....	105
<b>VII EJEMPLOS DE PROGRAMAS DE APLICACIÓN</b> .....	<b>106</b>
7.1 Programa de aplicación utilizando puertos de entrada y salida.....	106
7.2 Programa de aplicación utilizando la unidad de almacenamiento de datos UAD-PCMCIA.....	107
7.3 Programa de aplicación utilizando el reloj fechador, el módulo de despliegue y las instrucciones diversas.....	108
7.4 Programa de aplicación utilizando el monitoreo de señales.....	109
7.5 Programa de aplicación utilizando las sentencias de control de flujo y el monitoreo de señales.....	110
7.6 Ejemplo de aplicación: "Transferencia de productos en las líneas de producción".....	112
7.6.1 La automatización de sistemas y mecanismos de transferencia de productos en las plantas industriales.....	112
7.6.2 Transferencia de los palets a las salidas de las líneas de producción.....	112
7.6.3 Presentación general del sistema a automatizar .....	113
7.6.4 Estudio de la parte operativa .....	117
7.6.5 Estudio de la parte de comando o <i>software</i> .....	120
7.6.6 Elaboración de los esquemas .....	131
7.6.7 Beneficios obtenidos.....	133
<b>VIII CONCLUSIONES</b> .....	<b>134</b>
<b>BIBLIOGRAFÍA</b> .....	<b>136</b>
<b>ANEXO A: Especificaciones técnicas del prototipo del SAC</b> .....	<b>137</b>
<b>ANEXO B: Descripción de cables y conectores del SAC</b> .....	<b>139</b>
<b>ANEXO C: Información sobre archivos en formato S19</b> .....	<b>145</b>

## I INTRODUCCIÓN

En nuestros días, las necesidades de control, de medición de variables físicas y de adquisición de datos han cobrado gran importancia, ya que actualmente se requieren automatizar la mayor parte de los procesos.

En este sentido, la gran capacidad de memoria y la elevada velocidad de procesamiento de los circuitos integrados permiten muy diversas aplicaciones.

Por otra parte, actualmente existen muchos sistemas o dispositivos electrónicos destinados a tareas específicas como son: el monitoreo de señales, la programación de interfaces de potencia para motores, el control de procesos, el registro de eventos, la medición de variables físicas como temperatura, voltaje, presión etc. Sin embargo, estos sistemas o dispositivos son de propósito específico y por lo mismo, poco versátiles.

Es aquí donde encaja el Sistema de Adquisición y Control de Propósito General, denominado SAC, el cual se creó con el fin de obtener, como su nombre lo indica, un sistema de control de procesos y de adquisición de datos que funcionara de acuerdo a las necesidades del usuario, es decir, un sistema de uso múltiple.

Este proyecto surge de necesidades relacionadas con actividades de instrumentación del CENAPRED (Centro Nacional de Prevención de Desastres), en las cuales se requiere de un sistema "inteligente" de supervisión y/o control que permita de una manera desatendida diagnosticar el estado de operación de un proceso o equipo en particular. A fin de darle una mayor versatilidad al SAC, se propuso darle un carácter múltiple para que por programa y de manera fácil, se pudiera adecuar a cualquier tipo de tarea, tanto de adquisición de datos, como de control de procesos.

Se desarrolló un prototipo del SAC que, aún hallándose en una fase experimental, tiene la posibilidad de aplicarse en tareas tan diversas como son el monitoreo y la generación de señales, el control y la medición de variables físicas, la adquisición de datos, la supervisión y control de procesos entre otras.

El SAC, en cuanto a *hardware* se refiere está constituido por:

- Un microcontrolador dedicado a controlar las funciones y tareas para las cuales ha sido programado.
- Varios puertos de entrada/salida digitales y de potencia.
- Entradas de detección de eventos.
- Diversos puertos de comunicación que permiten el intercambio de información con una computadora personal PC.
- Sistema supervisor de flujo de programa y falla de alimentación.
- Memoria RAM no volátil, donde se almacena el programa que contiene las instrucciones de la tarea a realizar.
- Memoria EPROM donde reside el sistema operativo.

- Una unidad de almacenamiento y lectura de datos reescribible, no volátil tipo PCMCIA.
- Sistema adquirente de señales analógicas de 4 canales con niveles de voltaje de entrada seleccionables.
- Sistema temporizador.
- Reloj fechador de tiempo real respaldado.
- Módulo de despliegue de mensajes o avisos.

Además, con relación al *hardware*, el equipamiento básico del SAC se puede ampliar con módulos adicionales en caso de que las necesidades de la aplicación así lo requieran.

La característica más importante del SAC es que es programable, es decir, el usuario tiene la posibilidad de realizar muy diversas tareas programando al sistema de acuerdo a sus necesidades.

El *software* del SAC está conformado por:

- Un sistema operativo propio capaz de recibir y ejecutar programas enviados a través de una computadora personal o programas almacenados en una tarjeta de memoria tipo PCMCIA.
- Un menú de diagnóstico que permite inicializar, revisar y verificar el estado del sistema, por medio de un teclado de control y un módulo de despliegue.
- Un *software* de programación interactivo que por medio de una interfaz gráfica desarrollada para Windows denominada PROSAC (Programación del Sistema de Adquisición y Control de Propósito General) permite al usuario desarrollar diferentes programas de aplicación de una manera sencilla, fácil, rápida e intuitiva.
- Un gran número de instrucciones o funciones que lo hacen muy versátil.

El SAC se programa en un lenguaje lógico y sencillo y no hace falta que el usuario aprenda un lenguaje especial de programación. De esta forma si las tareas y las necesidades así lo requieren, puede fácilmente cambiarse el programa de aplicación del SAC acoplándose así a las nuevas tareas a realizar.

El desarrollo de este trabajo se presenta a través de 8 capítulos y 3 anexos que a continuación se describen:

En el capítulo 2 se describen los diferentes bloques de los cuales está constituido el SAC; los componentes electrónicos que integran cada bloque y sus principales características; es decir, una explicación completa del *hardware* del SAC.

En el capítulo 3 se muestra y explica el sistema operativo del SAC, así como las diferentes rutinas que lo componen.

En el capítulo 4 se trata el funcionamiento y manejo del SAC en sus diferentes modos de operación. la forma de revisar el estado de operación del sistema y la manera de realizar un diagnóstico del mismo.

En el capítulo 5 se describe el programa denominado PROSAC con su interfaz gráfica desarrollada para Windows, así como las diferentes instrucciones y sentencias de control de flujo. Además, se explica la forma de programar al SAC por medio de esta interfaz gráfica.

En el capítulo 6 se presentan el prototipo del SAC y el diagrama de interconexión; además, se describen la configuración del *hardware* y los puertos del SAC.

En el capítulo 7 se presentan algunos ejemplos de aplicación muy sencillos y un ejemplo de aplicación más elaborado sobre "Transferencia de productos en las líneas de producción". Estos ejemplos permitirán explicar mejor las instrucciones y sentencias de control de flujo, y podrán dar una buena idea al lector de la versatilidad del SAC y de lo fácil que es programarlo por medio de la interfaz gráfica.

En el capítulo 8 se presentan las conclusiones y los resultados obtenidos.

Por último, se incluyen tres anexos: en el anexo A se presentan las especificaciones técnicas del SAC, en el anexo B se describen los cables y conectores utilizados en las diferentes tarjetas y en el anexo C, la descripción del formato de archivos S19.

## II DESCRIPCIÓN DEL SISTEMA DE ADQUISICIÓN Y CONTROL DE PROPÓSITO GENERAL, SAC

### 2.1 CARACTERÍSTICAS GENERALES

Las características generales que se establecieron para el diseño del sistema de adquisición y control de propósito general, denominado SAC, fueron las siguientes:

- Flexible, capaz de realizar diferentes funciones de control y adquisición de datos por programa sin necesidad de hacer muchas modificaciones al *hardware*
- Capaz de manejar dispositivos externos por medio de puertos de entrada/salida
- Facilidad de manejar señales analógicas y digitales de entrada
- Facilidad de comunicación y programación por medio de una computadora personal
- Memoria de almacenamiento de datos mediante una tarjeta PCMCIA
- Autónomo y confiable en su operación
- Fácil manejo y programación
- Rutinas de diagnóstico del sistema
- Bajo consumo de energía, alimentado con 12 VCD
- Bajo costo

Para poder cumplir con las características anteriores se diseñaron varios módulos electrónicos alrededor de una tarjeta principal de control y procesamiento.

En la figura 2.1 se muestra el diagrama de bloques del SAC el cual se compone de los módulos e interfaces siguientes:

- 1) sistema de control y procesamiento
- 2) alimentación
- 3) batería de respaldo
- 4) módulo acondicionador de señales y puerto de detección
- 5) módulo de despliegue y teclado de control
- 6) puerto de entrada 1 y puerto de salida 1
- 7) puerto de entrada 2 y puerto de salida 2
- 8) puerto de comunicación serie asincrónica y puerto paralelo
- 9) unidad de almacenamiento de datos UAD tipo PCMCIA.

A continuación se presentará una descripción de cada uno de estos bloques.

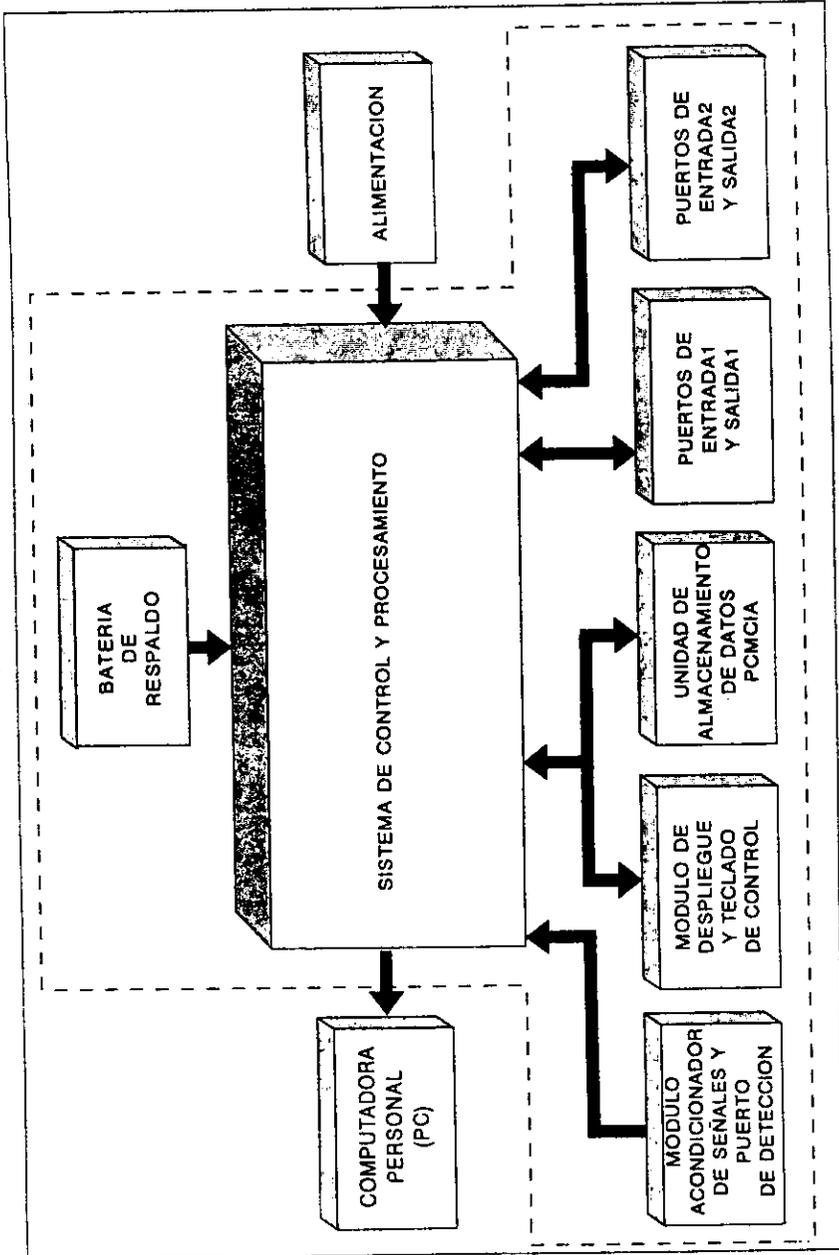


Figura 2.1. Diagrama de bloques del SAC

## 2.2 SISTEMA DE CONTROL Y PROCESAMIENTO

### Descripción general

El sistema de control y procesamiento es la parte más importante del SAC ya que, a través de él, se controlan las diferentes tareas tales como acceso al mundo real a través de puertos de entrada/salida, comunicación y transferencia de la información hacia una computadora y, además, el manejo de los distintos periféricos del sistema de adquisición y control.

Fue desarrollado con base en un microcontrolador el cual, por programa, controla y ejecuta las diferentes tareas o funciones.

Para el sistema de control y procesamiento se diseñó una tarjeta de circuito impreso especial con el fin de obtener una tarjeta que permita utilizar el microcontrolador teniendo acceso a todos los subsistemas que éste posee y utilizando el espacio de memoria en toda su capacidad. Además, se añadieron puertos de entrada/salida y periféricos externos. De esta manera, se obtuvo un sistema de control y procesamiento flexible y de fácil uso.

En la figura 2.2 se muestra un diagrama de bloques del sistema de control y procesamiento el cual se compone de los dispositivos siguientes:

Microcontrolador, memorias EPROM y RAM, regulador de voltaje, batería de respaldo, reloj fechor de tiempo real, circuito supervisor, puerto de comunicación serie asincrónica, puerto paralelo de comunicación, puerto de expansión para dispositivos externos, puerto especial para el acceso al convertidor analógico-digital, al puerto de comunicación serie síncrono y al sistema de tiempo, dos puertos de entrada y dos puertos de salida

En la memoria EPROM se almacena permanentemente el sistema operativo de SAC.

En la memoria RAM se almacena el programa hecho por el usuario.

La batería de respaldo, garantiza la integridad de los datos del reloj fechor y del programa hecho por el usuario.

El regulador de voltaje proporciona la alimentación principal al SAC que es de +5V a partir de +12V

El circuito supervisor permite mantener la operación autónoma del SAC.

Los diferentes puertos de entrada/salida y el puerto de expansión permiten la comunicación con dispositivos externos.

A través de los puertos de comunicación serie y paralela se recibe y envía información, generalmente con una computadora personal.

El puerto especial permite utilizar el convertidor analógico-digital y el sistema temporizador del microcontrolador.

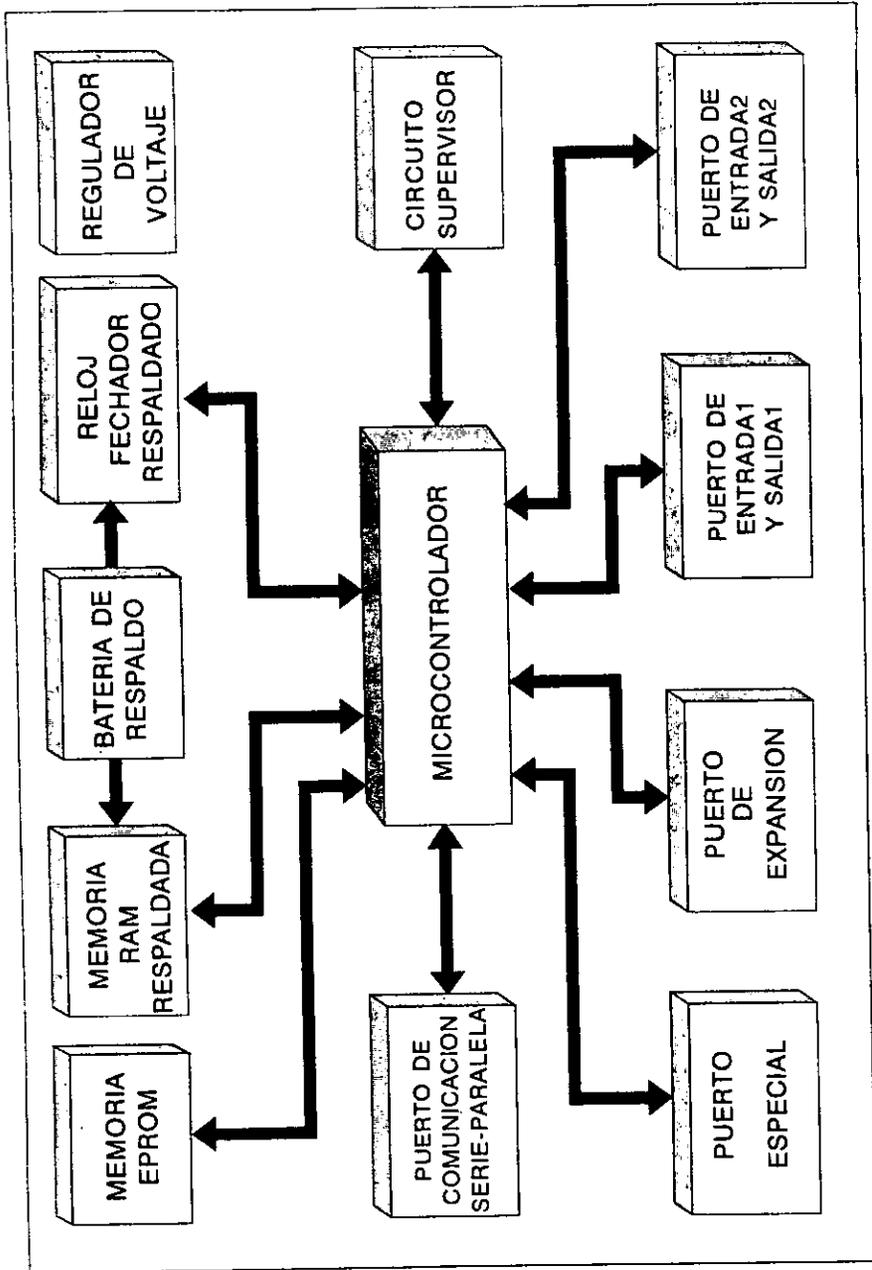


Figura 2.2. Diagrama del sistema de control y procesamiento.

En las figuras 2.3, 2.4, 2.5 y 2.6 se muestran los diagramas electrónicos de la tarjeta del sistema de control y procesamiento. A continuación se describirá con detalle cada subsistema.

### Microcontrolador

El microcontrolador utilizado en el SAC es el MC68HC11A1P perteneciente a la familia HC11 de Motorola. Es de 8 bits, de tecnología HCMOS (*High-density Complementary Metal Oxide Semiconductor*) y trabaja a una velocidad de bus de 2 MHz utilizando un cristal de cuarzo de 8 MHz.

Un diagrama simplificado de este circuito se muestra en la figura 2.7.

Las características y dispositivos que posee integrados el microcontrolador son:

- Unidad central de procesamiento MC68HC11 CPU
- 512 bytes de memoria EEPROM
- 256 bytes de memoria RAM estática
- 192 bytes de memoria ROM interna
- Un sistema temporizador de 16 bits
  - a) 4 canales de salida de comparación
  - b) 3 canales de entrada de captura
  - c) 1 canal de entrada de captura o de salida de comparación seleccionable
- Un acumulador de eventos de 8 bits
- Circuito de interrupción de tiempo real
- Sistema de supervisión de software (*COP Watchdog System*)
- Dos modos de operación de bajo consumo STOP y WAIT
- Interfaz de comunicación serie síncrona y asíncrona
- Convertidor analógico-digital de 8 bits de 4 canales
- Cinco puertos de entrada/salida programables
- Expansión de bus para dispositivos externos con espacio de direccionamiento de 64 Kbytes

Los dispositivos anteriores son controlados por un conjunto de registros internos del microcontrolador.

Aunque existen 4 modos de operación, modo *Test*, modo *Bootstrap*, modo *Single Chip* y modo *Expanded Multiplexed*, dependiendo del nivel lógico presente en las entradas MODA y MODB, el modo de operación utilizado es el modo expandido. Con este modo se tiene acceso a los buses de datos y de direcciones los cuales permiten manejar memorias y periféricos externos en un espacio de memoria de 64 Kbytes.

La parte alta del bus de direcciones se encuentra en las terminales de salida del puerto B. La parte baja del bus de direcciones y los 8 bits del bus de datos se encuentran multiplexados en las terminales de salida del puerto C. Mediante un circuito *latch* octal 74HC373 (U2 figura 2.3) y a la señal de salida AS (*adress strobe*) se demultiplexa la parte baja del bus de direcciones y el bus de datos del puerto C. De esta manera se pueden acceder dispositivos externos dentro del espacio de memoria direccionable.

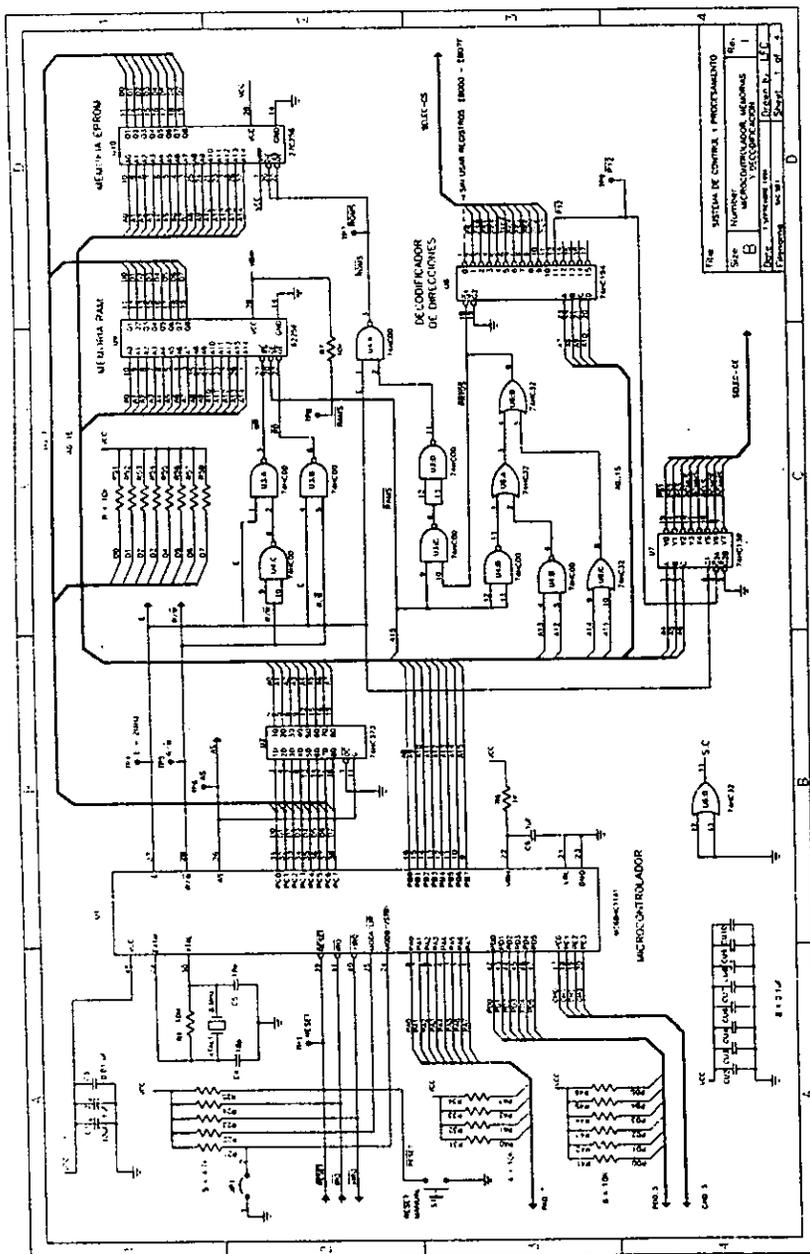


Figura 2.3. Sistema de control y procesamiento, microcontrolador, memorias y decodificación.

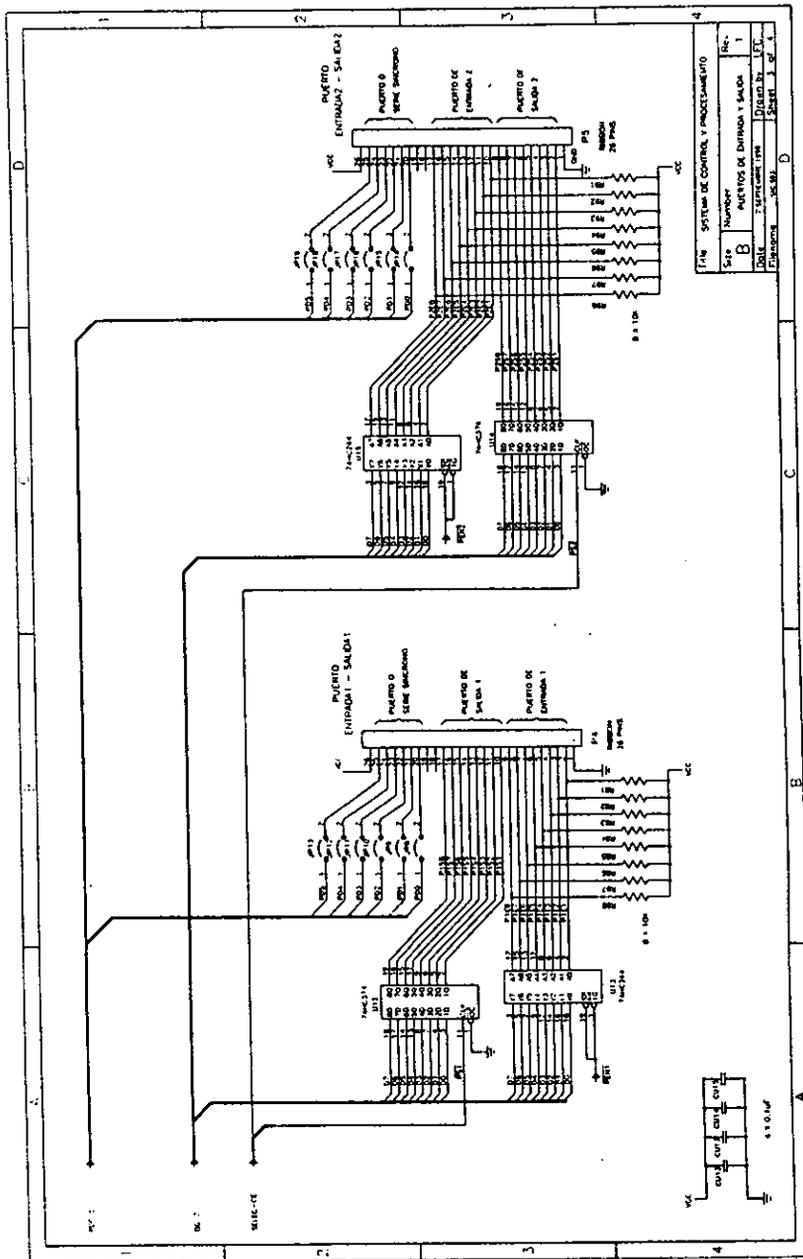


Figura 2.4. Puerto serie sincrono y puertos de entrada/salida 1 y 2.

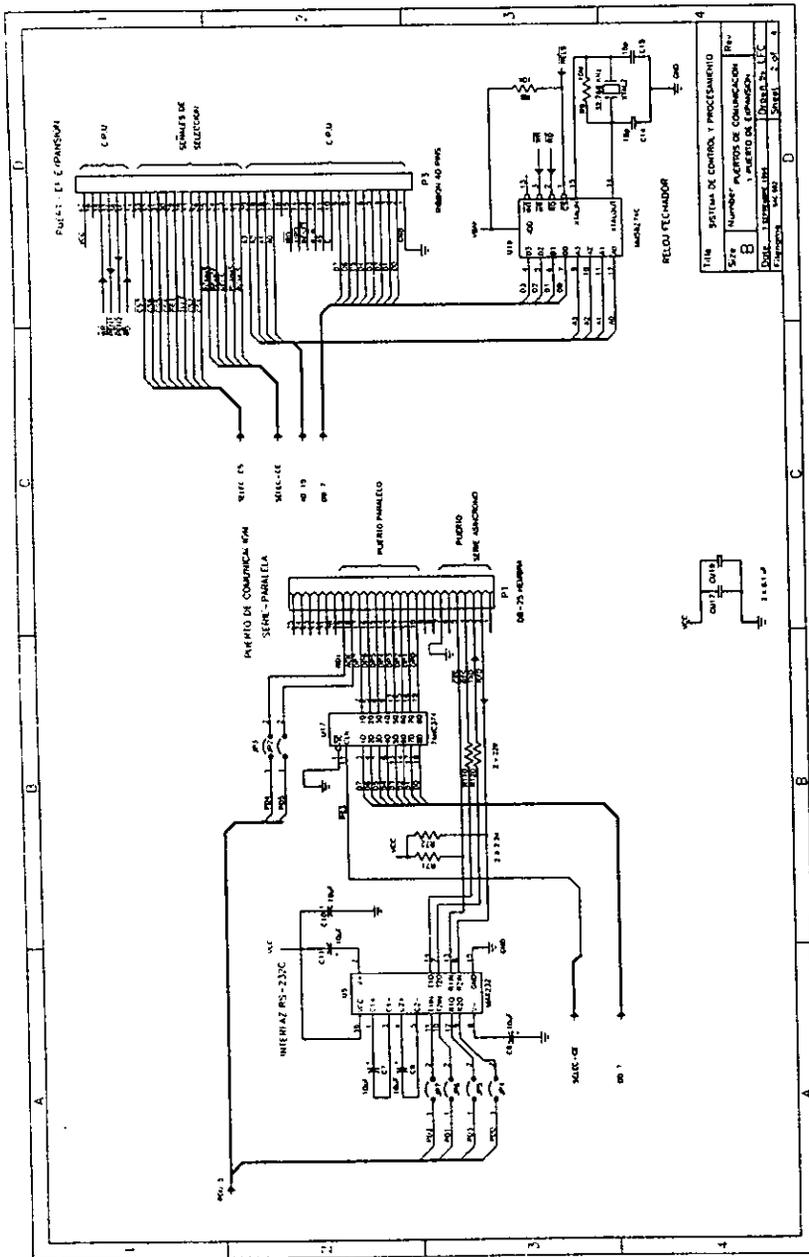


Figura 2.5. Puerto de expansión y puertos de comunicación.





### Mapa de memoria del sistema y decodificación

Dependiendo del modo de operación del sistema se determina su mapa de memoria.

Utilizando el modo de operación expandido, el microcontrolador MC68HC11A1P puede direccionar hasta 64Kbytes. El microcontrolador permite reacomodar el mapa de memoria original (modo *Single (chip)*) empleando para este fin el registro de control INIT.

Normalmente la dirección de *default* de la RAM interna es \$0000 y la de los registros de control es \$1000.

El registro INIT de 8 bits permite modificar las ubicaciones de *default* de los registros de control y de la RAM interna después de un RESET y sólo se puede acceder a este registro dentro de los primeros 64 ciclos de reloj que siguen a un RESET.

Con el registro CONFIG del microcontrolador se puede deshabilitar la memoria ROM interna.

La razón por la cual se remapean la RAM interna y los registros internos del microcontrolador es para hacer al microcontrolador compatible con el mapa de memoria deseado para el SAC, para el cual se desea la mayor cantidad posible de RAM externa, y también de memoria EPROM externa para almacenar el programa del sistema operativo de SAC.

El mapa de memoria del SAC se muestra en la figura 2.8, el rango de direcciones de los dispositivos externos y las señales de selección se muestran en la tabla 2.1. El SAC tiene dos espacios de memoria: la Memoria Principal del microcontrolador (de 64Kbytes) y una Memoria Extendida de 1Mbyte expandible a 16Mbytes. Esta última se describirá posteriormente.

Respecto a la memoria principal su mapa está organizado de la siguiente manera:

Los primeros 32 Kbytes (\$0000-\$7FFF) los ocupa la memoria RAM externa; los 12288 bytes siguientes (\$8000-\$AFFF) los ocupa la memoria externa EPROM; los registros internos del microcontrolador ocupan 64 bytes (\$B000-\$B03F); los direccionamientos externos ocupan 1472 bytes (\$B040-\$B5FF); la memoria EEPROM interna del microcontrolador abarca 512 bytes (\$B600-\$B7FF) y por último la memoria externa EPROM ocupa los 18432 bytes restantes (\$B800-\$FFFF).



RANGO DE DIRECCIONES	DISPOSITIVO SELECCIONADO	SEÑAL DE SELECCIÓN
\$0000 - \$7FFF	Memoria RAM 32 Kbytes	RAMS
\$8000 - \$AFFF	Memoria EPROM, 12288 bytes	ROMS
\$B000 - \$B03F	Registros internos del HC11, 64 bytes	---
\$B040 - \$B07F	Espacio disponible, 64 bytes	---
\$B080 - \$B0FF	Espacio disponible, 128 bytes	CS
\$B100 - \$B17F	Espacio disponible, 128 bytes	CS0
\$B180 - \$B1FF	Espacio disponible, 128 bytes	CS1
\$B200 - \$B27F	Espacio disponible, 128 bytes	CS2
\$B280 - \$B2FF	Teclado	TECF
\$B300 - \$B37F	Espacio disponible, 128 bytes	CS3
\$B380 - \$B3FF	Espacio disponible, 128 bytes	CS4
\$B400 - \$B47F	Espacio disponible, 128 bytes	CS5
\$B480 - \$B4FF	Espacio disponible, 128 bytes	CS6
\$B500	Visualizador	CS7, LCD-CNTL
\$B501	Visualizador	CS7, LCD-DATA
\$B502	Teclado	CS7, TECCR
\$B503	Puerto de entrada No. 1	PEN1
\$B504	Puerto de entrada No. 2	PEN2
\$B505	Espacio disponible, 1 byte	CE3
\$B506	Espacio disponible, 1 byte	CE4
\$B507 - \$B57F	Espacio disponible, 121 bytes	CE5
\$B580 - \$B58F	Puerto de salida No.1	PI2, PS1
\$B590 - \$B59F	Puerto de salida No.2	PI2, PS2
\$B5A0 - \$B5AF	Puerto paralelo o de salida No. 3	PI2, PS3
\$B5B0 - \$B5BF	PCMCIA	PI2, PCMLS
\$B5C0 - \$B5CF	PCMCIA	PI2, PCMMS
\$B5D0 - \$B5DF	RELOJ FECHADOR	PI2, RELS
\$B5E0 - \$B5EF	PCMCIA	PI2, PCMCE
\$B5F0 - \$B5FF	PCMCIA	PI2, PCMHS
\$B600 - \$B7FF	Memoria EEPROM interna, 512bytes	---
\$B800 - \$FFFF	Memoria EPROM externa, 18432 bytes	ROMS

Tabla 2.1 Decodificación de la memoria

### Memorias

Las memorias presentes en la tarjeta del sistema de control y procesamiento son varias. Las memorias RAM y ROM internas del microcontrolador se encuentran deshabilitadas y la memoria EEPROM interna del microcontrolador no se utiliza. Por ello las únicas dos memorias utilizadas en esta tarjeta son la RAM y EPROM externas.

#### a) Memoria EPROM

La memoria EPROM (*Erasable and electrically Programmable Read Only Memory*) es una memoria no volátil que, como su nombre lo indica, puede ser borrada mediante luz ultravioleta y es sólo de lectura una vez que ha sido programada.

La memoria EPROM (U10 figura 2.3) es el circuito 27C256 que tiene una capacidad de 32 Kbytes organizados en 32768 palabras de 8 bits cada una. Es en esta memoria donde reside permanentemente el programa del sistema operativo del SAC. El diagrama de tiempos del ciclo de lectura de la EPROM se muestra en la figura 2.9.

#### b) Memoria RAM

La memoria RAM (*Random Access Memory*) es una memoria que permite leer y escribir datos de manera aleatoria y es del tipo volátil. La memoria RAM (U9 figura 2.3) empleada es el circuito 62256 con una capacidad de 32 Kbytes organizados en 32768 palabras de 8 bits cada una. En esta memoria se almacena el programa elaborado por el usuario con la función o tarea deseada. La información contenida en la memoria RAM puede modificarse rápidamente y tantas veces como se precise. Sin embargo, tiene el inconveniente de que su contenido (el programa a realizar) se pierde si falta el voltaje de alimentación. Para evitar esto, se utilizó una batería de respaldo (ver figura 2.6).

En las figuras 2.10 y 2.11 se muestran los diagramas de tiempos de los ciclos de lectura y escritura de la memoria RAM, respectivamente.

#### Puertos de entrada

La tarjeta del sistema de control y procesamiento cuenta con 2 puertos de entrada digitales, puertos de entrada 1 y 2, cada uno con 8 líneas las cuales aceptan niveles de voltaje de entrada entre 0 y 5 Volts (ver figura 2.4).

La lectura de los puertos de entrada se realiza a través del microcontrolador. Físicamente el acceso a las líneas de entrada del puerto de entrada 1 se tiene a través del conector P4 y el acceso a las líneas de entrada del puerto de entrada 2 a través del conector P5 (ver anexo B: Descripción de cables y conectores del SAC).

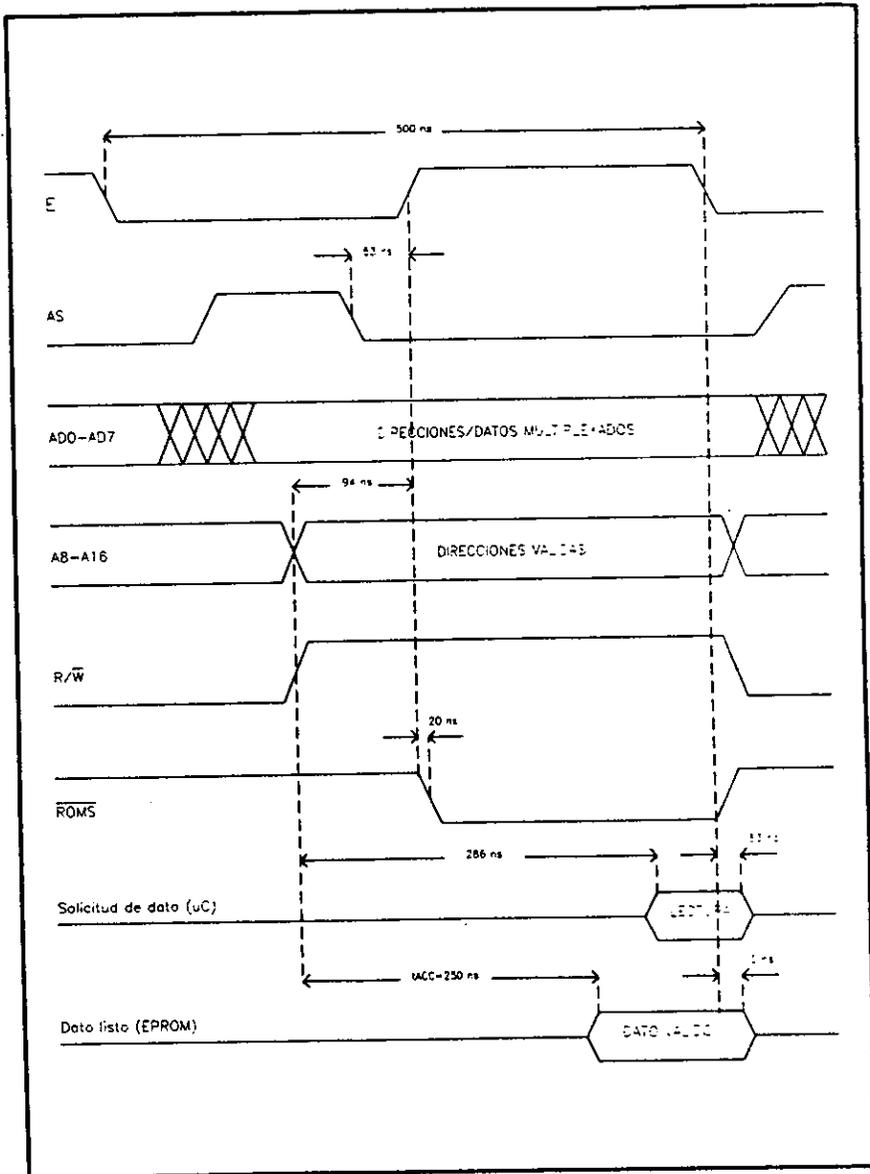


Figura 2.9. Ciclo de lectura de la memoria EPROM



### Puertos de salida

El sistema de control y procesamiento cuenta con 2 puertos de salida digitales, puertos de salida 1 y 2, cada uno cuenta con 8 líneas las cuales aceptan niveles de voltaje de entrada entre 0 y 5 Volts (ver figura 2.4).

La escritura de los puertos de salida se realiza a través del microcontrolador. Físicamente el acceso a las líneas de salida del puerto de salida 1 se tiene a través del conector P4 y las del puerto de salida 2 por el conector P5 (ver anexo B: Descripción de cables y conectores del SAC).

### Puertos de comunicación

#### a) Puerto paralelo

El puerto paralelo es exclusivamente de salida y de 8 bits (figura 2.5). Se ubica en la dirección \$B5A0 dentro del mapa de memoria y utiliza dos líneas de control o de *handshake* denominadas *ready* o listo (RDY) y *acknowledge* o reconocimiento (ACK), las cuales se toman del puerto D serie síncrono del microcontrolador (PD4=RDY y PD5=ACK). Estas líneas deben configurarse como entradas o salidas en el registro de dirección del puerto D (DDRD, 0=entrada, 1=salida) además de instalar los puentes JP2 y JP3 en la tarjeta del sistema de control y procesamiento.

El puerto paralelo se puede acceder a través del conector P1 (ver anexo B: Descripción de cables y conectores del SAC). Como el puerto paralelo utiliza las líneas PD4 y PD5 y, como se verá, estas líneas también son necesarias para el puerto serie síncrono, ambos puertos no se pueden utilizar a la vez.

#### b) Puerto serie asíncrono

El puerto serie asíncrono permite la comunicación con otros dispositivos externos a través de una interfaz RS-232C (ver figura 2.5). Normalmente este puerto conecta al SAC con una computadora personal PC para su programación e intercambio de datos.

La comunicación se lleva a cabo a través del UART (*Universal Asynchronous Receiver / Transmitter*) del microcontrolador, utilizando las líneas, RX para la recepción y TX para la transmisión derivadas de las líneas PD0 y PD1, respectivamente.

Con la finalidad de adecuar los niveles de voltaje de salida del puerto serie asíncrono del microcontrolador a los especificados por el estándar RS-232C se utilizó un circuito acondicionador de nivel MAX232 (U5 figura 2.5).

El puerto asíncrono tiene dos líneas de control denominadas *ready to send* (RTS) y *clear to send* (CTS) que se derivan de las líneas PD2 y PD3 respectivamente. Por ello el puerto serie síncrono y el asíncrono no pueden operar simultáneamente. Para utilizar estas dos líneas de control, es necesario configurar las líneas CTS y RTS una como salida y la otra como entrada en el registro de dirección del puerto D (DDRD, 0=entrada y 1= salida), instalar los puentes JP5 y JP7 y, además, configurar a los registros internos del microcontrolador.

El puerto serie asíncrono es un sistema *full-duplex* que utiliza el formato estándar NRZ (*Non-return-to-zero*). Para el SAC se utiliza un bit de inicio, ocho bits de datos, un bit de finalización y ninguna paridad. La velocidad de transmisión de datos es fija a 9600 bauds.

La figura 2.12 muestra el formato con el que se envían los datos.

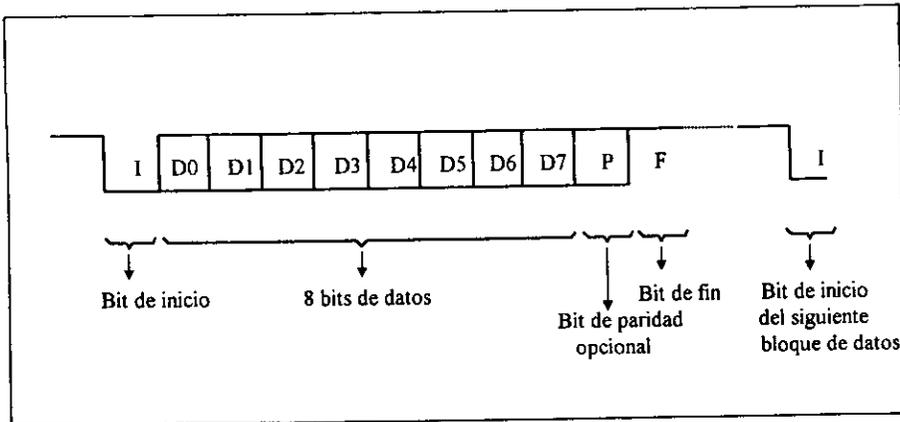


Figura 2.12. Formato de transmisión de datos.

### c) Puerto serie sincrónico<sup>1</sup>

El puerto serie sincrónico no se utilizó pero está disponible para aplicaciones y desarrollos futuros del SAC.

El puerto serie sincrónico puede manejar velocidades de transferencia seleccionables por *software*. La fase del reloj y su polaridad también son programables, lo cual le permite adaptarse a numerosos dispositivos periféricos.

Cuatro líneas están asociadas a este puerto, MOSI (*master-out-slave-in*) en la línea PD3, MISO (*master-in-slave-out*) en la línea PD2, el reloj serial SCK línea PD4 y SS (*slave select*) línea PD5. El acceso a estas líneas del puerto D es a través del conector P4 debiéndose instalar los puentes JP10, JP11, JP12 y JP13.

También, puede accederse al puerto D a través del conector P5 instalando los puentes JP16, JP17, JP18 y JP19. Como el puerto paralelo utiliza las líneas PD4 y PD5 para su funcionamiento y el puerto serie sincrónico también, ambos puertos no pueden utilizarse a la vez.

<sup>1</sup> No se pueden utilizar el puerto paralelo y el puerto serie sincrónico a la vez (ver el capítulo 6).

### **Puerto especial**

Este puerto, mostrado en la figura 2.6, es muy importante ya que a través de él se tiene acceso al puerto A del microcontrolador y al sistema temporizador que se describirá más adelante. Además, a través de este puerto se tiene acceso a los puertos de comunicación serie síncrona y asíncrona del microcontrolador y también al puerto E en el cual se encuentran las cuatro líneas de entrada analógicas del convertidor analógico-digital.

El puerto especial, se accesa a través del conector P2 (ver anexo B: Descripción de cables y conectores del SAC).

### **Puerto de expansión**

A través de este puerto se puede expandir el sistema y accesar distintos dispositivos externos como el módulos de despliegue, teclado, etc., ubicados en direcciones fijas dentro del mapa de memoria. Para ello tiene disponible la mayor parte de las señales de control y direccionamiento que maneja la tarjeta del sistema de control y procesamiento como son los 8 bits del bus de datos, las cuatro líneas mas bajas del bus de direcciones A0-A3, la línea de selección de lectura escritura *read-write* (R/W), la línea de demultiplexión de direcciones y datos *address-strobe* (AS), las líneas de interrupción no mascarable y mascarable XIRQ e IRQ, la señal de RESET o inicialización, la señal E de reloj del microcontrolador, las señales de alimentación de voltaje GND y VCC y varias líneas de selección (ver figura 2.6).

El puerto de expansión se puede accesar a través del conector P3 (ver anexo B: Descripción de cables y conectores del SAC).

### **Convertidor analógico-digital**

El microcontrolador contiene un convertidor analógico-digital A/D que permite convertir señales analógicas a valores digitales de 8 bits, por lo tanto su resolución es de una parte en 256, o sea, 48 dB.

El convertidor analógico-digital es de aproximaciones sucesivas de 4 canales de entrada multiplexados con *sample and hold* (muestreo y retención). Además, cuenta con dos líneas (VRH y VRL) para el voltaje de referencia de entrada, las cuales permiten incrementar la exactitud de la conversión del convertidor. Los niveles de voltaje de entrada son de 0 a 5 Volts.

El convertidor analógico-digital tiene una precisión de  $\pm 1$  bit (error de cuantización de  $\pm 1/2$  LSB y  $\pm 1/2$  de otros tipos). Cada conversión se realiza en 32 ciclos de reloj del microcontrolador, es decir, en  $16\mu\text{s}$  si la frecuencia del bus es de  $E=2$  MHz.

Dispone de 4 registros de resultados y cuatro tipos diferentes de secuencias de conversión:

- Convierte un solo canal cuatro veces consecutivas y luego se para. Los resultados de la secuencia se guardan en los registros de resultados.

- Convierte un solo canal continuamente, actualizando los registros de resultados de manera circular<sup>1</sup>
- Convierte consecutivamente los cuatro canales y para; almacena el resultado de la conversión en el registro correspondiente.
- Convierte continuamente los cuatro canales (de manera circular)<sup>1</sup> y almacena los resultados de la conversión en el registro correspondiente.

Las 4 líneas de entrada del convertidor analógico-digital están disponibles en el puerto especial. (ver figura 2.6) a través del conector P2 (ver anexo B: Descripción de cables y conectores del SAC).

### Temporizador

El microcontrolador tiene un sistema temporizador que cuenta con tres entradas de captura, cuatro salidas de comparación y una línea extra configurable como entrada o salida.

Las entradas de captura detectan cuando el tipo de transición seleccionada en la línea de entrada se ha presentado y también registran el tiempo cuando ocurrió esta transición. Esto es muy útil cuando se desea medir por ejemplo el período o el ancho del pulso de una señal.

Las salidas de comparación se utilizan para programar que ocurra una acción en alguna de ellas en un tiempo específico. Esto es útil cuando se desea producir un pulso de una duración dada o una señal de salida de determinada frecuencia.

Además de las funciones de entrada de captura y salidas de comparación, el sistema temporizador tiene asociadas otras actividades como la interrupción de tiempo real, el acumulador de eventos, y el sistema COP (*Computer Operating Properly watchdog*) las cuales aunque están disponibles, no se utilizaron en el SAC.

Las entradas y salidas del temporizador están disponibles en el puerto especial, (ver figura 2.6) a través del conector P2 (ver anexo B: Descripción de cables y conectores del SAC).

### Interrupciones

Además de la interrupción de RESET o inicialización, el microcontrolador HC11 permite manejar un gran número de interrupciones de *hardware* como de *software*. A continuación se describen las interrupciones utilizadas en el SAC.

---

<sup>1</sup> De manera circular, quiere decir que se convierte la señal del canal 1 y se guarda la muestra en el registro 1 y así sucesivamente para los cuatro canales. Luego se vuelve a convertir la señal del canal 1 y la muestra se guarda en el registro 1 sobrescribiendo el valor anterior y se continúa de igual manera para los otros canales.

### a) Reset o inicialización

Existen tres posibles fuentes de *reset* o inicialización del SAC: cuando se ha detectado una ausencia, transición o bajo nivel en el voltaje de alimentación, cuando el microcontrolador no está operando adecuadamente debido a un mal flujo del programa (función de *watchdog*), y cuando se ha dado un *reset* manual. En cualquiera de estos casos se pondrá en nivel bajo (0 Volts) la señal de  $\overline{\text{RESET}}$ .

Cuando el microcontrolador detecta un nivel bajo (0 Volts) en la entrada de  $\overline{\text{RESET}}$ , entonces detiene la ejecución del programa del sistema operativo y lo reinicia desde el principio.

### b) Interrupción $\overline{\text{IRQ}}$

La interrupción  $\overline{\text{IRQ}}$  o *interrupt request* es una interrupción de tipo mascarable. Cuando se detecta un nivel bajo en esta entrada, (debido a que se ha apretado la tecla "ON" del teclado de control) se interrumpe el flujo del programa del SAC y se da servicio a la rutina asociada de interrupción (la cual permite al usuario realizar un diagnóstico y revisión de las condiciones actuales del SAC entre otras funciones). Después de terminada esta rutina, se regresa al flujo normal de programa.

### c) Interrupción del puerto serie asíncrono

La interrupción de "dato recibido" por el puerto serie, es de gran utilidad, ya que permite parar la ejecución de un programa por medio del teclado de una computadora conectada al puerto serie asíncrono.

En el caso del SAC, cuando se ha detectado la recepción de un dato por el puerto serie asíncrono, se da servicio a una rutina de interrupción la cual verifica si el dato recibido corresponde a la tecla ESC (valor hexadecimal \$1B). En caso afirmativo, se para la ejecución del programa previamente cargado en la memoria de programa RAM, se ponen en nivel bajo todas las líneas de todos los puertos de salida y se espera el arribo de un nuevo programa. En caso negativo, se continúa con el flujo normal del programa previamente cargado.

### Reloj fechador de tiempo real

Una de las funciones más importantes a realizar por el SAC, es el registro correcto y preciso del tiempo. Para esto, se le incorporó a la tarjeta del sistema de control y procesamiento un reloj fechador de tiempo real. El reloj fechador utilizado es el circuito MM58274C (U19 figura 2.5).

Este circuito es directamente compatible con el bus de datos y direcciones del microcontrolador MC68HC11A1P y, además, cuenta con una base propia de referencia de tiempo basada en un cristal de 32.768 KHz.

El reloj fechador es un circuito independiente que trabaja de manera autónoma e ininterrumpida. Para ello cuenta con una batería de respaldo con la cual se garantiza el registro continuo de la fecha y hora al ocurrir una interrupción de la alimentación principal.

### Circuito supervisor

Con el fin de asegurar una operación autónoma y continua del SAC se le agregó al sistema un circuito MAX691CPE supervisor del microcontrolador (U16 figura 2.6). Este dispositivo realiza las funciones de supervisión del voltaje de alimentación y de autoinicialización comúnmente conocido como *watchdog* o "perro guardián" del SAC.

En ambos casos el circuito MAX691CPE genera una señal de  $\overline{\text{RESET}}$  la cual asegura que el microcontrolador se encienda en un estado conocido, y previene los errores de ejecución de programa cuando ocurre una falla en la alimentación.

Cuando el circuito realiza la función de supervisor del voltaje de alimentación VCC (+5V), activa la señal de  $\overline{\text{RESET}}$  (nivel bajo) en caso de que su valor caiga por debajo de 4.65 Volts.

Para la función de *watchdog* el MAX691CPE cuenta con una entrada WDI (*watchdog input*) la cual debe ser periódicamente activada por el microcontrolador.

Para el caso del SAC se conectó la entrada WDI a la línea E (señal de reloj) del bus del microcontrolador. Si WDI se mantiene en un nivel alto o bajo por más tiempo que el tiempo requerido por el *watchdog* (1.6 seg nominal) se activa la señal de  $\overline{\text{RESET}}$  reiniciando al microcontrolador y la ejecución del programa.

### Alimentación del sistema

El SAC, requiere de varias fuentes de alimentación para su operación.

En la figura 2.6 se muestra que el sistema se puede alimentar directamente con una fuente de voltaje externa de 5VCD, instalando el puente JA<sup>1</sup>, o también, se puede utilizar un voltaje de alimentación externo con valores entre 9 y 18 VCD instalando el puente JB<sup>1</sup>.

Al instalar el puente JB se hace uso del regulador de voltaje LM7805 quien proporcionará al sistema una corriente de hasta 1.0 A con el voltaje constante de 5VCD necesario para su funcionamiento.

Además de esta fuente, el SAC requiere de una batería de respaldo para la memoria RAM y el reloj fechador y otro sistema de alimentación de  $\pm 12$  VCD para el acondicionador de señales los cuales se describirán en las secciones siguientes.

### Batería de respaldo

Además de la fuente principal de alimentación de 5VCD, el SAC requiere de una batería de respaldo para la memoria RAM y el reloj fechador. Esta alimentación de respaldo se da mediante tres baterías alcalinas tipo AA que en conjunto proporcionan un voltaje de 4.5 volts (ver figura 2.6).

---

<sup>1</sup> **IMPORTANTE:** Jamás se deberán instalar los puentes o *jumpers* JA y JB simultáneamente (ver capítulo 6).

Como se dijo anteriormente, el programa hecho por el usuario y que ejecuta el sistema reside en la memoria RAM. Si fallase la alimentación, y dado que la memoria RAM es volátil, sus datos se perderían y en consecuencia se dejaría de ejecutar el programa allí residente.

Para llevar un registro constante y de manera ininterrumpida de la hora y fecha se utiliza también esta batería para respaldar el voltaje de alimentación del reloj fechador.

#### **Distribución de componentes y lista de material**

Como ya se mencionó, para el sistema de control y procesamiento se diseñó una tarjeta de circuito impreso especial. En la figura 2.13 se muestra la disposición de los componentes electrónicos tal y como se encuentran ubicados en esa tarjeta. En la tabla 2.2 se da una lista completa de los componentes que la integran.



REFERENCIA	DESCRIPCIÓN
U1	MC68HC11A1; microcontrolador
U2	74HC373; latch octal
U3, U4	74HC00; cuádruple compuerta NAND
U5	MAX232; adaptador de nivel
U6	74HC32; cuádruple compuerta OR
U7	74HC138; decodificador 3 X 8
U8	74HC154; decodificador 4 X 16
U9	62256; memoria RAM 32 Kbytes
U10	27C256; memoria EPROM 32 kbytes
U11	MC34064; detector de bajo voltaje
U12, U14, U17	74HC374; latch octal
U13, U15	74HC244; buffers tri-state
U16	MAX691; circuito supervisor
U18	LM7805; regulador de voltaje de 5V
U19	MM58274C; reloj fechador
XTAL1, XTAL2	Cristales de cuarzo de 8MHz y de 32.768KHz
R1, R9	Resistencia, 10M ohms
R21-R25	Resistencia, 4.7k ohms
R31-R34, R41-R46, R51-R58, R7, R8, R81-R88, R91-R98	Resistencia, 10k ohms
R6	Resistencia, 1k ohms
R71, R72	Resistencia, 2.2k ohms
RT10, RT20	Resistencia, 220 ohms
D1, D2	Diodos, D1 Schottky MBD701, D2 1N4148
C1, C7, C8, C9, C10, C11	Capacitor, 10 uF
C2, C6	Capacitor, 1 uF
C3	Capacitor, 0.01 uF
C4, C5, C14, C15	Capacitor, 18pF
C12, C13, CU2-CU10, CU12-CU15, CU17, CU19	Capacitor, 0.1 uF
JA, JB, JP1-JP19	Puentes

Tabla 2.2 Lista de componentes de la tarjeta de control y procesamiento

## 2.3 MÓDULO ACONDICIONADOR DE SEÑALES Y PUERTO DE DETECCIÓN

El sistema de adquisición y control de propósito general SAC cuenta con un módulo acondicionador de señales analógicas y puerto de detección de eventos como se puede ver en su diagrama de bloques (figura 2.1).

### Puerto de detección

El puerto de detección de eventos es básicamente un puerto de entrada digital el cual se conecta al puerto A del microcontrolador por medio de un circuito *buffer* 4050 (U6 figura 2.14). Este circuito permite manejar voltajes a la entrada entre 0 y 18 VCD. Por programa el usuario puede seleccionar el tipo de entrada a detectar, ya sea, un flanco de subida, uno de bajada o una transición; las resistencias R1-R3 mantienen las entradas a tierra cuando no hay señales conectadas al puerto. El acceso a las entradas de este puerto es a través del conector P12.

### Acondicionador de señales

El SAC cuenta con un convertidor analógico-digital que permite convertir cuatro señales analógicas de entrada de 0 a 5V CD a valores digitales de 8 bits.

Con el fin de poder manejar señales de entrada analógicas tanto unipolares como bipolares de distintos niveles de amplitud se diseñó un circuito acondicionador de señales de tres canales mostrado en la figura 2.14. Las señales analógicas entran por los conectores BNC1, 2, y 3 y luego de ser acondicionadas entran a través de P11 al puerto especial (ver figura 6.5 del capítulo 6).

Para poder tener un buen acoplamiento de impedancias entre las fuentes de las señales analógicas y el acondicionador, se utilizaron para cada uno de los canales de entrada unos amplificadores operacionales en configuración seguidor empleando el circuito LM358N (U1, U2 figura 2.14).

Para seleccionar los rangos de voltaje de entrada de las señales analógicas se tiene a la entrada del acondicionador un *switch* o interruptor rotativo. Los seis rangos de voltaje de entrada disponibles según la posición del *switch* rotativo son:

- 0 a 5 Volts
- 0 a 10 Volts
- 0 a 20 Volts
- 5 Volts pico a pico ( $\pm 2.5$  V)
- 10 Volts pico a pico ( $\pm 5$  V)
- 20 Volts pico a pico ( $\pm 10$  V)

Dado que el convertidor analógico-digital únicamente acepta voltajes de entrada de 0 a 5V, cuando se desean manejar señales unipolares entre 0 y 20 Volts, éstas se pasan primero por un divisor de voltaje que las reduce a los niveles adecuados. De allí las señales se llevan directamente al convertidor analógico-digital a través del conector P11.

Por el contrario, cuando se desean manejar señales de entrada analógicas bipolares entre 0 y 20 Volts pico a pico, se necesita, antes que nada, sumarle a la señal de entrada una componente de CD adecuada para convertir la señal en una señal unipolar que únicamente tenga una componente de voltaje positiva. Después, hay que atenuar la señal para que en la entrada del convertidor analógico-digital se tenga una señal analógica que varíe entre 0 y 5 Volts y que sea proporcional a la señal analógica original.

Para poder realizar esto se utilizaron unos amplificadores operacionales en configuración sumador e inversor empleando el circuito LM358N (U3, U4, U5 figura 2.14).

Por otra parte, los amplificadores operacionales requieren de una alimentación de  $\pm 12$  VCD de polarización para que puedan manejar las señales analógicas de entrada con los rangos de voltaje descritos.

En la tarjeta del sistema de control y procesamiento; únicamente se puede disponer, por medio de la fuente principal de alimentación; del voltaje de alimentación VCC, el cual es de 5 VCD, por lo que fue necesario agregar otra fuente de alimentación dedicada a la polarización de los amplificadores operacionales.

Puesto que se dispone sólo de una fuente de +5 VCD, para la alimentación de los amplificadores operacionales se utilizó el circuito integrado convertidor CD-CD MAX743 (ver figura 2.15) el cual con algunos circuitos externos y una alimentación única de + 5 VCD permite generar dos salidas de voltaje regulado independientes, una de + 12 volts a +100mA y la otra de -12 volts a -100mA.

En la tabla 2.3 se da una lista completa de los componentes que integran el módulo de acondicionador de señales y puerto de detección.

REFERENCIA	DESCRIPCIÓN
U1, U2, U3, U4, U5	LM358; doble amplificador operacional
U6	4050; buffer no inversor
U7	MAX743, regulador de voltaje de doble salida
SW1, SW2, SW3	Switch rotativo de 2 polos, 12 tiros
POT1, POT2, POT3	Potenciómetro Trimpot, 50K ohms
RC1, RC2, RC11, RC12, RC21, RC22	Resistencia, 1K ohms
R1, R2, R3, RC3, RC4, RC7, RC10, RC13, RC14, RC15, RC18, RC23, RC24, RC25, RC28	Resistencia, 10K ohms
RC5, RC16, RC26	Resistencia, 27K ohms
RC6, RC17, RC27	Resistencia, 3.3K ohms
RC8, RC19, RC29	Resistencia, 22K ohms
RC9, RC20, RC30	Resistencia, 39K ohms
BNC1, BNC2, BNC3	Conector BNC hembra
D1, D2	1N5817, diodo Schotcky
LX+, LX-	Inductancia, 100uH
C5, C6	Capacitor, 0.01uF
C4	Capacitor, 1uF
C3	Capacitor, 10uF
C1, C7, C8	Capacitor, 100uF
CU4, C2	Capacitor, 0.1 uF

Tabla 2.3 Lista de componentes de la tarjeta del módulo acondicionador de señales y puerto de detección

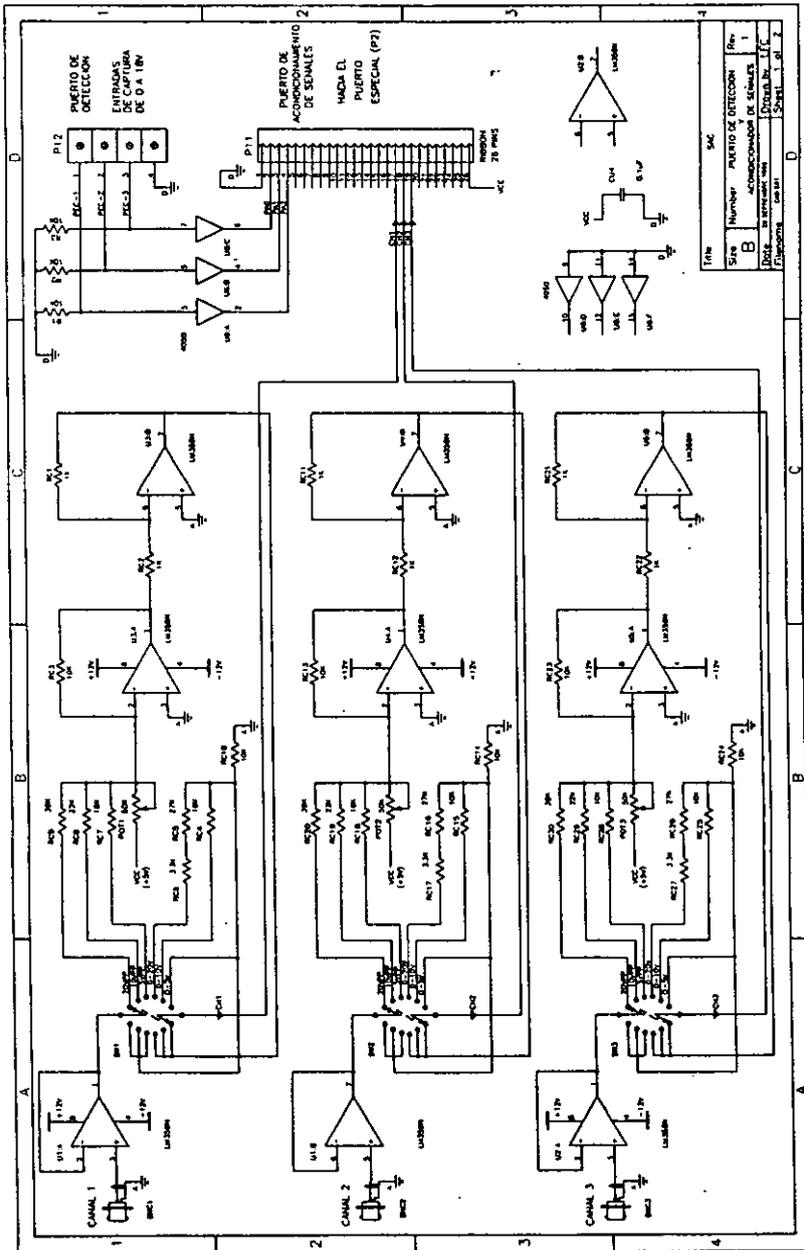
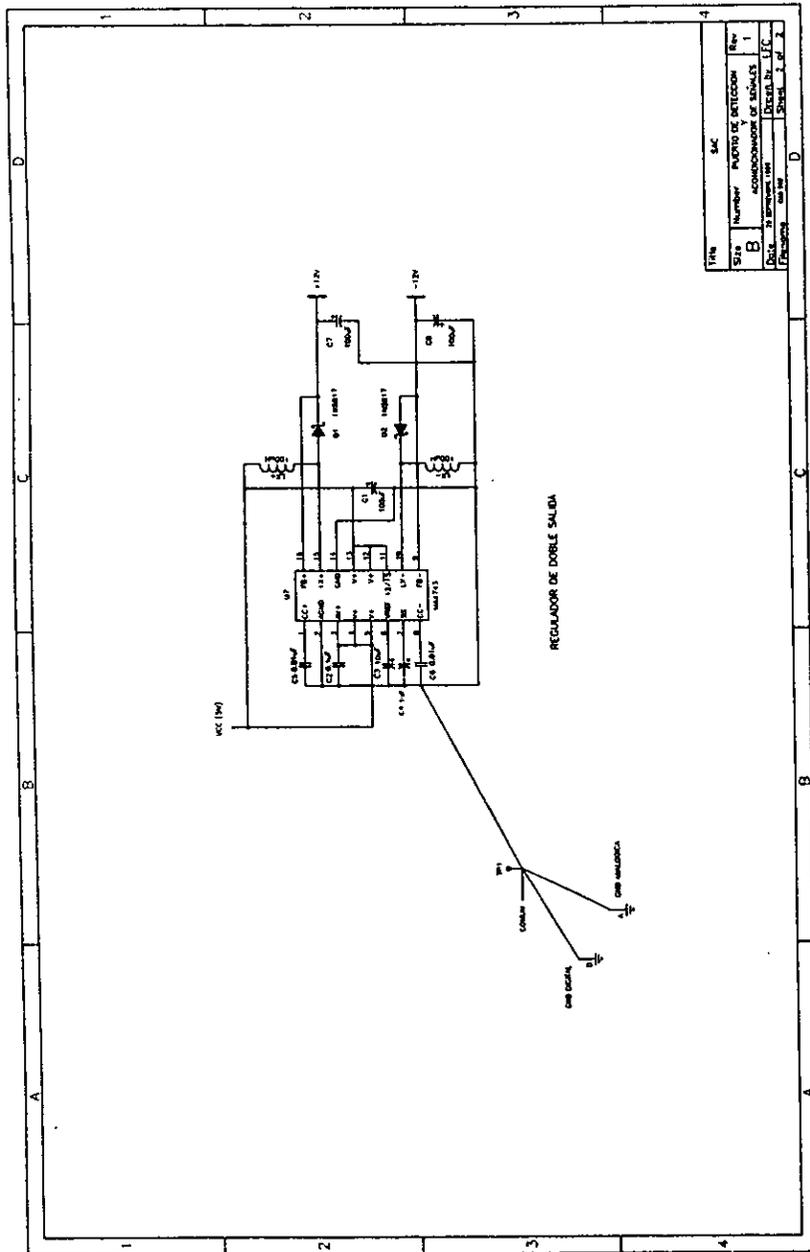


Figura 2.14. Acondicionador de señales y puerto de detección.



TITULO			
SAC			
Numero	Nombre	Fecha de Entrega	Rev.
B	COMANDO DE SEÑALES	1	1
Fecha de Entrega			
Fecha de Entrega			
Fecha de Entrega			

Figura 2.15. Regulador de doble salida de voltaje.

## 2.4 MÓDULO DE DESPLIEGUE Y TECLADO

El SAC cuenta con un módulo de despliegue y un teclado de control como se puede ver en el diagrama de bloques del SAC (ver figura 2.1).

### Módulo de despliegue

El SAC por medio del módulo de despliegue permite desplegar mensajes, avisos y valores de variables cuando el usuario por programa así lo desee. También, permite desplegar datos como la fecha y hora, el estado de la unidad de almacenamiento de datos (UAD-PCM CIA), el estado de un puerto de entrada, etc.

El dispositivo de visualización utilizado es un módulo AND491 (LCD1 figura 2.16). Este módulo tiene integrados un visualizador de cristal líquido de 2 líneas y 16 caracteres alfanuméricos por línea y un controlador que permite manejar fácilmente la información a desplegar.

El módulo AND491 se puede conectar directamente al bus de datos y direcciones de un microcontrolador siempre y cuando la velocidad del bus sea de 1 MHz. Puesto que, la velocidad del bus del microcontrolador HC11 es de 2 MHz, fue necesario realizar una interfaz especial por medio de 2 circuitos *flip-flop* tipo D 74HC374 (UTV2, UTV3, figura 2.16) los cuales permiten controlar y escribir datos al visualizador por medio de programa. En la figura 2.17 se muestra el ciclo de escritura del módulo de despliegue ya sea para enviar una instrucción o un dato.

En la tarjeta del módulo de despliegue y teclado de control también se colocaron 3 *leds* indicadores. El *led* verde es de estatus para indicar el estado de operación y los *leds* amarillo y rojo son de propósito general. Los tres *leds* pueden activarse por programa.

Una explicación más detallada sobre la forma de escribir mensajes y avisos, cómo desplegar variables en el visualizador y el funcionamiento de los *leds* indicadores se da en el capítulo 4.

### Teclado

El teclado tiene 15 teclas que se describen en la tabla 2.4.

Tecla	Descripción
"ON"	Tecla de encendido, con el cual se inicia el menú de diagnóstico
"←", "→"	Flechas de desplazamiento a la izquierda y a la derecha que permiten inicializar la fecha y hora, además de seleccionar el menú siguiente o el anterior
"C", "E"	Opciones de cancelar (C) o validar (E enter)
"0" a "9"	Permiten seleccionar cada uno de los menús de la rutina de diagnóstico e inicializar la fecha y hora

**Tabla 2.4 Descripción de las teclas**

Una explicación más detallada de los menús y opciones disponibles se da en los capítulos 3 y 4.



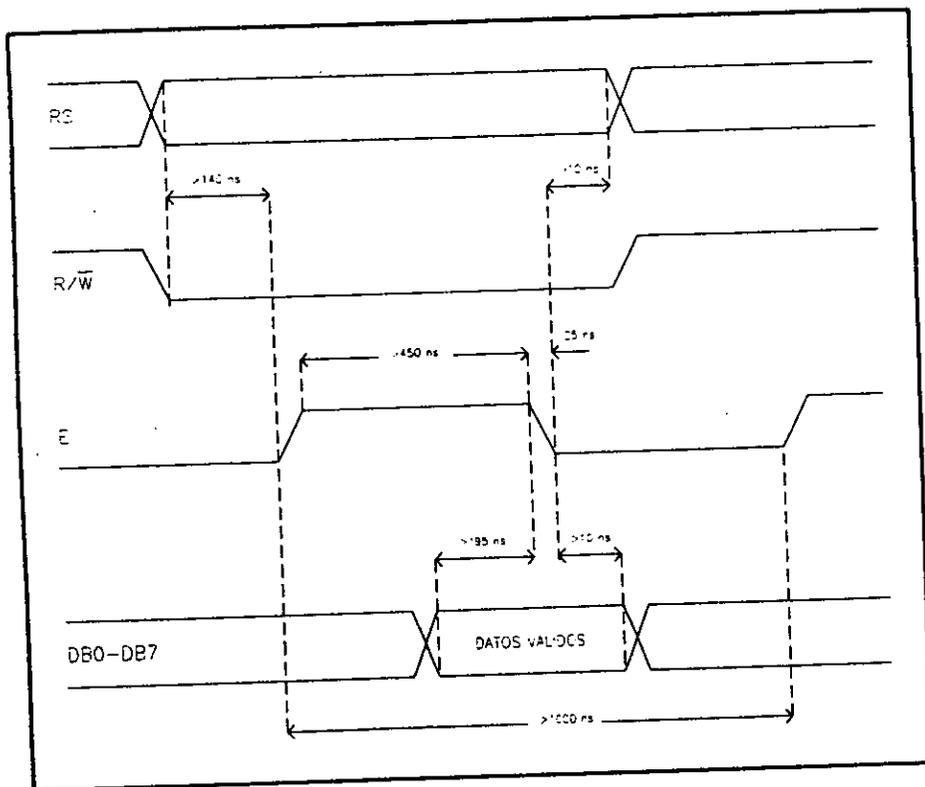


Figura 2.17. Ciclo de escritura del visualizador LCD1.

Una lista de los componentes que integran la tarjeta del módulo de despliegue y teclado se muestra en la tabla 2.5.

REFERENCIA	DESCRIPCIÓN
UTV1	74HC138; decodificador 3X8
UTV2,UTV3,UTV5	74HC374; <i>latch</i> octal
UTV4	74HC244; <i>buffer tri-state</i>
LCD1	AND-491, visualizador de cristal líquido de 2 líneas de 16 caracteres cada una.
"1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "→", "←", "C", "ON", "E"	Interruptores del teclado normalmente abiertos
Led1, led2, led3	<i>Leds</i>
RTV1-RTV8	Resistencia, 10K ohms
R1, R2, R3	Resistencia, 220 ohms
POTLCD	Potenciómetro <i>Trimpot</i> , 10K ohms
CUTV1-CUTV5	Capacitor, 0.1 uF

Tabla 2.5 Lista de componentes de la tarjeta del módulo de despliegue y teclado

## 2.5 PUERTOS DE ENTRADA Y DE SALIDA

El usuario tiene disponibles dos puertos de entrada (puertos de entrada 1 y 2) y dos puertos de salida (puerto de salida 1 y puerto de salida 2 con interfaz de potencia). La lectura y escritura de estos puertos se realiza por medio del programa de aplicación realizado por el usuario.

Los puertos de entrada 1 y de salida 1 se encuentran accesibles al usuario por medio del conector P4 de la tarjeta del sistema de control y procesamiento (ver figura 6.5 del capítulo 6).

SAC cuenta con una tarjeta que se conecta con la tarjeta del sistema de control y procesamiento por medio del cable 4 y que permite acceder al puerto de entrada 2 y al puerto de salida 2 con interfaz de potencia (ver figura 2.1 y figura 6.5 del capítulo 6). A continuación se describen cada uno de estos puertos.

### Puerto de entrada 1

El puerto de entrada 1 cuenta con 8 líneas de entrada digitales las cuales permiten voltajes de entrada entre 0 y 5 Volts, la lectura de este puerto se realiza por programa y el acceso al usuario es a través del conector P4 de la tarjeta del sistema de control y procesamiento (ver figura 6.5 del capítulo 6).

### Puerto de entrada 2

El puerto de entrada 2 cuenta con una entrada para conexión a tierra y 8 líneas de entrada digitales las cuales permiten voltajes de entrada entre 0 y 18 Volts DC gracias a un circuito *buffer* 4050 (ver figura 2.18). La lectura de este puerto se realiza por programa y el acceso al usuario es a través del conector P10 (ver figura 6.5 capítulo 6).

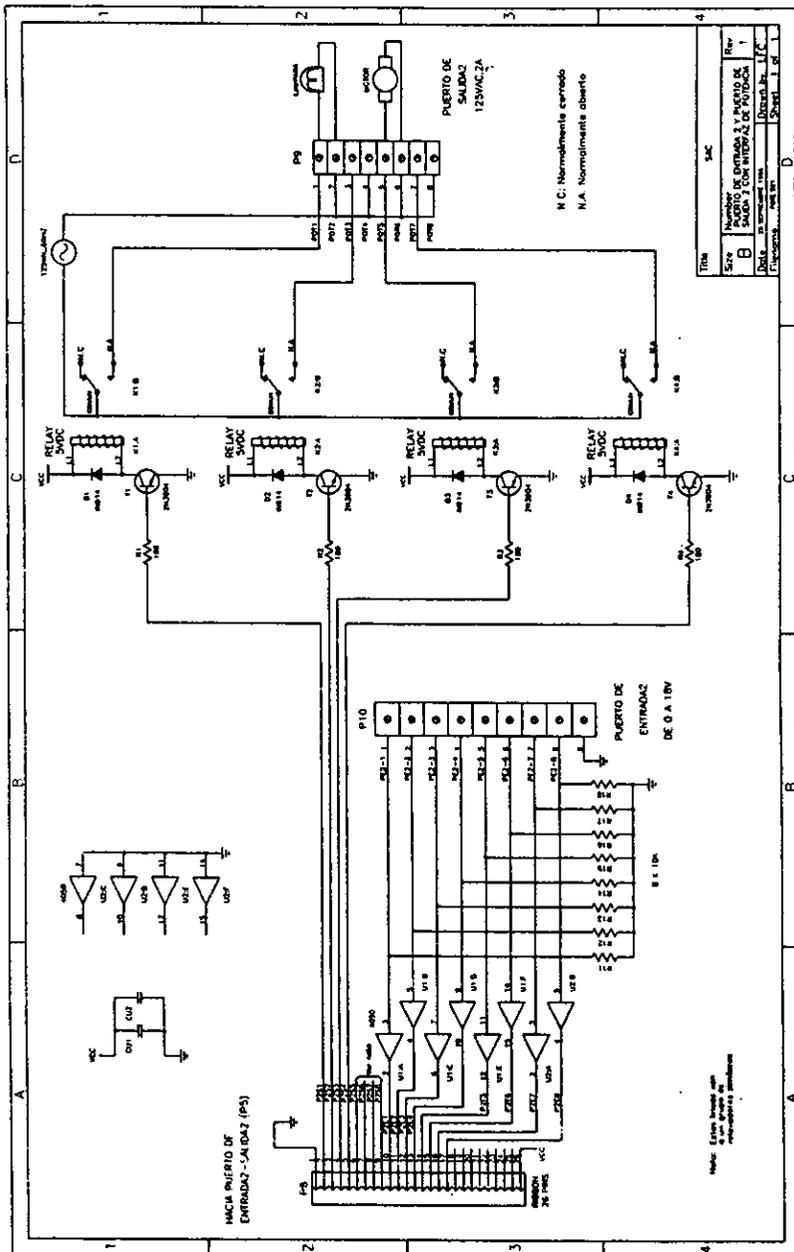


Figura 2.18. Puerto de entrada 2 y puerto de salida 2 con interfaz de potencia.

### Puerto de salida 1

El puerto de salida 1 cuenta con 8 líneas de entrada digitales las cuales permiten voltajes de salida entre 0 y 5 Volts, la escritura a este puerto se realiza por programa y el acceso al usuario es a través del conector P4 de la tarjeta del sistema de control y procesamiento (ver figura 6.5 del capítulo 6).

### Puerto de salida 2 con interfaz de potencia

El usuario también tiene a su disposición un puerto de salida 2, al cual se le agregó una interfaz de potencia constituida principalmente por unos transistores y unos relevadores de potencia de 5VCD a 125 VAC normalmente abiertos (ver figura 2.18). Los relevadores cuentan con tres terminales o contactos cada uno:

- N.A normalmente abierto
- N.C normalmente cerrado
- común.

Estos relevadores son capaces de manejar 2 amperes de corriente a 125 VAC.

Gracias a esta interfaz el usuario cuenta con 8 salidas que permiten manejar dispositivos externos de corriente alterna AC a 125 VAC y 2A de corriente máxima de carga por salida.

En la figura 2.18 se muestran 4 de las 8 salidas del puerto de salida 2.

La escritura a este puerto y la activación o desactivación de los dispositivos externos se realiza por medio del programa de aplicación realizado por el usuario, el acceso a este puerto se da a través del conector P9 (ver figura 6.5 del capítulo 6).

En la tabla 2.6 se muestra una lista de los componentes que integran la tarjeta del puerto de entrada 2 y puerto de salida 2 con interfaz de potencia.

REFERENCIA	DESCRIPCIÓN
U1, U2	4050; <i>buffer</i> no inversor
T1, T2, T3, T4	2N3904; transistor NPN de propósito general
K1, K2, K3, K4	DEG OUA-SS-1050; relevador de 5VCD a 125VAC/100VCD, 2 amperes
D1, D2, D3, D4	IN914; diodo
R1, R2, R3, R4	Resistencia, 180 ohms
R11-R18	Resistencia, 10k ohms
CU1, CU2	Capacitor, 0.1 uF

Tabla 2.6 Lista de componentes de la tarjeta del puerto de entrada 2 y puerto de salida 2 con interfaz de potencia

## 2.6 UNIDAD DE ALMACENAMIENTO DE DATOS UAD-PCMCIA<sup>1</sup>

El SAC, como sistema de adquisición, debe contar con un dispositivo mediante el cual los datos adquiridos por medio de los diferentes puertos de entrada, mensajes, datos de configuración o avisos se puedan almacenar de manera permanente en algún medio físico. Para ello el SAC cuenta con una unidad de almacenamiento de datos denominada UAD-PCMCIA.

La unidad de almacenamiento de datos UAD-PCMCIA se encuentra disponible en una tarjeta separada la cual se interconecta con la tarjeta del sistema de control y procesamiento a través del cable 3 (ver figura 6.5 del capítulo 6).

La UAD PCMCIA es una tarjeta de memoria PCMCIA del tipo I. Es una memoria RAM estática, no volátil, de 5 Volts, de muy bajo consumo, de 200ns de tiempo de acceso que cumple con todas las características del estándar para tarjetas de memoria PCMCIA (características eléctricas, mecánicas, de especificaciones de lectura y escritura etc.) y su información puede modificarse rápidamente tantas veces como se precise. Además, tiene la ventaja de que su contenido (el programa y/o los datos almacenados) no se pierden si falla el voltaje de alimentación, ya que trae incluida una batería de respaldo de litio que puede guardar la información por largos periodos de tiempo (hasta 9 años). Por lo tanto, la tarjeta con los datos e información almacenados se puede retirar en el campo y sustituirla por otra. De esta manera el usuario podrá tener acceso a la información recabada por el SAC para su procesamiento o revisión posterior en cualquier parte, por ejemplo en algún laboratorio u oficina, o en la estación misma.

El mapa de memoria de la UAD PCMCIA se muestra en la figura 2.8.

La capacidad actual de la UAD-PCMCIA es de un Mbyte, sin embargo, se puede incrementar hasta 16 Mbytes sin necesidad de hacer modificación alguna al *hardware*, únicamente se tiene que cambiar la tarjeta por una tarjeta de la capacidad deseada. Esta es una de las grandes ventajas de utilizar el estándar PCMCIA, ya que se pueden utilizar tarjetas de capacidad de memoria mayores (hasta 64 Megabytes) sin necesidad de cambiar de tipo de conector o el *hardware* asociado.

La tarjeta de memoria puede ser insertada en su conector especial o retirada del mismo aún cuando se encuentre presente el voltaje de alimentación, los datos almacenados en la memoria no sufrirán ninguna modificación. Además, cuenta con un sistema de protección contra escritura.

La UAD-PCMCIA cuenta con un gran número de comandos y funciones que ayudan al usuario a manejar una gran cantidad de datos fácilmente. Puede usarse para almacenamiento de datos, de mensajes o avisos, *data logging*, identificación del sistema etc.

Otra de las características importantes de la UAD-PCMCIA, es que en ella se puede depositar o guardar el programa del SAC. De esta manera, se pueden ejecutar distintos programas de operación del SAC en el campo sin necesidad de tener una computadora PC a la mano (ver capítulos 4 y 5).

---

<sup>1</sup> PCMCIA: Personal Computer Memory Card International Association



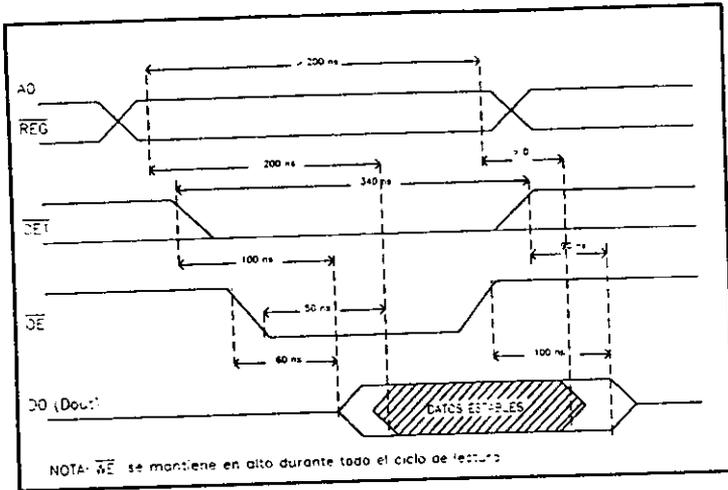


Figura 2.20. Ciclo de lectura de la UAD-PCMCIA.

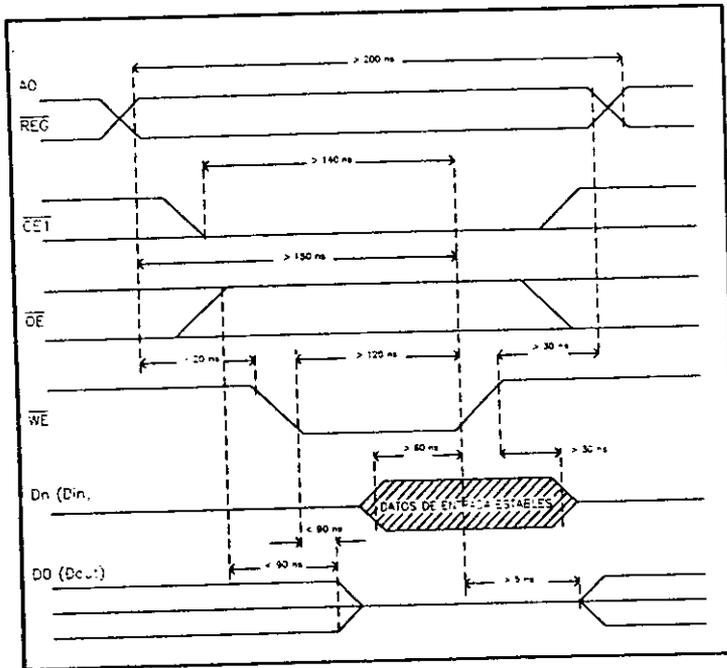


Figura 2.21. Ciclo de escritura de la UAD-PCMCIA.

Para poder manejar la unidad de almacenamiento de datos por medio del microcontrolador HC11 fue necesario realizar una interfaz entre la tarjeta de memoria y el microcontrolador. Esta interfaz se muestra en la figura 2.19.

Una lista del material utilizado para la tarjeta de la unidad de almacenamiento de datos UAD-PCMCIA se muestra en la tabla 2.7.

Los diagramas de los ciclos de lectura y escritura de la UAD-PCMCIA se muestran en las figuras 2.20 y 2.21.

REFERENCIA	DESCRIPCIÓN
UP5	SM9SRD1MP3200, tarjeta de memoria SRAM PCMCIA de 1 Megabyte y socket para tarjeta PCMCIA.
UP1, UP2, UP3	74HC374; <i>latch</i> octal
UP4	74HC4538; multivibrador monostable
RP1	Potenciómetro trimpot, 10k ohms
RP2, RP3	Resistencia, 10k ohms
CP1	Capacitor, 100 pF
CUP1-CUP4	Capacitor, 0.1 uF
JP2M, JP4M, JP8M, JP16M	Puentes o <i>jumpers</i> para expansión de la memoria hasta 16 MB

Tabla 2.7 Lista de componentes de la tarjeta de la unidad de almacenamiento de datos UAD-PCMCIA.

### III SISTEMA OPERATIVO DEL SAC

En este capítulo se describe el sistema operativo interno del SAC. Se explican tanto la estructura básica del sistema operativo, como los procedimientos de inicialización. Además, se presenta una breve descripción de los modos de operación del SAC.

#### 3.1 DESCRIPCIÓN

El sistema operativo es una de las partes fundamentales del SAC, ya que es el intermediario entre el *hardware* y la aplicación que el usuario desee implementar en el SAC por medio de un programa.

El sistema operativo es en sí, un programa escrito en lenguaje ensamblador y tiene una longitud de casi 2600 líneas: Se encuentra residente en la memoria EPROM (U10 figura 2.3) y ocupa 13.2 Kbytes. Gracias al sistema operativo, el microcontrolador puede almacenar y procesar la información, así como controlar los distintos dispositivos periféricos.

El sistema operativo está estructurado en cuatro rutinas principales que a continuación se describen brevemente. El diagrama de flujo general del sistema operativo se muestra en la figura 3.1.

- 1) Rutina I de Inicialización. En esta rutina se inicializan las diferentes variables, puertos, dispositivos externos y mensajes de inicio.
- 2) Rutina II o de recepción de un programa. El SAC se encuentra por *default* en espera del programa a ejecutar, pero una vez que se le envía por el puerto serie asíncrono un programa, esta rutina procede a recibirlo y guardarlo en la memoria RAM, para después ejecutarlo.
- 3) Rutina III o de ejecución del programa recibido. Esta rutina se encarga de ejecutar el programa que ha sido cargado en la memoria RAM; es decir, el control del flujo de programa se asigna al programa residente en la memoria RAM.
- 4) Rutina IV o de diagnóstico. Esta rutina se activa cuando se ha oprimido el botón de "ON" o de encendido del teclado y es básicamente una rutina de interrupción por medio de la cual el usuario puede hacer una revisión del estado actual del SAC, cargar un programa residente en la UAD-PCMCIA en la memoria de programa RAM, sincronizar el reloj fechador, leer puertos, borrar la memoria RAM y probar la UAD-PCMCIA, entre otros.

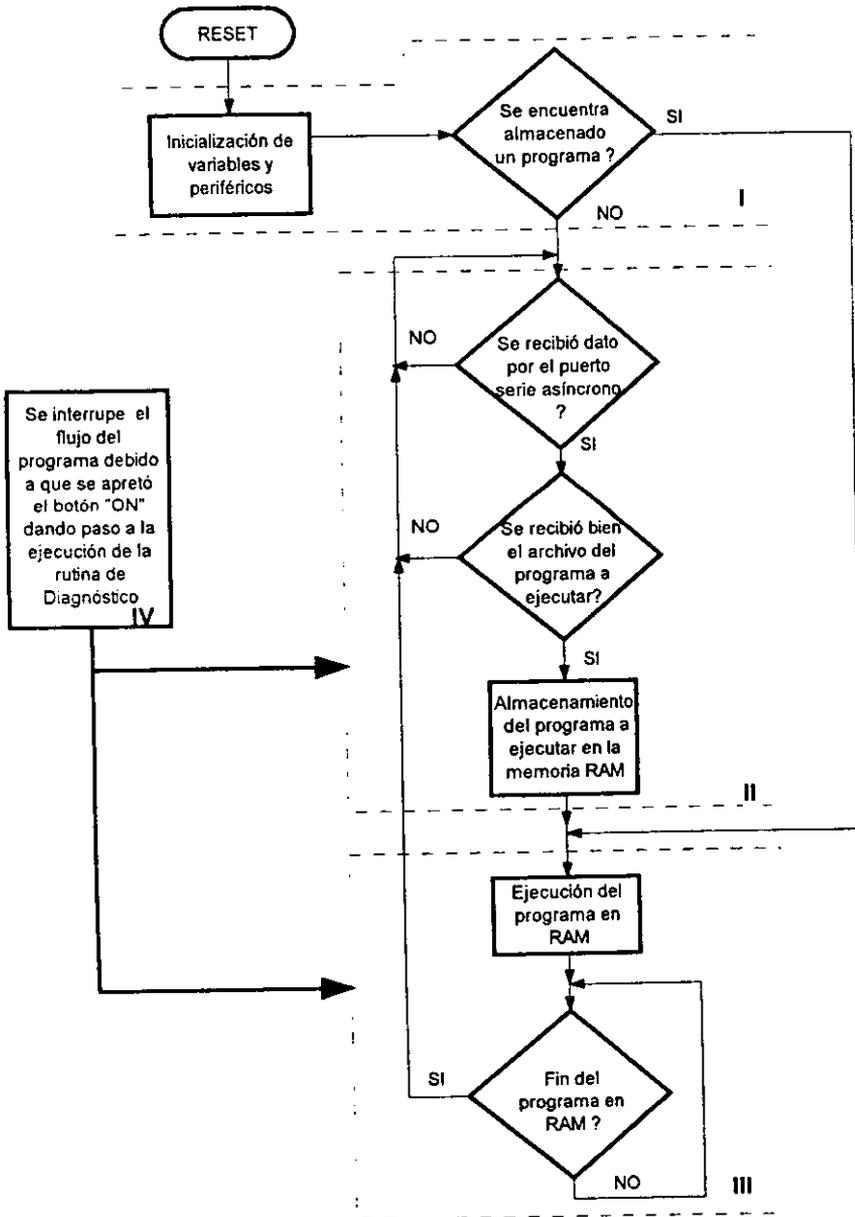


Figura 3.1 Diagrama de flujo general del sistema operativo de SAC

### 3.2 RUTINA I DE INICIALIZACIÓN

La rutina de inicialización se ejecuta después de que se ha encendido el SAC generando una señal de *reset* o cuando se ha generado la señal de *reset* por medio del circuito supervisor o de manera manual.

En esta rutina I, antes que nada, se inicializa el *stack pointer*. Luego, con el fin de adecuar el mapa de memoria del SAC, se mueven los registros de control y la memoria RAM interna del microcontrolador a la dirección \$B000. Después se ubica a la memoria RAM externa -la cual ocupa 32 Kbytes de la dirección \$0000 a la dirección \$7FFF- en la parte más baja del mapa de memoria. De esta manera, la memoria RAM externa se reserva para el almacenamiento del programa realizado por el usuario.

A continuación, se inicializa el puerto serie asíncrono con un formato de transmisión de 1 bit de inicio, 8 bits de datos, un bit de paro y sin paridad. Para el puerto serie asíncrono se decidió utilizar la máxima velocidad de transmisión para un cristal de 8 MHz y una frecuencia de reloj E de 2 MHz; es decir, 9600 bauds.

Enseguida se verifica si se encuentra almacenado un programa en la memoria RAM externa. Si es el caso, se procede a ejecutar el programa residente en memoria, es decir, el control del flujo del programa se direcciona a la RAM externa. Si no se encuentra almacenado un programa en la RAM, entonces se procede a ejecutar la rutina II o de recepción de un programa.

La manera de verificar si se encuentra almacenado un programa en la memoria RAM o no, se realiza de la siguiente manera:

Cuando se recibe un programa, ya sea por el puerto serie asíncrono o por la unidad de almacenamiento de datos UAD-PCM CIA y éste se almacena en la memoria RAM externa, entonces se escribe en un área de la RAM la etiqueta "RUN" (ejecutando) la cual indica que un programa se ha almacenado en la memoria RAM externa. Al terminar de ejecutarse el programa cargado, ya sea por el término del mismo o por el paro en su ejecución (se apretó la tecla ESC desde la PC) se borra la etiqueta de "RUN". De esta manera, se asegura la autonomía del sistema, ya que si falla la alimentación del SAC y después de un cierto tiempo la alimentación se vuelve a restablecer, el SAC seguirá operando normalmente, ejecutando desde el inicio el programa que el usuario realizó.

En la figura 3.2 se muestra el diagrama de flujo de la rutina I o de inicialización.

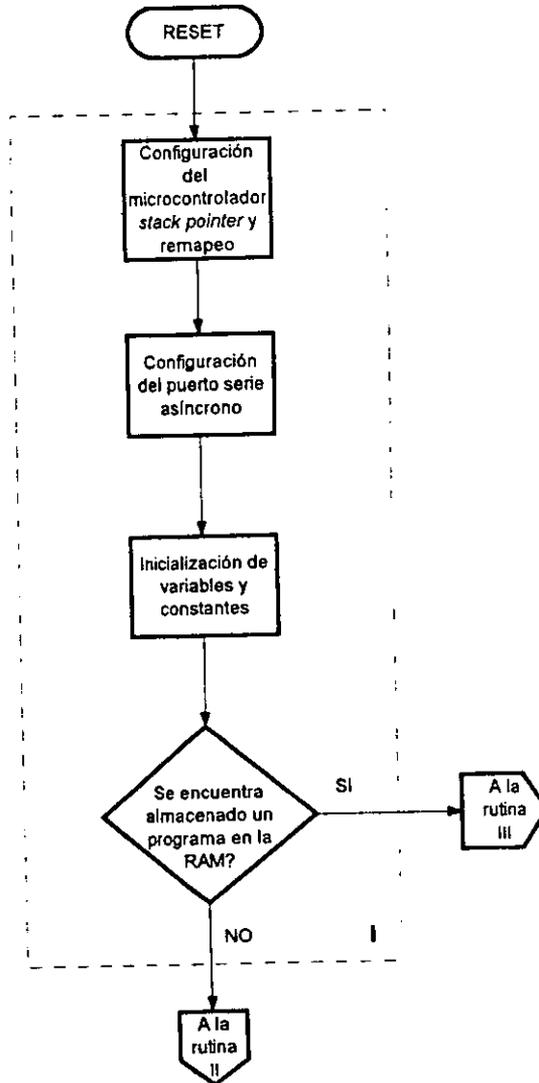


Figura 3.2. Diagrama de flujo de la rutina I de inicialización.

### 3.3 RUTINA II DE RECEPCIÓN DEL PROGRAMA

La rutina II de recepción de un programa, se ejecuta después de que la rutina I de inicialización ha terminado o cuando la rutina III de ejecución ha terminado de ejecutar el programa almacenado en la memoria RAM.

En la figura 3.3 se muestra un diagrama de flujo de la rutina II de recepción de un programa.

En esta rutina II, antes que nada, se manda un mensaje de presentación al puerto serie asíncrono y al módulo de despliegue LCD, indicando al usuario que el SAC esta listo para recibir el programa a ejecutar. Después, se lee constantemente el puerto serie asíncrono (modo de poleo) en espera de recibir algún dato.

Una vez que se recibe un dato por el puerto serie asíncrono, se verifica que los siguientes datos recibidos correspondan a un archivo en formato S19 y se procede a cargar los registros S1 y S9 (ver anexo C) enviados por la PC a la memoria RAM.

El sistema operativo únicamente puede recibir archivos con el programa a ejecutar en el formato S19, es decir, archivos formados por registros del tipo S1 y S9 (ver anexo C).

Cualquier dato antes del primer registro S1 se ignora. De allí en adelante, todos los registros deben ser del tipo S1 hasta que el registro S9 termina la transferencia de datos. La manera de verificar que el archivo recibido corresponda a un archivo S19 es la siguiente:

Primero se lee un dato del puerto serie asíncrono y se verifica que corresponda al carácter ASCII "S"; si no es así, se ignora y se espera el arribo de otro dato hasta que el carácter recibido sea "S".

Después que se recibió el carácter "S" se lee el siguiente y se compara tanto con el carácter "1" como con el carácter "9" para diferenciar los registros tipo S1 de los tipo S9.

#### a) Registros tipo S1

Si el segundo carácter leído corresponde a "1", entonces se lee el siguiente byte y se guarda en un contador; este byte indica el número de bytes que contiene todo el registro. Enseguida se leen los siguientes 2 bytes, los cuales indican la dirección en hexadecimal, donde los siguientes datos se almacenarán. Los siguientes pares de bytes, son los caracteres ASCII correspondientes a las instrucciones y datos del programa a ejecutar. El último byte del registro corresponde al byte de *checksum* del registro S1.

Los datos e instrucciones se almacenarán a partir de la dirección en hexadecimal leída en el paso anterior decrementando el contador cada vez que se almacene una instrucción. Los datos e instrucciones se van almacenando en la memoria RAM, verificando cada vez que los datos se hayan escrito bien y en una localidad válida dentro de la RAM; de otra manera se indica un error. Así, cuando el contador llega a cero, se verifica que no haya habido errores en el almacenamiento de los datos e instrucciones y se realiza un *checksum* de los datos almacenados (ver anexo C).

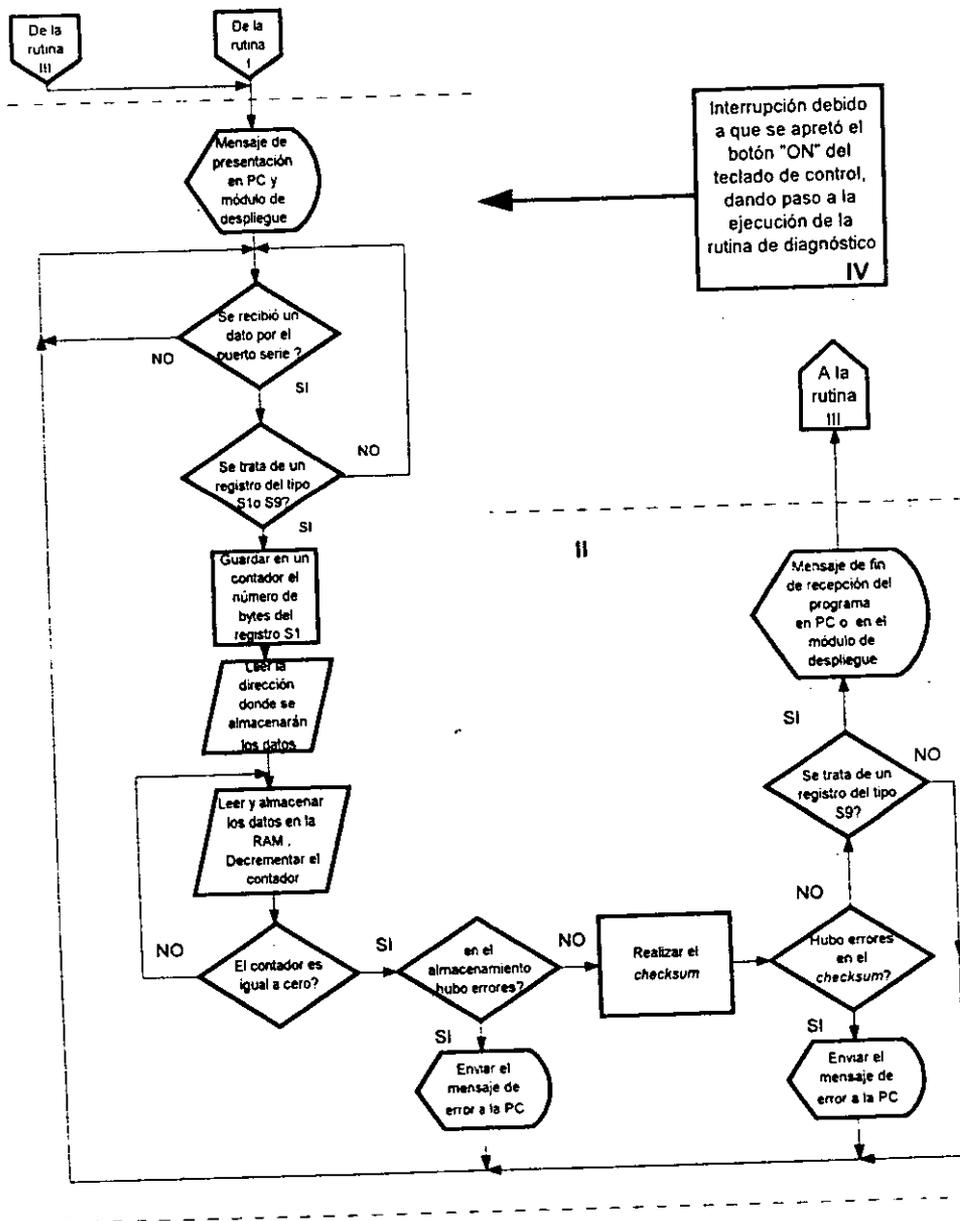


Figura 3.3. Diagrama de flujo de la rutina II de recepción de un programa.

### b) Registros tipo S9

Si el segundo carácter leído corresponde a "9", entonces se trata de un registro del tipo S9 el cual contiene 2 bytes de dirección y un byte de *checksum*. El registro S9 indica la terminación del bloque de registros S1, es decir, el fin del archivo del programa. Si no hubo errores de escritura o de *checksum* a lo largo de la recepción del archivo, entonces el archivo S19 con el programa a ejecutar por SAC fue bien recibido y se procede a ejecutar el programa.

## 3.4 RUTINA III DE EJECUCIÓN DEL PROGRAMA RECIBIDO

Una vez recibido y almacenado de manera exitosa el programa realizado por el usuario, se continúa con la rutina III, la cual se encarga de ejecutar el programa almacenado en la memoria RAM.

Dentro del programa realizado por el usuario siempre se incluye de manera automática y transparente para el usuario algunas instrucciones de configuración y habilitación de interrupciones, además de otras rutinas básicas. Es en el momento de la compilación que de manera automática (ver capítulo 4) se incluyen estas instrucciones y rutinas básicas sin que el usuario tenga nada que hacer para incluirlas. Por el momento se describirá la manera en que se ejecuta el programa almacenado en la memoria RAM.

Antes que nada, en el programa a ejecutar se definen los registros y variables del reloj fechador de tiempo real y del módulo de despliegue.

Después se inicializan los vectores de interrupción del puerto serie asincrónico y de la interrupción IRQ o *interrupt request*, definiendo las direcciones donde se encuentran localizadas las rutinas de las interrupciones.

Enseguida, se habilitan las interrupciones, se apagan (se ponen a nivel bajo 0 Volts) todas las salidas de todos los puertos de salida, y se escribe en un área de memoria de la RAM la etiqueta de "RUN" (ejecutando) con lo cual se le indica al sistema operativo que se tiene almacenado un programa en la memoria RAM: esto es muy importante ya que en caso de falla en la alimentación del SAC, el sistema operativo procederá a ejecutar el programa almacenado una vez que la alimentación se restablezca.

Después se procede a ejecutar el programa realizado por el usuario y cuando éste ha terminado, se borra la etiqueta de "RUN" (ejecutando) y se continúa con la rutina II, la cual se encuentra en espera de un nuevo programa.

En la figura 3.4 se muestra el diagrama de flujo de la rutina III de ejecución del programa recibido.

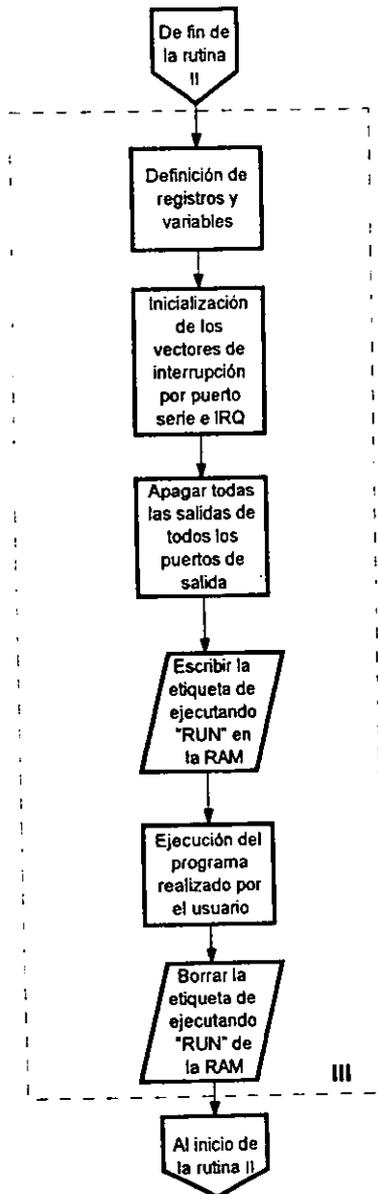


Figura 3.4 Diagrama de flujo de la rutina III de ejecución del programa recibido.

### 3.5 RUTINA IV DE DIAGNÓSTICO

La rutina IV de diagnóstico se ejecuta cuando se ha oprimido el botón "ON" del teclado y es una rutina de interrupción que permite al usuario leer información, modificar parámetros y activar algunas de las funciones disponibles con ayuda del teclado y del módulo de despliegue.

La rutina IV se ejecuta a partir de una interrupción del tipo mascarable, es decir, el microcontrolador por medio del *software* puede ignorarla. Para poder interrumpir al SAC y ejecutar la rutina de diagnóstico, el microcontrolador cuenta con una línea de entrada IRQ sensible a niveles bajos (0 Volts) para interrupciones mascarables. Cuando se oprime el botón "ON" del teclado, se presenta un nivel bajo en la entrada IRQ iniciando de esta manera la rutina de diagnóstico.

La forma de detectar cual de las teclas del teclado de control se ha oprimido, se realiza por *software* a través de una rutina que espera a que el usuario oprima una tecla, después la codifica, elimina los rebotes y guarda el resultado en una variable llamada "Tecla". Esta variable se utiliza para determinar qué menú, opción o dato se ha seleccionado.

En las figuras 3.5a, 3.5b, 3.5c, 3.5d se muestran de manera general los diagramas de flujo de la rutina IV de diagnóstico.

A continuación se describen de manera detallada cada uno de los menús y opciones que componen esta rutina.

Por medio del teclado de control y del módulo de despliegue se pueden seleccionar cada uno de los diez menús que integran la rutina de diagnóstico.

Dentro de la rutina de diagnóstico, lo primero que se realiza es presentar durante 2 segundos en el módulo de despliegue el mensaje de "\*DIAGNOSTICO\*" y el número asignado a la estación donde está ubicado el SAC "ESTACION n".

Enseguida se despliega el mensaje del menú 1 "1)FECHA Y HORA" y se espera a que el usuario oprima una tecla. Si se presiona la tecla "1", entonces se desplegará el valor actual de la fecha y hora en tiempo real durante 10 segundos y al término de este tiempo, se regresará al menú 1. Si la tecla oprimida corresponde a alguna de las teclas entre "0" y "9", entonces se desplegará el menú correspondiente a la tecla oprimida. Si la tecla oprimida es "→", entonces se desplegará el mensaje del menú siguiente, es decir, el menú 2.

Si el usuario seleccionó el menú 2, en el visualizador aparece el mensaje "2) CAMBIAR HORA" y se espera a que el usuario oprima una tecla. Si la tecla oprimida es "→", entonces se desplegará el mensaje del menú siguiente, es decir, el menú 3. Si se oprimió la tecla "2", en ese momento se desplegará un cursor y la fecha y hora en el formato siguiente: en el primer renglón *dd-mm-aa* y en el segundo renglón *hh:mm:ss.d* donde *dd* corresponde al día, *mm* al mes, *aa* al año, *hh* a la hora, *mm* a los minutos, *ss* a los segundos y *d* a las décimas de segundo.

Con ayuda de las teclas de desplazamiento a la izquierda "←", desplazamiento a la derecha "→" y las teclas numéricas "0" al "9", el usuario puede modificar fácilmente los valores de la fecha y hora del SAC y una vez que haya terminado de modificar los valores, al apretar la tecla "E" de validar (*enter*), en ese instante los valores de la fecha y hora modificados se transfieren al reloj fechador de tiempo real quedando actualizados de esta manera, la fecha y hora del SAC. Después de haber actualizado la fecha y hora se regresa al despliegue del mensaje del menú 2.

Si el usuario seleccionó el menú 3, en el visualizador aparece el mensaje "3) LEER PUERTO 1" y se espera a que el usuario oprima una tecla. Si la tecla oprimida es "→", entonces se desplegará el mensaje del menú siguiente, es decir, el menú 4. Si se oprimió la tecla "3", entonces se desplegará el valor actual de las entradas del puerto de entrada 1 con el formato siguiente: PUERTO1: XXXXXXXX, donde X puede ser igual a 0 o 1.

Si el valor de X es 0, entonces quiere decir que esa entrada tiene presente un voltaje de 0 Volts y si el valor de X es 1, entonces la entrada tiene presente un voltaje de 5 Volts. Una vez que se ha desplegado el valor actual de las entradas del puerto 1, se espera a que el usuario oprima una tecla. Si se aprieta la tecla "→", en ese instante se termina el despliegue del valor actual de las entradas del puerto 1 y regresa al despliegue del mensaje del menú 3, de otra forma se seguirá desplegando el valor actual de las entradas del puerto 1.

Para el caso en que se seleccione el menú 4, "4) LEER PUERTO 2" es exactamente igual que en el caso del menú 3 ("3) LEER PUERTO 1") sólo que los valores desplegados de las entradas corresponden al valor actual de las entradas del puerto de entrada 2.

Si el usuario seleccionó el menú 5, en el visualizador aparece el mensaje "5) PROBAR MEMORIA" y se espera a que el usuario oprima una tecla. Si la tecla oprimida es "→", entonces se desplegará el mensaje del menú siguiente, es decir, el menú 6. Si se oprimió la tecla "5", entonces se ejecutará una rutina que verifica si la memoria RAM está borrada (llena con el valor hexadecimal \$20) o si tiene un programa almacenado. En el caso de que exista un programa almacenado desplegará el mensaje "LA RAM TIENE UN PROGRAMA CARGADO" o bien "MEMORIA BORRADA" en el caso de que la memoria RAM esté vacía y se regresa al despliegue del mensaje del menú 5.

Si el usuario seleccionó el menú 6, en el visualizador aparece el mensaje "6) BORRAR MEMORIA" y se espera a que el usuario oprima una tecla. Si la tecla oprimida es "→", entonces se desplegará el mensaje del menú siguiente, es decir, el menú 7. Si se oprimió la tecla "6", entonces se desplegará el mensaje de "ESTA SEGURO?". Si el usuario oprime la tecla "E" de validar (*enter*), entonces se borra el programa contenido en la memoria RAM (se escribe el valor hexadecimal \$20 en todas las localidades de la RAM) y se despliega el mensaje "MEMORIA BORRADA". Si el usuario oprime la tecla "C" de cancelar, entonces no se borra el programa y se regresa al despliegue del mensaje del menú 6.

Si el usuario seleccionó el menú 7, en el visualizador aparece el mensaje "7) PROBAR UAD" y se espera a que el usuario oprima una tecla. Si la tecla oprimida es "→", entonces se desplegará el mensaje del menú siguiente, es decir, el menú 8. Si se oprimió la tecla "7", entonces se leerá cada byte de la unidad de almacenamiento de datos UAD-PCMCIA (de las localidades \$00000 a \$FFFFFF) y se compararán con el valor hexadecimal \$20 desplegando el mensaje "LA UAD TIENE DATOS EN MEMORIA", en el caso que cualquiera de los bytes leídos no sea igual al valor hexadecimal \$20, o desplegando el mensaje "UAD VACIA", en el caso de que todos los bytes leídos sean iguales al valor hexadecimal \$20.

Si el usuario seleccionó el menú 8, en el visualizador aparece el mensaje "8) BORRAR UAD" y se espera a que el usuario oprima una tecla. Si la tecla oprimida es "→", entonces se desplegará el mensaje del menú siguiente, es decir, el menú 9. Si se oprimió la tecla "8", entonces se desplegará el mensaje de "ESTA SEGURO?". Si el usuario oprime la tecla "E" de validar (*enter*), entonces se desplegará el mensaje "BORRANDO UAD...ESPERE POR FAVOR" y se borrarán los datos almacenados en la unidad de almacenamiento de datos UAD (se escribe el valor hexadecimal \$20 en todas las localidades de la UAD) y se despliega el mensaje "UAD VACIA". Si el usuario oprime la tecla "C" de cancelar, entonces no se borra el contenido de la UAD y se regresa al despliegue del mensaje del menú 8.

Si el usuario seleccionó el menú 9, en el visualizador aparece el mensaje "9) PROG EN PCMCIA" y se espera a que el usuario oprima una tecla. Si la tecla oprimida es "→", entonces se desplegará el mensaje del menú siguiente, es decir, el menú 0. Si se oprimió la tecla "9", entonces se despliega el mensaje "CARGANDO EL PROGRAMA" y se verifica que en la UAD-PCMCIA se tenga almacenado a partir de la localidad \$0000 un archivo en formato S19 con el programa a ejecutar por el SAC.

Si la UAD-PCMCIA efectivamente contiene un archivo en formato S19 con el programa a ejecutar, se leerá el programa y se intentará almacenarlo en la memoria RAM para después ejecutarlo. De lo contrario, si la UAD no tiene almacenado un programa a partir de la localidad \$0000, o bien, en el caso de que el programa no se hubiera podido almacenar correctamente en la memoria RAM, se desplegará el mensaje "NO SE PUDO CARGAR" durante 2 segundos y se regresará al despliegue del mensaje del menú 9.

Si el usuario seleccionó el menú 0, en el visualizador aparece el mensaje "0) SALIR" y se espera a que el usuario oprima una tecla. Si la tecla oprimida es "→", entonces se desplegará el mensaje del menú siguiente, es decir, el menú 1. Si se oprimió la tecla "0", entonces se desplegará el mensaje de "FIN DE REVISION?".

En el caso que el usuario oprima la tecla "C" de cancelar, entonces se continúa con la rutina IV de diagnóstico del SAC y se regresa al despliegue del mensaje del menú 1.

Si el usuario oprime la tecla "E" de validar (*enter*), entonces se despliega el mensaje de "FIN DEL DIAGNOSTICO" y se termina la rutina IV de diagnóstico de SAC, es decir, se termina con la interrupción.

En este caso, si antes de entrar a la rutina de diagnóstico el sistema operativo estaba ejecutando un programa, entonces se continúa con la ejecución del programa a partir de donde se interrumpió. En el caso de que no se hubiera estado ejecutando un programa en el momento de activar la rutina de diagnóstico, el sistema operativo continúa en espera de recibir un programa por el puerto serie asíncrono.

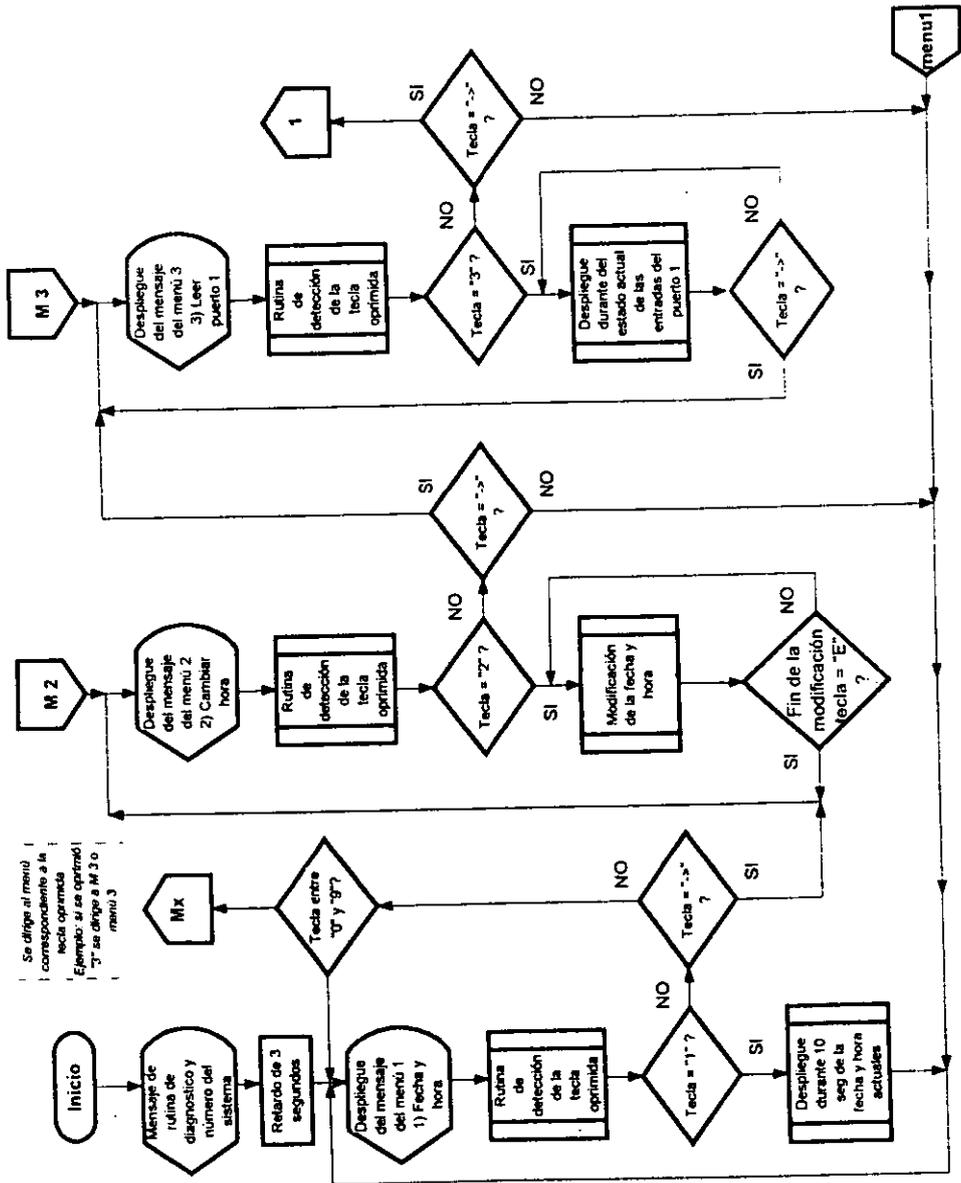


Figura 3.5a Diagrama de flujo general de la rutina IV de diagnóstico.

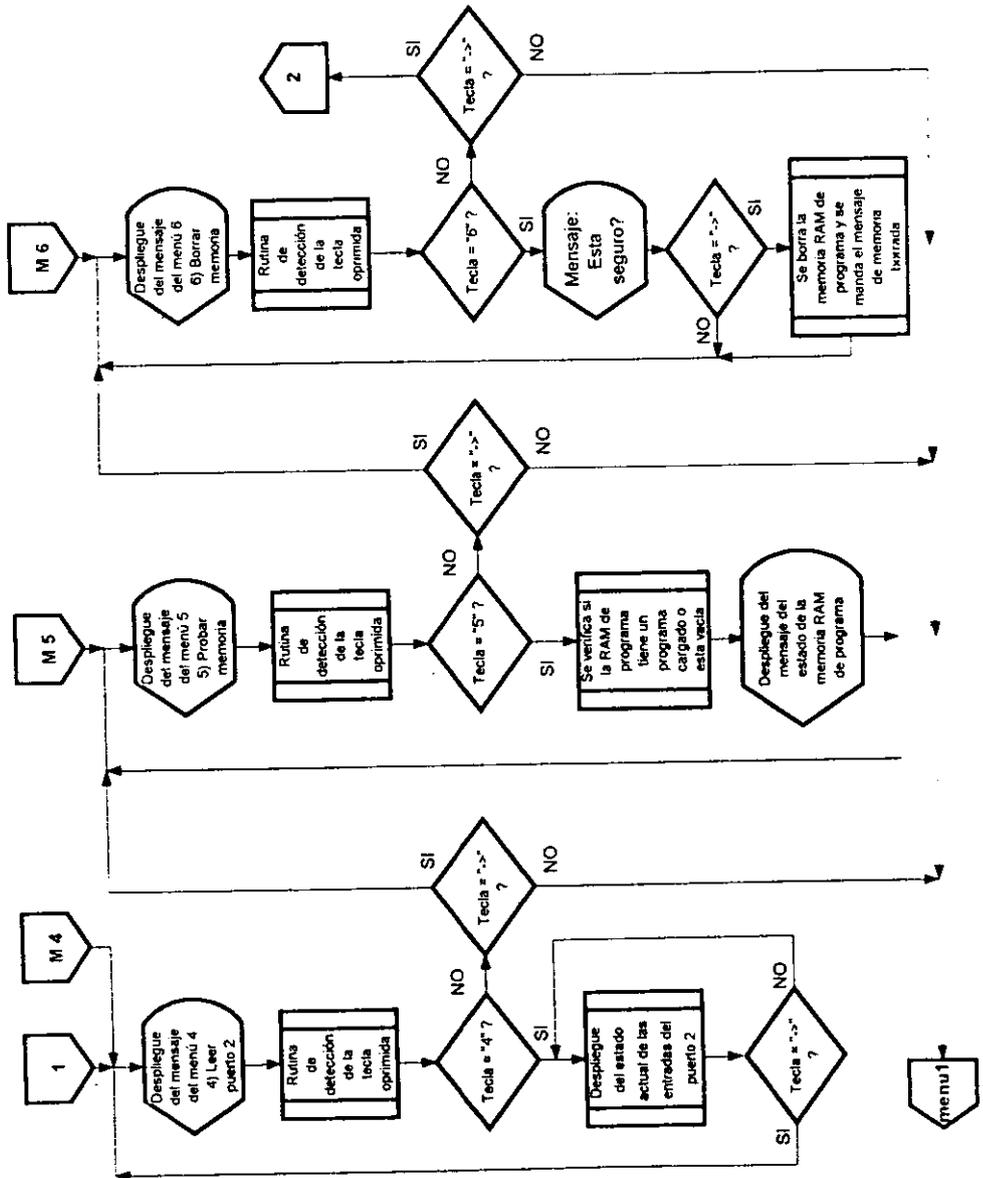


Figura 3.5b Diagrama de flujo general de la rutina IV de diagnóstico (continuación).

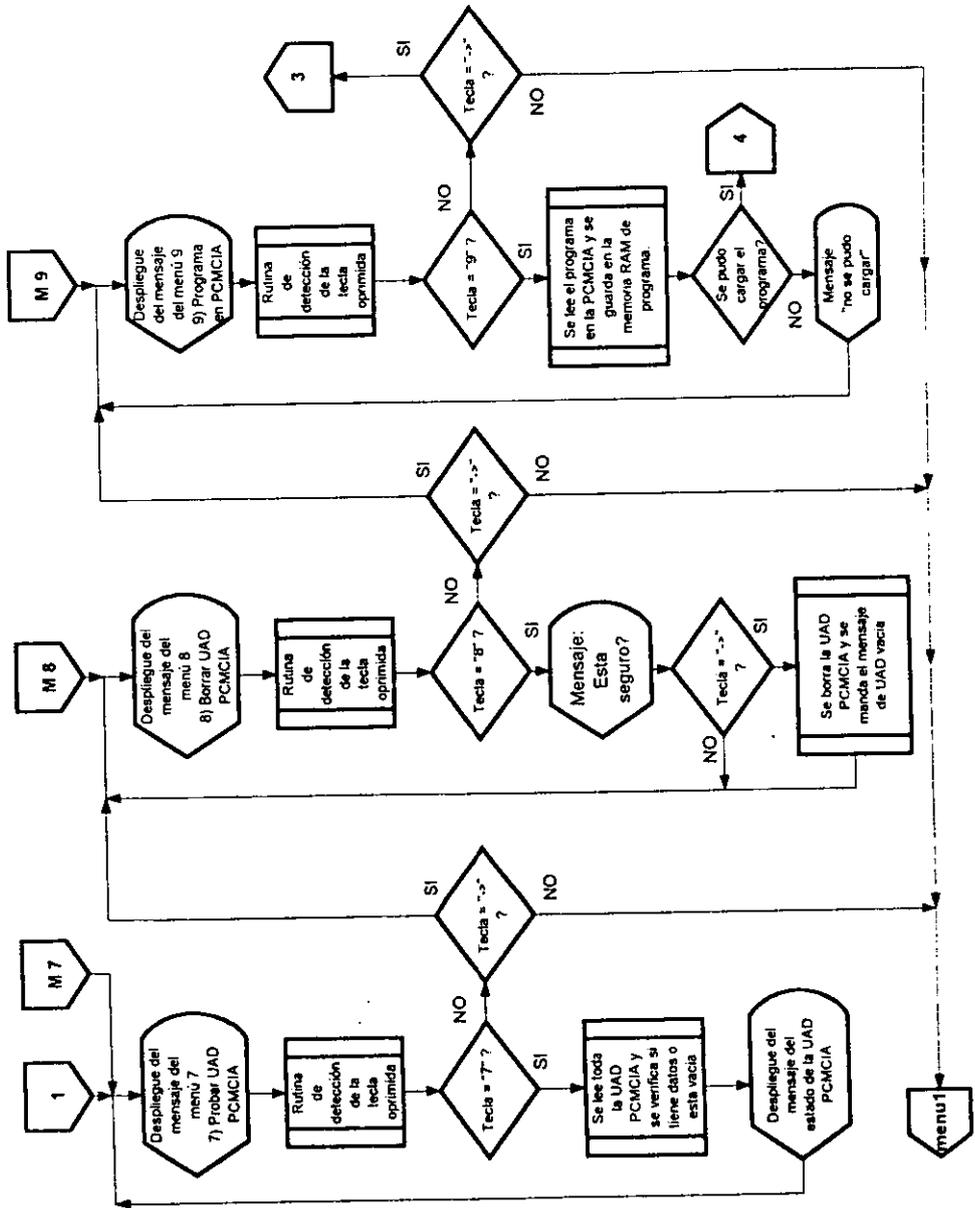


Figura 3.5c Diagrama de flujo general de la rutina IV de diagnóstico (continuación).

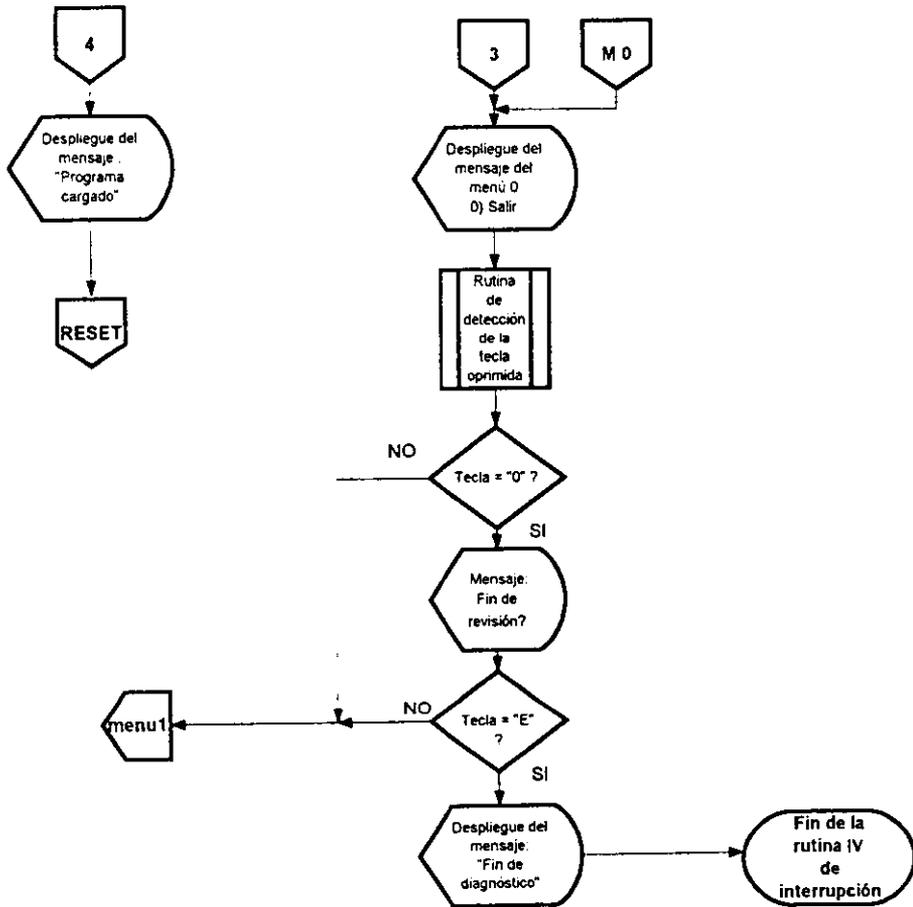


Figura 3.5d Diagrama de flujo general de la rutina IV de diagnóstico (continuación).

## IV OPERACIÓN Y MANEJO DEL SAC

En este capítulo se describirán las diferentes teclas del teclado, el despliegue de mensajes y datos en el visualizador, los modos de operación, el procedimiento de inicialización y funcionamiento del SAC entre otras funciones.

### 4.1 MÓDULO DE DESPLIEGUE Y TECLADO

El módulo de despliegue, el teclado y los *leds* indicadores se encuentran disponibles en una tarjeta prototipo que se conecta a la tarjeta del sistema de control y procesamiento (ver figura 6.5 del capítulo 6). En la figura 4.1 se muestra la disposición de los componentes de esta tarjeta.

#### Módulo de despliegue y leds indicadores

Para el despliegue de mensajes, avisos, variables y datos, el SAC cuenta con un visualizador alfanumérico de cristal líquido LCD de dos líneas de 16 caracteres por línea. El módulo de despliegue resulta muy útil cuando el SAC está operando en el modo de diagnóstico como se explicó anteriormente. Pero también puede desplegar mensajes y avisos indicados por el usuario, así como desplegar el contenido de algunas variables.

Por otra parte, el SAC cuenta con tres *leds* indicadores, uno de ellos es el llamado *led* de estatus que indica el estado actual del SAC y otros dos *leds* de propósito general, es decir, pueden ser programados para encenderse o apagarse durante el tiempo deseado. La descripción del estado de estos *leds* se presenta en la tabla 4.1.

LED/color	ENCENDIDO	APAGADO	PARPADEA	DESCRIPCIÓN
ESTATUS/verde	✓			El SAC está ejecutando un programa almacenado en la memoria RAM.
		✓		El SAC está en espera de un programa o en el modo de diagnóstico.
			✓	El SAC está recibiendo un programa por el puerto serie asíncrono o por la UAD-PCMCIA.
P.G./rojo	-	-	-	Leds de propósito general que pueden ser activados por programa.
P.G./amarillo	-	-	-	

Tabla 4.1. Descripción del estado de los leds indicadores.

### Teclado

En la tabla 2.4 del capítulo 2, se mostró una tabla descriptiva de las teclas del teclado. A continuación se describirán detalladamente las funciones de cada una ellas:



La tecla "ON" permite interrumpir al SAC y activar el modo de diagnóstico.



La tecla "C", de cancelar, permite anular la opción actual regresando a la opción anterior. También permite cancelar una opción en el caso de que el usuario no esté seguro de querer realizar alguna función.



La tecla "E", de validar (*enter*) en general sirve para aceptar o validar una opción y en el caso que el usuario esté en el menú 2 y haya modificado la fecha y hora, al oprimir esta tecla, el reloj fechador se sincronizará y quedará puesto a tiempo.



Esta tecla permite cambiar al menú siguiente. Resulta de gran utilidad cuando el usuario desea modificar la fecha y hora de SAC, ya que permite desplazar el cursor un lugar a la derecha seleccionando de esta manera el dato a modificar.



Cuando el usuario desea modificar la fecha y hora del SAC, esta tecla permite desplazar el cursor un lugar a la izquierda, seleccionando de esta manera el dato a modificar.



Las teclas numéricas del "0" al "9" permiten seleccionar el menú deseado y modificar la fecha y hora.



El botón de *reset*, aunque no forma parte del teclado de control por encontrarse ubicado en la tarjeta de control y procesamiento, es muy importante, ya que permite inicializar al SAC cuando se prende por primera vez o bien cuando por alguna razón se pierde el control del flujo del programa del sistema operativo.

Cabe mencionar que la detección, eliminación de rebotes e interpretación de las teclas oprimidas se hace bajo control de *software*.

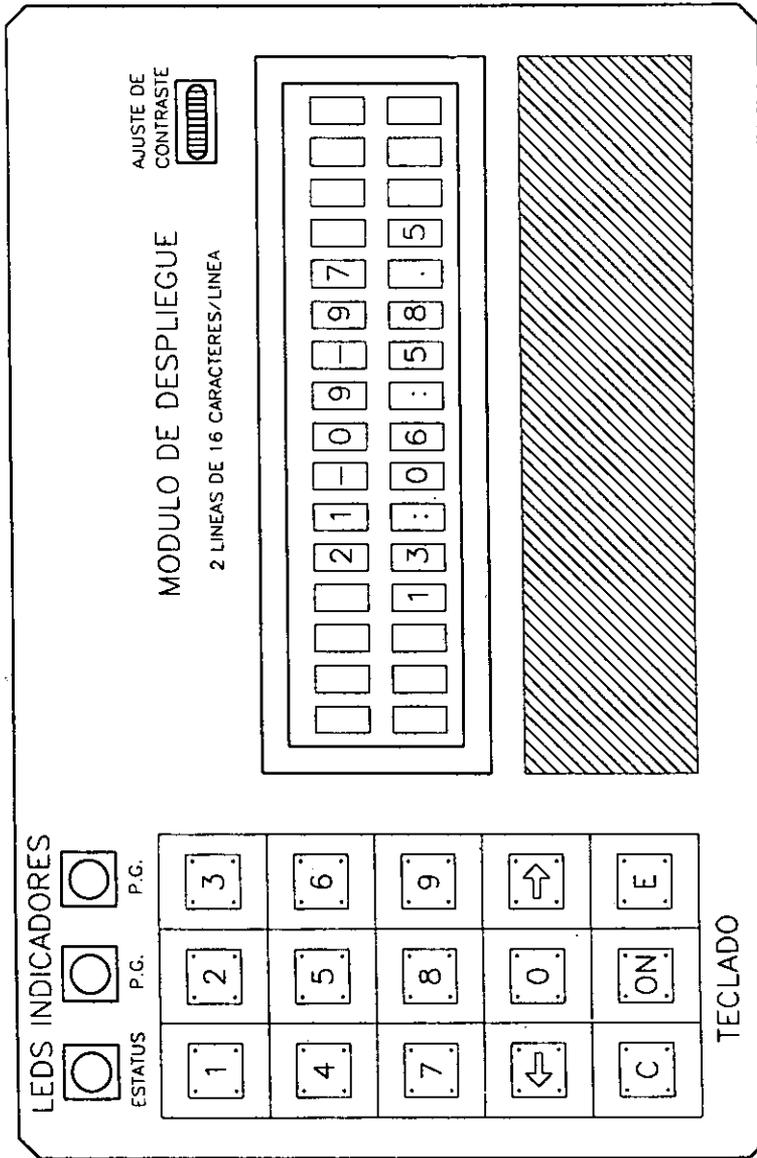


Figura 4.1. Módulo de despliegue, teclado y leds indicadores.

## 4.2 MODOS DE OPERACIÓN

Los modos de operación son cuatro:

- Modo de espera
- Modo de recepción del programa
- Modo de ejecución del programa
- Modo de diagnóstico

### Modo de espera

En este modo de operación el SAC se encuentra en espera de recibir por el puerto serie asincrono el programa de aplicación generado en una computadora personal, o bien, en espera de cargar un programa almacenado en la unidad de almacenamiento de datos UAD-PCMCIA para después ejecutarlo. Mientras el SAC se encuentra en el modo de espera, en el módulo de despliegue se presenta el mensaje de "\*\*SAC ver 1.0\*" y el *led* verde de estatus se encuentra apagado, indicando que el SAC está esperando que se le envíe el programa a ejecutar.

### Modo de recepción

Cuando el SAC se encuentra en el modo de recepción, el *led* verde de estatus parpadea indicando que el programa enviado a través de una PC o almacenado en la UAD-PCMCIA, se está recibiendo y almacenando en la memoria RAM. En el caso que el programa se haya almacenado exitosamente en la memoria RAM, el *led* verde ya no parpadeará más y se mantendrá encendido mientras el programa se esté ejecutando.

### Modo de ejecución

Una vez que el programa de aplicación ha sido almacenado de manera exitosa en la memoria RAM, se procede a ejecutarlo. Mientras un programa se está ejecutando, el *led* verde de estatus se mantiene encendido.

El usuario puede detener la ejecución de un programa oprimiendo la tecla ESC del teclado de la PC en el caso que se tenga conectada una computadora personal al puerto serie asincrono del SAC, o bien, entrando a la rutina de diagnóstico (oprimiendo la tecla "ON" del teclado de control) y seleccionando la opción 6 "6) BORRAR MEMORIA". Una vez que se ha terminado de ejecutar un programa o se ha detenido su ejecución el *led* verde de estatus se apaga indicando que se encuentra en el modo de espera.

### Modo de diagnóstico

Oprimiendo la tecla "ON" del teclado se interrumpe al SAC y se ejecuta la rutina de diagnóstico. Si el *led* verde de estatus se encontraba encendido indicando que el SAC estaba ejecutando un programa de aplicación, entrando al modo de diagnóstico el programa de aplicación se interrumpe y el *led* de estatus se apaga. Una vez que se ha terminado de realizar el diagnóstico, el programa de aplicación vuelve a ejecutarse a partir de donde se le interrumpió y el *led* de estatus se enciende.

La rutina de diagnóstico cuenta con diez menús que permiten verificar el estado de operación del SAC y realizar funciones como sincronizar el reloj fechador, borrar la memoria RAM y cargar en la memoria RAM un programa almacenado en la tarjeta de la UAD-PCMCIA, entre otras. La manera de realizar un diagnóstico del SAC se hace a través del teclado y el módulo de despliegue.

### 4.3 MANEJO DEL SAC

#### Puesta en operación

Cuando el SAC se ha encendido por primera vez, es recomendable oprimir el botón de *reset*. De esta manera automáticamente se ejecuta la rutina de inicialización la cual configura periféricos como el reloj fechador, el módulo de despliegue, el puerto de comunicación serie asíncrono, variables etc. Enseguida el SAC desplegará un mensaje de presentación y el *led* de estatus se mantendrá apagado indicando que el SAC se encuentra en modo de espera y listo para recibir el programa de aplicación a ejecutar.

#### Realización de un diagnóstico

Para verificar que el SAC y todos los módulos y periféricos asociados a él se encuentran en buen estado y listos para operar, se realiza un diagnóstico.

Oprimiendo la tecla "ON" del teclado de control, se entra al modo de diagnóstico y el *led* de estatus se apaga. Con ayuda del módulo de despliegue y el teclado el usuario podrá revisar y verificar el estado del SAC.

Se puede realizar un diagnóstico en cualquier momento siempre y cuando el SAC se encuentre en el modo de espera (*led* de estatus apagado) o en el modo de ejecución (*led* de estatus encendido). No se debe hacer un diagnóstico cuando el SAC se encuentre en el modo de recepción recibiendo un programa (*led* de estatus intermitente).

Cuando el SAC se encuentra en modo de espera, se puede realizar un diagnóstico y al término de éste, el SAC regresará automáticamente al modo de espera.

También, se puede realizar un diagnóstico cuando el SAC se encuentra en modo de ejecución (*led* de estatus encendido). En este caso, se interrumpirá la ejecución del programa de aplicación y se dará prioridad al diagnóstico (el *led* de estatus se apagará) y cuando se haya terminado de realizar el diagnóstico, se volverá a ejecutar el programa de aplicación a partir de donde se le interrumpió (el *led* de estatus se encenderá de nuevo)

Cuando el SAC se encuentra en modo de recepción, no se debe oprimir la tecla "ON" para entrar al modo de diagnóstico, ya que si se oprime, se le dará prioridad al diagnóstico pudiendo acceder a todos sus menús sin ningún problema. Sin embargo, el programa que se estuviese recibiendo quedará truncado y al término del diagnóstico el programa que se estaba recibiendo no se ejecutará y el SAC entrará automáticamente al modo de espera. Por lo tanto, **NO SE DEBE TRATAR DE REALIZAR UN DIAGNÓSTICO CUANDO EL SAC SE ENCUENTRA EN MODO DE RECEPCIÓN.**

#### a) Sincronización del reloj fechador

Para leer la fecha y hora actuales del SAC se debe acceder al menú 1 "1) FECHA Y HORA" con lo cual se desplegarán en tiempo real y durante 10 segundos los datos. De esta manera se puede verificar si el reloj fechador del SAC esta sincronizado o no. Para poner a tiempo la fecha y hora, estando en el modo de diagnóstico se debe acceder al menú 2 "2) CAMBIAR HORA". Enseguida se desplegarán la fecha y hora actuales y el cursor parpadeará en el dígito correspondiente a las decenas del día.

Cada uno de los dígitos indicativos de la fecha y hora se pueden modificar fácilmente desplazando el cursor una posición a la derecha o a la izquierda cada vez que se opriman las teclas "←" o "→" y oprimiendo la tecla numérica (teclas del "0" al "9") correspondiente al valor que se desea modificar.

La fecha y hora no quedarán actualizados sino hasta el momento en que el usuario oprima la tecla "E" de validar. En ese momento, se habrá puesto a tiempo la fecha y hora del SAC, y el reloj fechador quedará sincronizado y comenzará a operar normalmente. Con este procedimiento el reloj fechador se ajusta con una precisión de  $\pm 100$  ms.

Además de poder sincronizar el reloj fechador, el modo de diagnóstico nos permite verificar periféricos y realizar otras funciones. Los menús disponibles se muestran en las figuras 4.2a y 4.2b.

#### **Paro o interrupción de la ejecución de un programa**

La ejecución de un programa de aplicación puede interrumpirse principalmente por dos causas:

- El usuario.
- Fallas en la alimentación.

##### **a) Por el usuario**

Cuando el usuario ya no desea que el SAC siga ejecutando un programa de aplicación, puede parar la ejecución de tres maneras:

- 1) Puede hacerlo a través del teclado de control y el módulo despliegue, entrando al modo de diagnóstico, haciendo una revisión y al término de la revisión el programa de aplicación volverá a ejecutarse a partir de donde se le interrumpió.
- 2) Entrando al modo de diagnóstico y seleccionando el menú 6 "6) BORRAR MEMORIA" borrando de esta manera el programa de aplicación contenido en la memoria RAM. Al salir del diagnóstico el SAC entrará automáticamente en el modo de espera.
- 3) Por medio de una computadora conectada al puerto serie asincrónico del SAC y oprimiendo la tecla ESC (escape), parando así la ejecución del programa y forzando al SAC a entrar al modo de espera.

##### **b) Por fallas en la alimentación**

Cuando ocurre una perturbación eléctrica, por ejemplo un bajo nivel, una transición o falta del voltaje de alimentación, el programa de aplicación se interrumpe. Sin embargo, al restablecerse las condiciones normales de alimentación, el programa de aplicación automáticamente se vuelve a ejecutar desde el inicio y sin necesidad de que el usuario tenga que configurar algo o volver a enviarle al SAC el programa a ejecutar.

#### **4.4 MENÚS DISPONIBLES**

Cuando el SAC está operando en el modo de diagnóstico el usuario puede realizar diferentes funciones a través de los menús. En las figuras 4.2a y 4.2b se muestran los diferentes menús, las funciones disponibles, la forma de acceso a cada uno de ellos y las opciones con que cuenta el usuario para hacer un diagnóstico. En la sección 3.5 del capítulo 3 se describió detalladamente la rutina IV de diagnóstico.

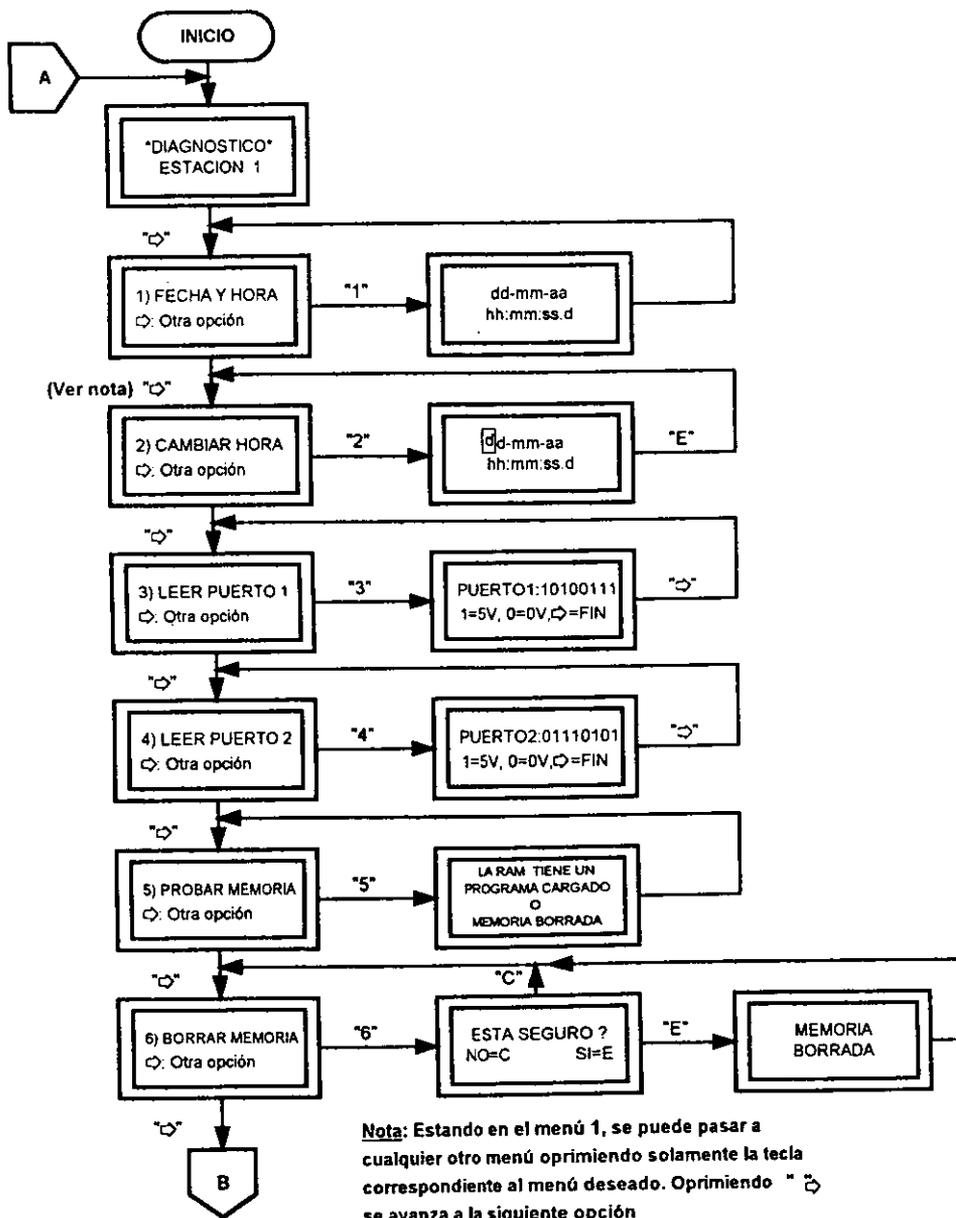


Figura 4.2a Menús disponibles

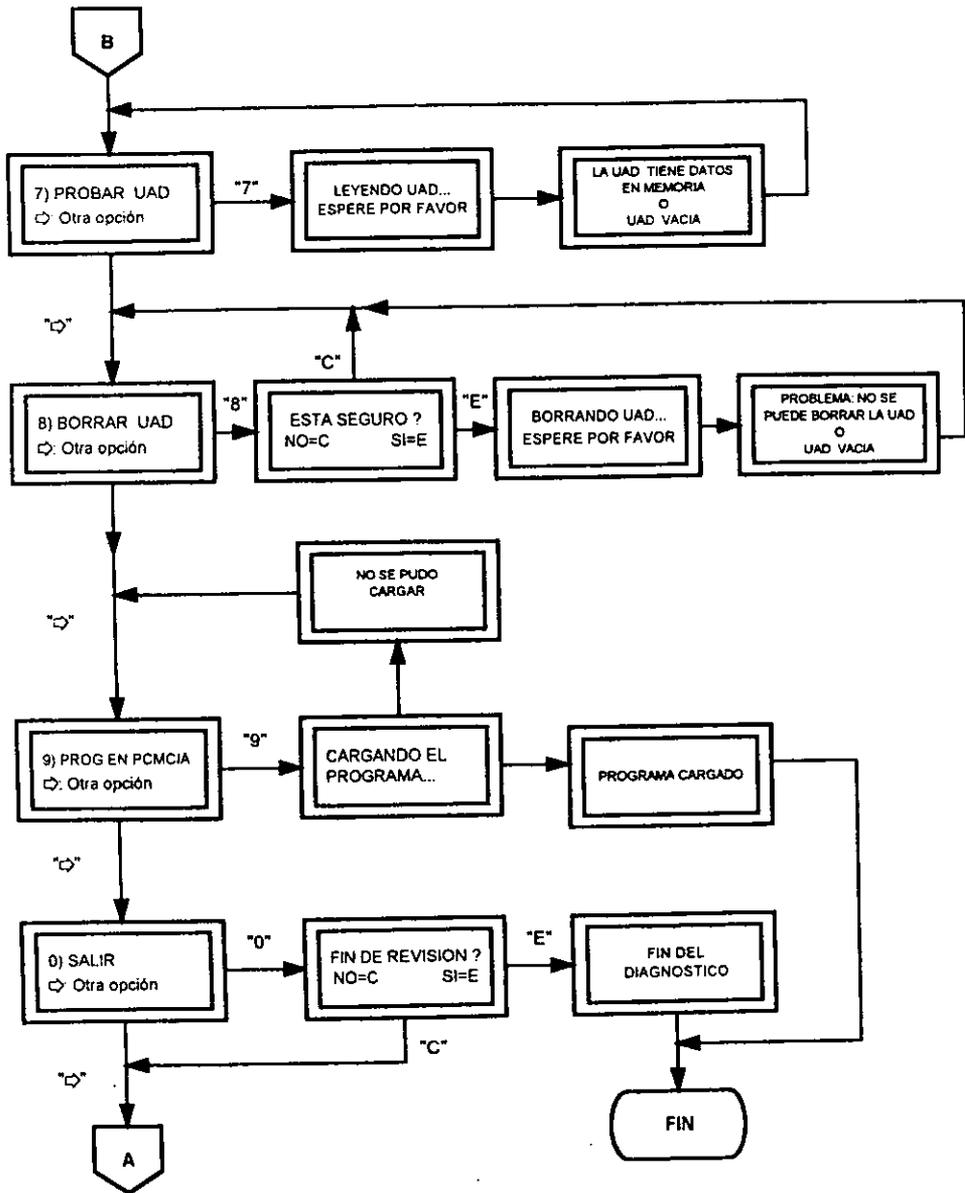


Figura 4.2b Menús disponibles (continuación)

## V GENERACIÓN DE PROGRAMAS DE APLICACIÓN DESDE UNA PC

### 5.1 Introducción

El SAC, como ya se ha explicado, es un sistema de adquisición y control de propósito general con lógica programable, cuya función o tarea a realizar está almacenada en la memoria RAM en forma de un programa de aplicación compuesto por instrucciones y sentencias de control de flujo. De acuerdo con estas instrucciones, el SAC permite leer los estados de los dispositivos y/o sensores conectados en los puertos de entrada, los combina entre sí, los procesa y entrega los resultados a los dispositivos conectados en los puertos de salida (ver figura 5.1).

El programa de aplicación que se desea ejecutar se presenta como una lista de instrucciones que se escriben en un lenguaje de programación sencillo y fácil de aprender denominado LENSAC. Las funciones complejas como temporizaciones y comparaciones, entre otras, se pueden representar fácilmente con este lenguaje.

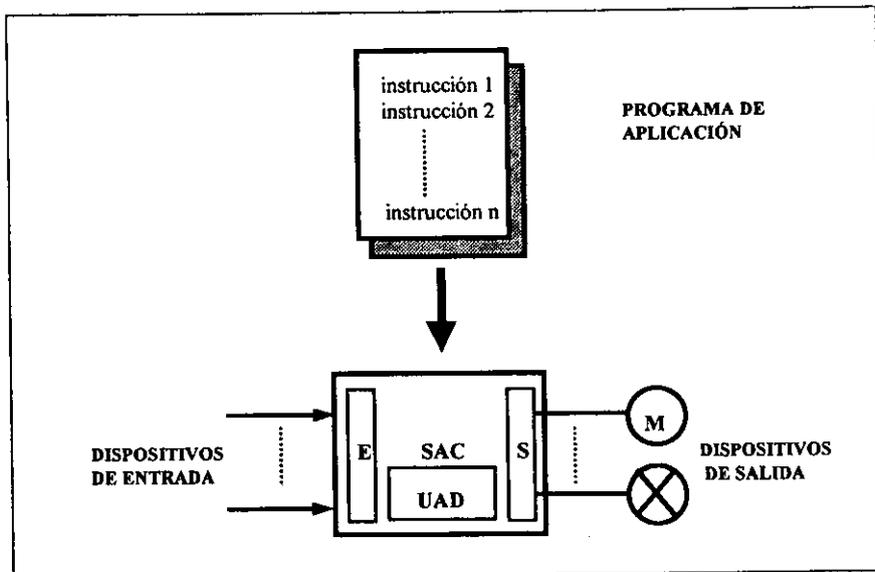


Figura 5.1. Programa de aplicación y SAC.

La generación del programa de aplicación es una tarea simple y fácil de realizar; basta introducir las instrucciones y sentencias de control de flujo necesarias. Seleccionando opciones dentro de los menús y con la ayuda de iconos es posible editar, modificar, borrar e intercalar instrucciones individuales o partes de programa. De este modo se simplifican y se acortan tanto la prueba, como la puesta en marcha del programa de aplicación.

## 5.2 Carga del programa de aplicación en el SAC

Por lo general, el programa de aplicación se escribe primero en la PC utilizando el lenguaje de programación LENSAC.

Después de escribir el programa de aplicación, éste se carga en el SAC en cualquiera de las dos formas siguientes:

### Programación con conexión al SAC ("in-line")

En este modo la computadora PC está unida directamente al SAC por medio de un cable para comunicación serie asíncrona como se ilustra en la figura 5.2.

El programa de aplicación se realiza en la PC, se edita, corrige y finalmente se transmite a la memoria interna RAM del SAC. La velocidad de transmisión es fija de 9600 bauds con un bit de inicio, 8 bits de datos, un bit de finalización y ninguna paridad.

### Programación sin conexión al SAC ("off-line")

Al igual que en el modo anterior el programa se realiza en la PC, luego se guarda en el disco duro o *diskette*. Utilizando el puerto o *drive* PCMCIA de la propia PC o una unidad similar externa (p. ej. CARDPRO<sup>1</sup> en el caso de una PC de escritorio) se transfiere el programa a una tarjeta de memoria RAM PCMCIA, es decir, a la UAD-PCMCIA.

A continuación la tarjeta con el programa almacenado se introduce en el *socket* para PCMCIA del SAC y se transfiere el programa a la memoria RAM interna del SAC. Para ello hay que seleccionar el menú 9 del modo de diagnóstico "9) PROG EN PCMCIA".

El proceso de la programación sin conexión al SAC ("off-line") se muestra en la figura 5.3.

Es muy importante que el programa de aplicación tenga como máximo una longitud de 32000 bytes y se almacene en la tarjeta de memoria UAD-PCMCIA a partir de la localidad hexadecimal \$0000, ocupando, de esta manera, desde la localidad \$100 hasta la localidad \$7D00 en la memoria interna RAM del SAC.

<sup>1</sup> El sistema CARDPRO se conecta al puerto paralelo de cualquier PC de escritorio y se maneja como un *floppy disk*, permitiendo leer y escribir datos a las tarjetas de memoria PCMCIA.

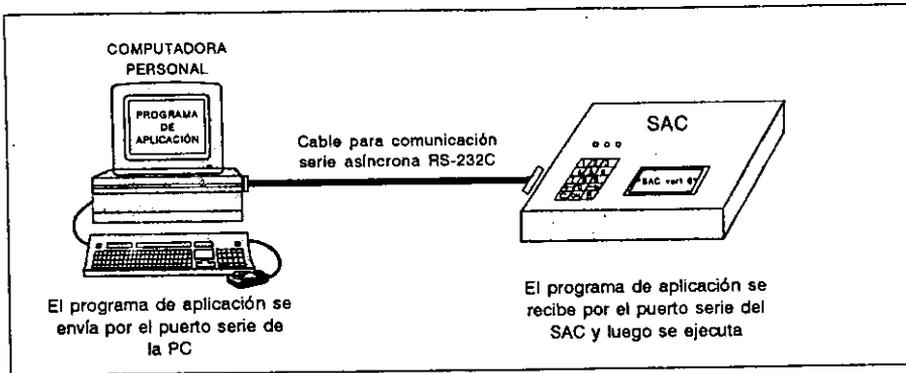


Figura 5.2 Programación con conexión al SAC ("in-line")

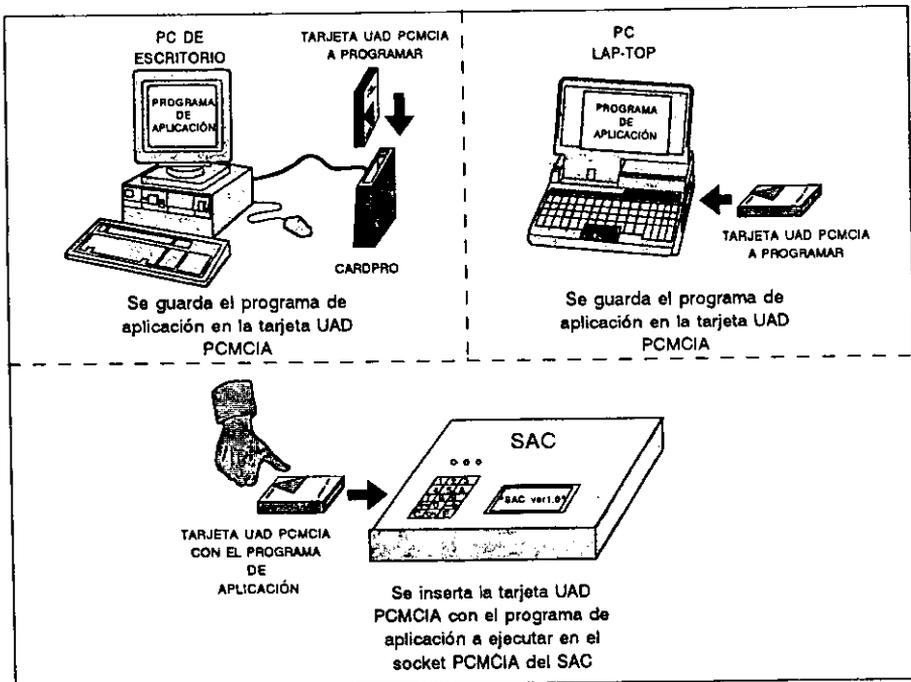


Figura 5.3 Programación sin conexión al SAC ("off-line")

### 5.3 Lenguaje de programación LENSAC

El lenguaje de programación denominado LENSAC se desarrolló con el fin de permitir al usuario generar programas de aplicación de una manera rápida y sencilla.

El lenguaje LENSAC cuenta con un amplio conjunto de instrucciones y sentencias de control enfocadas particularmente a la adquisición de datos y al control en tiempo real que el usuario puede utilizar sin conocimiento previo de lenguajes de programación o la necesidad de herramientas especiales.

#### 5.3.1 Formato de las instrucciones y sentencias de control del LENSAC

La lista de instrucciones representa la tarea a realizar. El formato de las instrucciones es el siguiente:

**INSTRUCCIÓN ( *parámetros* ) ENTER**

donde:

INSTRUCCIÓN	corresponde al nombre de la acción o sentencia de control a realizar y se indica con letras mayúsculas:
( ... )	encierra un campo de parámetros.
<i>parámetros</i>	denotan puertos, variables, operadores, direcciones etc; y se indican con letras minúsculas.
ENTER	tecla "↵" o "RETURN" del teclado de la PC.

#### Observaciones:

- 1) Dentro del campo de parámetros (...), cuando se trata de parámetros asociados a instrucciones, éstos van separados por comas. Cuando se trata de parámetros asociados a sentencias de control de flujo, éstos van separados por un espacio o tabulador.
- 2) Dependiendo del tipo de parámetro, los números pueden indicarse como enteros, hexadecimal o referirse a un bit de salida. Para definir números enteros sólo se escribe el número sin prefijos, por ej., 235. Para definir números hexadecimales se antepone el prefijo "hx" al número, por ej., hx1b56f. Para definir un bit de salida se antepone el prefijo "b" al bit de salida, por ej., b5.
- 3) Para fines de claridad en la programación, se establece que las instrucciones y sentencias de control se escriban con letras mayúsculas y los parámetros con letras minúsculas. Los operadores lógicos "Y", "O" siempre deberán escribirse con letras mayúsculas.
- 4) Un máximo de 255 caracteres por línea pueden ser escritos para instrucciones y sentencias de control.
- 5) Únicamente se permite escribir una instrucción o sentencia de control por línea.

## 5.3.2 Parámetros asociados a las instrucciones y sentencias de control

Los parámetros pueden ser puertos, variables y operadores, entre otros. Los parámetros se escriben con letras minúsculas y los operadores lógicos con letras mayúsculas. Su descripción se da en la tabla 5.1.

PARÁMETRO	DESCRIPCIÓN
<i>puerto</i>	Cualquier puerto de los siguientes: pts1, ptac, ptop, pte1, ptoe2.
<i>dato</i>	Cualquier valor hexadecimal entre hx00 y hxff. Cualquier bit o salida de b1 a b8. Cualquier valor entero de 0 a 255.
<i>var</i>	Cualquier variable de las siguientes: var1, var2, evc1, evc2, evc3, mues, mus1, mus2, mus3, mus4, dsem, dano, uano, dmes, umes, dda, udda, dhurs, uhurs, dmin, umin, dseg, useg.
<i>entrada</i>	Las entradas del puerto de detección son: 1, 2 o 3.
<i>transición</i>	Los tipos de transición son: 1 (flanco de subida), 2 (flanco de bajada), 3 (transición, es decir, flanco de subida o flanco de bajada).
<i>dirini</i>	Dirección de inicio. Cualquier valor hexadecimal entre hx00000 y hxffff.
<i>dirfin</i>	Dirección de fin. Cualquier valor hexadecimal entre hx00000 y hxffff.
<i>texto</i>	Una cadena de caracteres alfanuméricos.
<i>diasem</i>	Cualquier día de la semana del 1 al 7: 1 (Lunes), 2 (Martes), 3 (Miércoles), 4 (Jueves), 5 (Viernes), 6 (Sábado), 7 (Domingo).
<i>fecha</i>	El formato de la fecha es: dd/mm/aa, donde: dd va de 01 a 31, mm va de 01 a 12, aa va de 00 a 99.
<i>hora</i>	El formato de la hora es: hh:mm:ss, donde: hh va de 00 a 24, mm va de 00 a 59, ss va de 00 a 59.
<i>ren</i>	Las opciones son: 1 (indica el renglón de arriba o 1 del visualizador LCD), 2 (indica el renglón de abajo o 2 del visualizador LCD).
<i>pos</i>	La posición de los caracteres en el visualizador LCD las opciones son cualquier número entre 1 y 16.
<i>unidad</i>	Se refiere a las unidades de tiempo. Las unidades válidas son: ms (milisegundos), ds (décimas de segundo), seg (segundos), min (minutos), hrs (horas). En el caso del monitoreo de señales también se puede escoger: "sin" lo cual indica que el monitoreo se hará con el mínimo tiempo posible entre muestra y muestra.
<i>veces</i>	Indica el número de veces que se realizará una tarea. Los valores posibles están entre 1 y 30000.
<i>tiempo</i>	Indica el tiempo <u>en milisegundos</u> que se mantendrá apagada o encendida una salida cuando se utiliza la instrucción INTERMIT(...). Los valores posibles están entre 1 y 30000 ms.
<i>canal</i>	El sistema cuenta con 4 canales de entrada a los cuales se les pueden conectar cuatro señales analógicas diferentes. Los valores que puede tomar son: 1 (indica canal 1 o señal 1), 2 (canal 2 o señal 2), 3 (canal 3 o señal 3), 4 (canal 4 o señal 4).
REL	Indica un operador relacional válido como: = (igual a), != (diferente a), > (mayor que), < (menor que), <= (inferior o igual a), >= (superior o igual a).
LOG	Indica un operador lógico indicado en <u>letras mayúsculas</u> como : Y ,o bien, O .
<i>led</i>	Cualquier LED de los siguientes: led1 (led verde), led2 (led rojo), led3 (led amarillo).
<i>nombre</i>	Cadena de caracteres indicando el nombre de la etiqueta a la cual se va a direccionar el flujo del programa.

Tabla 5.1 Descripción de los parámetros asociados a las instrucciones y sentencias de control

### 5.3.3 Puertos, variables y operadores

Los puertos y variables son los objetos básicos que se utilizan en un programa de aplicación. Su definición y descripción se da en las tablas 5.2 y 5.3, respectivamente. Los operadores que soporta el LENSAC se dan en la tabla 5.4

PUERTO	DESCRIPCIÓN
pts1	Puerto de salida 1 con 8 salidas disponibles que manejan niveles de voltaje de salida de 0 y 5 VCD @ 35 mA por salida
ptac	Puerto de salida 2 con 8 salidas disponibles que permite manejar dispositivos de CA 120 VCA @ 2A.
ptop	Puerto paralelo de salida con 8 salidas con niveles de voltaje de salida de 0 y 5 VCD @ 35 mA por salida
pte1	Puerto de entrada 1 con 8 entradas disponibles que permiten manejar niveles de voltaje de entrada de 0 y 5 VCD compatibles con TTL
pte2	Puerto de entrada 2 con 8 entradas disponibles que permiten manejar niveles de voltaje de entrada entre 0 y 18 VCD.
PTOD	<p>Puerto de detección con 3 entradas disponibles que permiten manejar niveles de voltaje de entrada entre 0 y 18 VCD; pero que a excepción de los demás puertos, <u>nunca</u> se utilizará como parámetro.</p> <p><b>Nota importante:</b> la forma de hacer uso del puerto de detección es a través de la función DETECTRANS(1,1). Por lo tanto no se debe indicar PTOD como parámetro asociado a alguna instrucción, <u>ninguna instrucción acepta PTOD como parámetro.</u></p>

Tabla 5.2 Descripción de los puertos de entrada y salida

VARIABLE	DESCRIPCIÓN
var1, var2	Variables de 8 bits de propósito general
evc1, evc2, evc3	Variables correspondientes a la detección de eventos. Ejemplo: Si se detectó un evento en la entrada 1 del puerto de detección entonces evc1=1 si no evc1=0.
mues	Variable de 8 bits que contiene el valor actual de la muestra cuando se selecciona únicamente una señal de entrada.
mus1, mus2, mus3, mus4	Variables de 8 bits que contienen el valor actual de las muestras de cada uno de los cuatro canales.
dsem, dano, uano, dmes, umes, dda, udda, dhrs, uhrs, dmin, umin, dseg, useg	Variables de 8 bits c/u que indican la fecha y hora del sistema, Por ejemplo: PONRELOJ(1, 15/07/96 14:32:57) corresponde a : Dsem=1 (lunes), dda=1, udda=5, dmes=0, umes=7, dano=9, uano=6, dhrs=1, uhrs=4, dmin=3, umin=2, dseg=5, useg=7.

Tabla 5.3 Descripción de las variables

OPERADOR	DESCRIPCIÓN
=	Igual a
!=	Diferente a
>	Mayor que
<	Menor que
>=	Mayor o igual a
<=	Menor o igual a
Y	Conectivo lógico "Y" <sup>1</sup>
O	Conectivo lógico "O" <sup>1</sup>

Tabla 5.4 Descripción de los operadores relacionales y lógicos

<sup>1</sup> Los operadores lógicos "Y", "O" siempre deben escribirse en letra mayúscula.

### 5.3.4 Lista de instrucciones y sentencias de control

Todas las instrucciones y sentencias de control están enfocadas al control en tiempo real y a la adquisición de datos. Sus nombres dan una buena idea al usuario de cuál es la función que realizan.

Las instrucciones se dividen en 6 grupos:

- Instrucciones asociadas a los puertos de entrada y salida
- Instrucciones asociadas al almacenamiento de datos
- Instrucciones asociadas al reloj fechador
- Instrucciones asociadas al módulo de despliegue
- Instrucciones asociadas al monitoreo de señales
- Instrucciones diversas

Su definición, descripción y ejemplos se dan en las tablas 5.5 – 5.10.

INSTRUCCIONES ASOCIADAS A LOS PUERTOS DE ENTRADA Y SALIDA		
INSTRUCCIÓN	DESCRIPCIÓN	EJEMPLO
PRENDETS( )	Prende todas las salidas de todos los puertos de salida.	PRENDETS( )
APAGATS( )	Apaga (pone a 0 volts) todas las salidas de todos los puertos de salida.	APAGATS( )
PRENDELED( <i>led</i> )	Prende el led monitor deseado.	PRENDELED( <i>led</i> 1)
APAGALED( <i>led</i> )	Apaga el led monitor deseado.	APAGALED( <i>led</i> 3)
ESCRIBEPS( <i>puerto, dato ó var</i> )	Escribe un dato o variable en el puerto de salida seleccionado.	ESCRIBEPS( <i>pts</i> 1, <i>hxf</i> ) ESCRIBEPS( <i>pts</i> 1, <i>var</i> 1) ESCRIBEPS( <i>ptac</i> ,255)
PRENDESAL( <i>puerto, dato</i> )	Prende algunas de las salidas del puerto de salida seleccionado dejando prendidas o apagadas las salidas que ya estaban activas.	PRENDESAL( <i>ptac</i> , <i>hx</i> 5d) PRENDESAL( <i>pts</i> 1, <i>b</i> 2) PRENDESAL( <i>ptac</i> , <i>var</i> 2) PRENDESAL( <i>pts</i> 1, <i>l</i> 23)
APAGASAL( <i>puerto, dato</i> )	Apaga algunas de las salidas del puerto de salida seleccionado dejando prendidas o apagadas las salidas que ya estaban activas.	APAGASAL( <i>ptop</i> , <i>hx</i> 3b) APAGASAL( <i>pts</i> 1, <i>b</i> 6) APAGASAL( <i>pts</i> 1, <i>var</i> 1) APAGASAL( <i>ptac</i> ,67)
LEEPE( <i>puerto, var</i> )	Lee el dato del puerto de entrada seleccionado y guarda el dato en la variable deseada.	LEEPE( <i>pte</i> 1, <i>var</i> 1) LEEPE( <i>PTE</i> 2, <i>var</i> 2)
DETECTRANS( <i>entrada, transición</i> )	Detecta el tipo de transición deseada en una de las entradas del puerto de detección.	DETECTRANS(1,2) DETECTRANS(2,3)
TXSER( <i>dato</i> )	Envía un dato o variable al puerto serie asíncrono. (1)	TXSER( <i>hx</i> 76) TXSER( <i>var</i> 1)
RXSER( <i>var</i> )	Guarda un dato recibido por el puerto serie en la variable deseada. (1)	RXSER( <i>var</i> 1) RXSER( <i>var</i> 2)

Tabla 5.5.

INSTRUCCIONES ASOCIADAS AL ALMACENAMIENTO DE DATOS		
INSTRUCCIÓN	DESCRIPCIÓN	EJEMPLO
BORRAUAD( )	Borra la UAD PCMCIA.	BORRAUAD( )
TEXTOUAD( <i>dirini, texto</i> )	Escribe hasta 240 caracteres en la UAD PCMCIA a partir de la dirección de inicio.	TEXTOUAD(hx0000, !hola!) TEXTOUAD(hx1fb, Aquí es..)
LEEUAD( <i>dirini, dato</i> )	Lee y envía por el puerto serie asincrono hasta 240 caracteres de la UAD PCMCIA a partir de la dirección de inicio especificada. Nota: El dato deberá darse como número entero. (1)	LEEUAD(hxa0000,157) LEEUAD(hxc0000,240) LEEUAD(hxda237f,55)
LEEBLOQUAD( <i>dirini, dirfin</i> )	Lee y envía por el puerto serie asincrono a partir de la dirección de inicio y hasta la dirección de fin un bloque de datos de la UAD PCMCIA. (1)	LEEBLOQUAD(hxa0000, hxFFFF) LEEBLOQUAD(hxAB000, hxAB123) LEEBLOQUAD(hxAC100, hxAC67F)
TRANSUAD( )	Envía por el puerto serie asincrono toda la información contenida en la tarjeta de la UAD-PCMCIA. (1)	TRANSUAD( )
ESCRIBEUAD( <i>dirini, dato</i> )	Escribe un dato o el valor de una variable en la UAD PCMCIA en la dirección deseada.	ESCRIBEUAD(hxa0000,hx4f) ESCRIBEUAD(hxa123d,var2)
UADAVAR( <i>dirini, var</i> )	Lee un dato de la UAD PCMCIA en la dirección especificada y lo guarda en una variable.	UADAVAR(0xFF0E0,var1) UADAVAR(0xF1230,var2)

Tabla 5.6.

INSTRUCCIONES ASOCIADAS AL RELOJ PECHADOR		
INSTRUCCIÓN	DESCRIPCIÓN	EJEMPLO
PONRELOJ( <i>diasem, fecha, hora</i> )	Inicializa el día de la semana, la fecha y la hora del sistema.	PONRELOJ(2, 24/01/97, 13:55:30)
LEERELOJ( )	Lee la fecha y hora del sistema y las envía por el puerto serie asincrono con el formato siguiente: <aa/mm/dd, día, hh:mm:ss.dseg>	LEERELOJ( )
RELOJAUAD( <i>dirini</i> )	Escribe la fecha y hora del sistema en la UAD a partir de la dirección especificada. El formato de almacenamiento ocupa 25 bytes y es el siguiente: <aa/mm/dd, día, hh:mm:ss.dseg>	RELOJAUAD(hxb0000) RELOJAUAD(hx12345) RELOJAUAD(hxd6c5f)

Tabla 5.7.

INSTRUCCIONES ASOCIADAS AL MÓDULO DE DESPLIEGUE		
INSTRUCCIÓN	DESCRIPCIÓN	EJEMPLO
BORRAVIS( )	Borra la pantalla del visualizador de cristal líquido LCD.	BORRAVIS( )
ESCRIBEVIS( <i>ren, pos, texto</i> )	Escribe en el visualizador LCD un texto de hasta 16 caracteres en el renglón y posición deseada.	ESCRIBEVIS(1,8,!Hola!) ESCRIBEVIS(1,5.Como estas?)
VARAVIS( <i>var</i> )	Despliega en el visualizador LCD el valor actual de var1 o var2 en binario y hexadecimal.	VARAVIS(var1) VARAVIS(var2)

Tabla 5.8.

INSTRUCCIONES ASOCIADAS AL MONITOREO DE SEÑALES		
INSTRUCCIÓN	DESCRIPCIÓN	EJEMPLO
M1SENAL( <i>canal</i> )	Monitorea una señal; es decir, toma UNA muestra de la señal del canal indicado y la envía por el puerto serie asíncrono. (1)	M1SENAL(1) M1SENAL(2) M1SENAL(4)
M4SENAL()	Monitorea cuatro señales; es decir, toma sucesivamente las muestras de las señales de los canales 1,2,3 y 4 y las envía de igual forma al puerto serie asíncrono. (1)	M4SENAL()
G1SENAL( <i>canal, dirini, dirfin, dato, unidad</i> )	Guarda las muestras de una señal; es decir, toma las muestras de la señal del canal seleccionado y las almacena en la UAD a partir de la dirección de inicio y hasta alcanzar la dirección de fin. También, permite seleccionar el tiempo entre muestra y muestra.	G1SENAL(2,hx0000,hx7f000,3, min) G1SENAL(4,hxab123,hxfe000,5, ms) G1SENAL(1,hxd5f00,hxde000,3, seg)
G4SENAL( <i>dirini, dirfin, dato, unidad</i> )	Guarda las muestras de las cuatro señales; es decir, toma sucesivamente las muestras de las señales de los canales 1,2,3 y 4 y las almacena de igual forma en la UAD a partir de la dirección de inicio y hasta alcanzar la dirección de fin. También, permite seleccionar el tiempo entre cada cuatro muestras. (3)	G4SENAL(hxa0000,hxb0017,15,seg) G4SENAL(hxd0007,hxe001e,1, min) G4SENAL(hxc0005,hxf001c, 1, hrs)
MG1SENAL( <i>canal, dirini, dirfin, dato, unidad</i> )	Monitorea y guarda las muestras de una señal; es decir, toma las muestras de la señal del canal seleccionado, las almacena en la UAD a partir de la dirección de inicio y hasta alcanzar la dirección de fin. Además envía las muestras al puerto serie asíncrono.(1) También, permite seleccionar el tiempo entre muestra y muestra.	MG1SENAL(2, hx0000,hx7f000, 3, min) MG1SENAL(4, hxab123,hxfe000, 5, ms) MG1SENAL(1, hx45f00,hxa5000, 1, ds)
MG4SENAL( <i>dirini, dirfin, dato, unidad</i> )	Monitorea y guarda las muestras de las cuatro señales; es decir, toma sucesivamente las muestras de las señales de los canales 1,2,3 y 4 y las almacena de igual forma en la UAD a partir de la dirección de inicio y hasta alcanzar la dirección de fin. Además envía las muestras al puerto serie asíncrono. (1) También, permite seleccionar el tiempo entre cada cuatro muestras.(3)	MG4SENAL(hxca000,hxca007, 13, seg) MG4SENAL(hxdb000,hxdb00,25, ms) MG4SENAL(hxac000,hxac007,25, seg)

Tabla 5.9.

INSTRUCCIONES DIVERSAS		
INSTRUCCIÓN	DESCRIPCIÓN	EJEMPLO
INCVAR( <i>var, dato</i> )	Incrementa o decrementa la variable seleccionada con el valor dado por dato Nota: El dato deberá darse como número entero.	INCVAR(var1,10) INCVAR(var2,-10) INCVAR(dsem,1)
DATOAVAR( <i>var, dato ó var</i> )	Guarda un dato o el valor de una variable en otra variable. var1=hex7f equivale a DATOAVAR(var1,hex7f).	DATOAVAR(var2,hx57) DATOAVAR(var1,38) DATOAVAR(var2,dsem)
PAUSA( <i>dato, unidad</i> )	Genera un retardo o pausa en milisegundos, décimas de segundo, segundos, minutos u horas. Nota: El dato deberá darse como número entero.	PAUSA(100.ms) PAUSA(75,ds) PAUSA(2,hrs)
TEXTASER( <i>texto</i> )	Envía por el puerto serie asincrono un texto de hasta 200 caracteres.(1)	TEXTASER( ! Hola mundo ! ) TEXTASER(El valor es: )
INTERMIT( <i>puerto, dato, veces, tiempo</i> )	De manera intermitente prende y apaga las salidas del puerto descado tantas veces y durante el tiempo dado en milisegundos.	INTERMIT(PTS1,hx4f,10,500) INTERMIT(ptac,b7,125,10000) INTERMIT(PTS1,hxaa,75,300)

Tabla 5.10.

Nota (1): Velocidad de transmisión fija de 9600 bauds. Formato: un bit de inicio, 8 bits de datos, un bit de finalización y ninguna paridad.

Nota (2): El formato de despliegue es fijo y es así como aparece en el visualizador LCD:

v	a	r	1	:	1	0	1	0	1	1	1	1	B	I	N
					S	C	F						H	E	X

Nota (3): Como se están almacenando de 4 en 4 muestras, la dirección de fin debe ser mayor a la dirección de inicio y además la diferencia entre la dirección de fin y la dirección de inicio deberá ser un múltiplo de 4.

Las sentencias de control de flujo del lenguaje LENSAC especifican el orden en que se realizan las instrucciones. A continuación se muestran las sentencias de control de flujo. Al igual que las instrucciones se presenta la sentencia y sus parámetros asociados, una explicación de la función que realiza y ejemplos de utilización.

Un resumen de las sentencias de control de flujo disponibles se da en la tabla 5.11

**La sentencia SI(*expresión*) - FINSI()**

Esta sentencia de control sirve para tomar decisiones. Se evalúa la *expresión*; si es verdadera, se ejecutan las instrucciones siguientes (instrucciones 1, 2 y 3); si es falsa, se ejecutan las instrucciones que se encuentran después de FINSI() (instrucciones 4, 5).

Ejemplo:     SI(mues >= 250 O mues <= 57 Y var1 != 10 Y var2 != hx7E)  
                   instrucción 1  
                   instrucción 2  
                   instrucción 3  
               FINSI()  
                   instrucción 4  
                   instrucción 5

**La sentencia REPITE(*dato*) - FINREPITE()**

Sirve para realizar iteraciones. Las instrucciones siguientes a la sentencia REPITE(*dato*) (instrucciones 1, 2 y 3), se ejecutarán el número de veces indicado por *dato* hasta encontrar la sentencia FINREPITE() .

Ejemplo:     REPITE (55)  
                   instrucción 1  
                   instrucción 2  
                   instrucción 3  
               FINREPITE()  
                   instrucción 4  
                   instrucción 5

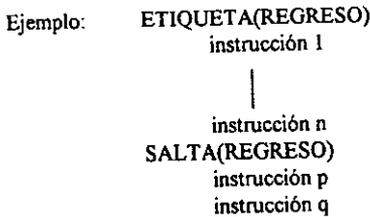
**La sentencia MIENTRAS(*expresión*) - FINMIENTRAS()**

Esta sentencia sirve para ejecutar instrucciones ciclicamente. Se evalúa la *expresión*; si es verdadera, se ejecutan las instrucciones siguientes repetidamente hasta que la expresión sea falsa (instrucciones 1, 2 y 3); si la expresión es falsa, se ejecutan las instrucciones que se encuentran después de FINMIENTRAS() (instrucciones 4 y 5).

Ejemplo:     MIENTRAS(mus1 <= 250 Y mus2 <= 100 Y mus3 <= 49)  
                   instrucción 1  
                   instrucción 2  
                   instrucción 3  
               FINMIENTRAS()  
                   instrucción 4  
                   instrucción 5

**Las sentencias SALTA(nombre) y ETIQUETA(nombre)**

Estas sentencias de control sirven para direccionar el flujo del programa, es decir, permiten saltos a cualquier etiqueta que se encuentre dentro del programa de aplicación.



<b>SENTENCIAS DE CONTROL DE FLUJO</b>		
<b>SENTENCIA</b>	<b>DESCRIPCIÓN</b>	<b>EJEMPLO</b>
<b>SI</b> (var REL dato LOG var REL dato) instrucción l . . instrucción n <b>FINSI</b> ( )	Si la expresión lógica es cierta, se ejecutarán los instrucciones siguientes hasta encontrar la sentencia <b>FINSI</b> ( ).  <b>FINSI</b> ( ) indica el fin de la sentencia " <b>SI</b> (expresión)".	<b>SI</b> (mues >= 250 O mues <= 57 Y var1 !=10 Y var2!=hx7E) instrucción l . . instrucción n <b>FINSI</b> ( )
<b>REPITE</b> (dato) instrucción l . . instrucción n <b>FINREPITE</b> ( )	Las instrucciones siguientes, hasta encontrar la sentencia <b>FINREPITE</b> ( ), se ejecutarán el número de veces indicado. Nota: El dato deberá ser un número entero entre 0 y 30000.  <b>FINREPITE</b> ( ) indica el fin de la sentencia " <b>REPITE</b> (dato)".	<b>REPITE</b> (30000) instrucción l . . instrucción n <b>FINREPITE</b> ( )
<b>MIENTRAS</b> (var REL dato LOG var REL dato...) instrucción l . . instrucción n <b>FINMIENTRAS</b> ( )	Permite ejecutar las instrucciones ubicadas entre la sentencia <b>MIENTRAS</b> (expresión) y <b>FINMIENTRAS</b> ( ), hasta que la expresión sea falsa.  <b>FINMIENTRAS</b> ( ) indica el fin de la sentencia " <b>MIENTRAS</b> (expresión)".	<b>MIENTRAS</b> (mus1 <= 250 Y mus2 <=100 Y mus3 <= 49) instrucción l . . instrucción n <b>FINMIENTRAS</b> ( )
<b>ETIQUETA</b> (nombre) instrucción l . . instrucción n <b>SALTA</b> (nombre)	<b>SALTA</b> (nombre) salta o direcciona el flujo del programa a la etiqueta deseada.  <b>ETIQUETA</b> (nombre) pone un etiqueta a la cual se va a direccionar el flujo del programa.	<b>ETIQUETA</b> (REGRESO) instrucción l . . instrucción n <b>SALTA</b> (REGRESO)

Tabla 5.11.

## 5.4 Programa PROSAC para Windows

### Formas tradicionales de generar programas de aplicación

Existen dos formas tradicionales de generar programas de aplicación :

- 1) Generando programas de aplicación en código ensamblador. Desafortunadamente, escribir el código fuente (archivo. s) es una labor tediosa, que requiere de mucho tiempo y propicia errores. Además, el usuario tiene que programar, probar y depurar todas y cada una de las funciones que desee implementar. Para los usuarios no familiarizados con el lenguaje ensamblador, la tarea será ardua y complicada, ya que el código fuente del lenguaje ensamblador es muy complejo.
- 2) Generando programas de aplicación en lenguaje de alto nivel C. El lenguaje C es uno de los más poderosos lenguajes de alto nivel. Si el usuario desea implementar programas de aplicación para el SAC en el lenguaje C, se encontrará con herramientas muy útiles y poderosas que le permitirán realizar tareas y funciones complejas. Sin embargo, es obvio que el usuario tiene que conocer el lenguaje C, y para los usuarios no familiarizados con él, la tarea será, si no muy complicada; si lenta, ya que tendrán que aprender la sintaxis del lenguaje, los tipos de operadores, sentencias de control etc. Por otra parte, aún conociendo el lenguaje de programación C, el usuario tendrá que programar, probar y depurar todas y cada una de las funciones que desee implementar.

### PROSAC, una nueva forma de generar programas de aplicación

Una forma alterna de realizar programas de aplicación para el SAC es generando archivos de texto (archivo.txt) que contengan las instrucciones y sentencias de control correspondientes al lenguaje de programación LENSAC utilizando para ello el programa PROSAC. En este caso el usuario no requiere conocimientos previos de ningún lenguaje de programación y podrá generar un programa de aplicación rápidamente y en un ambiente de trabajo versátil como Windows. El usuario cuenta con muchas instrucciones enfocadas al control y adquisición en tiempo real, además de sentencias de control para realizar rápidamente y de manera lógica e intuitiva el programa de aplicación que desee. Por ser un ambiente gráfico, únicamente tendrá que seleccionar las instrucciones y sentencias de control haciendo "click" con el ratón sobre el botón o ícono correspondiente a la instrucción o sentencia de control deseada. Con PROSAC el usuario cuenta con las herramientas necesarias que le permiten editar, depurar, probar y ejecutar programas de aplicación de una manera sencilla y eficiente.

#### 5.4.1 Descripción

PROSAC (**PRO**gramación del **S**istema de **A**dquisición y **C**ontrol de propósito general) es un programa para Windows que permite al usuario realizar y transferir fácilmente programas de aplicación, así como recibir información proveniente del SAC. Básicamente, es una interfaz gráfica que permite generar el conjunto de instrucciones necesarias para realizar programas de aplicación por medio de una computadora personal PC. PROSAC está escrito en el lenguaje Visual BASIC y está ligado directamente al lenguaje de programación LENSAC descrito anteriormente.

El objetivo principal de PROSAC es: obtener un archivo que contenga las instrucciones necesarias para que el SAC pueda ejecutar la tarea deseada, a partir de un programa formado por un conjunto de instrucciones y sentencias de control pertenecientes al lenguaje de programación LENSAC.

Para alcanzar este objetivo, en el desarrollo de PROSAC se combinaron los conceptos de intérprete, compilador y ensamblador.

#### 5.4.2 Estructura del programa PROSAC

En la figura 5.3 se muestra a manera de diagrama de bloques la estructura del PROSAC.

#### 5.4.3 Editor e intérprete

El programa PROSAC cuenta con un editor y un intérprete de instrucciones y sentencias de control de flujo (ver bloque 1 de la figura 5.3).

##### a) Editor

PROSAC cuenta con una ventana de edición que permite al usuario escribir las instrucciones y sentencias de control de flujo de un programa de aplicación. Dentro de la ventana de edición, el usuario tiene dos posibilidades:

- Escribir en cada una de las líneas las instrucciones y sentencias de control del lenguaje de programación LENSAC, o bien,
- Introducir fácilmente el conjunto de instrucciones y sentencias de control por medio de la barra de botones con iconos únicamente haciendo "click" en el botón izquierdo del ratón (*mouse*).

Si el usuario no tiene acceso a una computadora con el programa PROSAC, puede utilizar el editor de textos de su preferencia para generar el programa de aplicación deseado. Únicamente tendrá que guardar el programa como un archivo de texto con extensión ".txt".

Cabe mencionar, que si el usuario hace uso de la ventana de edición del PROSAC, podrá generar rápidamente y de manera sencilla programas de aplicación. También podrá abrir, guardar e imprimir archivos; cortar, pegar, borrar, buscar y remplazar instrucciones, además de insertar instrucciones o sentencias de control entre líneas de una manera fácil.

Los menús con las opciones disponibles y la barra de botones del editor se describirán en las secciones siguientes.

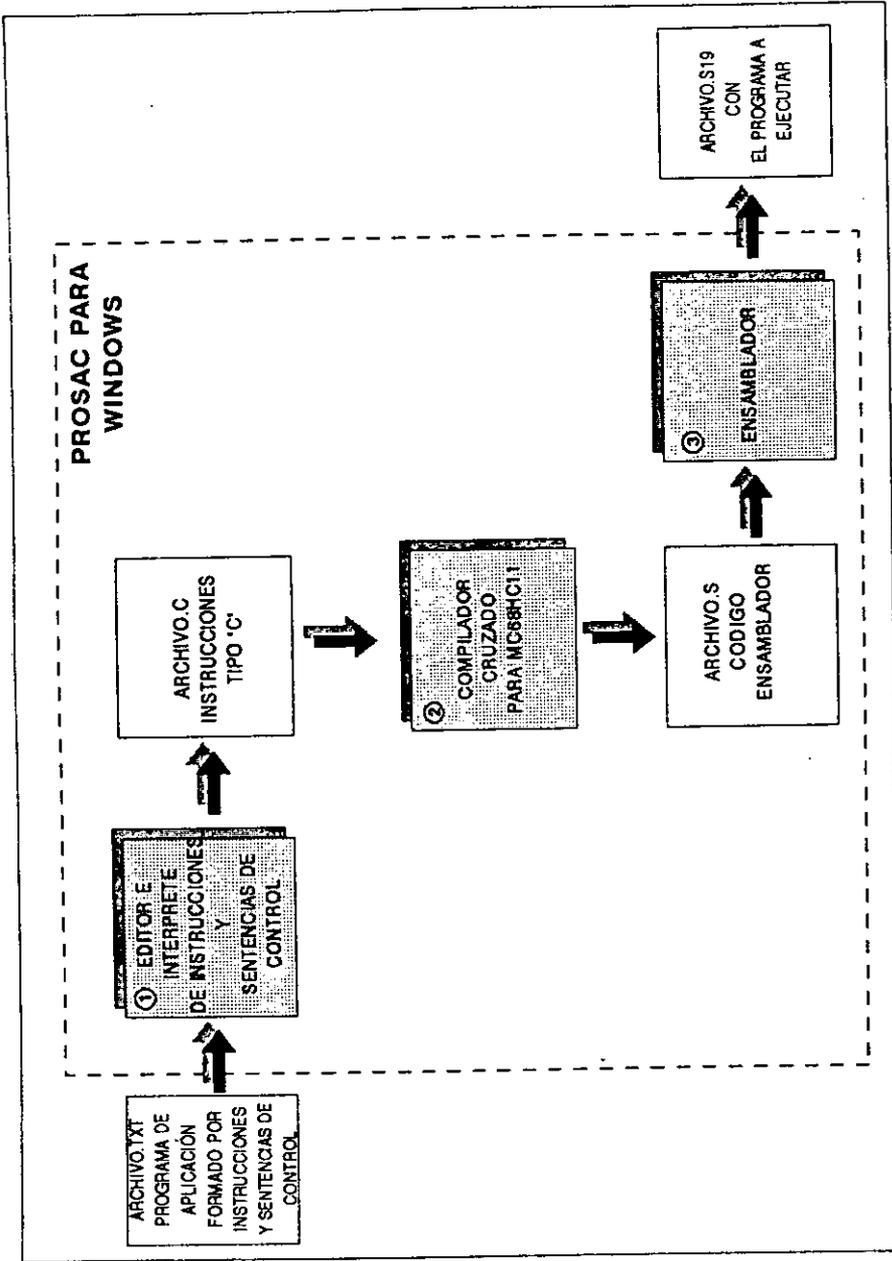


Figura 5.3. Estructura del programa PROSAC.

### b) Intérprete

El programa intérprete de PROSAC recibe un archivo de entrada con extensión ".txt" (archivo.txt) y genera un archivo de salida con extensión ".c" (ver figuras 5.3 y 5.4).

En términos simples, el intérprete lee el programa de aplicación en lenguaje de programación LENSAC (archivo ".txt") una línea a la vez y genera su correspondiente código en lenguaje "C". De esta manera, si dentro del programa de aplicación se encuentran errores de sintaxis, nombres, variables o valores no válidos, entonces el intérprete desplegará en la pantalla de la PC el mensaje del error correspondiente, las posibles causas y sugerencias para su corrección. Además, indicará el número de línea en donde se encontró el error y cuando sea posible, lo iluminará para que el usuario pueda corregirlo. Finalmente, cuando el programa de aplicación esté libre de errores el programa intérprete generará un archivo con extensión ".c" que contiene el programa de aplicación traducido al lenguaje "C".

Cabe mencionar que el editor y el programa intérprete son programas originales no comerciales que fueron desarrollados en su totalidad en lenguaje Visual BASIC, tienen una longitud de más de 3000 líneas de código y no están basados en ningún otro editor, programa intérprete o compilador.

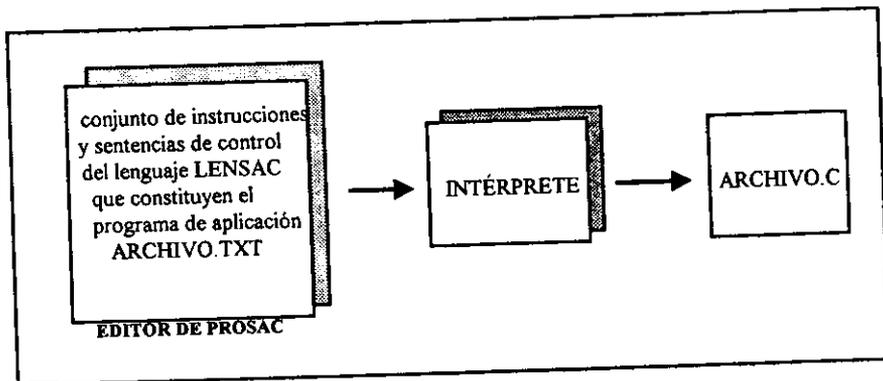


Figura 5.4 Editor e intérprete

#### 5.4.4 Compilador cruzado para MC68HC11

El compilador cruzado para MC68HC11 permite traducir el programa de aplicación de lenguaje "C" en un código de ensamblador para microcontroladores MC68HC11 (ver figura 5.3 bloque 2 ). Para realizar esta tarea, se utilizó el compilador cruzado para MC68HC11 versión "freeware" de ImageCraft el cual es del dominio público pudiéndose obtener a través de la red INTERNET. Consiste de los programas siguientes: un programa manejador (icc11.exe), un preprocesador de C (icpp.exe) y un compilador (iccom11.exe).

El compilador cruzado es un sistema compilador tradicional que acepta y se ajusta al lenguaje ANSI C. El compilador cruzado invoca al programa manejador (icc11.exe), a las funciones necesarias y al archivo que contiene las rutinas básicas de ensamblador (bas6.s) para poder procesar el archivo en C (archivo.c). Si el compilador cruzado no detecta ningún error, produce un archivo objeto, que contiene el código en ensamblador (archivo.s). Si hubo algún error se le indica al usuario.

El programa manejador (icc11.exe) direcciona al preprocesador (icpp.exe) para que busque en los directorios especificados los encabezados de los archivos que contienen las funciones que se han desarrollado en C y que son necesarias para generar el archivo con el código en ensamblador.

El preprocesador (icpp.exe) agrega al archivo en C las funciones necesarias (*header files*) generando un archivo en lenguaje de alto nivel C (archivo.i).

Finalmente el compilador (iccom11.exe) traduce el código del archivo en lenguaje de alto nivel C (archivo.i) en un código fuente para el ensamblador.

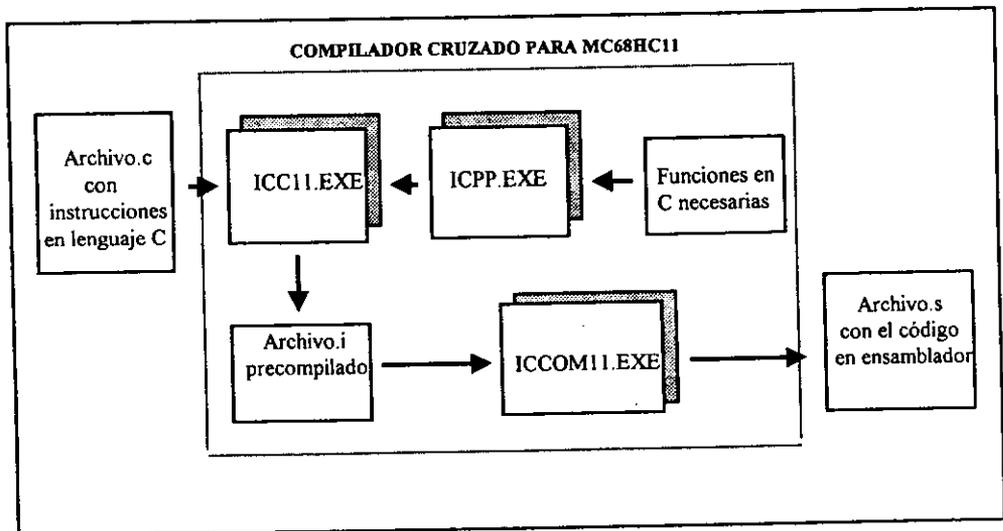


Figura 5.5 Compilador cruzado para MC68HC11

### 5.4.5 Ensamblador

El ensamblador utilizado es el IAS11.exe y es la versión "freeware" de Motorola el cual es del dominio público y se puede obtener también en la red INTERNET (ver figura 5.3 bloque 3).

La función del ensamblador es convertir el archivo fuente (archivo.s) con el código en ensamblador en código leíble por el SAC (archivo.s19). Durante el proceso de ensamblado se ensamblan los archivos archivo.s y bas6.s.

El archivo bas6.s es un fragmento de código en ensamblador que contiene varias rutinas básicas en lenguaje ensamblador. Define el código de inicialización y de configuración de variables y etiquetas, las direcciones de inicio para datos, la dirección de inicio del programa, el valor inicial del *stack pointer* y después salta a la función *main* del programa en C.

Una vez que los archivos archivo.s y bas6.s han sido ensamblados, el código resultante, llamado código objeto (archivo.s19), es una cadena de dígitos binarios (unos y ceros) que pueden ser ejecutados directamente por el microcontrolador MC68HC11A1. Debido a que existe una relación directa entre el código fuente (archivo.s) y el código objeto (archivo.s19), el código producido por el ensamblador es usualmente muy compacto y eficiente.

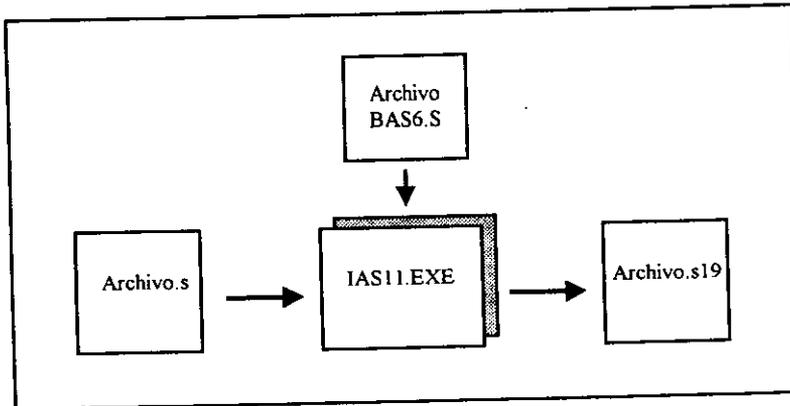


Figura 5.6 Ensamblador

#### 5.4.6 Pantallas de PROSAC

PROSAC es una interfaz gráfica que funciona bajo Windows que permite al usuario realizar programas de aplicación fácilmente, ya que cuenta con varios menús y un conjunto de botones con los cuales se puede programar rápidamente la función deseada.

A continuación se presentan las pantallas disponibles en el programa PROSAC.

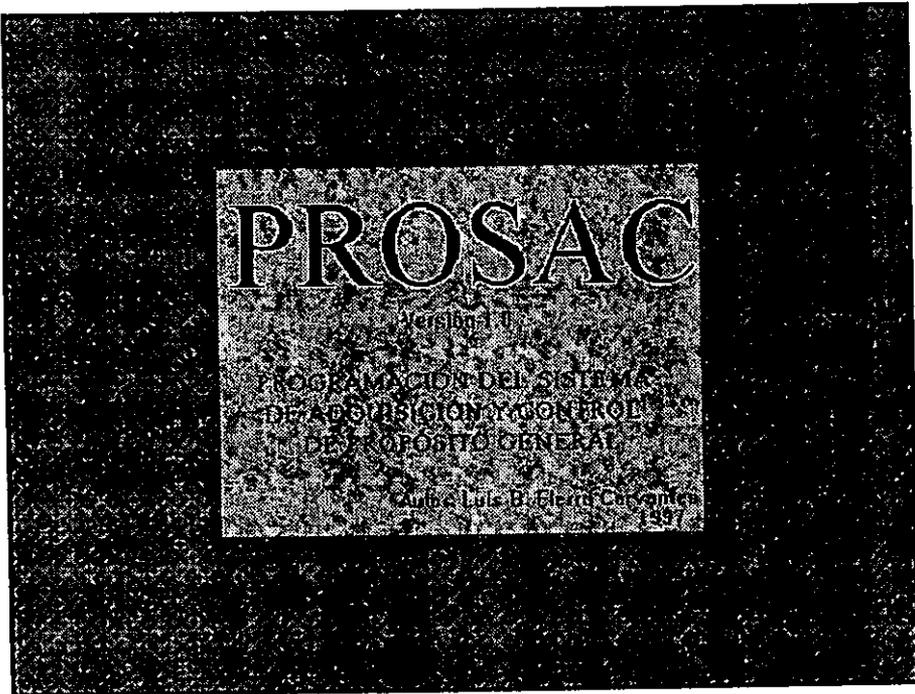


Figura 5.7 Pantalla de inicio del programa PROSAC

La pantalla de presentación (figura 5.7) se despliega en el monitor de la PC durante 5 segundos y después se presenta la pantalla principal de PROSAC, la cual consta de 4 menús, una ventana de edición, una ventana de ayuda y una barra de botones o iconos (ver figura 5.8).

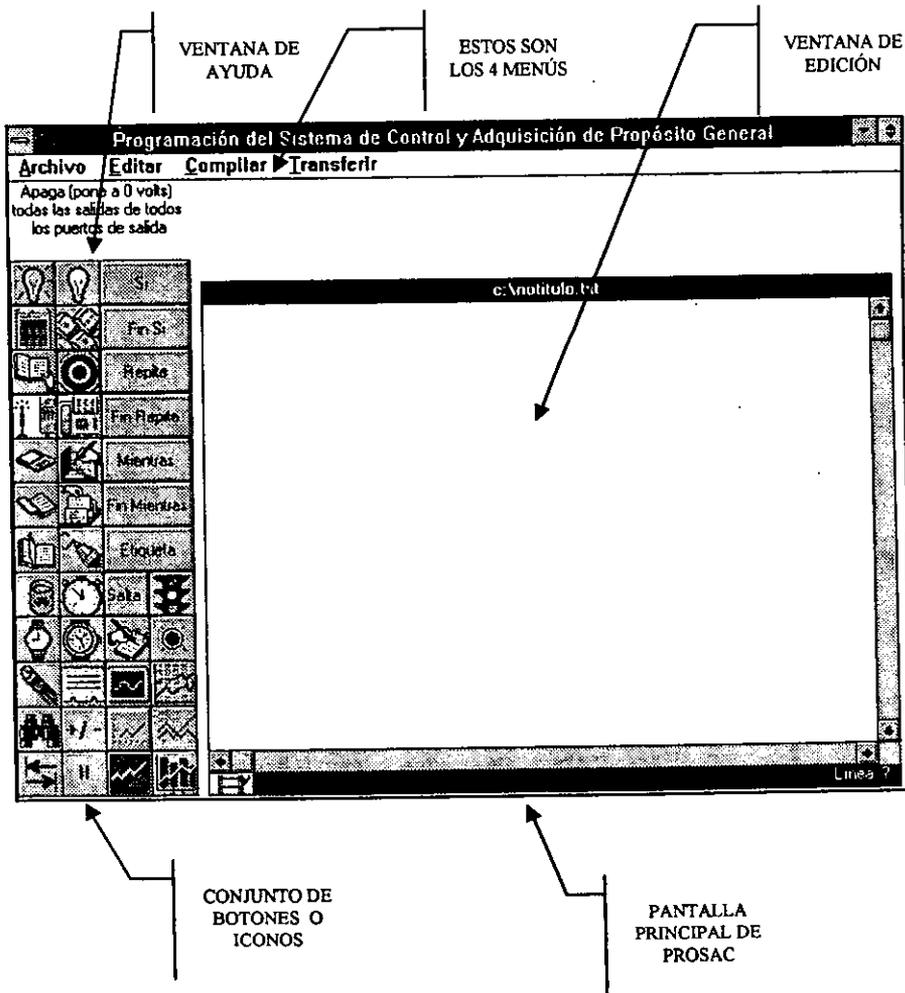


Figura 5.8 Pantalla principal del programa PROSAC

5.4.7 Botones o íconos

Los botones o iconos permiten seleccionar por medio del ratón, las instrucciones y sentencias de control para generar un programa de aplicación (ver figura 5.9). Al pasar el cursor del ratón sobre estos iconos aparecerá en la ventana de ayuda una breve descripción de la función que se realiza. Si se hace "click" sobre alguno de estos iconos o botones, sobre la ventana de edición aparecerá la ventana de parámetros con la cual el usuario podrá seleccionar y modificar los parámetros asociados a la instrucción o sentencia de control seleccionada. Una vez que los parámetros han sido seleccionados, el usuario podrá escoger si desea ACEPTAR, CANCELAR o INSERTAR la instrucción o sentencia dentro de la ventana de edición. Si el usuario hace "click" sobre el botón de ACEPTAR, la instrucción o sentencia de control se agregará en la línea siguiente. De esta manera se pueden ir seleccionando las instrucciones o sentencias y automáticamente formar el programa de aplicación. El botón de CANCELAR, es útil cuando el usuario no desea agregar la instrucción o sentencia de control que había seleccionado en la ventana de edición. La opción de INSERTAR se utiliza cuando se desea insertar por ej., entre dos instrucciones una nueva instrucción. La instrucción seleccionada se insertará a partir de la posición en donde se encuentre el cursor de la ventana de edición.

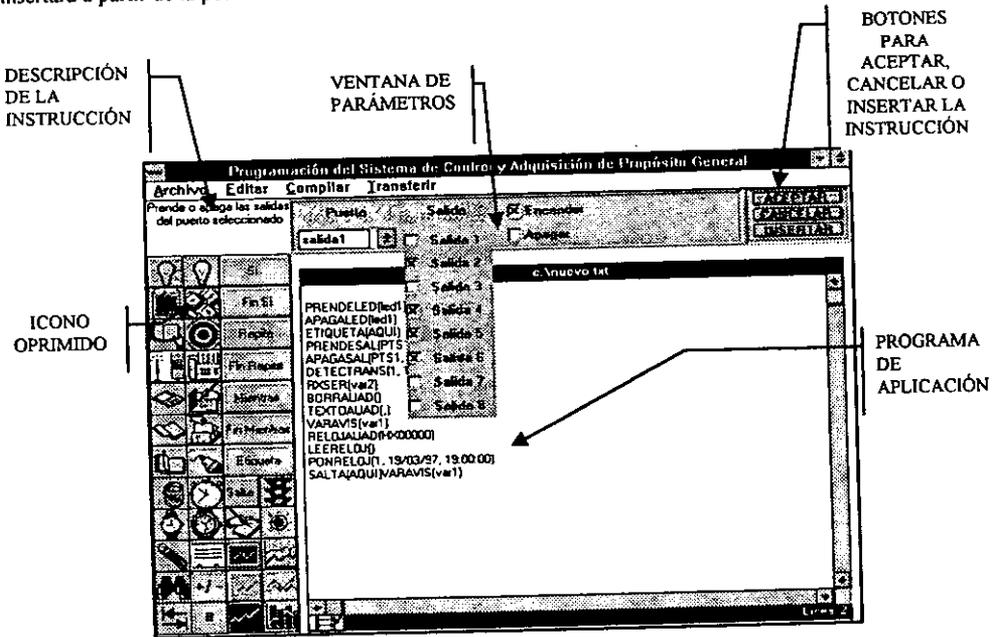


Figura 5.9. Botones o iconos.

1) Iconos de las instrucciones asociadas a los puertos de entrada y salida

PRENDETS()  
 ESCRIBEPS(*puerto, dato* o *var*)  
 LEEPE(*puerto, var*)  
 TXSER(*dato*)  
 PRENDELED(*led*) o  
 APAGALED(*led*)



APAGATS()  
 PRENDESAL(*puerto, dato*) o  
 APAGASAL(*puerto, dato*)  
 DETECTTRANS(*entrada, transición*)  
 RXSER(*var*)

2) Iconos de las instrucciones asociadas al almacenamiento de datos

BORRAUAD()  
 LEEUAD(*dirini, dato*)  
 TRANSUAD()  
 UADAVAR(*dirini, var*)



TEXTOAUAD(*dirini, texto*)  
 LEEBLOQUAD(*dirini, dirfin*)  
 ESCRIBEUAD(*dirini, dato*)

3) Iconos de las instrucciones asociadas al reloj fechador

LEERELOJ()



PONRELOJ(*diasem, fecha, hora*)  
 RELOJAUAD(*dirini*)

4) Iconos de las instrucciones asociadas al módulo de despliegue

BORRAVIS()  
 VARAVIS(*var*)



ESCRIBEVIS(*ren, pos, texto*)

**5) Iconos asociados a las instrucciones del monitoreo de señales**

MISENAL( <i>canal</i> )		M4SENAL()
G1SENAL( <i>canal, dirini, dirfin, dato, unidad</i> )		G4SENAL( <i>dirini, dirfin, dato, unidad</i> )
MG1SENAL( <i>canal, dirini, dirfin, dato, unidad</i> )		MG4SENAL( <i>dirini, dirfin, dato, unidad</i> )

**6) Iconos asociados a las instrucciones diversas**

INCVAR( <i>var, dato</i> )		DATOAVAR( <i>var, dato ó var</i> )
PAUSA( <i>dato, unidad</i> )		TEXTASER( <i>texto</i> )
INTERMIT( <i>puerto, dato, veces, tiempo</i> )		

**7) Iconos asociados a las sentencias de control de flujo**

SI( <i>var REL dato LOG var REL dato ...</i> )	
FINSI()	
REPITE( <i>dato</i> )	
FINREPITE()	
MIENTRAS( <i>var REL dato LOG var REL dato...</i> )	
FINMIENTRAS()	
ETIQUETA()	
SALTA( <i>nombre</i> )	

5.4.8 Ventana de edición

Dentro de la ventana de edición (figura 5.10) se encuentran todas las instrucciones y sentencias de control que forman un programa de aplicación. En esta ventana, el usuario puede visualizar las instrucciones y sentencias de control que forman el programa de aplicación. Puede seleccionar, copiar, pegar o borrar el texto que desee y también puede escribir y modificar las instrucciones, las sentencias y los parámetros asociados a cada una de ellas.

La ventana de edición cuenta con dos barras de desplazamiento ("scroll bars") una vertical y una horizontal que permiten desplazarse a lo largo y ancho del programa de aplicación. También, cuenta en la parte inferior con un indicador de la línea en la que se encuentra actualmente el cursor. Proporciona además un botón ubicado en la esquina inferior izquierda que, al apretarlo, y después de que el programa de aplicación ha sido compilado, presenta la lista de instrucciones en el lenguaje LENSAC y la lista de instrucciones correspondientes en el lenguaje C.

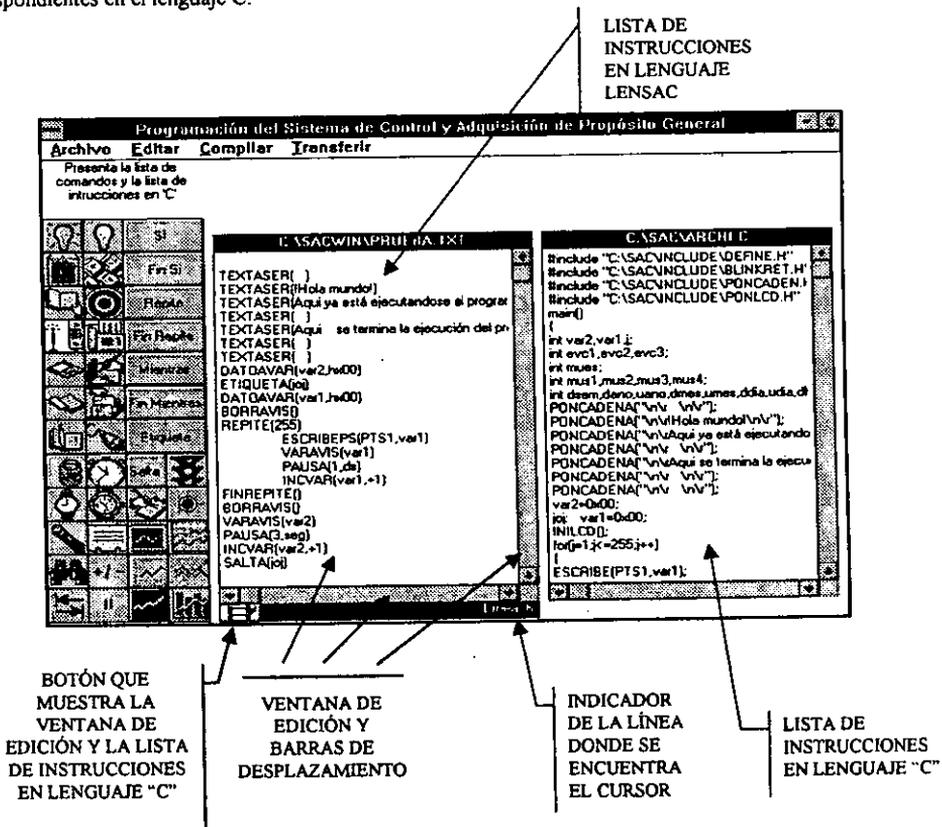


Figura 5.10. Ventana de edición

### 5.4.9 Menú Archivo

El menú de **Archivo** cuenta con 6 opciones y se accesa con el ratón o con las teclas **ALT+A**.

#### 1) Nuevo

La opción **Nuevo**, permite abrir un nuevo archivo (por *default* se pone el título c:\nuevo.txt) para poder generar a su vez un nuevo programa de aplicación. Se puede seleccionar rápidamente estando dentro del menú **Archivo** con la tecla **N**.

#### 2) Abrir archivo

Seleccionando **Abrir** con el ratón o con **archivo** con la tecla **A**, abrirá una ventana de diálogo que permite al usuario seleccionar algún archivo de las unidades que dispone la computadora para de ésta manera poder verlo, editarlo, compilarlo o transferirlo al SAC.

#### 3) Guardar

Con la opción **Guardar**, el usuario puede salvar los cambios que se le hubieran hecho a algún programa de aplicación. Esta opción está disponible con la tecla **G** o con el ratón.

#### 4) Guardar como

Si el usuario desea guardar los cambios hechos a un programa en un archivo, por medio de una ventana de diálogo esta opción le permitirá cambiar el nombre del archivo y escoger la unidad de la computadora en la que desea guardarlo. La tecla rápida es **C**.

#### 5) Imprimir el archivo

Seleccionando esta opción, las instrucciones y sentencias de control (el programa de aplicación) que se encuentren en la ventana de edición se imprimirán en la impresora conectada a la computadora.

#### 6) Salir

Cuando el usuario selecciona la opción de **Salir**, inmediatamente se guardan los últimos cambios hechos al programa de aplicación y la aplicación de **PROSAC** termina.

#### 5.4.10 Menú Editar

El menú de **Editar** se accesa por medio de las teclas **ALT+E**. Este menú cuenta con 6 opciones que se pueden manejar, ya sea por el ratón o desde el teclado con las teclas de acceso rápido. Las opciones válidas son las siguientes:

**1) Cortar    Ctrl+X**

Para cortar o borrar un texto de la ventana de edición, basta con seleccionarlo con el ratón y seleccionar también la opción de cortar del menú editar, o bien apretando simultáneamente las teclas **Ctrl** y **X** del teclado.

**2) Copiar    Ctrl+C**

Con esta opción el usuario puede copiar el texto deseado de la ventana de edición para después pegarlo en cualquier parte dentro de la ventana de edición.

**3) Pegar    Ctrl+V**

Cuando el usuario previamente ha copiado un texto, con esta opción podrá pegarlo en cualquier parte dentro de la ventana de edición.

**4) Borrar    Del ó Supr**

Al igual que la opción cortar **Ctrl+X**, el usuario puede cortar o borrar un texto de la ventana de edición únicamente seleccionando el texto a borrar y apretando la tecla **Del** o **Supr** del teclado.

**5) Buscar    Ctrl+F**

Con esta opción, el usuario puede buscar el texto que desee dentro de la ventana de edición. El usuario puede seleccionar buscar una sola vez el texto deseado o buscar de nuevo la aparición de éste dentro de la ventana de edición.

**6) Reemplazar Ctrl+R**

Reemplaza automáticamente todas las ocurrencias (dentro de la ventana de edición) del texto que se desee reemplazar por otro texto seleccionado por el usuario.

### 5.4.11 Menú Compilar

El menú de **Compilar** se activa por medio del ratón o del teclado con las teclas **ALT+C**. Este menú cuenta solamente con una opción:

#### 1) Iniciar la compilación

Una vez que se ha terminado de escribir el programa de aplicación en la ventana de edición, es recomendable que el usuario lo guarde con la opción de **Guardar** del menú de **Archivo**. Una vez hecho esto, se puede iniciar la compilación del programa.

Al iniciar la compilación del programa se siguen los pasos del diagrama de bloques de la estructura de PROSAC (figura 5.3), es decir, se ejecuta el intérprete de instrucciones, luego el compilador cruzado y al final el ensamblador. A lo largo de este proceso de compilación, PROSAC se comunica con el usuario por medio de ventanas de mensajes indicándole si hubo errores y cuando sea posible, en dónde ocurrieron y también las posibles causas, proponiendo sugerencias para su corrección. El error se destaca iluminando el lugar donde ocurrió. Finalmente, PROSAC manda un mensaje al usuario cuando el programa de aplicación se compiló de manera exitosa y el usuario está listo para transferir el programa de aplicación (archivo.s19) al SAC (ver figura 5.11).

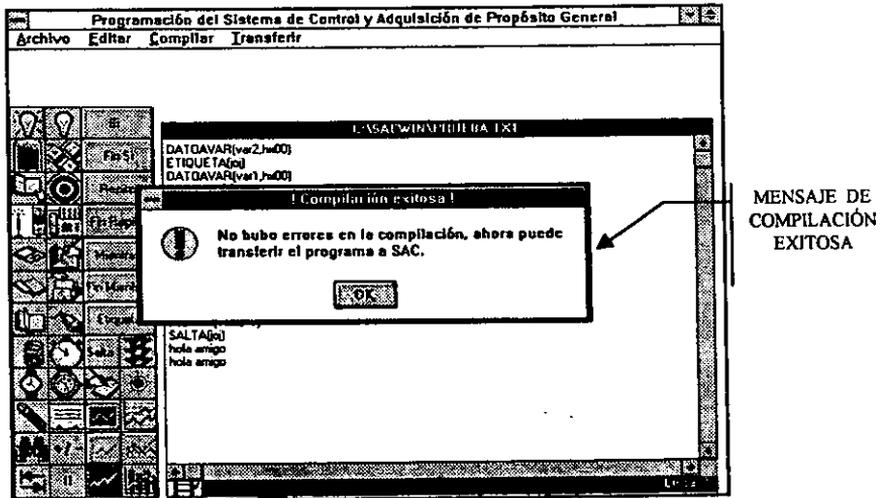


Figura 5.11. Compilación del programa.

5.4.12 Menú Transferir

Una vez que el programa de aplicación ha sido compilado exitosamente, el archivo resultante de la compilación (archi.s19) deberá transferirse al SAC para que éste pueda ejecutar la tarea deseada. Anteriormente se mencionó que existen dos formas de transferir el programa de aplicación al SAC: mediante programación con conexión al SAC ("in-line") y mediante programación sin conexión a SAC ("off-line").

En el caso de la opción ("off-line") sin conexión al SAC, el usuario deberá copiar el archivo resultante de la compilación (c:\sac\archi.s19) a la tarjeta de memoria PCMCIA a partir de la dirección \$00000. Para esto, la computadora deberá contar con una unidad (drive) que permita leer y escribir tarjetas de memoria PCMCIA y por supuesto con los programas que manejan este tipo de unidades.

En el caso de la opción ("in-line") con conexión al SAC, el usuario está conectado directamente al SAC por medio de un cable para comunicación serie asincrónica RS-232C. En este caso se utiliza el menú de Transferir el cual se activa por medio del ratón o del teclado con las teclas ALT+T. Este menú cuenta solamente con una opción:

1) Transferir el programa

Para transferir el programa de aplicación compilado al SAC, el usuario únicamente tiene que seleccionar la opción de transferir el programa y PROSAC se encargará de transferirlo. Cuando el usuario aprieta la opción de transferir el programa, PROSAC activa la ventana del programa TERMINAL de Windows y configura todos los parámetros necesarios (como formato, velocidad, nombre del archivo etc.) para transferir el programa al SAC. Una vez que el SAC ha recibido bien el programa de aplicación, aparecerá en el programa TERMINAL, el mensaje de "OK !!" y automáticamente el SAC comenzará a ejecutarlo.

En las pantallas mostradas en las figuras 5.12 y 5.13 se muestra el programa TERMINAL de Windows durante la transferencia del programa y después de recibirlo y comenzar a ejecutarlo.

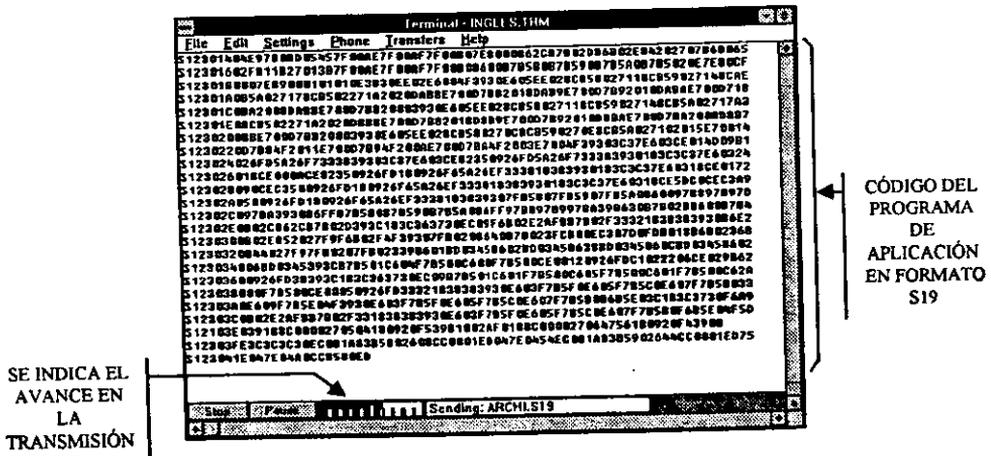


Figura 5.12. Transferencia del programa.

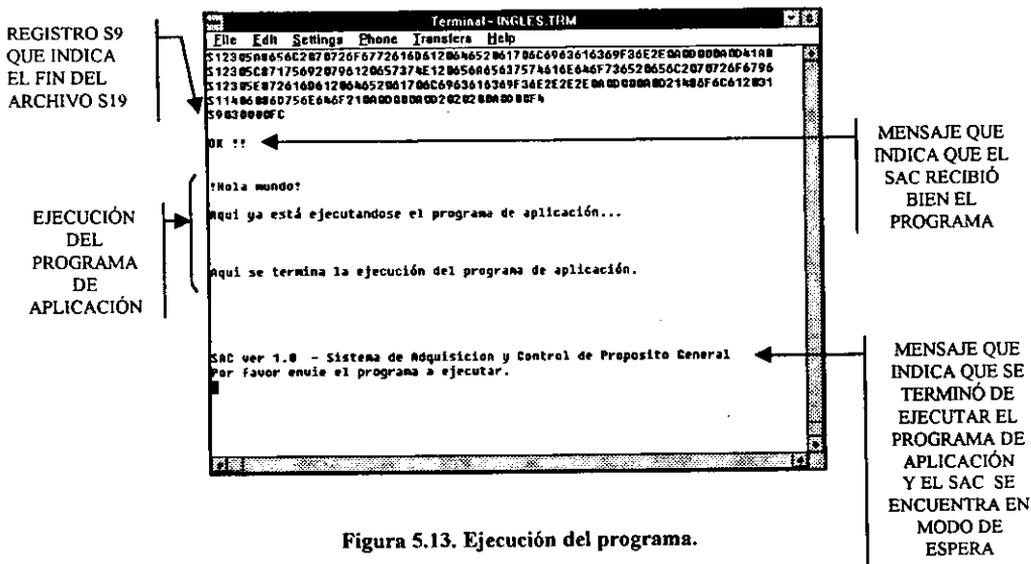


Figura 5.13. Ejecución del programa.

### 5.4.13 Requerimientos y limitaciones de PROSAC

#### Requerimientos

Los requerimientos mínimos de *hardware* y *software* que necesita la aplicación de PROSAC, son:

- Microsoft Windows 3.0 o superior incluyendo Windows 95, corriendo en modo estándar o modo aumentado
- Microprocesador 80286 o superior
- Monitor de video CGA, EGA, VGA, 8514, Hércules, o cualquier otro monitor compatible con Windows
- 4 MB de memoria o más
- Ratón o "mouse"

Es posible escribir programas de aplicación para el SAC sin el ratón. En este caso, el usuario necesitará trabajar con las teclas de acceso rápido válidas.

## Limitaciones de PROSAC

### a) Limitaciones de las instrucciones y sentencias de control

No hay límite en el número de veces que se puede utilizar o escribir una instrucción o una sentencia de control, sin embargo, se deberá respetar el formato especificado en la sección 5.3.1. Únicamente se permite escribir una instrucción o sentencia de control por línea y en cuanto al número de parámetros que se pueden escribir en una sentencia de control o instrucción la limitante es de un máximo de 255 caracteres por línea.

### b) Limitaciones en la extensión del programa de aplicación

Como PROSAC es un programa que permite al usuario generar sus propios programas de aplicación para el SAC, resulta difícil definir la máxima extensión que puede alcanzar el programa de aplicación. Sin embargo, en PROSAC la ventana de edición puede contener un máximo de 64 Kbytes; es decir, únicamente se pueden escribir programas de aplicación (archivos.txt) que ocupen máximo 64 Kbytes.

Por otra parte, se debe tener en cuenta que en la memoria interna RAM se almacenará finalmente el programa de aplicación en forma de un archivo en formato S19, y la capacidad máxima de ésta memoria es de 32 Kbytes. Por lo anterior, el archivo en formato S19 más grande que puede almacenarse en la memoria RAM es de 32 Kbytes.

Finalmente, cuando el SAC está en modo de recepción, recibiendo un programa de aplicación y éste rebasa la capacidad de la memoria interna RAM, entonces se envía un mensaje de error a la PC indicándole al usuario que el programa de aplicación es demasiado grande.

### Limitaciones de Windows

Muchas de las limitaciones de PROSAC, están impuestas por Microsoft Windows. Estas limitaciones cambian cuando se instala una versión diferente de Windows. PROSAC ha sido probado bajo Windows 3.11 y Windows 95' (versiones en Español e Inglés) y ha funcionado sin ningún problema.

Sin embargo, se debe tomar en cuenta que cada vez que se abre una nueva ventana en Windows se utilizan recursos del sistema, por lo que es recomendable verificar el porcentaje de recursos del sistema Windows utilizados (ver opción "About" del menú "Help" en el "Program manager" o "File manager"). Para reducir este porcentaje de recursos utilizados, se deben cerrar ventanas y terminar con las aplicaciones que estén ejecutándose.

## VI INTEGRACIÓN E INSTALACIÓN DEL SAC

En este capítulo se muestra cómo se integra, interconecta, configura e instala el SAC. Además, en las figuras 6.1, 6.2, 6.3 y 6.4 se muestran fotografías de distintas vistas del prototipo del SAC y del programa PROSAC.

### 6.1 Diagrama de interconexión

Como se indicó en el capítulo 2, el SAC es un sistema de adquisición y control de propósito general que está formado por varias tarjetas interconectadas entre sí (ver figura 6.2). Las tarjetas que constituyen al SAC son las siguientes:

- Tarjeta del sistema de control y procesamiento
- Tarjeta del módulo acondicionador de señales y puerto de detección (PTOD)
- Tarjeta del módulo de despliegue y teclado de control
- Tarjeta del puerto de entrada 2 (pte2) y puerto de salida 2 con interfaz de potencia (ptac)
- Tarjeta de la unidad de almacenamiento de datos UAD-PCMCIA

La tarjeta principal es la de control y procesamiento y las demás tarjetas están conectadas a ella a través de sus puertos. Estos son:

- Puerto serie-paralelo (conector P1)
- Puerto especial (conector P2)
- Puerto de expansión (conector P3)
- Puerto de entrada 1 - salida 1 (conector P4)
- Puerto de entrada 2 - salida 2 (conector P5)

A estos puertos se tiene acceso a través de los conectores P1 a P5 arriba señalados. Además, la tarjeta principal cuenta otros 2 conectores (P13 y P14) para la alimentación del sistema y la batería de respaldo, respectivamente. En la figura 6.5 se muestra el diagrama de interconexión.

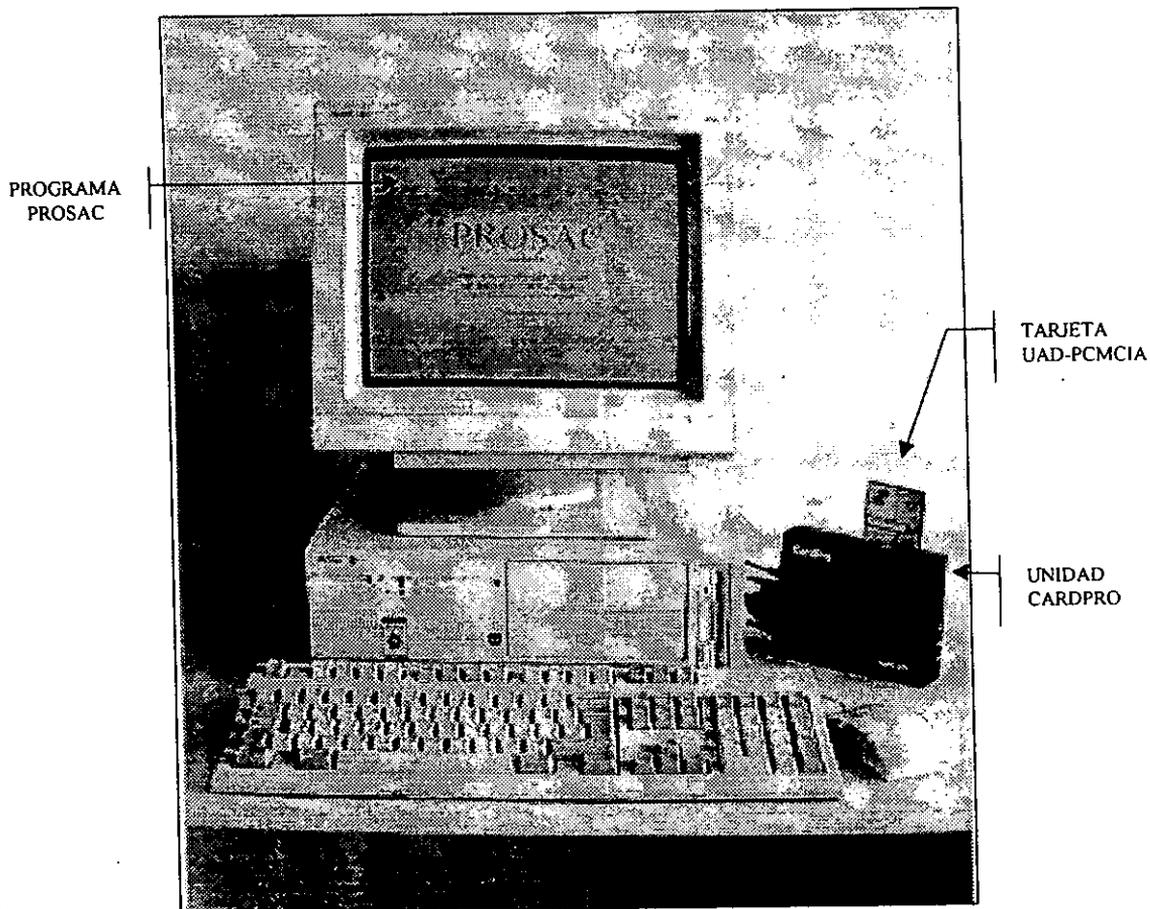


Figura 6.1. Vista de la ventana de inicio del programa PROSAC, unidad CARDPRO y tarjeta UAD-PCMCIA.

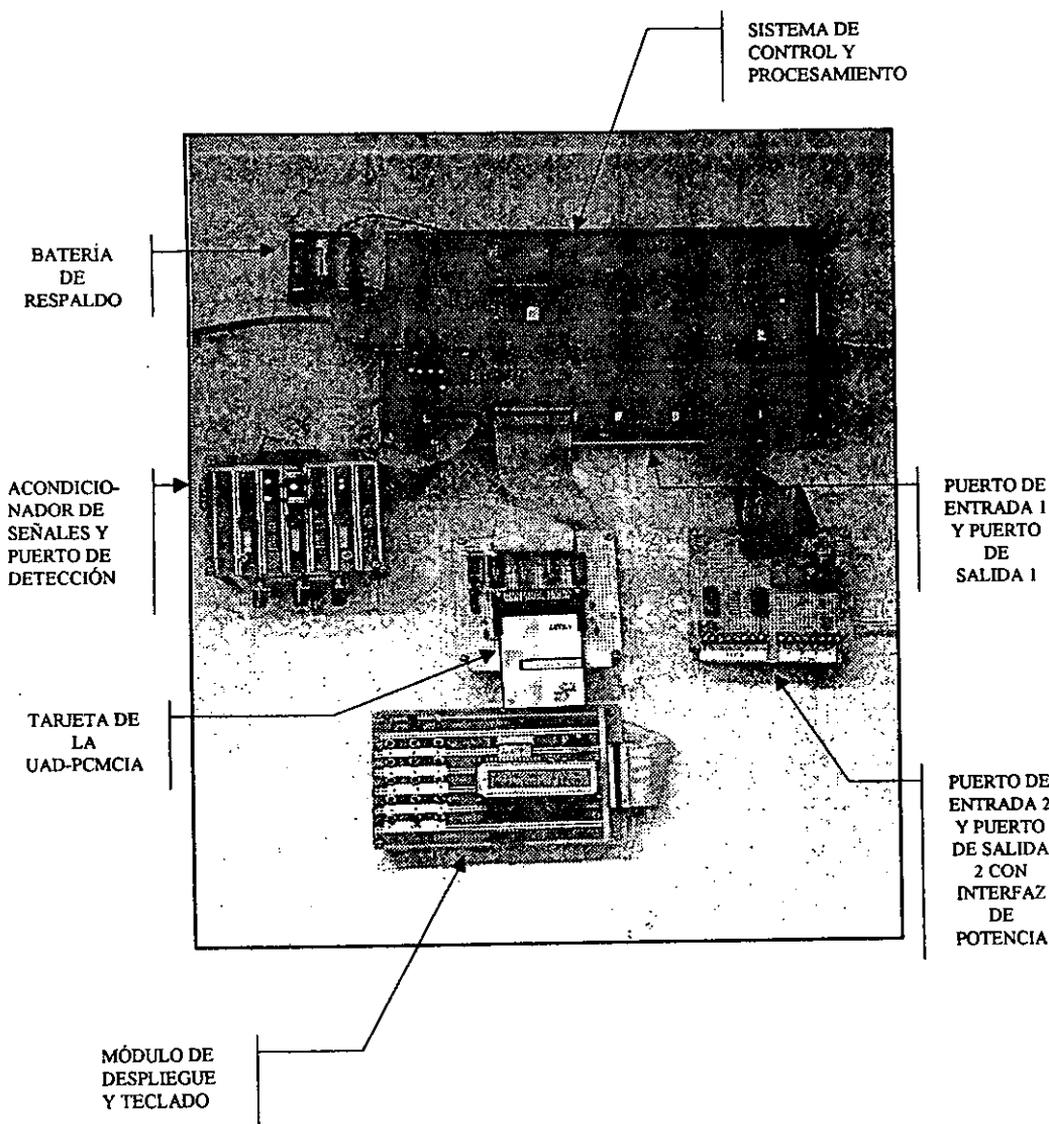


Figura 6.2. Prototipo del SAC.

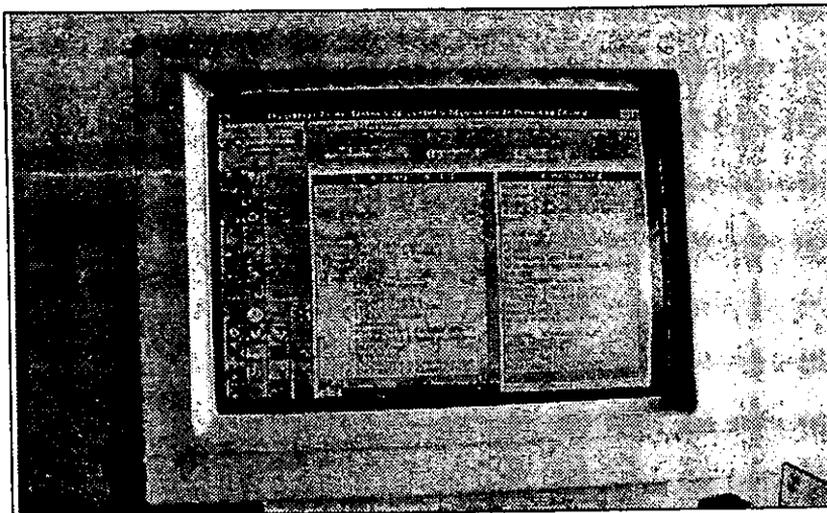


Figura 6.3. Ventana de edición del programa PROSAC.

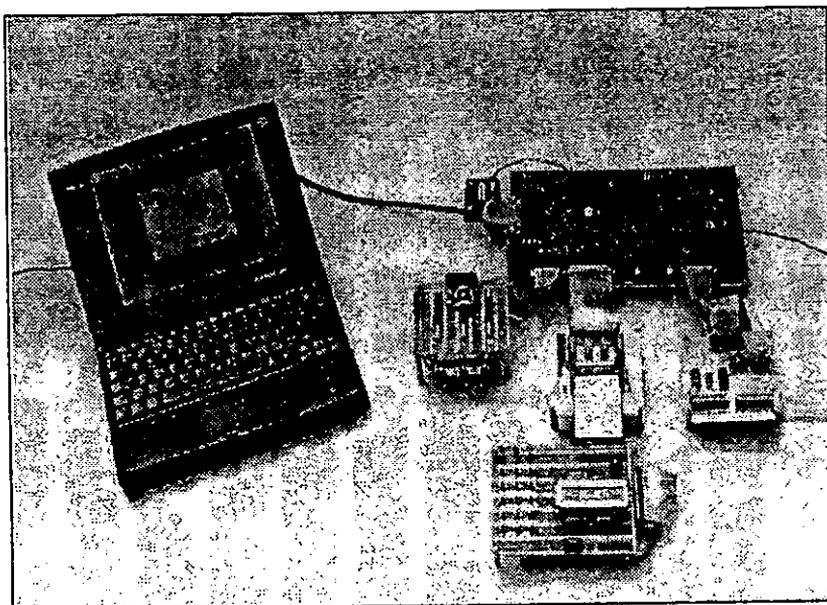


Figura 6.4. El programa PROSAC y el SAC.



## 6.2 Configuración del Hardware

Antes de poder utilizar el SAC es necesario configurar y preparar adecuadamente algunos de sus componentes, tales como puentes, conectores y cables (ver figuras 6.2 y 6.5).

En la tabla 6.1 se indica la función de los puentes y su estado normal.

PUENTE	FUNCIÓN	ESTADO NORMAL
JP1	No puesto: habilitación del modo expandido del microcontrolador. Puesto: habilitación del modo especial del microcontrolador.	NO PUESTO
JP2, JP3	Puestos habilitan las líneas de control RDY y ACK del puerto paralelo.	NO PUESTOS
JP4 - JP7	Habilitan la comunicación del SAC con una PC por el puerto serie asíncrono.	PUESTOS: JP4 y JP6 NO PUESTOS: JP5 y JP7
JP8 - JP13	Habilitan la presencia del puerto serie sincrónico en el conector P4	NO PUESTOS
JP14 - JP19	Habilitan la presencia del puerto serie sincrónico en el conector P5	NO PUESTOS
JA	Puesto: Voltaje de alimentación externo de +5 VCD	A decidir dependiendo de la fuente externa de alimentación
JB	Puesto: Voltaje de alimentación externo entre +9 VCD y +18 VCD	A decidir dependiendo de la fuente externa de alimentación

**IMPORTANTE: JA Y JB NO SE DEBEN INSTALAR SIMULTÁNEAMENTE**

**Tabla 6.1. Función de los puentes.**

El puente JP1 permite seleccionar el modo de operación del microcontrolador. El modo de operación normal del microcontrolador es el modo expandido, por lo que el puente JP1 no deberá instalarse.

La colocación de los puentes JP2 y JP3 habilitan las líneas de control RDY (*ready* o listo) y ACK (*acknowledge* o reconocimiento) del puerto paralelo. No se pueden utilizar simultáneamente el puerto paralelo y el puerto serie sincrónico por lo que no se deben instalar estos puentes cuando se encuentran colocados los puentes JP8 a JP13, o bien, los puentes JP14 a JP19. Normalmente los puentes JP2 y JP3 no se instalan.

Los puentes JP4 al JP7 habilitan la comunicación entre el puerto serie asíncrono del SAC (la interfaz RS232-C) y el puerto serie de una PC. Normalmente la comunicación entre el SAC y la PC es una comunicación serie asíncrona sin protocolo, por lo que únicamente se manejan las líneas de TX, RX y GND y en este caso se instalan solamente los puentes JP4 y JP6.

Los puentes JP8 a JP13 al instalarse permiten habilitar la presencia del puerto serie síncrono en el conector P4. No se deben instalar estos puentes cuando se encuentran instalados los puentes JP2 y JP3 (puerto paralelo), y los puentes JP14 a JP19 (puerto serie asíncrono en el conector P5). Normalmente los puentes JP8 a JP13 no se instalan.

Los puentes JP14 a JP19 permiten habilitar la presencia del puerto serie síncrono en el conector P5. No se deben instalar cuando se encuentran instalados los puentes JP2 y JP3 (puerto paralelo), y los puentes JP8 a JP13 (puerto serie asíncrono en el conector P4). Normalmente los puentes JP14 a JP19 no se instalan.

Los puentes JA y JB se usan para seleccionar el voltaje de alimentación que será aplicado al SAC. Si se instala el puente JA, el usuario deberá suministrar al SAC con un voltaje de entrada fijo de 5VCD@200 mA a través del conector P13. Si no se cuenta con una fuente de 5VCD, se podrá alimentar al SAC con otra fuente de voltaje entre 9 y 18 VCD@200 mA a través del conector P13 instalando el puente JB y quitando el puente JA. Es muy importante que **únicamente se instale uno de los dos puentes, nunca se deben instalar los dos puentes simultáneamente.**

### 6.3 Puertos y conectores no accesibles al usuario

Existen varios conectores que el usuario no puede utilizar ya que éstos interconectan a la tarjeta principal del sistema de control y procesamiento con los distintos módulos o periféricos del SAC. Una descripción de cada cable y su conector correspondiente se da en el Anexo B.

A continuación se indica cuales son los conectores no accesibles al usuario y qué módulos o periféricos se pueden interconectar a la tarjeta principal (ver figura 6.5).

Los conectores P2 y P11 permiten la interconexión (a través del cable 2) de la tarjeta del sistema de control y procesamiento con la tarjeta del módulo acondicionador de señales y puerto de detección (PTOD).

Los conectores P3 y P6 permiten la interconexión (a través del cable 3) de la tarjeta del sistema de control y procesamiento con la tarjeta del módulo de despliegue y teclado.

Los conectores P3 y P7 permiten la interconexión (a través del cable 3) de la tarjeta del sistema de control y procesamiento con la tarjeta de la unidad de almacenamiento de datos UAD-PCMCIA.

Los conectores P5 y P8 permiten la interconexión (a través del cable 3) de la tarjeta del sistema de control y procesamiento con la tarjeta de los puertos de entrada 2 (pte2) y puerto de salida 2 con interfaz de potencia (ptac).

El conector P13 permite la interconexión de la fuente externa de alimentación ( 5VCD@200 mA; o bien, de 9 a 18VCD@200 mA) al SAC.

El conector P14 permite la interconexión de una batería (3 pilas tipo AA de 1.5V) para las funciones de respaldo del sistema.

#### 6.4 Puertos y conectores accesibles al usuario

Como se indicó en el capítulo 5 (ver tabla 5.2), los puertos y entradas a los que se les pueden conectar dispositivos externos y que están accesibles al usuario son los siguientes:

- “pts1” puerto de salida 1 con 8 salidas disponibles (0-5VCD @ 35mA por salida).
- “ptac” puerto de salida 2 con 8 salidas disponibles (125VCA @ 2A por salida).
- “ptop” puerto paralelo de salida con 8 salidas disponibles (0-5VCD @ 35mA por salida).
- “pte1” puerto de entrada 1 con 8 entradas disponibles (0-5VCD compatibles con TTL).
- “pte2” puerto de entrada 2 con 8 entradas que permiten manejar voltajes de entrada entre 0 y 18VCD.
- “PTOD” puerto de detección con 3 entradas disponibles que permiten manejar voltajes de entrada entre 0 y 18VCD.
- 4 canales de entrada que permiten manejar señales analógicas con diferentes rangos de amplitud y frecuencia.

El usuario puede tener acceso a estos puertos diversos a través de los siguientes conectores:

El conector P1 (a través del cable 1) permite la interconexión del SAC con una computadora PC (puerto serie asíncrono y puerto paralelo “ptop”).

El conector P4 permite al usuario tener acceso al puerto de entrada 1 (“pte1”) y a puerto de salida 1 (“pts1”).

El conector P9 permite al usuario tener acceso al puerto de salida 2 con interfaz de potencia (“ptac”).

El conector P10 permite al usuario tener acceso al puerto de entrada 2 (“pte2”).

El conector P12 permite al usuario tener acceso a las 3 entradas del puerto de detección (“PTOD”).

Los 4 canales de entrada que manejan señales analógicas y las perillas SW1 a SW4 que seleccionan el rango del voltaje de entrada de cada una, se encuentran en la tarjeta del módulo acondicionador de señales y puerto de detección (ver figuras 2.14 y 6.5).

Por otra parte, el botón de RESET localizado en la tarjeta de control y procesamiento, permite al usuario inicializar el programa del SAC (ver figura 6.5).

#### 6.5 Procedimiento de instalación

Para poder programar al SAC, se requiere de una fuente de voltaje, una batería de respaldo y una computadora personal.

**a) Fuente de voltaje y batería de respaldo**

La interconexión de la fuente de voltaje de alimentación con SAC se realiza a través del conector P13.

Además, de la fuente de la fuente de voltaje, el usuario deberá conectar la batería (3 pilas AA de 1.5V) en el conector P14 para que el SAC pueda realizar las funciones de respaldo (ver figuras 6.1, 6.5, sección 2.2 y anexo B).

**b) Conexión del SAC a una computadora personal PC**

La interconexión del SAC con una computadora personal PC para una comunicación serial compatible con el estándar RS232-C, se realiza por medio del cable 1 (ver figuras 6.1, 6.5 y anexo B).

El SAC también puede conectarse a una PC o a otros dispositivos externos para la comunicación de datos a través del puerto paralelo. Las líneas de control y de datos del puerto paralelo se encuentran disponibles al usuario en el conector P1 (ver anexo B).

**6.6 Cables y conectores**

Los distintos cables y conectores requeridos para el SAC se describen con detalle en el anexo B.

## VII EJEMPLOS DE PROGRAMAS DE APLICACIÓN

En este capítulo, se presentan algunos ejemplos de programas de aplicación muy sencillos que permitirán conocer mejor las instrucciones y sentencias de control de flujo. Posteriormente se presenta un ejemplo de aplicación más completo denominado "Transferencia de productos en las líneas de producción", el cual dará una buena idea al lector de lo fácil que es programar al SAC por medio de la interfaz gráfica, y lo útil y versátil que resulta para resolver problemas en el área industrial.

### 7.1 Programa de aplicación utilizando puertos de entrada y salida

A continuación se presenta un listado del programa de aplicación denominado "puertos.txt", el cual ejemplifica la utilización de las instrucciones asociadas a los puertos de entrada y salida del SAC. Por medio del texto que aparece dentro de los paréntesis de la instrucción TEXTASER(...) se explica el flujo del programa.

#### "PUERTOS.TXT"

```

TEXTASER(1.- PRENDER TODAS LAS SALIDAS DE LOS PUERTOS DE SALIDA)
  PRENDETS()
  PAUSA(5,seg)
TEXTASER(2.- APAGAR TODAS LAS SALIDAS DE LOS PUERTOS DE SALIDA )
  APAGATS()
  PAUSA(5,seg)
TEXTASER(3.- PRENDER EL LED ROJO)
  PRENDELED(led2)
  PAUSA(5,seg)
TEXTASER(4.- PRENDER EL LED AMARILLO)
  PRENDELED(led3)
  PAUSA(5,seg)
TEXTASER(5.- APAGAR EL LED ROJO)
  APAGALED(led2)
  PAUSA(5,seg)
TEXTASER(6.- APAGAR EL LED AMARILLO)
  APAGALED(led3)
  PAUSA(5,seg)
TEXTASER(7.- ESCRIBIR EL DATO HXF0 EN EL PUERTO DE SALIDA PTS1 )
  ESCRIBEPS(PTS1,HXF0)
  PAUSA(5,seg)
TEXTASER(8.- ESCRIBIR EL DATO 15 EN EL PUERTO DE SALIDA PTS1 )
  ESCRIBEPS(PTS1,15)
  PAUSA(5,seg)
TEXTASER(9.- PRENDER LAS SALIDAS 8 Y 5 DEL PUERTO DE POTENCIA PTAC)
  PRENDESAL(PTAC, 144)
  PAUSA(5,seg)
TEXTASER(10.- APAGAR LAS SALIDAS 8 Y 5 DEL PUERTO DE POTENCIA PTS1)
  APAGASAL(PTAC, 144)
  PAUSA(5,seg)
TEXTASER(11.- LEER EL PTE1 Y GUARDAR EL DATO EN LA VARIABLE VAR1)
  LEEPE(PTe1,var1)
  PAUSA(5,seg)
TEXTASER(12.- PONER EL VALOR DE LA VARIABLE VAR1 EN EL PUERTO PTS1)
  ESCRIBEPS(PTS1,var1)

```

```

    PAUSA(5,seg)
TEXTASER(13.- DETECTAR UNA TRANSICION EN LA ENTRADA 1 DEL PUERTO DE DETECCION)
TEXTASER(    Y LUEGO PONER EN EL PUERTO DE SALIDA PTS1 EL VALOR DE EVC1)
    DETECTRANS(1, 3)
    ESCRIBEPS(PTS1,EVC1)
    PAUSA(5,seg)
TEXTASER(14.- TRANSMITIR EL DATO HX3F = "7" AL PUERTO SERIE)
    TXSER(HX3F)
    PAUSA(5,seg)
TEXTASER(15.- RECIBIR UN DATO POR EL PTO SERIE Y A GUARDARLO EN LA VARIABLE VAR2)
TEXTASER(    PARA LUEGO DESPLEGARLO EN PTS1 Y RETRANSMITIRLO AL PTO SERIE)
    TXSER(VAR2)
    ESCRIBEPS(PTS1,var2)
    PAUSA(5,seg)

```

## 7.2 Programa de aplicación utilizando la unidad de almacenamiento de datos UAD-PCMCIA

El siguiente programa, denominado "uad.txt", ejemplifica la utilización de las instrucciones asociadas al almacenamiento de datos. Al igual que en el ejemplo anterior, por medio del texto que aparece dentro de los paréntesis de la instrucción TEXTASER(...) se explica el flujo del programa.

### "UAD.TXT"

```

TEXTASER(1.- BORRA LA UNIDAD DE ALMACENAMIENTO DE DATOS UAD-PCMCIA)
    BORRAUAD()
    PAUSA(5,seg)
TEXTASER(2.- ESCRIBE EL TEXTO "!! HOLA MUNDO !! EN LA UAD- PCMCIA )
TEXTASER(    A PARTIR DE LA DIRECCION HEXADECIMAL: HXE0000 )
    TEXTTOAUAD(HXE0000,!! HOLA MUNDO !!)
    PAUSA(5,seg)
TEXTASER(3.- LEE EL BLOQUE DE DATOS DE LA UAD-PCMCIA DE LA DIRECCION DE INICIO
HXE0000)
TEXTASER(    Y HASTA LA DIRECCION DE FIN HXE0012 , Y ENVIA LOS DATOS AL PUERTO
SERIE)
    LEEBLOQUAD(HXE0000,HXE0012)
    PAUSA(5,seg)
TEXTASER(4.- GUARDA EN LA DIRECCION HXA123D DE LA UAD-PCMCIA EL DATO: HX4F)
TEXTASER(    Y EN LA DIRECCION HXE4000 EL VALOR DE LA VARIABLE VAR2)
    ESCRIBEUAD(HXA123D,HX4F)
    ESCRIBEUAD(HXE4000,var2)
    PAUSA(5,seg)
TEXTASER(5.- LEE EL DATO QUE SE ENCUENTRA EN LA DIRECCION HXE4000 DE LA )
TEXTASER(    UAD-PCMCIA Y LO GUARDA EN LA VARIABLE VAR1)
    UADAVAR(HXE4000,var1)
    PAUSA(5,seg)
TEXTASER(6.- ENVIA AL PUERTO SERIE TODA LA INFORMACION CONTENIDA EN LA UAD-
PCMCIA)
    TRANSUAD()
    PAUSA(5,seg)

```

### 7.3 Programa de aplicación utilizando el reloj fechador, el módulo de despliegue y las instrucciones diversas

A continuación se presenta un listado del programa de aplicación denominado "relojlcd.txt", el cual ejemplifica la utilización de las instrucciones asociadas al reloj fechador, al módulo de despliegue y las instrucciones diversas. Como se vio en los ejemplos anteriores, por medio del texto que aparece dentro de los paréntesis de la instrucción TEXTASER(...) se explica el flujo del programa.

#### "RELOJLCD.TXT"

```

TEXTASER(1.-INICIALIZA LA FECHA Y HORA DEL SAC CON:)
TEXTASER( Domingo 31 de Dic de 1999 , 23:59:55)
    PAUSA(5,seg)
    PONRELOJ(7, 31/12/99, 23:59:55)
    PAUSA(5,seg)
TEXTASER(2.-LEE LA FECHA Y HORA DEL SAC Y LA ENVIA AL PUERTO SERIE)
    PAUSA(5,seg)
    LEERELOJ()
    PAUSA(5,seg)
TEXTASER(3.-LEE LA FECHA Y HORA DEL SAC Y LA GUARDA EN LA UAD-PCMCIA)
TEXTASER( A PARTIR DE LA DIRECCION HXEDF00 )
    PAUSA(5,seg)
    RELOJAUAD(hxedf00)
    PAUSA(5,seg)
TEXTASER(4.-LEE LOS PRIMEROS 25 DATOS DE LA UAD-PCMCIA A PARTIR DE LA )
TEXTASER( DIRECCION HXEDF00 )
    PAUSA(5,seg)
    LEEUAD(hxedf00,25)
    PAUSA(5,seg)
TEXTASER(5.- EN 3 seg SE BORRARA LA PANTALLA DEL VISUALIZADOR LCD)
    PAUSA(3,seg)
    BORRAVIS()
    PAUSA(2,seg)
TEXTASER(6.-PONER EL TEXTO 'IHOLA MUNDO!' EN EL 1er RENGLON DEL LCD)
    PAUSA(3,seg)
    ESCRIBEVIS( 1, 3,IHOLA MUNDO!)
    PAUSA(3,seg)
TEXTASER(7.-PONER EL TEXTO ' How are you?' EN EL 2do RENGLON DEL LCD)
    PAUSA(2,seg)
    ESCRIBEVIS( 2, 3,How are you?)
    PAUSA(2,seg)
TEXTASER(8.-PONER var1=hex00 E INCREMENTAR var1 HASTA EL VALOR hxff)
TEXTASER( DESPLEGANDO SU VALOR EN EL PUERTO PTS1 Y EL VISUALIZADOR LCD)
    BORRAVIS()
    PAUSA(5,seg)
    DATOAVAR(var1,hex00)
    REPITE(256)
        ESCRIBEPTS(PTS1,var1)
        VARAVIS(var1)
        INCVAR(var1,+1)
    PAUSA(1,ds)
    FINREPITE()
    PAUSA(5,seg)

```

```

TEXTASER(8.-PONER var2=hexFF Y DECREMENTAR var2 HASTA EL VALOR hex00 )
TEXTASER( DESPLEGANDO SU VALOR EN EL PUERTO PTS1 Y EL VISUALIZADOR LCD)
    BORRAVIS()
    PAUSA(5,seg)
    DATOAVAR(var2,hexff)
    REPITE(256)
        ESCRIBEPS(PTS1,var2)
        VARAVIS(var2)
        INCVAR(var2,-1)
        PAUSA(1,ds)
    FINREPITE()
    PAUSA(5,seg)
TEXTASER(9.-PRENDER Y APAGAR LAS SALIDAS 1,3,5 y 7 DEL PUERTO 10 VECES. )
TEXTASER( CON UNA DURACION DE ENCENDIDO Y APAGADO DE 500 ms )
    PAUSA(5,seg)
    INTERMIT(PTS1, 85,10,500)
    PAUSA(5,seg)

```

#### 7.4 Programa de aplicación utilizando el monitoreo de señales

El siguiente programa denominado "monitor.txt", ejemplifica la utilización de las instrucciones asociadas al monitoreo de señales. Al igual que en los ejemplos anteriores, por medio del texto que aparece dentro de los paréntesis de la instrucción TEXTASER(...) se explica el flujo del programa.

#### "MONITOR.TXT"

```

TEXTASER(1.-LEER 12 MUESTRAS DE LA SEÑAL DEL CANAL 1)
    REPITE(12)
    MISENAL( 1)
    FINREPITE()
    PAUSA(5,seg)
TEXTASER(2.-LEER 20 MUESTRAS DE LA SEÑAL DEL CANAL 4, CON UN TIEMPO)
TEXTASER( ENTRE MUESTRA Y MUESTRA DE 500ms)
    REPITE(20)
    MISENAL( 4)
    PAUSA(5,ds)
    FINREPITE()
    PAUSA(5,seg)
TEXTASER(3.-LEER 4 MUESTRAS DE LAS SEÑALES DE LOS 4 CANALES )
    M4SENAL()
    PAUSA(5,seg)
TEXTASER(4.-LEER 32 MUESTRAS DE LOS 4 CANALES CON TIEMPO ENTRE )
TEXTASER( MUESTRA Y MUESTRA DE 1seg )
    REPITE(8)
    M4SENAL()
    PAUSA(1,seg)
    FINREPITE()
    PAUSA(5,seg)
TEXTASER(5.-ALMACENAR 15 MUESTRAS DEL CANAL 1 EN LA UAD A PARTIR DE LA )
TEXTASER( DIRECCION hexdac00, TOMANDO EL TIEMPO ENTRE MUESTRAS DE 1 seg)
    GISENAL( 1,hexdac00,hexdac0f,1,seg)
    PAUSA(3,seg)

```

```

TEXTASER(6.-ALMACENAR 20 MUESTRAS DE LOS 4 CANALES EN LA UAD A PARTIR DE)
TEXTASER( LA DIRECCION hxf0000, TOMANDO EL TIEMPO ENTRE MUESTRAS DE 1 seg)
      G4SEÑAL(hxf0000,hxf0013,1,seg)
      PAUSA(3,seg)
TEXTASER(7.-LEER Y ALMACENAR EN LA UAD 20 MUESTRAS DE LA SEÑAL DEL CANAL 2 )
TEXTASER( A PARTIR DE LA DIRECCION hxabc00, ADEMAS TOMANDO EL TIEMPO )
TEXTASER( ENTRE MUESTRAS DE 500 ms)
      MG1SEÑAL( 2,hxabc00,hxabc13,5,ds)
TEXTASER(8.-LEER 32 MUESTRAS DE LOS 4 CANALES Y ALMACENARLAS EN LA UAD A )
TEXTASER( PARTIR DE LA DIRECCION hxa0000, ADEMAS TOMANDO EL TIEMPO )
TEXTASER( ENTRE MUESTRAS DE 1 seg)
      MG4SEÑAL(hxa0000,hxa001f,1,seg)
      PAUSA(5,seg)

```

### 7.5 Programa de aplicación utilizando las sentencias de control de flujo y el monitoreo de señales

A continuación se presenta un listado del programa de aplicación denominado "control.txt", el cual ejemplifica la utilización de las instrucciones asociadas al monitoreo de señales y las sentencias de control de flujo. Por medio del texto que aparece dentro de los paréntesis de la instrucción TEXTASER(...) se explica el flujo del programa.

#### "CONTROL.TXT"

```

TEXTASER(1.-LEER LAS MUESTRAS DEL CANAL 3, SI EL VALOR DE LA MUESTRA ES)
TEXTASER( SUPERIOR O IGUAL A 122=hex7A, O SI EL VALOR DE LA MUESTRA ES)
TEXTASER( INFERIOR O IGUAL A 48=hex30 ENTONCES SE DESPLEGARA UN MENSAJE)
TEXTASER( Y SE PRENDERAN 3 VECES LAS SALIDAS DEL PUERTO PTS1. )
PAUSA(5,seg)

```

```

ETIQUETA(CANAL3)
MISEÑAL( 3)

```

```

SI( mues <= hex30 )
      TEXTASER(!!! EL VALOR DE LA MUESTRA ES <= hex30 !!!)
      LEERELOJ()
      INTERMIT(PTS1, 255,3,1000)
      SALTA(OPCION2)

```

```

FINSI()

```

```

SI( mues >= hex7A )
      TEXTASER(!!! EL VALOR DE LA MUESTRA ES >= hex7A !!!)
      LEERELOJ()
      INTERMIT(PTS1, 255,3,1000)
      SALTA(OPCION2)

```

```

FINSI()
SALTA(CANAL3)

```

```

ETIQUETA(OPCION2)

```

```

TEXTASER(2.-LEER LAS MUESTRAS DE LOS 4 CANALES Y SI LAS 4 SON )
TEXTASER( MAYORES O IGUALES A hex5A SE DESPLEGARA UN MENSAJE)
TEXTASER( Y SE PRENDERAN 3 VECES LAS SALIDAS 2,4,6 y 8 DEL PUERTO PTS1)
PAUSA(5,seg)

```

```

ETIQUETA(CANALES)
M4SENALQ

SI( mus1 >= HX5A Y mus2 >= HX5A Y mus3 >= HX5A Y mus4 >= HX5A )
    TEXTASER(!!! EL VALOR DE LAS 4 MUESTRAS ES >= hx5A !!!)
    LEERELOJQ
    INTERMIT(PTS1, 170.3,1000)
    SALTA(REPETIR)
FINSIQ

SALTA(CANALES)

ETIQUETA(REPETIR)
TEXTASER(3.- DESPLEGAR EL VALOR DE var2 256 VECES EN EL VISUALIZADOR LCD)
TEXTASER( Y PUERTO DE SALIDA PTS1)
PAUSA(5,seg)
DATOAVAR(var2,hx00)
BORRAVISQ

REPITE(256)
    VARAVIS(var2)
    ESCRIBEVIS( 2, 14, ms)
    ESCRIBEVIS( 1, 1,Valor)
    ESCRIBEPS(PTS1,var2)
    INCVAR(var2,+1)
    PAUSA(1,ds)
FINREPITEQ

PAUSA(5,seg)
TEXTASER(4.-MIENTRAS LA MUESTRA DEL CANAL 2 SEA INFERIOR A hx7A, ENTONCES )
TEXTASER( VEER SU VALOR EN EL PUERTO PTS1)
PAUSA(5,seg)

MIENTRAS( mues <= hx7a )
    MISENAL(2)
    ESCRIBEPS(PTS1,mues)
FINMIENTRASQ

TEXTASER(FIN DEL PROGRAMA >:D )

```

Con este último programa se dan por terminados los ejemplos sencillos de aplicación.

A continuación se explica a detalle un ejemplo de aplicación más completo llamado "Transferencia de productos en las líneas de producción".

Este ejemplo permitirá conocer mejor las instrucciones y sentencias de control de flujo, y dará una buena idea al lector de la versatilidad del SAC.

## 7.6 Ejemplo de aplicación: " Transferencia de productos en las líneas de producción "

### 7.6.1 *La automatización de sistemas y mecanismos de transferencia de productos en las plantas industriales.*

Las necesidades de competitividad en la industria, conducen a automatizar las cadenas de producción. En particular, las transferencias de productos de máquina a máquina se automatizan.

Esos medios de transferencia de productos pueden tomar formas variadas, según el problema a resolver:

- Una simple rampa que transfiere por gravedad el producto de una máquina a otra
- Robot de transferencia que permite tomar el producto en una posición y presentarlo en otra posición a la máquina siguiente.

Dentro del cuadro de una instalación de acondicionamiento de productos lácteos, la realización presentada es una transferencia de palets o tarimas (base de madera donde se ponen los envases de leche) al final de las líneas de producción. Un dispositivo de nave móvil enteramente automatizado dirige los palets hacia una línea de acondicionamiento.

En este caso, el SAC se utiliza para automatizar la transferencia de los productos en las líneas de producción, es decir, para controlar los movimientos de transferencia de los palets, para dar prioridad a las llamadas emitidas por las líneas de producción, y tomar en cuenta los procedimientos de seguridad necesarios. Este ejemplo es una ilustración de la adaptación y versatilidad del SAC en los problemas de automatización que exigen una alta seguridad.

### 7.6.2 *Transferencia de los palets a las salidas de las líneas de producción*

En la figura 7.1 se esquematiza una cadena de una fábrica de acondicionamiento de leche. La leche se introduce en envases de cartón que son enseguida agrupados en 12 por una empaquetadora y finalmente dispuestos automáticamente sobre un palet (720 envases de 1 litro) para su transporte.

La planta de la fábrica comprende varias líneas de producción en paralelo. Una nave móvil está encargada de transferir los paletas que llegan al final de cada una de las líneas de producción hacia una línea de salida para el acondicionamiento final de los palets (envoltura de la paleta en un plástico) y su evacuación.

Este equipamiento ilustrado en la figura 7.2 comprende:

- un transbordador de paleta a la salida de cada línea de producción
- una nave móvil para la transferencia hacia la línea de reagrupamiento
- un transbordador de salida sobre la línea de reagrupamiento

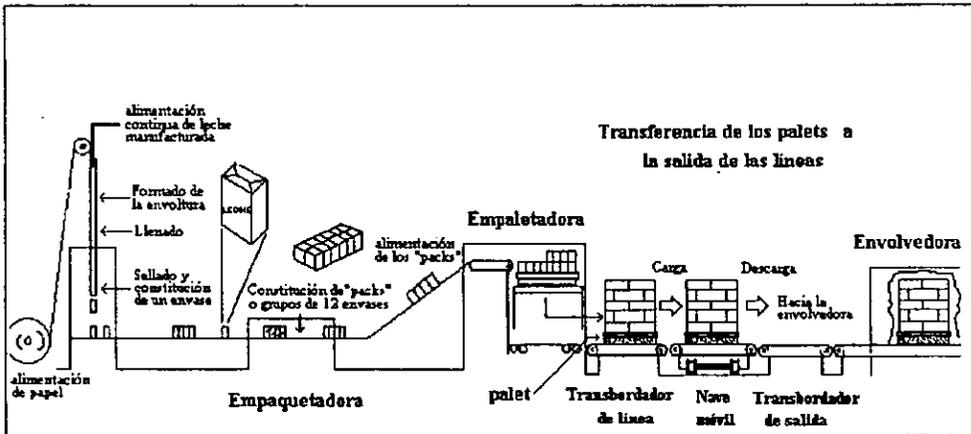


Figura 7.1 Línea de producción en una fábrica de acondicionamiento de leche

### 7.6.3 Presentación general del sistema a automatizar

#### a) Objetivos de la automatización

##### Modo de producción normal

La instalación deberá funcionar en modo automático, sin operador en el cuadro de una producción normal.

##### Problemas de explotación

Las acondicionadoras de los envases a la cabeza de las líneas de producción son difíciles de poner en marcha después de un paro eventual. Por lo tanto, el mantener un flujo de producción regular sobre cada una de las líneas de producción exige la evacuación de las paletas de una manera regular.

La automatización del equipo deberá respetar las prioridades de llamadas de transferencia de cada línea.

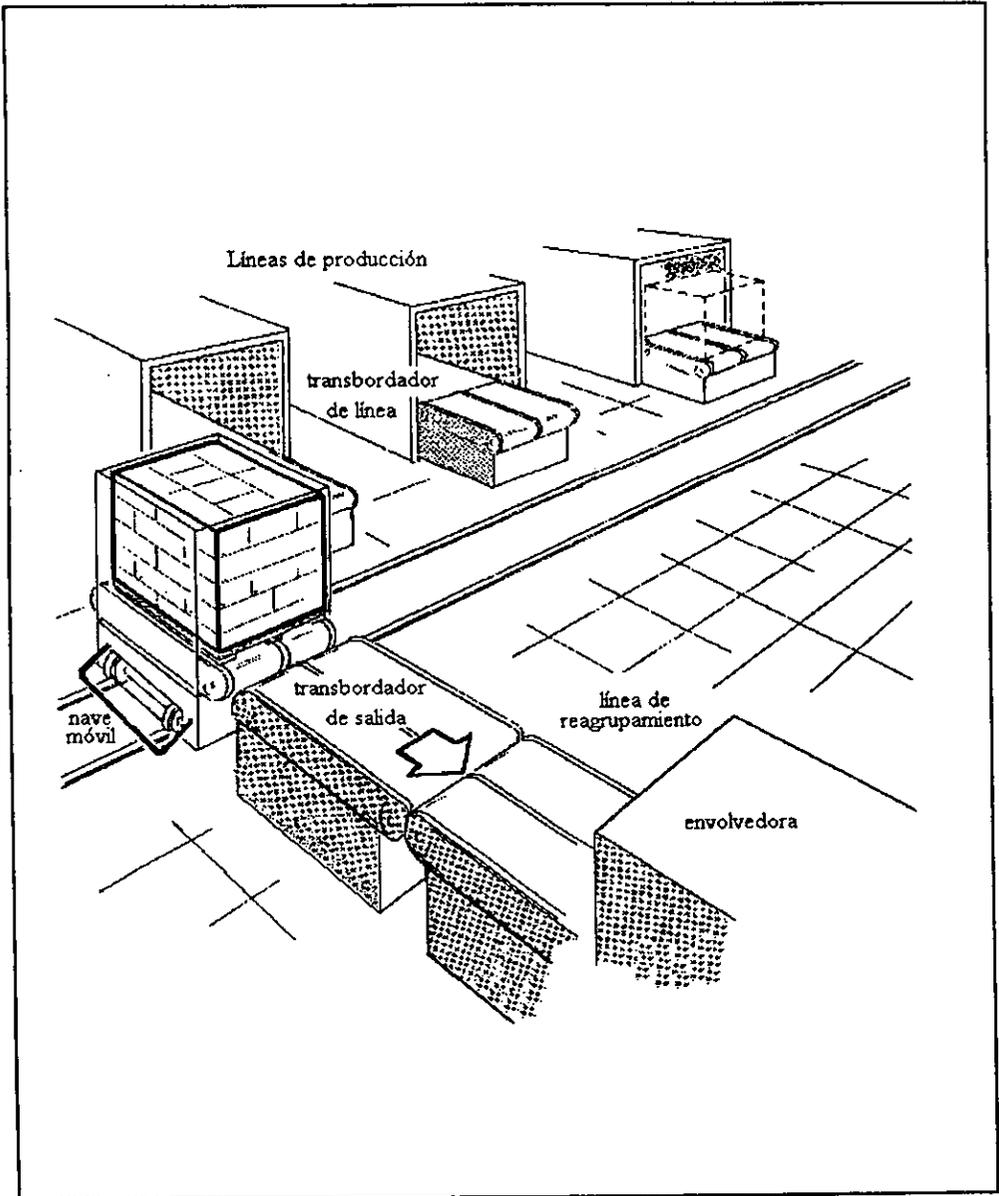


Figura 7.2 Equipamiento de transferencia de los palets

### b) Descripción de la instalación

La descripción de la instalación se muestra en la figura 7.2. Esta instalación de transferencia tiene por fin la evacuación de los palets que contienen los envases de leche y que llegan de las diferentes líneas de producción hacia el equipamiento único de la envolvedora.

El equipamiento presentado en la página siguiente (figura 7.3) comprende:

- un transbordador de línea al final de cada línea de producción, sobre el cual son dirigidas los palets y el cual asegura su transferencia hacia la nave móvil;
- una nave móvil, que tiene igualmente un transbordador para la carga y descarga de los palets. La nave móvil circula perpendicularmente a las líneas de producción y permite la evacuación de los palets hacia la línea de reagrupamiento;
- un transbordador sobre la línea de salida para la evacuación de los palets hacia la envolvedora.

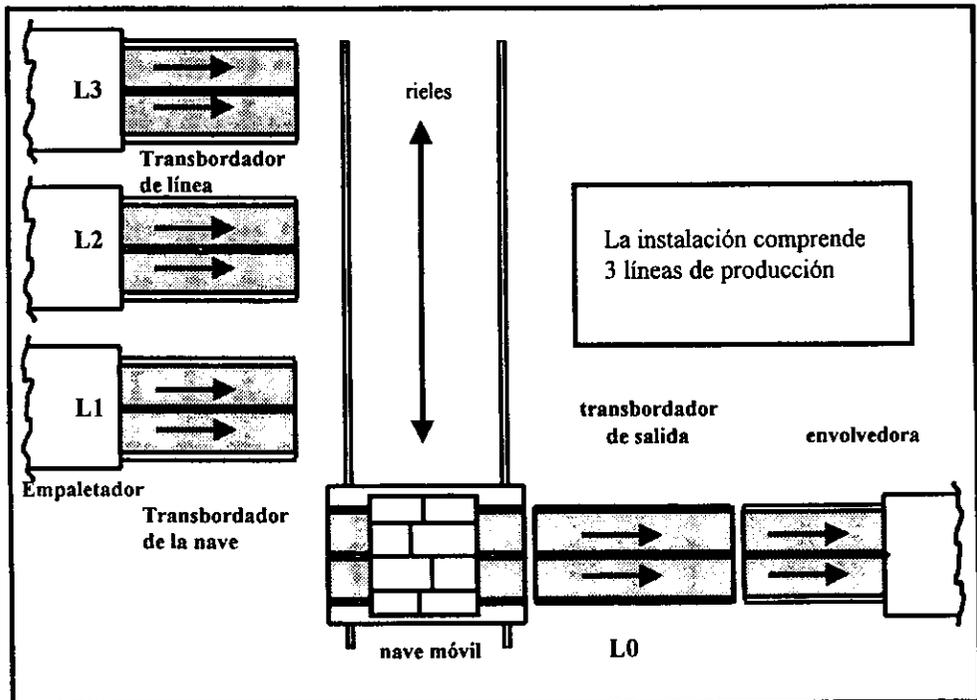


Figura 7.3 La instalación en su posición inicial

c) **Definición de las funciones y de los problemas**

En la tabla 7.1 se describen las funciones a realizar y las restricciones de operación.

Funciones	Problemas
Carga y transferencia	<ul style="list-style-type: none"> <li>• La capacidad del palet (<math>\approx 720</math> envases de leche)</li> <li>• Tomar en cuenta el tiempo de evacuación y la disponibilidad de la línea de producción</li> </ul>
Explotación	<ul style="list-style-type: none"> <li>• No se toleran desbordamientos en las cargas y descargas de palets</li> <li>• Prioridad en las llamadas</li> </ul>
Diálogo	Información del flujo o evolución de las tareas que se están realizando
Comunicación	<ul style="list-style-type: none"> <li>• Información de transbordadores libres</li> <li>• Autorización de descarga en el transbordador de reagrupamiento</li> </ul>
Seguridad de funcionamiento	• Seguridad, análisis de riesgos

Tabla 7.1 definición de las funciones y de los problemas

d) **Principio de funcionamiento**

El modo de producción normal, MPN, corresponde a la situación de espera de una llamada de una de las líneas de producción para la evacuación de un palet terminado. El primer llamado provoca que la nave móvil se dirija a la línea que hizo el llamado. Una vez que la nave móvil se encuentra al lado de la línea correspondiente, la carga del palet sobre la nave se efectúa por medio de la puesta en marcha simultánea del transbordador de la línea y del transbordador de la nave. Cuando ha sido cargado el palet sobre la nave, la nave regresa al lado de la línea de salida L0. Si la línea de salida está libre (autorización de descarga), la descarga del palet se efectúa por medio de la puesta en marcha simultánea del transbordador de la línea de salida y del transbordador de la nave. Cuando el palet se ha transferido de la nave a la línea de salida, la nave se encuentra disponible para atender el llamado siguiente. El diagrama de flujo general de transferencia de productos en las líneas de producción explica el funcionamiento del equipamiento (ver figuras 7.5a, 7.5b, 7.5c y 7.5d).

e) **Seguridad de funcionamiento**• **Seguridad**

La protección de las personas y de los bienes conduce a tomar las soluciones para los problemas siguientes:

- dos "caza-cuerpos" puestos en cada extremo de la nave móvil deben permitir un paro inmediato de seguridad en caso de una colisión
- prohibición de transbordo si la nave no está presente al lado de la línea
- se prohíbe que la nave sobrepase los extremos fijados en las vías
- respeto de las protecciones eléctricas

- **Disponibilidad**

Aparte de los paros en la producción debidos a fallos en los materiales, una de las causas principales de inutilización se debe al desbordamiento de la paleta, por lo que ningún desbordamiento de la paleta se podrá tolerar.

- **Análisis de riesgos**

Para este tipo de equipamiento, los riesgos de daños en las personas y bienes tienen como causas principales:

- una colisión con la nave móvil, tomando en cuenta su velocidad y su peso;
- un desbordamiento del palet después de un transbordo, en caso de un mal posicionamiento al lado de la línea o durante el desplazamiento de la nave;
- una falla en la protección de las partes eléctricas.

**f) Modos de producción**

Las condiciones de operación vistas para este tipo de equipamiento conducen a prever dos modos de producción principales:

- **un modo de producción normal MPN** que debe permitir la transferencia de los palets según la demanda de las diferentes líneas sin intervención de un operador. Este modo de producción debe respetar los problemas de seguridad citados anteriormente. El paro de urgencia deberá traducirse por el paro de todos los accionadores.
- **un ciclo automático de puesta en referencia** de la parte operativa deberá ser accesible cuando el operador así lo demande, bajo la reserva de las condiciones de seguridad a precisar (conducción de la parte operativa al MPN en caso de falla en la seguridad de los bienes o personas).

#### 7.6.4 Estudio de la parte operativa

El estudio de la parte operativa se limita aquí a la selección del equipo necesario para después proseguir con la parte de comando o *software*.

**a) Selección de los accionadores**

La transferencia de las paletas sobre los transbordadores de línea deberá hacerse a velocidad reducida y constante. Los motoredutores eléctricos satisfacen de manera simple y económica estas exigencias.

Cada transbordador utiliza un motoreductor de un solo sentido de marcha (M0, M1...Mm)

La nave móvil comprende:

- un motoreductor M, con dos sentidos de marcha y frenado a falta de corriente, para asegurar el movimiento sobre los rieles (movimientos: avance MA y regreso MR);
  - un motoreductor Mn de un solo sentido de marcha para el accionamiento del transbordador
- La potencia nominal de cada motor es de 3KW.



### Selección del tipo de detector de posición

Entre los diferentes tipos de detectores de posición posibles (detectores electrónicos de proximidad, detectores fotoeléctricos,...), la selección fue el interruptor de posición de contacto.

Este tipo de detector de posición, en efecto, se adapta bien por varias razones:

- facilidad de adaptación mecánica a las líneas de producción o a la nave móvil
- existe una gran variedad en este tipo de interruptores y son fáciles de adaptar
- umbral de mínimo esfuerzo
- fiabilidad y robustez
- conmutación simultánea de varios contactos independientes eléctricamente, lo que facilita la implementación de la seguridad y la lógica de comando.

#### c) Resumen de la selección de los accionadores, de los preaccionadores y de los detectores de posición.

En la tabla 7.2 se muestran los accionadores, preaccionadores y detectores de posición seleccionados.

PARTE OPERATIVA	NAVE MÓVIL		TRANSBORDADOR DE LÍNEA		TRANSBORDADOR DE SALIDA	
Accionadores	<ul style="list-style-type: none"> <li>• 1 motoreductor de dos sentidos de marcha               <ul style="list-style-type: none"> <li>- Avance</li> <li>- Regreso</li> </ul> </li> <li>• 1 motoreductor de un sentido de marcha</li> </ul>	M MA MR Mn	<ul style="list-style-type: none"> <li>• motoreductores de un sentido de marcha</li> </ul>	M1 M2 M3	<ul style="list-style-type: none"> <li>• motoreductor de un sentido de marcha</li> </ul>	M0
Preaccionadores	<ul style="list-style-type: none"> <li>• 2 contactores-inversores</li> <li>• 1 contactor-disyuntor</li> </ul>	KMA KMR KMN	<ul style="list-style-type: none"> <li>• 3 contactores-disyuntores</li> </ul>	KM1 KM2 KM3	<ul style="list-style-type: none"> <li>• 1 contactor-disyuntor</li> </ul>	KM0
Detectores de posición	<ul style="list-style-type: none"> <li>• presencia de la paleta sobre la nave</li> <li>• detección de cuerpos</li> </ul>	Pn CC1 CC2	<ul style="list-style-type: none"> <li>• presencia de la paleta sobre transbordadores</li> <li>• presencia de la nave frente a las líneas</li> <li>• sobrecurso de ida</li> <li>• sobrecurso de regreso</li> </ul>	P1 P2 P3 C1 C2 C3 CF1 CF2	<ul style="list-style-type: none"> <li>• presencia de la paleta sobre el transbordador</li> <li>• Presencia de la nave enfrente de la línea 0</li> </ul>	P0 C0

Tabla 7.2. Resumen de los accionadores, preaccionadores y detectores de posición.

Además, se tiene un accionador de la alarma auditiva "AL" y otros dos detectores: botón de producción normal "PN" y botón de paro de urgencia "PU".

## d) Repartición de las entradas y salidas

Con el fin de simplificar al máximo las adaptaciones futuras que se le puedan hacer al equipamiento, se hizo una repartición de las entradas y salidas en los puertos disponibles en el SAC (ver tablas 7.3 y 7.4).

ACCIONADORES	
Equipamiento	Salida-Puerto
• Transbordadores de cada línea	
línea 0 M0	salida1 - PTAC
línea 1 M1	salida2 - PTAC
línea 2 M2	salida3 - PTAC
línea 3 M3	salida4 - PTAC
• La nave móvil	
transbordador Mn	salida5 - PTAC
un motor con 2 sentidos de	
marcha:	salida6 - PTAC
avance MA	salida7 - PTAC
regreso MR	salida8 - PTAC
• Alarma auditiva AL	

Tabla 7.3. Asignación de los accionadores a los puertos del SAC.

DETECTORES DE POSICIÓN			
Equipamiento	entrada-puerto	Equipamiento	Entrada-puerto
• Detección de posición de la nave		• Detección de presencia de paletas	
nave junto a L0 C0	entrada 1 - PTE1	paleta sobre transbordador de L0 P0	entrada 1 - PTE2
nave junto a L1 C1	entrada 2 - PTE1	paleta sobre transbordador de L0 P1	entrada 2 - PTE2
nave junto a L2 C2	entrada 3 - PTE1	paleta sobre transbordador de L0 P2	entrada 3 - PTE2
nave junto a L3 C3	entrada 4 - PTE1	paleta sobre transbordador de L0 P3	entrada 4 - PTE2
		paleta sobre la nave móvil Pn	entrada 5 - PTE2
• Botón de paro de urgencia PU	entrada 5 - PTE1	• Autorización de transbordo At	entrada 6 - PTE2
• Botón de producción normal PN	entrada 6 - PTE1	• Seguridad de bienes y personas	
• Seguridad de bienes y personas		Sobrepaso de los extremos del riel CF1	entrada 7 - PTE2
Colisión CC1	entrada 7 - PTE1	Sobrepaso de los extremos del riel CF2	entrada 8 - PTE2
Colisión CC2	entrada 8 - PTE1		

Tabla 7.4. Asignación de los detectores de posición a los puertos del SAC.

## 7.6.5 Estudio de la parte de comando o software

En las secciones 7.3.4 a 7.3.5 se explicaron el principio de funcionamiento, la seguridad y los modos de producción. A continuación se presenta el diagrama de flujo general que resume el funcionamiento del equipamiento que tendrá que automatizarse por medio del SAC (ver figuras 7.5a, 7.5b, 7.5c, 7.5d).

Diagrama de flujo general del ejemplo de "Transferencia de productos en las líneas de producción"

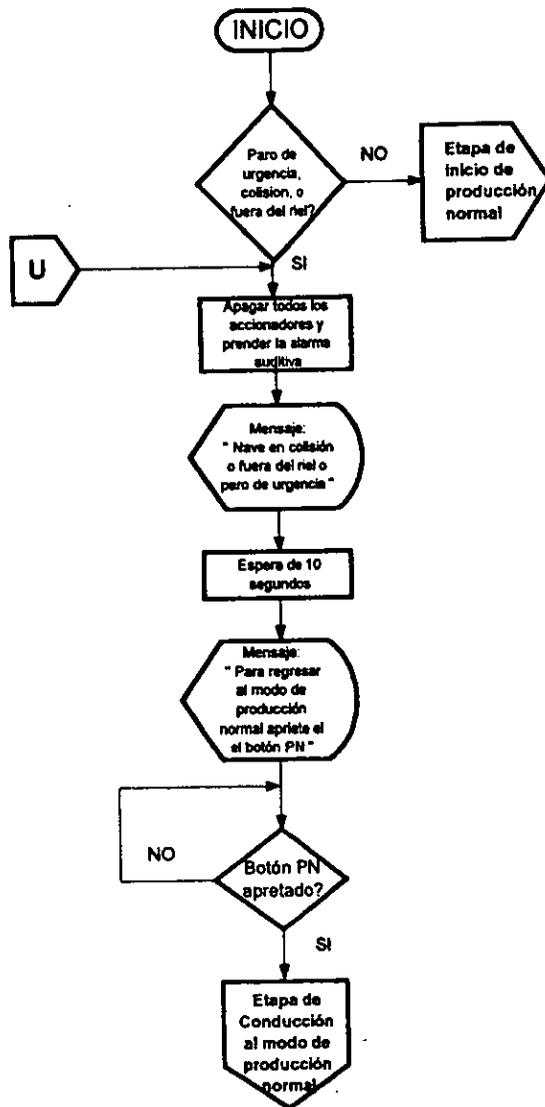


Figura 7.5a Diagrama de flujo general

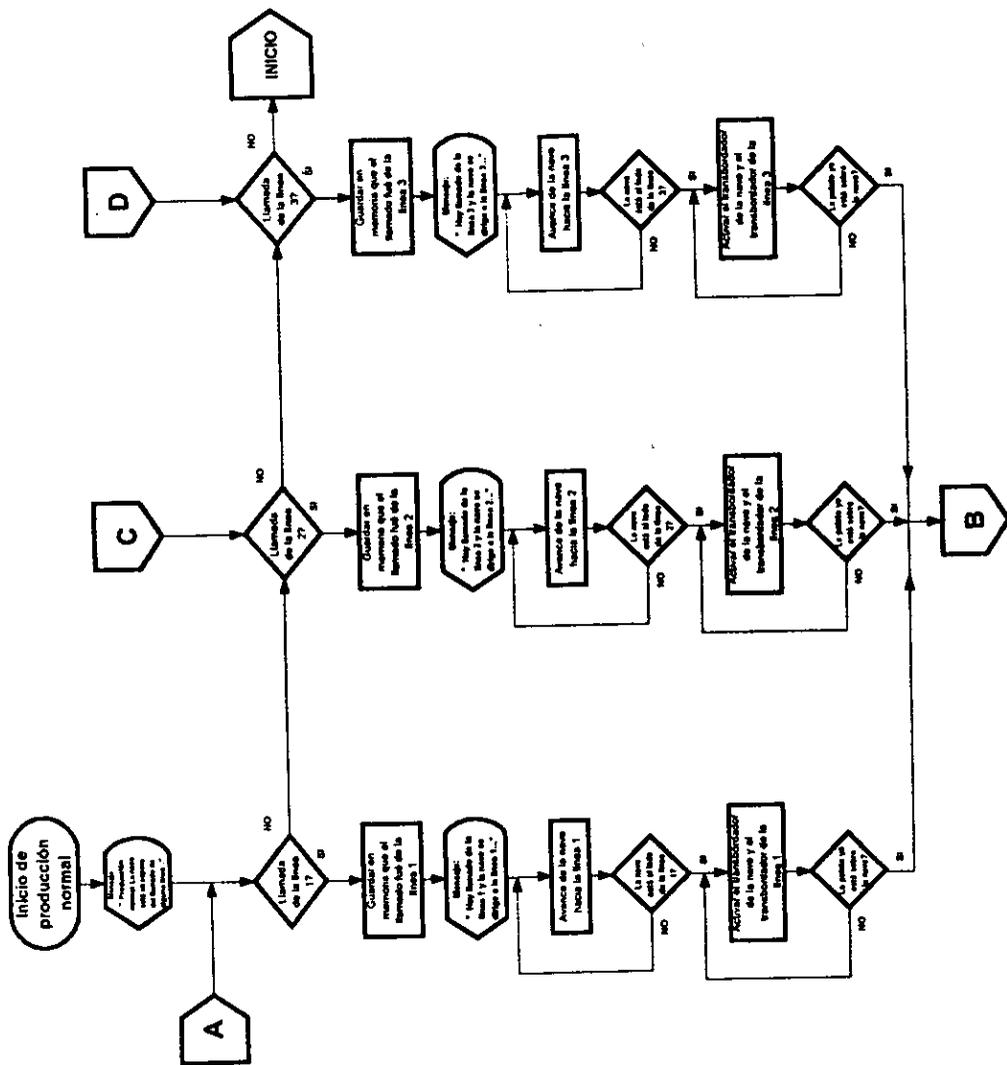


Figura 7.5b Diagrama de flujo general (continuación)

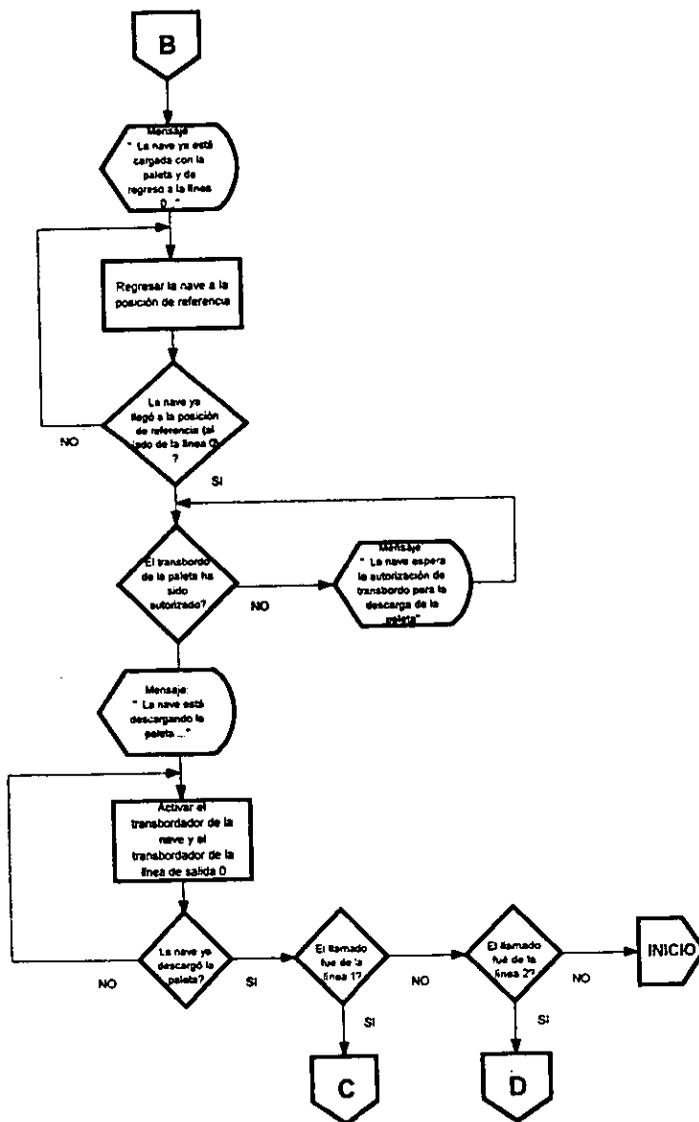


Figura 7.5c Diagrama de flujo general (continuación)

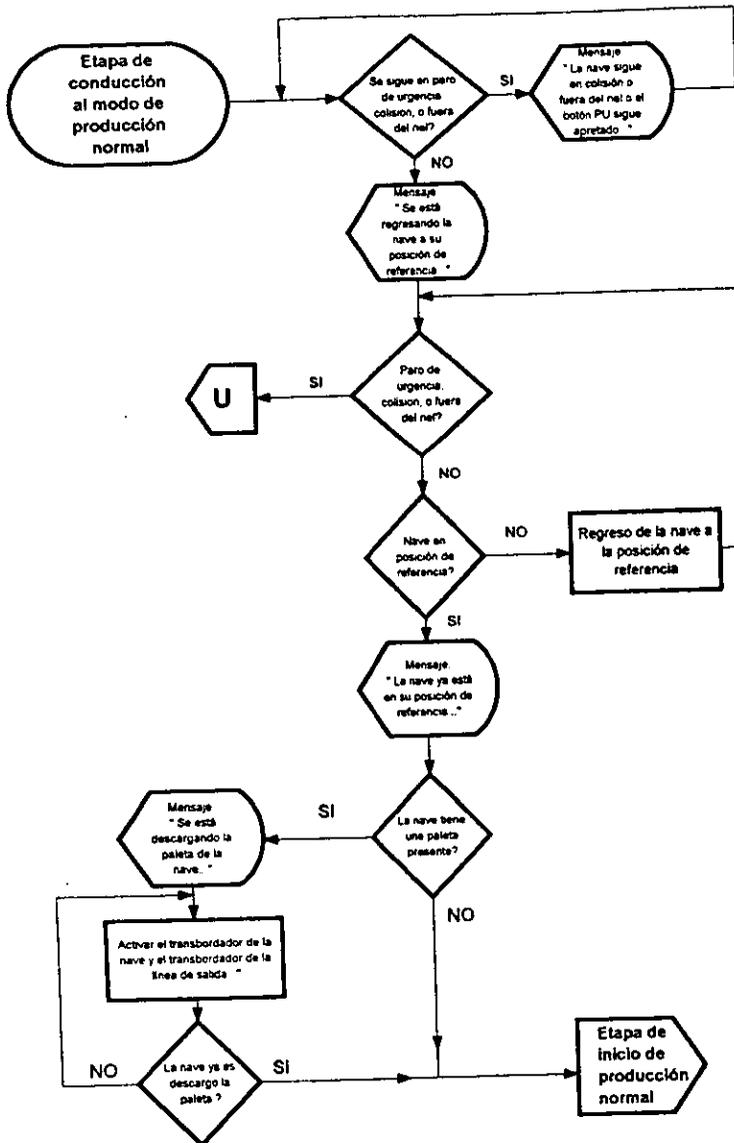


Figura 7.5d Diagrama de flujo general (continuación)

A continuación se presenta el listado del programa de aplicación llamado "leche.txt", el cual fue hecho en PROSAC y que permitirá al SAC coordinar y automatizar las funciones del equipamiento para que se puedan transferir los palets entre las líneas de producción.

### "LECHE.TXT"

BORRAVIS()  
ETIQUETA(inicio)

*MPN: Modo de producción normal.  
Nave en espera de una llamada de alguna línea. (detección de presencia de paleta en algún transbordador de línea).*

APAGATS()  
ESCRIBEVIS( 1, 1,MPN, nave en es-)  
ESCRIBEVIS( 2, 1,pera de llamada.)  
LEEPE(PTE1,var1)  
LEEPE(PTE2,var2)  
SI( var1 >= hx20 O var2 >= hx40 )

*Estado de urgencia debido a una colisión , a que la nave se salió de los extremos del riel o se apretó el botón de urgencia PU .*

ETIQUETA(urgencia)  
APAGATS()  
PRENDELED(led3)  
PRENDESAL(PTAC,128)  
ETIQUETA(botonpn)  
BORRAVIS()  
ESCRIBEVIS( 1, 2,!! URGENCIA !!)  
ESCRIBEVIS( 2, 1,Hay un problema.)  
PAUSA(2,seg)  
BORRAVIS()  
ESCRIBEVIS( 1, 1,Verifique si hay)  
ESCRIBEVIS( 2, 1,colision, si ta )  
PAUSA(2,seg)  
BORRAVIS()  
ESCRIBEVIS( 1, 1,nave esta fuera)  
ESCRIBEVIS( 2, 1,dcl riel o PU...)  
PAUSA(2,seg)  
BORRAVIS()  
ESCRIBEVIS( 1, 1,resuelva el pro-)  
ESCRIBEVIS( 2, 1,blema y despues)  
PAUSA(2,seg)  
BORRAVIS()  
ESCRIBEVIS( 1, 1,apriete PN para)  
ESCRIBEVIS( 2, 1,regresar al MPN)  
PAUSA(2,seg)  
LEEPE(PTE1,var1)

SI( var1 >= hx10 Y var1 <= hx1F O var1 >= hx30 Y var1 <= hx3f  
O var1 >= hx50 Y var1 <= hx5f O var1 >= hx90 Y var1 <= hx9f  
O var1 >= hxd0 Y var1 <= hxdf )  
BORRAVIS()  
ESCRIBEVIS( 1, 1,Se apreto PN...)  
ESCRIBEVIS( 2, 1,conduccion a MPN)  
PAUSA(2,seg)  
SALTA(CONDMPN)  
FINSI()

*Se aprieta el botón PN (producción normal), para que se realicen automáticamente las operaciones necesarias para que la nave regrese al MPN.*

SALTA(botonpn)  
FINSI()

ETIQUETA(MPN)

*MPN: Modo de producción normal.  
Nave en espera de una llamada de alguna línea.*

LEEPE(PTE2,var2)

ETIQUETA(1etraa)

*Paleta presente en el transbordador de la línea 1.*

SI( var2 = hx02 O var2 = hx03 )

ESCRIBEUAD(hx100,hx31)

BORRAVIS()

ESCRIBEVIS( 1, 1,Paleta en L1, la)

ESCRIBEVIS( 2, 3,nave va a L1)

PRENDESAL(PTAC, 16)

ETIQUETA(navea11)

LEEPE(PTE1,var1)

LEEPE(PTE2,var2)

*Nave dirigiéndose a la línea 1.*

SI( var1 &gt;= hx20 O var2 &gt;= hx40 )

SALTA(urgencia)

FINSI()

SI( var1 = hx02 )

BORRAVIS()

ESCRIBEVIS( 1, 1,La nave ya llego)

ESCRIBEVIS( 2, 3,a la línea L1.)

PAUSA(2.scg)

BORRAVIS()

ESCRIBEVIS( 1, 1,Cargando la pal-)

ESCRIBEVIS( 2, 1,eta en la nave..)

APAGASAL(PTAC, 16)

PRENDESAL(PTAC, 66)

ETIQUETA(trans11)

LEEPE(PTE1,var1)

LEEPE(PTE2,var2)

*Arribo de la nave a la línea 1.*

*Carga de la paleta en la nave.*

SI( var1 &gt;= hx20 O var2 &gt;= hx40 )

SALTA(urgencia)

FINSI()

SI( var2 &gt;= hx10 Y var2 &lt;= hx20)

APAGASAL(PTAC, 66)

BORRAVIS()

ESCRIBEVIS( 1, 1,Paleta de L1 )

ESCRIBEVIS( 2, 1,cargada en nave)

SALTA(contimpn)

FINSI()

SALTA(trans11)

FINSI()

SALTA(navea11)

FINSI()

*Paleta cargada en la nave*

ETIQUETA(letrac)

*Paleta presente en el transbordador de la línea 2.*

```

SI( var2 = hx04 O var2 = hx05 )
ESCRIBEUAD(hx100,hx32)
BORRAVIS()
ESCRIBEVIS( 1, 1,Paleta en L2, la)
ESCRIBEVIS( 2, 3,nave va a L2)
PRENDESAL(PTAC, 16)
ETIQUETA(naveal2)
LEEPE(PTE1,var1)
LEEPE(PTE2,var2)

```

*Nave dirigiéndose a la línea 2.*

```

SI( var1 >= hx20 O var2 >= hx40 )
SALTA(urgencia)
FINSIQ()

```

```

SI( var1 = hx04 )
BORRAVIS()
ESCRIBEVIS( 1, 1,La nave ya llego)
ESCRIBEVIS( 2, 3,a la línea L2.)
PAUSA(2,seg)
BORRAVIS()
ESCRIBEVIS( 1, 1,Cargando la pal-)
ESCRIBEVIS( 2, 1,eta en la nave..)
APAGASAL(PTAC, 16)
PRENDESAL(PTAC, 68)
ETIQUETA(transl2)
LEEPE(PTE1,var1)
LEEPE(PTE2,var2)

```

*Arribo de la nave a la línea 2.**Carga de la paleta en la nave.*

```

SI( var1 >= hx20 O var2 >= hx40 )
SALTA(urgencia)
FINSIQ()

```

```

SI( var2 >= hx10 Y var2 <= hx20)
APAGASAL(PTAC, 68)
BORRAVIS()
ESCRIBEVIS( 1, 1,Paleta de L2 )
ESCRIBEVIS( 2, 1,cargada en nave)
SALTA(contimpn)
FINSIQ()
SALTA(transl2)

```

*Paleta cargada en la nave*

```

FINSIQ()
SALTA(naveal2)

```

FINSIQ()

ETIQUETA(letrad)

*Paleta presente en el transbordador de la línea 3.*

SI( var2 = hx08 O var2 = hx09 )  
 ESCRIBEUAD(hx100,hx33)  
 BORRAVIS()  
 ESCRIBEVIS( 1, 1,Paleta en L3, la)  
 ESCRIBEVIS( 2, 3,nave va a L3)  
 PRENDESAL(PTAC, 16)  
 ETIQUETA(naveal3)  
 LEEPE(PTE1,var1)  
 LEEPE(PTE2,var2)

*Nave dirigiéndose a la línea 3.*

SI( var1 >= hx20 O var2 >= hx40 )  
 SALTA(urgencia)  
 FINSI()

SI( var1 = hx08 )  
 BORRAVIS()  
 ESCRIBEVIS( 1, 1,La nave ya llego)  
 ESCRIBEVIS( 2, 3,a la línea L3.)  
 PAUSA(2,seg)  
 BORRAVIS()  
 ESCRIBEVIS( 1, 1,Cargando la pal-)  
 ESCRIBEVIS( 2, 1,eta en la nave..)  
 APAGASAL(PTAC, 16)  
 PRENDESAL(PTAC, 72)  
 ETIQUETA(transl3)  
 LEEPE(PTE1,var1)  
 LEEPE(PTE2,var2)

*Arribo de la nave a la línea 3.**Carga de la paleta en la nave.*

SI( var1 >= hx20 O var2 >= hx40 )  
 SALTA(urgencia)  
 FINSI()

SI( var2 >= hx10 Y var2 <= hx20 )  
 APAGASAL(PTAC, 72)  
 BORRAVIS()  
 ESCRIBEVIS( 1, 1,Paleta de L3 )  
 ESCRIBEVIS( 2, 1,cargada en nave)  
 SALTA(contimprn)  
 FINSI()  
 SALTA(transl3)

*Paleta cargada en la nave*

FINSI()  
 SALTA(naveal3)

FINSI()

SALTA(inicio)

ETIQUETA(contimpn)	<i>MPN: Modo de producción normal (continuación )</i>
BORRAVISQ	
ESCRIBEVIS( 1, 1,Paleta cargada y)	<i>La nave cargada con la paleta se dirige a la</i>
ESCRIBEVIS( 2, 3,nave va a L0)	<i>línea 0</i>
PRENDESAL(PTAC, 32)	
ETIQUETA(regresonave)	
LEEPE(PTE1,var1)	
SI( var1 = hx01 )	
APAGASAL(PTAC, 32)	
BORRAVISQ	
ESCRIBEVIS( 1, 1,Paleta en nave)	<i>Nave en espera de autorización de descarga de</i>
ESCRIBEVIS( 2, 1,espera de AT...)	<i>paleta</i>
ETIQUETA(autorizacion)	
LEEPE(PTE2,var2)	
SI( var2 >= hx30 Y var2 <= hx3f)	
BORRAVISQ	
ESCRIBEVIS( 1, 1,Nave descargando)	<i>Se ha dado la autorización y la nave procede a</i>
ESCRIBEVIS( 2, 1,la paleta...)	<i>descargar la paleta</i>
PRENDESAL(PTAC, 65)	
ETIQUETA(descarga)	
LEEPE(PTE2,var2)	
SI( var2 = hx01 O var2 = hx03 O var2 = hx05 O var2 = hx09 O var2 =hx07 O var2 = hx0b O var2 =	
hx0d O var2 = hx0f )	
APAGASAL(PTAC, 65)	
UADAVAR(hx100,var1)	
LEEPE(PTE2,var2)	
SI(var1 = hx31)	<i>Le nave ha descargado la paleta, se procede a</i>
SALTA(1etrac)	<i>esperar a atender el llamado de una línea</i>
FINSIQ	
SI(var1 = hx32)	
SALTA(1etrad)	
FINSIQ	
SALTA(inicio)	
FINSIQ	
SALTA(descarga)	
FINSIQ	
SALTA(autorizacion)	
FINSIQ	
SALTA(regresonave)	
ETIQUETA(CONDMPN)	<i>CONDMPN: Conducción al MPN. Después de</i>
APAGASAL(PTAC, 128)	<i>haber apretado el botón Pn debido a una</i>
LEEPE(PTE1,var1)	<i>urgencia, se regresa la nave a su posición inicial.</i>
APAGALED(led3)	
LEEPE(PTE2,var2)	
SI( var1 >= hx20 O var2 >= hx40 )	<i>Se verifica si se ha corregido el problema que</i>
APAGATSQ	<i>ocasionó la urgencia</i>
PRENDELED(led3)	
PRENDESAL(PTAC, 128)	
BORRAVISQ	

```

ESCRIBEVIS( 1, 1, ISE CONTINUA EN)
ESCRIBEVIS( 2, 1, EDO DE URGENCIA!)
PAUSA(1,seg)
BORRAVIS()
ESCRIBEVIS( 1, 1, !! CORRIJA EL)
ESCRIBEVIS( 2, 1, PROBLEMA !!)
PAUSA(1,seg)
SALTA(CONDMPN)

```

```

FINSI()
BORRAVIS()
ESCRIBEVIS( 1, 1, OK!, se regresa)
ESCRIBEVIS( 2, 1, la nave a L0...)
PAUSA(5,seg)
ETIQUETA(navcent0)
LEEPE(PTE1,var1)
LEEPE(PTE2,var2)

```

```

SI( var1 >= hx20 O var2 >= hx40 )
SALTA(urgencia)
FINSI()

```

```

SI( var1 = hx01 )
APAGASAL(PTAC, 32)
BORRAVIS()
ESCRIBEVIS( 1, 1, La nave esta al )
ESCRIBEVIS( 2, 1, lado de L0.)
PAUSA(2,seg)
LEEPE(PTE2,var2)

```

```

SI( var2 >= hx10 Y var2 <= hx1f)
BORRAVIS()
ESCRIBEVIS( 1, 1, Paleta en nave)
ESCRIBEVIS( 2, 1, espera de AT...)
ETIQUETA(esperauto)
LEEPE(PTE2,var2)

```

```

SI( var2 >= hx30 Y var2 <= hx3f )
BORRAVIS()
ESCRIBEVIS( 1, 1, Descargando la)
ESCRIBEVIS( 2, 1, paleta...)
PRENDESAL(PTAC, 65)
ETIQUETA(descargada)
LEEPE(PTE2,var2)

```

```

SI( var2 = hx01 O var2 = hx03 O var2 = hx05
O var2 = hx09 O var2 = hx07 O var2 = hx0b
O var2 = hx0d O var2 = hx0f )
APAGASAL(PTAC, 65)
SALTA(MPN)
FINSI()
SALTA(descargada)

```

```

FINSI()
SALTA(esperauto)

```

```

FINSI()
SALTA(MPN)

```

```

FINSI()
PRENDESAL(PTAC, 32)
SALTA(navcent0)

```

*El problema ha sido corregido, ahora si la nave no estaba en la posición inicial, se regresa la nave al lado de L0.*

*La nave está en su posición inicial (al lado de L0)*

*Si hay una paleta sobre la nave se espera la autorización de descarga. de lo contrario se regresa al MPN.*

*La autorización se ha dado y se procede a descargar la paleta*

*La nave ha descargado la paleta, se procede a esperar a atender el llamado de una línea*

## 7.6.6 Elaboración de los esquemas

## a) Elaboración de los esquemas de potencia

En el caso de los motores eléctricos, los preaccionadores y la alimentación eléctrica se deducen directamente de:

- las características de los motores seleccionados;
- las recomendaciones del usuario;
- las características de distribución de energía eléctrica y de las reglamentaciones de seguridad

Cada línea del motor está equipada con un contactor-disyuntor que integra:

- la función de comando (contactor)
- la función de protección del motor contra los cortocircuitos (disyuntor), contra las sobrecargas y sobrecorrientes (magneto-térmico)
- la función de señalización de errores y diálogo con el SAC.

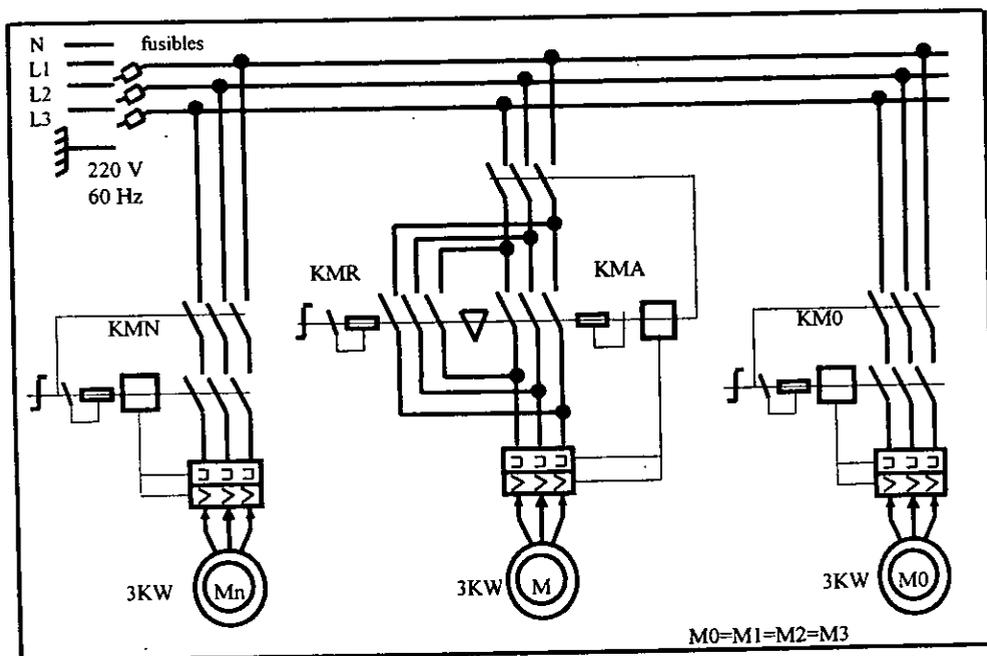


Figura 7.5 Esquema de potencia

## b) Elaboración de los esquemas de comando

A continuación se muestran los esquemas de conexión de las entradas y salidas del SAC con los detectores de posición, preaccionadores y accionadores.

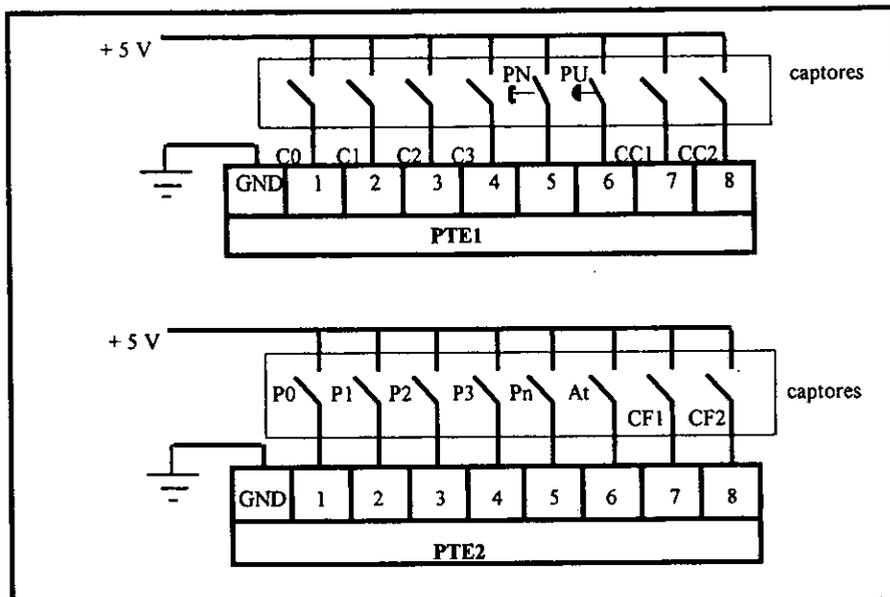


Figura 7.6 Esquema de conexión de los detectores de posición a las entradas del SAC.

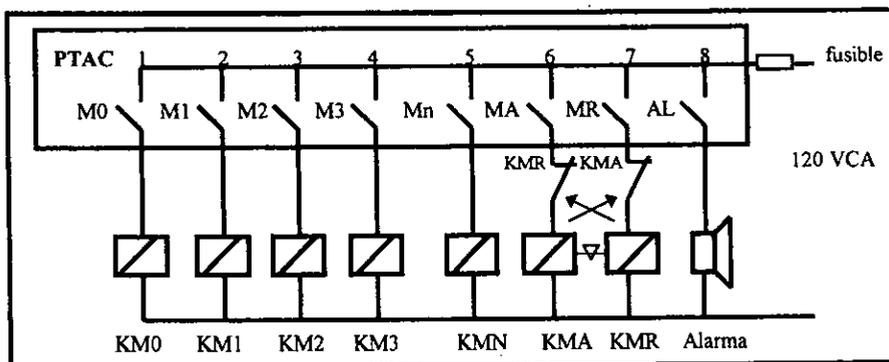


Figura 7.7 Esquema de conexión de los preaccionadores y accionadores a las salidas del SAC

### 7.6.7 Beneficios obtenidos

En este caso, el SAC se utilizó para automatizar la transferencia de los productos en las líneas de producción, es decir, para controlar los movimientos de transferencia de los palets, para dar prioridad a las llamadas emitidas por las líneas de producción, y tomar en cuenta los procedimientos de seguridad necesarios. Este ejemplo es una ilustración de la adaptación y versatilidad del SAC en los problemas de automatización que exigen una alta seguridad.

Con la automatización de la fábrica de productos lácteos utilizando al SAC, se obtuvieron los siguientes beneficios:

- Un valor agregado a la materia prima
- Un crecimiento de la productividad del sistema; es decir, aumento de la cantidad de productos elaborados en un tiempo determinado. Este crecimiento en la productividad da como resultado una ganancia en el valor agregado bajo la forma de una mayor rentabilidad y de una mayor competitividad
- Un mejoramiento de la calidad del producto gracias a una mejor repetición del valor agregado
- Una adaptación a contextos particulares como:
  - adaptación a medios hostiles de trabajo para el hombre ( frío, calor, etc).
  - adaptación a tareas físicas o intelectuales difíciles para el hombre (manipulación de cargas pesadas, tareas repetitivas, paralelizadas).
- Un aumento en la seguridad tanto de los bienes como de las personas.

## VIII CONCLUSIONES

El Sistema de Adquisición y Control de Propósito General denominado SAC, es un prototipo aun en fase experimental. Requiere un gabinete protector adecuado, una revisión minuciosa de conectores, puertos e interfaces y además, diseñar las tarjetas de circuitos impresos correspondientes. Fuera del laboratorio no ha tenido usos ni aplicaciones específicas; sin embargo, en las pruebas ya realizadas los resultados han sido muy satisfactorios.

El SAC es flexible; es decir, puede usarse sin necesidad de acoplar una gran cantidad de *hardware* adicional. Por otra parte, el equipamiento básico también se puede ampliar en el caso de que las necesidades de la aplicación así lo requieran.

De esta manera los beneficios obtenidos en cuanto a *hardware* son los siguientes:

- Sistema autónomo y confiable.
- Sistema de bajo consumo de energía.
- Permite avisar y tener un control de las tareas que se realizan, con la posibilidad de almacenar la información o desplegarla en la pantalla de una PC o de un visualizador LCD.
- Conexión directa a sensores y actuadores.
- Permite realizar diversas funciones de alarma.
- Permite monitorear señales analógicas y adquirir datos.
- Sistema flexible: cualquier modificación de la lógica de control puede hacerse con una simple reprogramación en vez de obligar al cambio de componentes, los cuales exigen mucho tiempo y gastos extras.

La característica más importante del SAC es que es programable. Gracias a ella, el usuario tiene la posibilidad de realizar muy diversas tareas de acuerdo a sus necesidades.

Por medio del programa denominado PROSAC (Programación del Sistema de Adquisición y Control de Propósito General) se buscó que el usuario contara con un ambiente de trabajo en donde encontrara todas las herramientas de *software* necesarias para generar programas de aplicación de manera sencilla, lógica e intuitiva sin necesidad de conocer otros lenguajes de programación.

Aunque el programa PROSAC requiere todavía de una revisión en su estructura general, así como en las rutinas de las instrucciones y sentencias de control, las ventajas obtenidas con el *software* fueron las siguientes:

- El mismo usuario puede realizar la función necesaria para su aplicación a través de un programa.
- Rápido desarrollo de los programas de aplicación sin necesidad de conocer otros lenguajes de programación.
- Realización de tareas de manera rápida y sin errores.
- Herramientas de *software* sencillas y fáciles de utilizar enfocadas a la adquisición de datos y al control en tiempo real.
- Diferentes formas de programación: "in-line" y "off-line".
- Posibilidad de revisar y verificar el estado del sistema por medio de un diagnóstico.

Finalmente, el SAC tiene un gran campo de aplicación y puede utilizarse para resolver diversas tareas entre las que se encuentran:

- Monitoreo de señales
- Control y medición de variables físicas
- Adquisición de datos
- Supervisión y control de procesos
- Procesamiento de alarmas
- Generación de señales

Entre otras.

En conclusión, se lograron los objetivos de diseño inicialmente propuestos: se obtuvo un sistema sencillo, robusto en su diseño, versátil, fácil de programar y de uso múltiple. Gracias a todo esto, el SAC podrá realizar diferentes tareas según las necesidades del usuario.

**BIBLIOGRAFÍA**

1. Aho Alfred V, Sethi Ravi, Ullman Jeffrey D, "Compilers Principles, Techniques, and Tools", Addison-Wesley Publishing Company, 1988
2. Kernighan Brian W, Ritchie Dennis M, "El lenguaje de programación C", Prentice-Hall Hispanoamericana, 1986.
3. Purdum Jack, "C Programmer's Toolkit, 2nd Edition", Que Corporation., 1992.
4. "Microsoft Visual Basic Programmer's Guide", Microsoft Corporation., 1993
5. Spencer Donald D, "Matemáticas para la Ciencia de la Computación", C.E.C.S.A., 1984.
6. Coughlin Robert F, Driscoll Frederick F, "Circuitos Integrados Lineales y Amplificadores Operacionales", Prentice-Hall., 1982.
7. Franco Sergio, "Design with Operational Amplifiers and Analog Integrated Circuits", McGraw-Hill International Editions., 1988.
8. Wood Steve, "Using Turbo Pascal", McGraw Hill, 1986.
9. "MC68HC11 Reference Manual", Motorola Inc., 1991.
10. "MC68HC11E9, Technical Data", Motorola Inc., 1991.
11. "Microprocessor, Microcontroller and Peripheral Data Book", (Vols.I y II) Motorola Inc., 1988.
12. "MC68HC11EVBU Universal Evaluation Board User's Manual", Motorola Inc., 1990.
13. "PCMCIA PC Card Standard Release 2.01", PCMCIA., 1992.
14. "MAXIM New Released Data Book", Maxim Integrated Products Inc., 1990.
15. "AND Display Products Catalog", William J. Purdy Company, 1990.
16. "IC Memories Data Book", Hitachi America Ltd., 1987.
17. "Linear Data Book", National Semiconductor Corporation., 1988.
18. "CMOS Logic Data", Motorola Inc., 1988.
19. "High-Speed CMOS Logic Data", Motorola Inc., 1988.

## ANEXO A

## ESPECIFICACIONES TÉCNICAS DEL PROTOTIPO DEL SAC

- Función:** Sistema de adquisición y control de propósito general.
- CPU:** MC68HC11A1@ 2 MHz con 32Kbytes de memoria EPROM para el sistema operativo y 32 Kbytes de memoria RAM para el programa de aplicación.
- Memoria del programa:** Memoria RAM interna de 32 Kbytes, respaldada por batería.
- Medio de almacenamiento de datos (UAD-PCMCIA):** Tarjeta de memoria SRAM PCMCIA respaldada con batería con capacidad de 1 Mbyte expandible hasta 16 MB.
- Tiempo de retención de datos de la UAD-PCMCIA:** 9 años máximo.
- Registro de tiempo:** Reloj fechador interno, precisión  $8 \times 10^{-6}$  con respaldo de batería.
- Puerto para PC y transmisión de datos:**
  - Puerto serie RS232-C configuración fija: 9600 bauds, NRZ, 1 bit de inicio, 8 bits de datos, un bit de paro, sin paridad.
  - Puerto paralelo de 8 bits con señales de control.
- Tiempo de transmisión de datos:** 18 minutos para un Mbyte a través del puerto serie.
- Visualización de datos:** Módulo de despliegue LCD de 2 líneas y 16 caracteres por línea.
- Puertos de entrada:**
  - 4 entradas analógicas con rangos de entrada seleccionables entre: 0 y 5 V, 0 y 10 V, 0 y 20 V,  $\pm 2.5$  V,  $\pm 5$  V y  $\pm 10$  V. Resistencia de entrada por canal de 100 K $\Omega$ . Las frecuencias máximas admisibles en las entradas analógicas dependiendo de la función deseada son:

Entradas (número de señales analógicas)	Monitoreo (envío continuo por el puerto serie de las muestras)	Almacenamiento continuo de las muestras en la UAD	Monitoreo y almacenamiento de las muestras
1 sola señal	Fmáx= 480 Hz	Fmáx= 1.5 KHz	Fmáx= 480 Hz
4 señales simultáneas	Fmáx= 120 Hz por señal	Fmáx= 850 Hz por señal	Fmáx= 120 Hz por señal

<b>Puertos de entrada (continuación):</b>	<ul style="list-style-type: none"><li>• Puerto de entrada con 8 entradas disponibles que permiten manejar niveles de voltaje de entrada de 0 y 5 VCD compatibles con TTL.</li><li>• Puerto de entrada con 8 entradas disponibles que permiten manejar niveles de voltaje de entrada entre 0 y 18 VCD.</li><li>• Puerto de detección con 3 entradas disponibles que permiten manejar niveles de voltaje de entrada entre 0 y 18 VCD.</li></ul>
<b>Puertos de salida:</b>	<ul style="list-style-type: none"><li>• Un puerto de salida con 8 salidas que manejan niveles de voltaje de salida de 0 y 5 VCD 35 mA por salida.</li><li>• Un puerto de salida con 8 salidas disponibles que permiten manejar dispositivos de corriente alterna a 125 VCA @ 2A por salida.</li></ul>
<b>Modos de operación:</b>	<ul style="list-style-type: none"><li>• Modo de espera. Durante este modo el SAC se encuentra en espera de recibir un programa de aplicación.</li><li>• Modo de recepción. En este modo el SAC indica que esta recibiendo y cargando en memoria un programa de aplicación.</li><li>• Modo de ejecución. En este modo se indica que el SAC está ejecutando un programa de aplicación.</li><li>• Modo de diagnóstico. En este modo se puede verificar el estado de operación, modificar la fecha y hora y activar varias funciones.</li></ul>
<b>Programa de utilería PROSAC:</b>	Programa para Windows que permite al usuario crear, editar, compilar y transferir programas de aplicación al SAC, así como recibir información proveniente de el.
<b>Alimentación:</b>	<ul style="list-style-type: none"><li>• Externa y seleccionable: + 5 VCD; o de 9 a 18 VCD. Nominal 12 VCD.</li><li>• 3 Baterías alcalinas tamaño AA de 1.5V para respaldo del reloj fechador y de la memoria de programa.</li></ul>
<b>Consumos:</b>	200 mA máximo (con todos los sistemas activos) con alimentación de 12 VCD.
<b>Tecnología:</b>	Estado sólido CMOS
<b>Rango de temperatura:</b>	0 a 40 °C.

## ANEXO B

### DESCRIPCIÓN DE CABLES Y CONECTORES DEL SAC

#### Descripción de los cables

A continuación se describen los cables utilizados en el SAC (ver figuras B.1 a B.4).

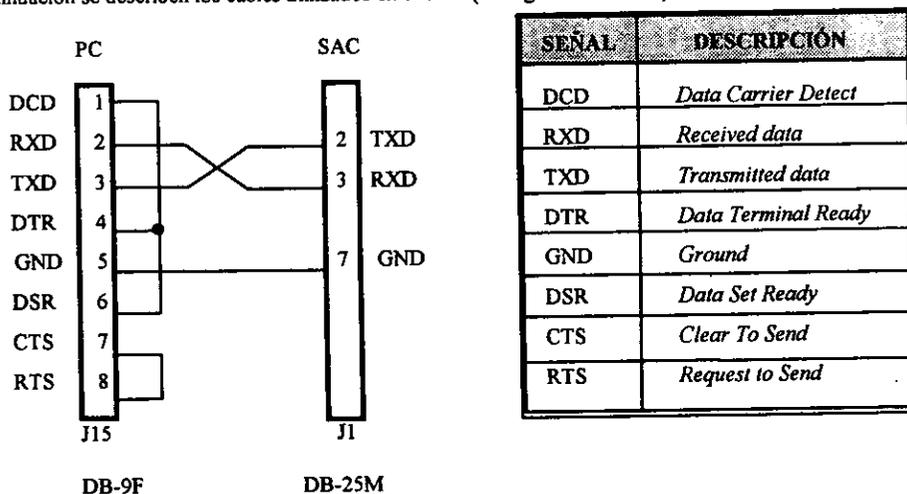


Figura B.1. CABLE 1 Interconexión del SAC y una PC mediante comunicación serie RS-232C



Figura B.2. CABLE 2 Interconexión del sistema de control y procesamiento con el módulo acondicionador de señales y puerto de detección.

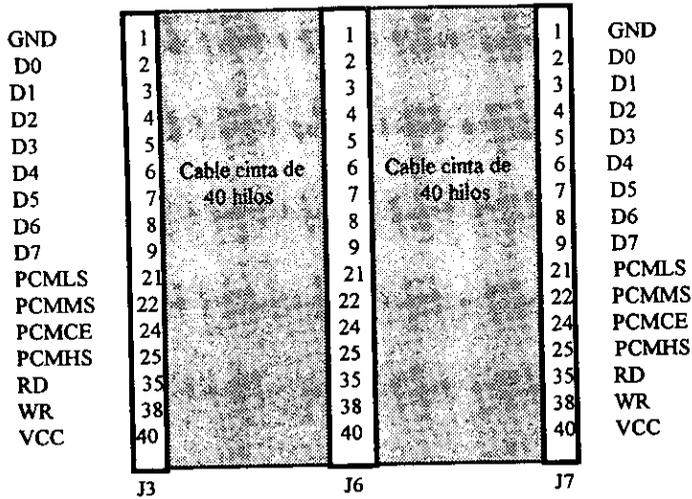


Figura B.3. CABLE 3 Interconexión del sistema de control y procesamiento con la tarjeta UAD-PCMCIA y el módulo de despliegue y teclado.

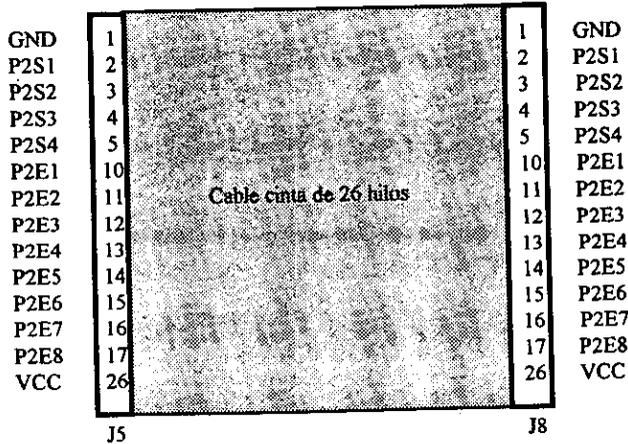


Figura B.4. CABLE 4 Interconexión del sistema de control y procesamiento con el puerto de entrada 2 y puerto de salida 2 con interfaz de potencia.

Descripción de los conectores

A continuación se describen los conectores del SAC (ver figuras B.5 a B.9).

**PUERTO SERIE-PARALELO**

CONECTOR P1 DE 25 PINS	
PIN	DESCRIPCION
1	S.C
2	RXD
3	TXD
4	RTS
5	CTS
6	S.C
7	GND
8	S.C
9	S.C
10	DP0
11	DP1
12	DP2
13	DP3
14	DP4
15	DP5
16	DP6
17	DP7
18	ACK
19	RDY
20	S.C
21	S.C
22	S.C
23	S.C
24	S.C
25	S.C

**PUERTO ESPECIAL**

CONECTOR P2 DE 26 PINS	
PIN	DESCRIPCION
1	GND
2	PA0
3	PA1
4	PA2
5	PA3
6	PA4
7	PA5
8	PA6
9	PA7
10	PD0
11	PD1
12	PD2
13	PD3
14	PD4
15	PD5
16	PE0
17	PE1
18	PE2
19	PE3
20	S.C
21	S.C
22	S.C
23	S.C
24	S.C
25	S.C
26	VCC

**PUERTO DE EXPANSION**

CONECTOR P3 DE 40 PINS	
PIN	DESCRIPCION
1	GND
2	D0
3	D1
4	D2
5	D3
6	D4
7	D5
8	D6
9	D7
10	E
11	AS
12	R/W
13	RESET
14	XIRQ
15	IRQ
16	S.C
17	A0
18	A1
19	A2
20	A3
21	PCMLS
22	PCMMS
23	RELS
24	PCMCE
25	PCMHS
26	CS1
27	CS2
28	TECF
29	CS3
30	CS4
31	CS5
32	CS6
33	CS7
34	S.C
35	RD
36	PEN2
37	PEN1
38	WR
39	S.C
40	VCC

NOTA:  
S.C : SIN CONECTAR

**PUERTOS ENTRADA1-SALIDA 1**

CONECTOR P4 DE 26 PINS	
PIN	DESCRIPCION
1	GND
2	P1E1
3	P1E2
4	P1E3
5	P1E4
6	P1E5
7	P1E6
8	P1E7
9	P1E8
10	P1S1
11	P1S2
12	P1S3
13	P1S4
14	P1S5
15	P1S6
16	P1S7
17	P1S8
18	S.C
19	S.C
20	PD0
21	PD1
22	PD2
23	PD3
24	PD4
25	PD5
26	VCC

**PUERTOS ENTRADA2-SALIDA 2**

CONECTOR P5 DE 26 PINS	
PIN	DESCRIPCION
1	GND
2	P2S1
3	P2S2
4	P2S3
5	P2S4
6	P2S5
7	P2S6
8	P2S7
9	P2S8
10	P2E1
11	P2E2
12	P2E3
13	P2E4
14	P2E5
15	P2E6
16	P2E7
17	P2E8
18	S.C
19	S.C
20	PD0
21	PD1
22	PD2
23	PD3
24	PD4
25	PD5
26	VCC

**Figura B.5. Conectores de la tarjeta del sistema de control y procesamiento**

CONECTOR P13 DE 2 ENTRADAS CON TORNILLOS DE PRESIÓN	
PIN	DESCRIPCIÓN
(+)	V. ALIMENTACIÓN
(-)	TIERRA

CONECTOR P14 DE 2 ENTRADAS	
PIN	DESCRIPCIÓN
(+)	POSITIVO DE LA BATERÍA
(-)	NEGATIVO DE LA BATERÍA

**Conectores de alimentación y batería de respaldo (continuación)**

CONECTOR P6 40 PINS	
PIN	DESCRIPCIÓN
1	GND
2	D0
3	D1
4	D2
5	D3
6	D4
7	D5
8	D6
9	D7
10	E
11	S.C
12	S.C
13	S.C
14	S.C
15	IRQ
16	S.C
17	A0
18	A1
19	A2
20	S.C
21	S.C
22	S.C
23	S.C
24	S.C
25	S.C
26	S.C
27	S.C
28	TECF
29	S.C
30	S.C
31	S.C
32	S.C
33	CS7
34	S.C
35	S.C
36	S.C
37	S.C
38	S.C
39	S.C
40	VCC

NOTA:

S.C : SIN CONECTAR

**Figura B.6. Conector de la tarjeta del visualizador y teclado**

CONECTOR P11 26 PINS	
PIN	DESCRIPCION
1	GND
2	PA0
3	PA1
4	PA2
5	S.C
6	S.C
7	S.C
8	S.C
9	S.C
10	S.C
11	S.C
12	S.C
13	S.C
14	S.C
15	S.C
16	S.C
17	PE1
18	PE2
19	PE3
20	S.C
21	S.C
22	S.C
23	S.C
24	S.C
25	S.C
26	VCC

**PUERTO DE DETECCIÓN**

CONECTOR P12 DE 4 ENTRADAS CON TORNILLOS DE PRESION	
PIN	DESCRIPCION
1	PEC-1
2	PEC-2
3	PEC-3
4	GND

CONECTOR P7 40 PINS	
PIN	DESCRIPCION
1	GND
2	D0
3	D1
4	D2
5	D3
6	D4
7	D5
8	D6
9	D7
10	S.C
11	S.C
12	S.C
13	S.C
14	S.C
15	S.C
16	S.C
17	S.C
18	S.C
19	S.C
20	S.C
21	PCMLS
22	PCMMS
23	S.C
24	PCMCE
25	PCMHS
26	S.C
27	S.C
28	S.C
29	S.C
30	S.C
31	S.C
32	S.C
33	S.C
34	S.C
35	RD
36	S.C
37	S.C
38	WR
39	S.C
40	VCC

Figura B.7. Conectores de la tarjeta del módulo acondicionador de señales y puerto de detección

Figura B.8. Conector de la tarjeta de la unidad de almacenamiento de datos UAD-PCMCIA

CONECTOR P2 26 INS	
PIN	DESCRIPCION
1	GND
2	P2S1
3	P2S2
4	P2S3
5	P2S4
6	S.C
7	S.C
8	S.C
9	S.C
10	P2E1
11	P2E2
12	P2E3
13	P2E4
14	P2E5
15	P2E6
16	P2E7
17	P2E8
18	S.C
19	S.C
20	S.C
21	S.C
22	S.C
23	S.C
24	S.C
25	S.C
26	VCC

**PUERTO DE SALIDA 2  
CON INTERFAZ DE POTENCIA**

CONECTOR P3 DE 8 ENTRADAS CON TORNILLOS DE PRESION	
PIN	DESCRIPCION
1	POT1
2	POT2
3	POT3
4	POT4
5	POT5
6	POT6
7	POT7
8	POT8

**PUERTO DE ENTRADA 2**

CONECTOR P16 DE 9 ENTRADAS CON TORNILLOS DE PRESION	
PIN	DESCRIPCION
1	PE2-1
2	PE2-2
3	PE2-3
4	PE2-4
5	PE2-5
6	PE2-6
7	PE2-7
8	PE2-8
9	GND

Figura B.9. Conectores de la tarjeta del puerto de entrada 2 y puerto de salida 2 con interfaz de potencia

## ANEXO C

## INFORMACIÓN SOBRE ARCHIVOS EN FORMATO S19

## INTRODUCCION

El formato de archivos S19 fue creado por Motorola con el propósito de codificar programas o archivos de datos en un formato sencillo para la transferencia de datos entre sistemas de computadoras. El proceso de transportación puede entonces ser monitoreado visualmente y los archivos en formato S19 formados por registros S pueden ser editados fácilmente.

## CONTENIDO DE LOS REGISTROS "S"

Los registros "S" son esencialmente cadenas de caracteres constituidas por diferentes campos que permiten identificar el tipo de registro, la longitud del registro, la dirección de memoria, los códigos o datos y el *checksum*. Cada byte de datos binarios está codificado por dos caracteres que representan un número hexadecimal: El primer carácter representa los 4 bits más significativos del byte (MSB), y el segundo representa los 4 bits menos significativos (LSB).

Los 5 campos que forman un registro S son:

TIPO	LONGITUD	DIRECCIÓN	CÓDIGO/DATOS	CHECKSUM
------	----------	-----------	--------------	----------

Los campos están compuestos de la siguiente manera (ver tabla C.1):

CAMPO	NÚMERO DE CARACTERES	DESCRIPCIÓN
TIPO	2	Tipo de registro S: S0, S1, S9, etc.
LONGITUD	2	Indica el número de pares de caracteres en el registro, excluyendo el tipo y la longitud
DIRECCIÓN	4, 6 u 8	Los 2, 3, o 4 bytes de dirección que indican la dirección de memoria en donde se cargara en campo de datos
CÓDIGO/DATOS	0-2n	De 0 a n bytes de código ejecutable, datos cargables en memoria, o información descriptiva
CHECKSUM	2	El byte menos significativo del complemento a uno de la suma de los valores representados por los pares de caracteres, tomando en cuenta los campos de longitud del registro, el de dirección y el de código/datos

Tabla C.1. Campos de un registro S

Cada registro S puede ser terminado con un CR/LF/NULL. La precisión de la transmisión se asegura por el campo de longitud del registro (cuenta del número de bytes) y el campo de *checksum*.

### TIPOS DE REGISTROS "S"

Ocho tipos de registros S han sido definidos para satisfacer las necesidades de codificación, transferencia y decodificación de datos. Estos se dan en la tabla C.2. El SAC únicamente utiliza los registros tipo S1 y S9 para la recepción de un programa de aplicación, cualquier dato presente antes del primer registro S1 es ignorado. De allí en adelante, todos los registros deberán ser del tipo S1 hasta que un registro tipo S9 indique la terminación de la transferencia de datos.

Registro	Descripción
S0	Este tipo de registro no es utilizado por el SAC.
S1	Es un registro que contiene código y datos y los 2 bytes de dirección en donde el código y los datos van a residir.
S2-S8	Estos tipos de registros no son utilizados por el SAC.
S9	Indica la terminación de un bloque de registros S1 y no contiene campo de código/datos.

Tabla C.2. Tipos de registros S.

Solamente un registro de terminación S9 puede ser usado en cada bloque de registros S.

### CREACIÓN DE UN REGISTRO "S"

La creación del programa con formato de registros S (archivo.S19) es generado por el programa PROSAC a través del ensamblador IAS11. El sistema operativo del SAC ha sido desarrollado para poder recibir el programa de aplicación como archivo S19.

### EJEMPLO DE UN ARCHIVO S19

A continuación se muestra un ejemplo de archivo S19, formado por registros S1 y S9:

```
S1230BD76E2064656C2070726F6772616D612064652061706C6963616369F36E2E0A0D00D9
S1230BF70A0D4171756920796120657374E120656A65637574616E646F736520656C207051
S1230C17726F6772616D612064652061706C6963616369F36E2E2E0A0D000A0D21486FA0
S1170C376C61206D756E646F210A0D000A0D2020200A0D00CF
S9030000FC
```

El archivo S19 anterior consiste de 4 registros de código y datos S1 y de un registro de terminación S9.

A continuación se explica el primer registro S1 de código y datos:

- S1 Registro S1, indicando un registro de código y datos que será cargado y verificado en 2 bytes de dirección (campo de TIPO).
- 23 Número hexadecimal \$23 (decimal 35), que indica que enseguida se encuentran 35 pares de caracteres que representan 35 bytes de datos binarios (campo de DIRECCIÓN).
- 0BD7 Cuatro caracteres que indican 2 bytes de campo de dirección; dirección hexadecimal \$0BD7 que indica la dirección en donde se almacenarán los siguientes datos. Los siguientes 33 pares de caracteres son los bytes ASCII del actual programa de código y datos. En este ejemplo de ensamblado los códigos de operación hexadecimales del programa están escritos en la misma secuencia en que aparecen en el registro S1.

CÓDIGO DE OPERACIÓN Y DATOS	INSTRUCCIÓN
6E 20 64 65 6C 20 70 72 6F 67 72 61 6d 61 20 64	FCB 'n,32,'d,'e,'l,32,'p,'r,'o,'g,'r,'a,'m,'a,32,'d
65 20 61 70 6C 69 63 61 63 69 F3 6E 2E 0A 0D 00	FCB 'e,32,'a,'p,'l,'i,'c,'a,'c,'i,'ó,'n,46,10,13,0
D9	Checksum del primer registro S1

El segundo y tercer registros S1 contienen cada uno \$23 (35) pares de caracteres y finalizan con checksums de S1 y A0, respectivamente. El cuarto registro S1 contiene \$17 (23) pares de caracteres y tiene un checksum de CF.

El registro de terminación S9 se explica a continuación:

- S9 Registro tipo S9 que indica la terminación de un bloque de registros S1.
- 03 Número hexadecimal \$03 (03 decimal) que indica que tres caracteres pares se encuentran a continuación.
- 0000 Cuatro caracteres que indican 2 bytes de campo de dirección; dirección hexadecimal \$0000.
- FC Checksum del registro S9.

Cada caracter de un registro S está codificado en hexadecimal por lo que tiene asociados a él los bits binarios que están siendo transmitidos. Por ejemplo, el primer Registro S1 es enviado como se muestra a continuación:

TIPO		LONGITUD		DIRECCIÓN				CÓDIGO/DATOS				CHECKSUM																
S	1	2	3	0	B	D	7	6	E	2	0	D	9															
5	3	3	1	3	2	3	3	3	0	4	2	4	4	3	7	3	6	4	5	3	2	3	0	...	4	4	3	9
0101	0011	0011	0001	0011	0010	0011	0011	0011	0000	0100	0010	0100	0100	0011	0111	0011	0110	0100	0101	0011	0010	0011	0000	...	0100	0100	0011	1001