

UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERIA



Prototipo Experimental de un Sistema de
Adquisición de Datos de Tiempo Diferido
Compatible con una Computadora IBM

TRABAJO PRESENTADO POR:

Carlos Roberto Zapata
Oscar Wenceslao Rivas

COMO REQUISITO
PARA OPTAR AL TITULO DE:

INGENIERO ELECTRONICO

AGOSTO 1991

UNIVERSIDAD DON BOSCO

PRESIDENTE DEL CONSEJO DIRECTIVO:
Reverendo Padre Luis Ricardo Chinchilla.

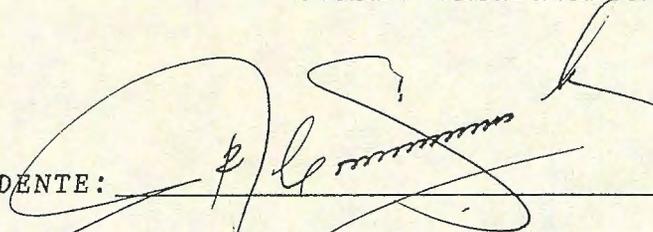
RECTOR:
Licenciado Gilberto Aguilar Avilés.

DECANO DE LA FACULTAD DE INGENIERIA:
Ing. Roberto Armando Cruz.

SECRETARIO GENERAL DE LA UNIVERSIDAD DON BOSCO:
Dr. José Gerardo Liévano Chorro.

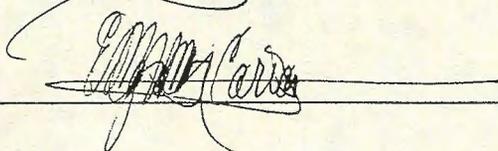
JURADO CALIFICADOR

PRESIDENTE:

A large, stylized handwritten signature in black ink, written over a horizontal line. The signature is highly cursive and difficult to decipher.

MIEMBRO :

:

A smaller handwritten signature in black ink, written over a horizontal line. The signature is also cursive and partially obscured by the 'MIEMBRO' label.

ABSTRACTO

DISEÑO E IMPLEMENTACION DE PROTOTIPO EXPERIMENTAL DE UN SISTEMA DE ADQUISICION DE DATOS COMPATIBLE CON UNA COMPUTADORA IBM.

Por:

Oscar Wenceslao Rivas
Carlos Roberto Zapata

En el campo de estudio de la geotermia, la Comisión Ejecutiva del Río Lempa (CEL) realiza proyecto de investigación, que consiste en tomar muestras de presión y temperatura en los pozos geotérmicos de Ahuachapan para determinar las condiciones termales en ellos. Para lograrlo se requiere de un dispositivo electrónico que pueda tomar esos datos a las condiciones ambientales del pozo y ser programado por un operario. Es así como la tesis tiene como finalidad diseñar e implementar un prototipo experimental que mida las variables físicas de presión y temperatura dentro del pozo almacenándolas digitalmente esa información, para que después de procesar esos datos en el pozo, será transmitida la información a una computadora donde un operario interpretará los datos obtenidos.

El sistema anteriormente descrito se conoce como sonda electrónica y para este caso se busca que el prototipo sea económico, portátil y compatible con cualquier computadora IBM que tenga puerto serie via RS-232C. Para lograr que la sonda electrónica almacene los datos de presión y temperatura se valdrá del uso de transductores, convertidor analógico-digital, memorias y microprocesador que realizará las decisiones de cuando tomar datos en el pozo. Además un programa que consta de una serie de menus donde la sonda pueda ser fácilmente programada en base a las necesidades que la empresa CEL requiere.

De esta manera el prototipo cumple con los requerimientos establecidos por CEL para que posteriormente pueda llevarse a cabo un estudio de como proteger a la sonda de las condiciones ambientales de un pozo geotérmico.

DEDICACION.

- Al nuestro Dios Jehová, a su hijo Jesucristo y que por su Santo Espiritu hemos sido inspirados para la elaboración de las ideas de la tesis y alcanzar este triunfo que desde el principio es dedicado para la gloria de su nombre.
- A mis padres, quien porporcionaron los medios necesarios a través de estos cinco años para ver coronada la carrera, quienes tuvieron paciencia para conmigo instandome en todo tiempo a superarme.
- A mis hermanos en Cristo que mediante sus oraciones me proporcionaron el aliento necesario para no desmayar.
- A mi novia quien fue paciente para conmigo y dándome fuerzas y motivándome a continuar para llegar a la meta final.

Mi agradecimiento:

- A la Divina Providencia por proveerme de todo cuanto he necesitado, por darme la voluntad de alcanzar mis metas y por guiarme en los momentos precisos de más exigente labor.
- A María Santísima, madre de Dios, por su incesante atención, auxilio y protección.
- A mis papás Vicente Oscar Rivas y Ana María Zaldaña de Rivas y a mi tía Haydée Elena Zaldaña, por que jamás han medido esfuerzos ni sacrificios en buscar mi bienestar y superación.
- A la comunidad Salesiana del Instituto Técnico Ricaldone por haberme permitido usar el excelente equipo del taller de mantenimiento.
- En este espacio deseo agradecer a todos aquellos que de una u otra manera nos ayudaron en este trabajo; al mencionarlos a todos sería una lista demasiado larga y aun así la memoria puede fallarme y pueda olvidar a alguno; pero lo importante es que ellos y Yo sabemos quienes son y cuanto nos ayudaron a facilitarnos la tarea, cuando no fue que sin esa ayuda no hubieramos finalizado con éxito este proyecto. En una manera especial he de mencionar a Luis Roberto Granados Paz que nos ayudó en el simulador de presiones y en el acople neumático de nuestro sistema.
- Al Ing. Manuel Monterrosa que aunque no consiguió que CEL nos ayudara tanto como deseábamos, he de reconocer y agradecer por que la ayuda en elementos que nos proporcionó, en buena medida alivió nuestra carga económica.
- Y finalmente a Carlos Roberto Zapata, por su dedicación al trabajo.

Oscar Wenceslao Rivas Zaldaña

TABLA DE CONTENIDO

Lista de figuras, x
Lista de tablas, xii
Prefacio, xiii

1 Sistema de adquisición de datos.

1.1 Descripción global, 1
1.2 Antecedentes del proyecto, 1
1.3 Descripción general del circuito, 4
1.4 Operación de funcionamiento, 7

2 Software de la PC.

2.1 Funciones del sistema y generalidades, 8
2.2 Rutina de programación del sistema, 9
 2.2.1 Justificación de la calibración, 12
2.3 Rutina de lectura y presentación de datos, 34
2.4 Direccionamiento del puerto serie de la PC, 41

3 Hardware del sistema.

3.1 Diseño del circuito del prototipo experimental, 44
3.1.1 Sistema digital del sistema de adquisición de datos, 45
 3.1.1-1 Sección de control, 45
 3.1.1-2 Sección de memorias y decodificador de direcciones, 48
 3.1.1-3 Sección de interfaces, 50
 3.1.1-4 Convertidor Analogo-Digital, 51
 3.1.1-5 Circuito para pruebas. Control externo, 53
 3.1.1-6 Circuito para pruebas. Visualización, 54
3.1.2 Parte analógica del sistema, 54
 3.1.2-1 Transductores, 54
 3.1.2-2 Circuito de la sección analógica, 58
 3.1.2-3 Proceso del diseño del circuito en el sistema analógico, 61
 3.1.2-4 Proceso de calibración, 64
Esquemas, 66

4 Software del sistema. Programa monitor

- 4.1 Estructuración de las rutinas principales, 73
 - 4.1.1 Inicio, 73
 - 4.1.2 Estado1, 74
 - 4.1.3 Vector de subrutinas, 74
 - 4.1.4 Programar, 74
- 4.2 Estructuración de las subrutinas, 75
 - 4.2.1 Transmisión, 76
 - 4.2.2 Prueba1, 76
 - 4.2.3 Autocalibración, 77
 - 4.2.4 Adquisición, 77
 - 4.2.5 TESTRAM, 79
 - 4.2.6 RESETRAM, 80
 - 4.2.7 Verificar, 80
 - 4.2.8 TX, 80
 - 4.2.9 Retardo, 80

5 Diseño del sistema de adquisición de datos de tiempo diferido. Modelo real.

- 5.1 Software de la PC, 96
- 5.2 Hardware del sistema, 97
 - 5.2.1 Sistema digital, 97
 - 5.2.2 Convertidor Analogo-Digital, 98
 - 5.2.3 Transductores, 100
 - 5.2.4 Sistema analógico. Diseño 102
 - 5.2.5 Proceso de autocalibración, 108
- 5.3 Análisis del consumo de potencia, 109
- 5.4 Sugerencias, 112
- 5.5 Cambios en el programa que gobierna al sistema de adquisición de datos, 114
 - 5.5.1 Rutina de adquisición de datos, 119
 - 5.5.2 Autocalibración, 122
 - 5.5.3 TESTRAM, 122
 - 5.5.4 RESETRAM, 123

6 Aplicaciones alternativas del sistema de adquisición de datos.

- 6.1 Descripción de la opción, 135
- 6.2 Cambios en el programa de la PC, 135
- 6.3 Cambios en el hardware, 135
- 6.4 Cambios en el programa monitor, 136

- Apendice A. Listado del programa monitor.
- Apendice B. Listado del programa en Turbo Basic.
- Apendice C. Hojas técnicas.

LISTA DE FIGURAS

CAPITULO 1.

1.1 Diagrama a bloques del sistema

CAPITULO 2.

- 2.1 Menu de programación de la PC.
- 2.2 Presentación de menus de la PC.
- 2.3 TestRAM, Pruebal y Autocalibración cuando se programa al sistema.
- 2.4 Mensajes automáticos de la programación.
- 2.5 (a) Inicialización de la programación.
- 2.5 (b) Selección de la variable a medir.
- 2.5 (c) Ventana de programación de tiempo antes de empezar.
- 2.5 (d) Ventana de programación de tiempo entre muestras.
- 2.5 (e) Selección de la toma de datos.
- 2.5 (f) Preparar la cantidad de muestras.
- 2.5 (g) Cálculo de la cantidad de datos a tomar.
- 2.5 (h) Selección del número de estaciones.
- 2.5 (i) Resumen de las variables de programación.
- 2.5 (j) Transmisión de los datos de programación.
- 2.5 (k) Verificación.
- 2.5 (l) Resumen de los datos programados y datos sonda.
- 2.5 (m) Ventana de retransmisión de datos de programación.
- 2.5 (n) Selección de borrado de los datos en la sonda.
- 2.6 Parámetros de programación del sistema.
- 2.7 Transmisión y verificación de los datos en la sonda.
- 2.8 Menu de lectura de datos y presentación en pantalla.
- 2.9 (a) Inicialización de lectura de datos en la sonda.
- 2.9 (b) Recibiendo los datos de la sonda.
- 2.9 (c) Unidades para temperatura.
- 2.9 (d) Unidades para presión.
- 2.9 (e) Menu del programa de recepción.
- 2.9 (f) Tabla de Temperatura VRS Tiempo.

CAPITULO 3.

- 3.1 Mapa de memorias del sistema de adquisición de datos.
- 3.2 mapa de periféricos del sistema de adquisición de datos.
- 3.3 Características R vrs t de los termistores
- 3.4 Diagrama de conexiones para el transmisor, receptor y fuente de alimentación.
- 3.5 Puente de impedancias.
- 3.6 Circuito de acondicionamiento de señal de temperatura.
- 3.7 Circuito de acondicionamiento de señal de presión.
- 3.8 Esquema de la sección de los voltajes de referencia.
- 3.9 Red de corrección de offset en un amplificador inversor.
- 3.10 Sección de control. Prototipo.
- 3.11 Sección de memorias y decodificador de direcciones.
- 3.12 Sección de interfaces. Prototipo.
- 3.13 Sección de control. Prototipo.
- 3.14 Sección de visualización. Circuito de prueba del prototipo.
- 3.15 Sección de acondicionamiento de señal.
- 3.16 Sección de amplificador común y voltajes de referencia.

CAPITULO 4.

- 4.1 Programa monitor.
- 4.2 Rutina de programación de la sonda.
- 4.3 Subrutina de transmisión.
- 4.4 Subrutina de prueba de línea.
- 4.5 Subrutina de calibración I.
- 4.6 Subrutina de calibración II.
- 4.7 Subrutina de adquisición de datos I.
- 4.8 Subrutina de adquisición de datos II.
- 4.9 Subrutina de adquisición de datos III.
- 4.10 Subrutina de prueba de RAM (TESTRAM).
- 4.11 Subrutina de borrado de memoria RAM (RESETRAM).
- 4.12 Subrutina de verificación y transmisión.
- 4.13 Subrutina de retardo de tiempo.

CAPITULO 5.

- 5.1 Amplificador de instrumentación.
- 5.2 Montaje del transductor de temperatura RTD en el puente de Wheatstone.
- 5.3 Arreglo de la etapa de entrada de la sección de presión.
- 5.4 Flujograma de adquisición de datos I.
- 5.5 Adquisición de datos II.
- 5.6 Adquisición de datos III.
- 5.7 Adquisición de datos IV.
- 5.8 Subrutina de TESTRAM I.
- 5.9 Subrutina de TESTRAM II.
- 5.10 Subrutina de RESETRAM.
- 5.11 Sección de control. Sonda.
- 5.12 Sección de memorias y decodificador de direcciones. Sonda
- 5.13 Sección de interfaces.
- 5.14 Sección de acondicionamiento de señal.

CAPITULO 6.

- 6.1 Eliminador de rebote.
- 6.2 Subrutina de opción manual.

LISTA DE TABLAS

CAPITULO 2.

2.1 Problemas de transmisión y soluciones.

CAPITULO 3.

3.1 Tabla de verdad del decodificador de direcciones.

Parte de memorias. La línea de control IO/M=0.

3.2 Tabla de verdad del decodificador de direcciones.

Parte de periféricos. La línea de control IO/M=1.

3.3 Problemas y soluciones.

CAPITULO 4.

4.1 Resumen de los códigos de programación.

4.2 Problemas y soluciones.

CAPITULO 5.

5.1 Consumo de potencia de los elementos de la sonda.

5.2 Resumen de las memorias utilizadas en el sistema.

PREFACIO

Esta tesis es acerca de un sistema de adquisición de datos digital con el microprocesador 8088 para recolectar información de variables físicas tales como presión y temperatura dentro de un pozo geotérmico y la lectura de éstas a través de un programa en Turbo Basic para presentar la información en forma de tabla de datos. Este documento incluye el proceso de diseño tanto analógico como digital para la implementación del prototipo experimental.

ORGANIZACION .

Se divide en cuatro partes. En la primera (Capítulo 1) se describe en que consiste la tesis, su alcance, su importancia y el funcionamiento general.

La segunda parte consiste de los capítulos 2 al 4 que detallan exclusivamente el software y el hardware del prototipo experimental. Se ilustran con flujogramas y esquemas eléctricos de la circuitería. El capítulo 2 hace referencia al programa realizado en Turbo Basic, en el capítulo 3 se desarrollan las técnicas de diseño empleadas para el circuito, tanto analógicas como digitales y en el capítulo 4 explica el desarrollo del programa monitor que gobierna al sistema de adquisición de datos con el microprocesador 8088.

En la tercera parte correspondiente al capítulo 5 se hacen notar los cambios necesarios para pasar del prototipo al sistema real conocido como "sonda". Esta sección es proyectada para un futuro cuando se desee implementar el circuito para operar a las condiciones ambientales en un pozo geotérmico.

Finalmente el capítulo 6 provee una aplicación alternativa del sis-

tema de adquisición de datos.

ASUNCIONES.

Para hacer efectiva la lectura de la tesis, se debe estar familiarizado con elementos y circuitos digitales, una comprensión de los sistemas binarios y hexadecimal, un profundo conocimiento del microprocesador 8088 o estar familiarizado con el 8088 ó 8085, conocer el funcionamiento de operacionales, convertidores analogo/digital y transductores de temperatura tipo termistor y RTD y de presión tipo transmisor diferencial y strain gage. Comprensión del lenguaje BASIC.

RECONOCIMIENTOS .

Agradecemos el apoyo bibliográfico y técnico que proporcionaron las siguientes personas: Rodolfo Machón, quien nos proporcionó material suficiente para la comprensión del micro y enlaces de comunicación con el puerto serie de la computadora y utilizar su computadora para editar cierta parte de este documento y Francisco Arbizú proporcionando sugerencias en cuanto al software.

Especial agradecimiento a mi novia Lissette Zapata y su madre Zoila Aguilar quienes contribuyeron en gran medida de prestar su computadora para realizar este documento y a la impresión del mismo, que sin su ayuda hubiera resultado menos exitosa.

Carlos Roberto Zapata.

CAPITULO 1

SISTEMA DE ADQUISICION DE DATOS.

temperaturas superiores a los 400°C, y además el computador que interpreta los resultados es una computadora de superficie que ha sido diseñada solo para esa función; y el costo del computador y la sonda es de alrededor de \$35,000.00.

Lo anteriormente expuesto motivo al departamento de investigación para crear un diseño alternativo que cumpliera con los requisitos anteriores y cuyo rango de funcionamiento, como temperatura, supere al que ya posee y se adecúe a los pozos geotérmicos y además resultare de bajo costo y utilizarse una computadora de uso comercial.

El departamento de instrumentación hizo un estudio previo de la operación del sistema, evaluando entre dos posibilidades, es decir:

- a. Electrónica de fondo, esto es, que gran parte del sistema (circuitaría) debe estar dentro del pozo.
- b. Electrónica de superficie, es decir, que sólo los transductores estarán dentro del pozo.

Dentro de la alternativa de electrónica de fondo existe las posibilidades de toma de datos de tiempo diferido o de tiempo real. La tiempo diferido es aquella donde solo se almacenana los datos en la sonda mientras está en el pozo, y cuando se se ha sacado se recuperan para su procesamiento; esto implica que no existe ninguna conexión eléctrica entre la sonda y el computador de superficie durante la medición. La de tiempo real transfiere y procesa inmediatamente la información para su procesamiento final y presentación en la computadora de superficie, esto implica una permanente conexión eléctrica entre la sonda y el computador de superficie.

El departamento de instrumentación se inclinó por el sistema de electrónica de fondo de tiempo diferido, debido a que el costo de un

cable que estuviera protegido a las condiciones de un pozo geotérmico es de precio elevado y además la profundidad de un pozo puede sobrepasar los 1000 mts.

En términos generales el sistema consta de dos partes:

- a. Sistema electrónico.
- b. Sistema termodinámico-mecánico.

El sistema electrónico es todo el hardware de adquisición de datos y su transferencia al computador, así como el software de la computadora.

El sistema termodinámico-mecánico es la cubierta que lleva el sistema electrónico para protegerlo de las altas temperaturas y presiones así como de la corrosión y también de la disipación de calor de los sistemas internos. Con lo anteriormente expuesto el departamento de instrumentación ha establecido un medio por el cual estudiantes de ingeniería a nivel de tesis, elaboren proyectos que dicho departamento tiene proyectado ejecutar y que no han sido elaborados por falta de material humano capacitado en diversas especialidades; y como el sistema de adquisición de datos necesita a personas con conocimientos en electrónica, termodinámica, química y otros; el departamento ha establecido que la sonda sea ejecutada en dos fases, lo primero es elaborar el diseño e implementar a nivel de laboratorio la circuitería electrónica y la programación que ella requiere, ésta la realizarán personas especializadas en el área de electrónica, de esta primer fase se obtendrán datos importantes tales como las dimensiones reales de la circuitería dentro de la sonda, así como la disipación de potencia del circuito y demás especificaciones del mismo. Estos datos son muy importantes en el diseño de la cubierta protectora para obtener las características óptimas en ésta. El diseño de la cubierta protectora y ensamble final está

comprendido en la segunda fase del proyecto la cual piensa ser encomendada a personas o empresas con los recursos y la especialización que ello requiere.

1.3 Descripción general del circuito

En la fig.1.1 se muestra el diagrama a bloques del sistema de adquisición de datos de tiempo diferido. El circuito dentro de la sonda debe medir variables físicas propias de los pozos geotérmicos, en el diseño que se implementa se trabaja con mediciones de temperatura y presión. El sistema funciona de la siguiente manera:

Primeramente debe traducir los valores de presión y temperatura a un código digital, para eso se vale de transductores, circuiteria analógica asociada a ellos y un convertidor analogo/digital y un multiplexor dado que son dos señales analógicas a medir y convertir. El sistema tiene como base un microprocesador el cual se encarga de recoger los datos del código digital correspondiente a los valores medidos, procesarlos y luego guardarlos en una memoria; todo este proceso se efectua en base a un programa maestro (monitor) interno al sistema pero que emplea datos asignados desde el exterior (programación externa previa a la operación) que involucran la variable tiempo y diversas formas de operación del sistema. El sistema lo completan circuitos de direccionamiento, de temporización y la interfaz de comunicación serie en formato RS-232C.

Además el circuito de la sonda, para obtener la información de la computadora se tiene un interfaz y un programa de comunicación y procesamiento final y presentación de los datos; y es por medio de un programa donde el usuario selecciona el modo de operación del sistema, lo cual involucra los siguientes parámetros:

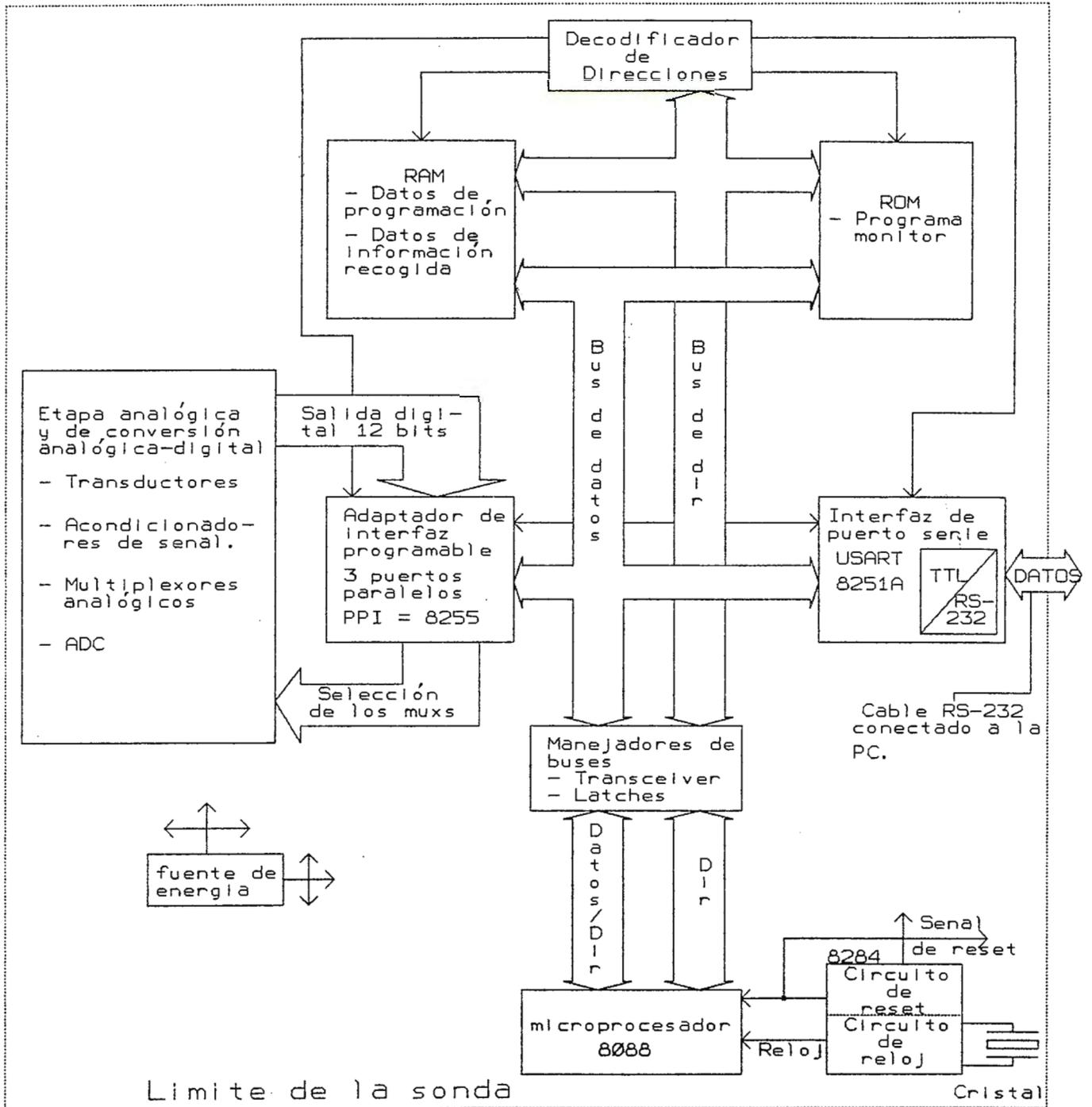


Fig. 1.1 Diagrama a bloques del sistema

- a. Cuales son las variables a medir.
- b. Cuál es el tiempo antes de empezar la toma de datos.
- c. Cuanto tiempo debe esperar entre cada toma de datos en el pozo.
- d. Cuántas es el número de muestras por variable.
- e. Cómo se realizará la toma de datos: Simple o Promedio.
- f. Cuántas estaciones se realizaran dentro del pozo.

Todos los literales anteriores son los que el usuario utiliza durante la programación a través de una PC.

En resumen el trabajo de tesis comprende lo siguiente:

1.- Diseño del Hardware y el Software. El hardware comprenderá los transductores, acondicionador de señal, ADC, microprocesador, memorias RAM, EPROM, adaptador de interfaz y reloj del sistema. El software se divide en dos secciones que corresponden a uno en Ensamblador y otro en lenguaje de alto nivel, que para este caso es el Turbo Basic; esto permitirá al usuario una programación que le sea fácil de entender.

2.- Diseño y quemado del programa monitor. Es el que se encargará de controlar las funciones de la sonda en el pozo.

3.- Diseño de la interfaz. Es el hardware y el software que requiere para establecer una comunicación con el computador a través del puerto serie. (RS-232C)

4.- Especificaciones del consumo de potencia de todo el sistema.

5.- Sugerencias en cuanto al uso de baterías y otras en cuanto a la circuiteria. Esta en función de los problemas que se presentaron a lo largo del proyecto.

Queda establecido que se trata de un prototipo experimental y que la implementación que se llevo a cabo se realizó a nivel de breadboard, esto es lo que lo diferencia con el sistema real proyectado de la sonda

2.1 Funciones del sistema y generalidades. XXXXXXXXXXXXXXXXXXXX

El Software de la PC consiste en un programa (realizado en alto nivel) que tendrá una serie de opciones donde un operador le indicará al sistema como va a realizar su trabajo. Las opciones como: Variable a medir, número de muestras a tomar, cómo será la toma de datos, etc; son convertidas a una serie de códigos hexadecimales que el sistema pueda entender, y una vez listos cada uno de los códigos, son transmitidos por medio del puerto serie RS-232C de la computadora. El Software no solo incluye la programación del sistema sino también la recuperación de los datos almacenados en la sonda. Este programa consiste en que el computador le pedirá a la sonda toda la información recabada, esa información por si sola no significa mucho; por lo cual es necesario interpretar esos resultados antes de ser mostrados al operador en forma de tabla de datos.

Resumiendo, el Software de la PC consiste en dos grandes programas que son de programación y lectura; en el primero a través de códigos se le dice a la sonda que es lo que va a hacer, y en el segundo, se le piden los datos obtenidos y son interpretados y pasados a unidades físicas que el operador pueda entender.

CAPITULO 2

SOFTWARE DE LA PC.

y ese único valor lo almacenará en memoria.

f. **Cuanto tiempo piensa dejar la sonda en el pozo (Horas, minutos).** Este corresponde al tiempo de trabajo de la sonda. El programa en base a la información anterior calculará la máxima cantidad de datos que puede obtener con esos parámetros de programación y al mismo tiempo el programa decidirá si puede aceptar esa cantidad de datos, de lo contrario tendrá que disminuir el tiempo que la sonda permanecerá en el pozo. Si el cálculo está dentro del rango, que es de 2000 datos de temperatura y 2000 datos de presión ó 4000 de temperatura o presión, entonces le dirá el máximo número de estaciones que puede realizar en el pozo.

g. **Cuántas estaciones va a realizar.** Este dato se refiere al número de paradas que la sonda hará dentro del pozo durante su trabajo.

Con toda esta información que el operador ha programado, el paso a seguir es transmitirla a la sonda. Esa información que la conforman códigos decimales son transmitidos y recibidos en formato hexadecimal. El programa monitor del sistema se encargará de almacenarlos en su memoria.

La rutina de programación además contiene otras funciones importantes al sistema, y se explican a continuación:

1. **TESTRAM.** Esta rutina se encarga de verificar el estado de la memoria de la sonda. En este caso un código es enviado a la sonda y cuando ésta lo recibe ejecuta un programa donde verifica c/u de las localidades de memoria, si la prueba fue pasada, la sonda enviará un código a la PC donde éste esperará recibir ese código, entendiéndose que la memoria del sistema no tiene daños. Si el código que espera el programa no es el correcto entonces el programa pondrá un mensaje en la pantalla, que dice: **La memoria del sistema no es confiable.** Esto implica que los chips de memoria tiene daños internos.

2. PRUEBAL. Para asegurarse que la línea de comunicación se encuentran en buen estado, fue necesario crear una rutina que ejecutara una prueba de la línea y tomara decisiones en base a los resultados obtenidos de dicha prueba; es así como Pruebal cumple con estos requisitos, y la forma que ejecuta su trabajo es:

Es necesario primero pedirle a la sonda que una Prueba de Línea va a realizarse, por tanto se le envía el código respectivo de Pruebal, la sonda al recibirlo y entenderlo se queda en espera de recibir los datos de prueba. Pruebal entonces comienza a enviar números decimales enteros de 1 a 100, al terminar de transmitirlos se queda en espera de recibirlos y preparar un contador de errores (Errores). Cuando la sonda recibió los 100 datos, ésta los almacena en su memoria y ahora los tomará y comenzará a transmitir uno a uno. Pruebal espera a recibir en el mismo orden que transmitió; de haber disparidad en los datos recibidos Errores se va incrementando. Al finalizar la secuencia de recepción, Pruebal hace la decisión si existe el 10% de error, si esto ocurre muestra en pantalla **Existen problemas en la línea**. Con este debe entender que bajo las actuales condiciones no se puede transmitir y/o recibir confiablemente. Al existir menos del 10% de error el programa continua en su ejecución normal. Pareciera ser que permitir un 10% de error es bastante alto en el caso de comunicaciones, pero con este porcentaje se asegura que una rutina que se llama **obtener la mayoría** funcione correctamente, el uso de esta rutina se explica más adelante y al final, en el cuadro de **problemas y soluciones** se explica el ¿por qué? es necesario utilizarla.

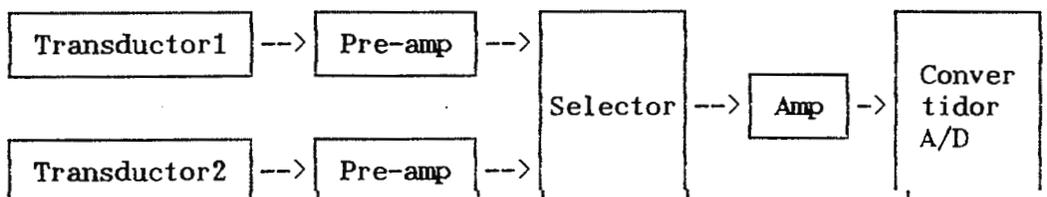
3. AUTOCALIBRACION. El trabajo de esta rutina consiste en crear un archivo de los códigos enviados por la sonda, pero antes de realizar esto, se ejecutan una serie de cálculos. Como la sonda almacena sus datos

en forma de bytes, donde una palabra, contiene la información de un dato de calibración. El programa recibirá $2*n$ datos, donde n =# de códigos, y ese número de códigos es nueve. Por tanto la rutina de Autocalibración después de haber convertido a decimal los datos, debe de formar de los dos datos uno solo, y estos serán pasados por un proceso de comparación. Los nueve datos deben estar en un rango del 10% de los esperados, para asegurar que la parte analógica se encuentra trabajando bien; si tan solo un dato no se encuentra en ese rango entonces la rutina de Autocalibración vuelve a ejecutarse, naturalmente esto implica un contador que estará monitoreando cuantas veces se repite la rutina, si se encuentra que el contador excede un cierto rango, el programa mostrará en pantalla: **La parte Analógica ya no es confiable.** Nos da entender que no es posible obtener datos reales en la medición y el sistema se echo a perder, pero si el contador no sobrepasa el límite establecido hará el archivo de los códigos recibidos.

Como dentro de los códigos recibidos también se encuentran el estado de las baterías, la rutina verificará si las baterías están o no agotadas de ocurrir el caso de que esten agotadas se le dirá al operador: **Mal estado de las baterías,** causando de inmediato el cambio del banco de baterías.

2.2.1 Justificación de la calibración.

El Sistema que se ha diseñado, y en general, la mayoría de los sistemas en existencia siguen un proceso como el que sigue:



La razón por la cual una calibración es necesaria, es porque todo fabricante hace los ajustes necesarios para que cada valor de la variable de salida del transductor sea representada por un código binario más seguro de manejar (menos distorsión por ruido) y procesar que en etapas posteriores se recuperará y presentará como el valor medido en las unidades correspondientes. Puesto que la etapa conocida como acondicionamiento de la señal está compuesta de amplificadores de corriente continua se encuentra expuesta a sufrir variaciones de respuesta debido al tiempo y a la temperatura, lo mismo es válido para el convertidor Analogo/Digital, e incluso para el transductor.

La forma usual de calibración es usando señales de referencia, es decir, señales de valor conocido y constante con respecto a cualquier variable, luego se aplica esta señal a cierta parte del sistema y se recupera en alguna de las etapas siguientes por lo que la señal debe pasar; y entonces comparar con el valor obtenido con el resultante típico esperado.

Si el resultado es aceptable el sistema o la etapa sensada está operando bien, pero si existe un error, el sistema deberá proceder a corregirlo, esto implica tomar una de las dos alternativas:

- a) Hacer una realimentación, es decir, modificar una variable en la etapa que introduce error, a fin de eliminarlo o reducirlo.
- b) Compensar la señal de salida de la etapa a fin de llevarla al valor deseado.

Ambas alternativas pueden ser llevadas a cabo en varias formas tales como: Proporcionalmente, por niveles, etc.

II. Secuencia de como operan cada una de las funciones.

El programa realizado es ejecutable y el archivo principal para correr el programa se llama LIZA.EXE, que además se cuenta con otros dos programas más que son llamados desde LIZA, estos son los programas TECLAS.TBC Y RECIBA.TBC, sin estos archivos no se puede ejecutar completamente el programa. Además antes de echar a andar el programa ejecutable se recomienda que la sonda ya se encuentre conectada al puerto serie del computador pues cuando se selecciona la opción de programar éste inmediatamente empieza a correr la rutina de TESTRAM y para ello la sonda debe estar conectada, si por alguna razón el operador no la ha conectado entonces el programa después de cierto tiempo pondrá un mensaje en la pantalla indicándole que la sonda debe conectarse al puerto serie. Si la computadora con que esta trabajando tiene dos puertos serie utilice siempre el puerto COM1, pues el programa LIZA sólo reconoce la dirección de este puerto para establecer la comunicación. Además si el computador no cuenta con puerto serie formato RS-232C, pero si con puerto de mouse cuya configuración es de nueve pines, entonces se desconectan las conexiones del puerto serie RS-232C cuyos pines son el 2, 3 y 7 y se conecta respectivamente en el mismo orden en el mouse.

Para ejecutar el programa, si se encuentra en la unidad A, digite lo siguiente:A>LIZA esto inmediatamente seguirá la secuencia mostrada en los diagramas presentados desde la Fig.2.1 hasta la Fig.2.8.

En la Fig.2.1 se muestra el flujograma de la secuencia de decisiones que el programa realiza en espera de recibir información desde el teclado y en la Fig.2.2 se muestra la ventana tal y como se ve en el computador que corresponde a la Fig.2.1.

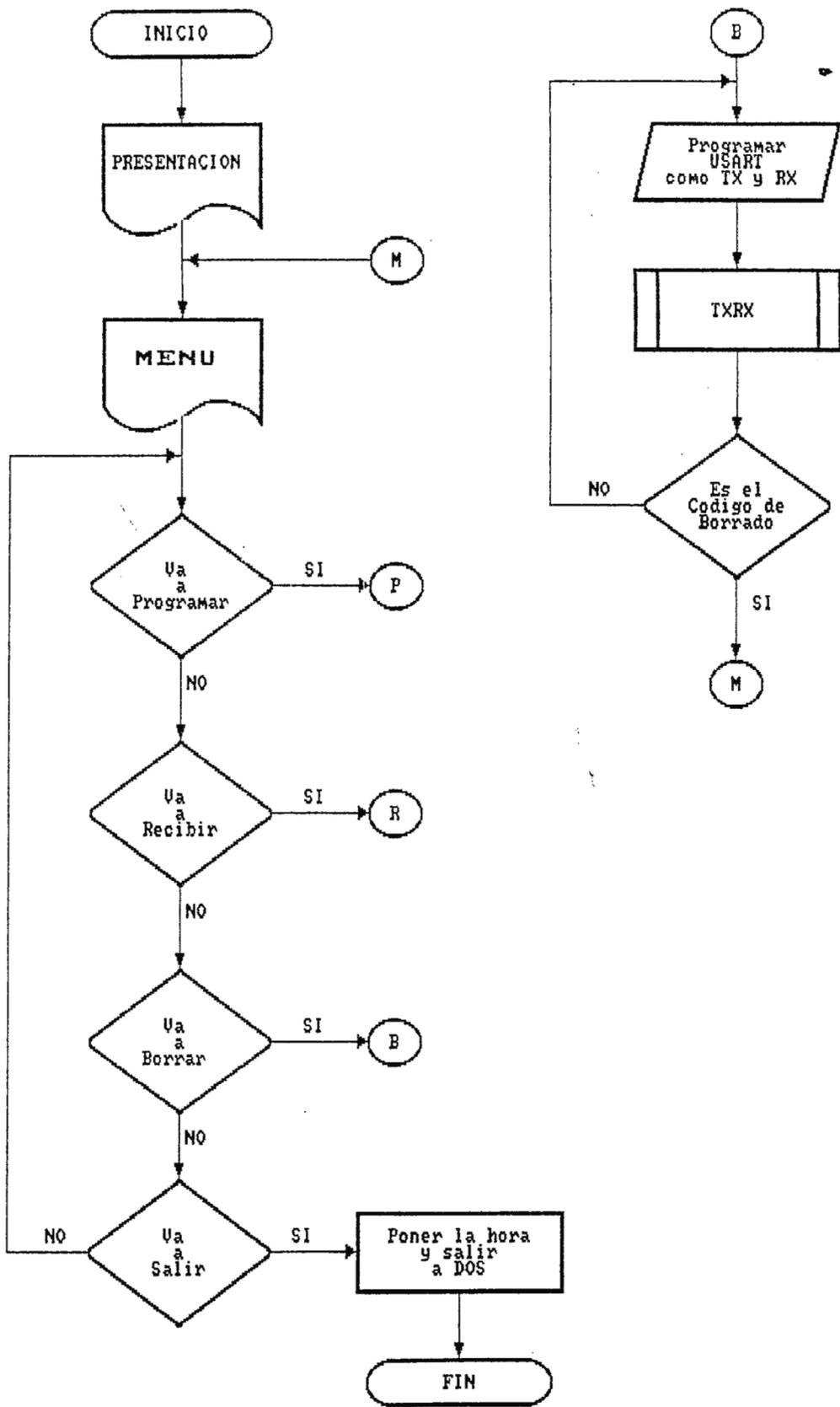


Fig.2.1 Menu de Programación de la PC.

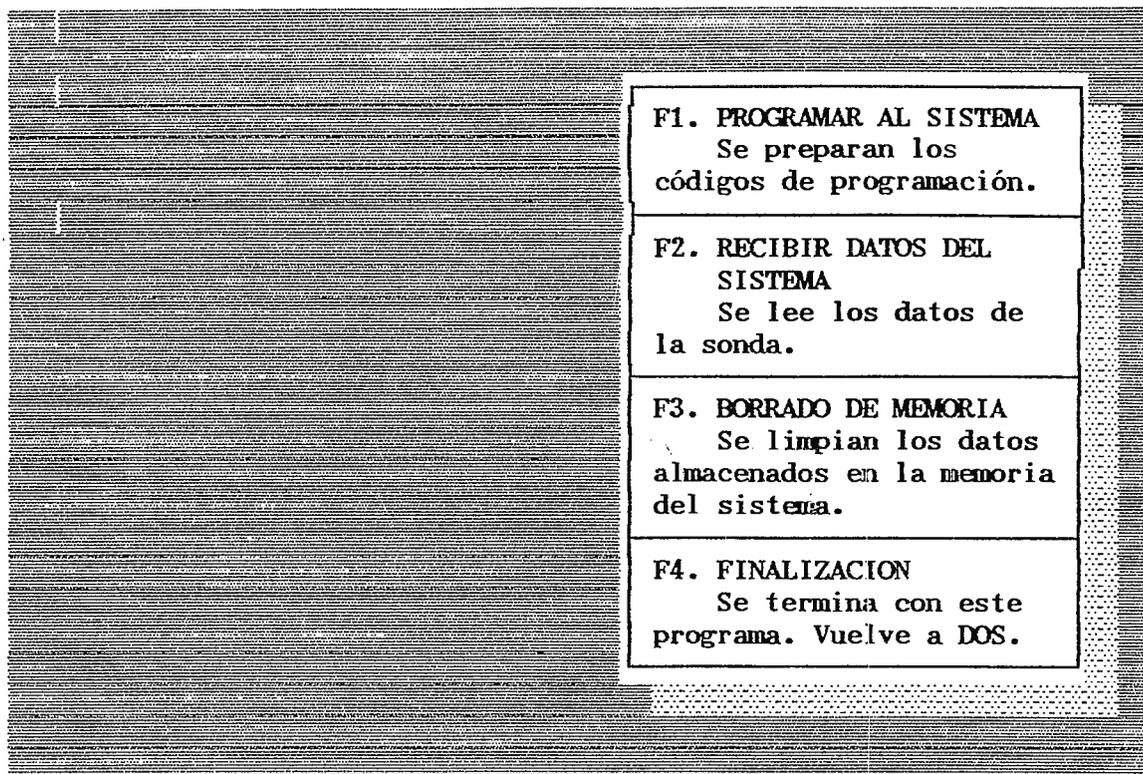


Fig.2.2. Presentación de menus de la PC.

En la ventana (Fig.2.2) puede apreciarse cuatro opciones diferentes, de las cuales el programa espera la respectiva selección (F1 a F4).

Si selecciona F1, que es programar al sistema el programa inmediatamente presentará en pantalla una serie de mensajes correspondientes a las rutinas que está ejecutando (Fig.2.3). En pantalla se mostrará una pequeña ventana que dice: **Revisando la memoria del Sistema**, esto es, la rutina **TESTRAM** (Fig.2.4), se carga el código para **TESTRAM** y se va a la subrutina **TXRX**; al retornar de la subrutina se pregunta si es el código esperado, de ser falso se verá en pantalla: **La memoria del sistema no es utilizable**, y retornará al **MENU**, pues de lo contrario se ejecutará **PREUBAL**. Si la prueba de línea es satisfactoria se ejecutará a continuación **AUTOCALIBRACION** y el operador verá en pantalla: **Efectuando la Autocalibración**.

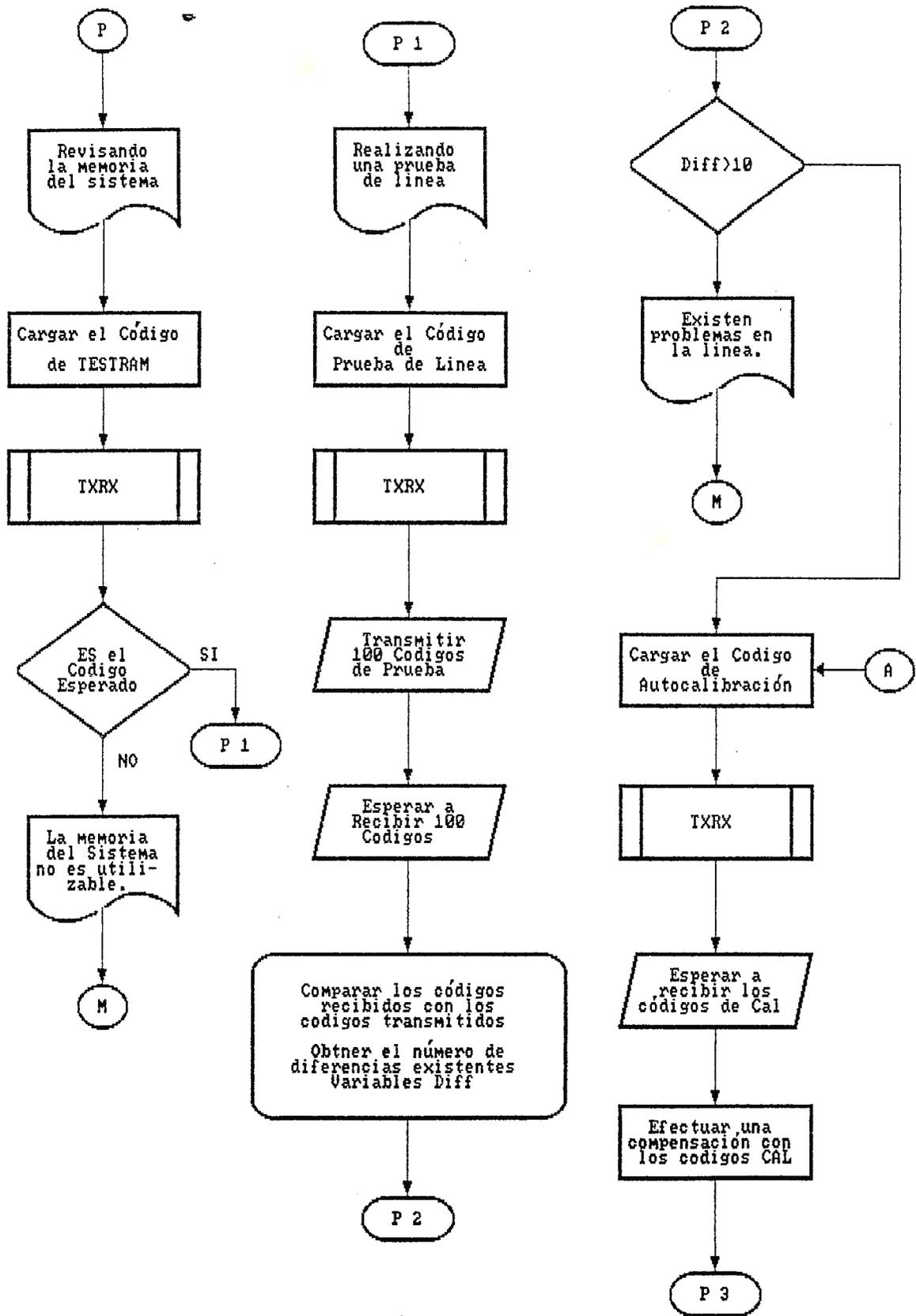


Fig.2.3 TestRAM, Prueba de línea y Autocalibración cuando se programa al sistema

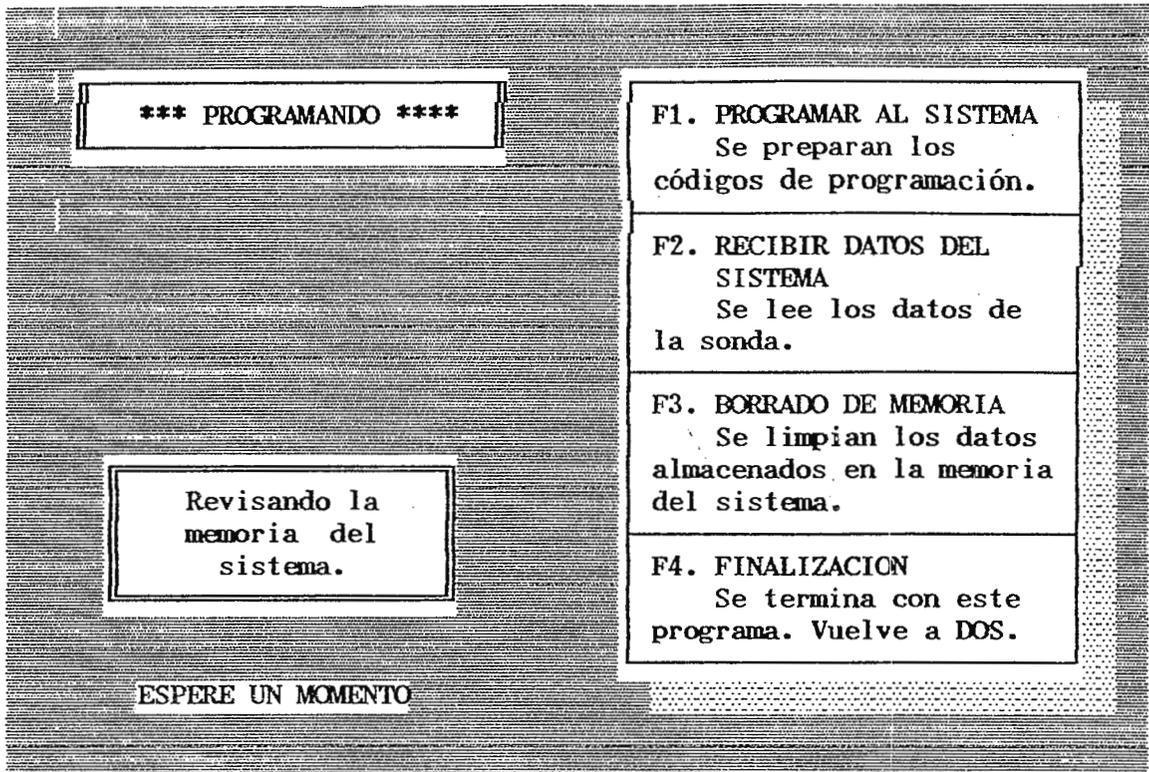


Fig.2.4 Mensajes automáticos antes de la programación.

Hasta el momento el operador ha observado los cambios en la pantalla y las rutinas TESTRAM, PRUEBAL y AUTOCALIBRACION se han estado ejecutando sin que el operador intervenga. Cuando la rutina de Autocalibración ha concluido el programa solicita al operador que le suministre la información (I. Esquema Funcional).

*** PROGRAMANDO ****	<p>F1. PROGRAMAR AL SISTEMA Se preparan los códigos de programación.</p>
	<p>F2. RECIBIR DATOS DEL SISTEMA Se lee los datos de la sonda.</p>
	<p>F3. BORRADO DE MEMORIA Se limpian los datos almacenados en la memoria del sistema.</p>
	<p>F4. FINALIZACION Se termina con este programa. Vuelve a DOS.</p>

Cuantas variables va a medir: _

Fig.2.5(a).

La Fig.2.5(a) muestra la primera información que el programa solicita al operador. El sistema está definido para operar con solamente dos variables, así de esta forma el operador digitará un máximo de 2. Si digita un número mayor a éste o menor a 1 el programa no lo aceptará y continuará en la misma pantalla hasta que digite el dato correcto.

En la Fig.2.6 se muestra el flujograma que conlleva la secuencia de la introducción de la información.

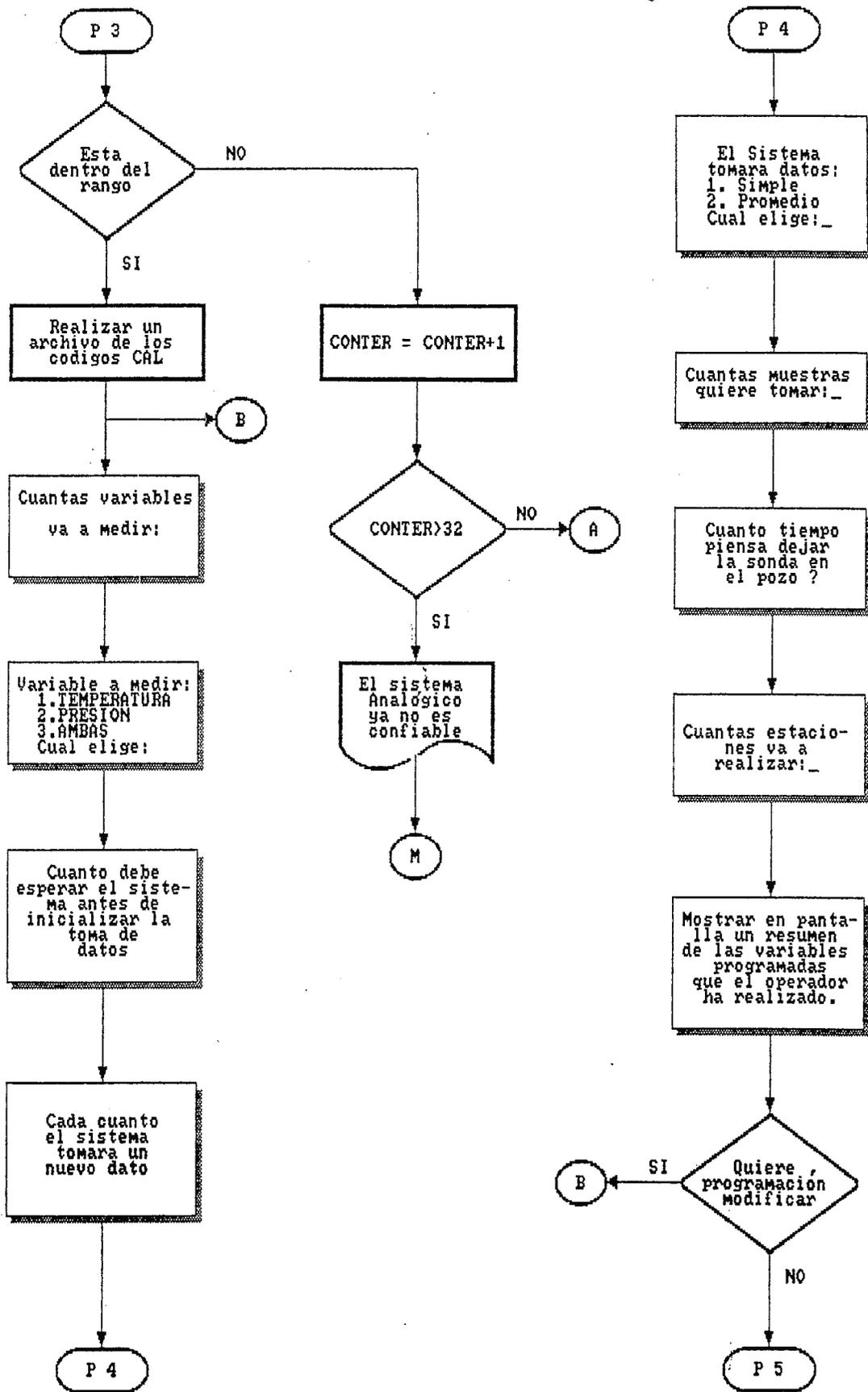


Fig.2.6 Parametros de programación del sistema.

*** PROGRAMANDO ****	<p>F1. PROGRAMAR AL SISTEMA Se preparan los códigos de programación.</p>
	<p>F2. RECIBIR DATOS DEL SISTEMA Se lee los datos de la sonda.</p>
	<p>F3. BORRADO DE MEMORIA Se limpian los datos almacenados en la memoria del sistema.</p>
	<p>F4. FINALIZACION Se termina con este programa. Vuelve a DOS.</p>

<p>Variable a medir: 1. TEMPERATURA. 2. PRESION. 3. AMBAS. Cual elige: _</p>
--

Fig.2.5(b)

En la Fig.2.5(b) se muestra una ventana donde el programa solicita del operador cual es la variable que va a medir. Para este caso se tiene que digitar un número entre 1 y 3, cualquier otro número causará un pitillo indicando que no es válida la opción seleccionada por el operador y nuevamente continua en esta pregunta hasta que se digite el número esperado por el programa. En la Fig.2.5(a), el operador selecciono el número de variables con que va a trabajar, luego si seleccionó 2 variables y solo quiere trabajar con Temperatura o Presión esto causará un error y el programa mostrará en pantalla: **Usted ha programado dos variables,** siendo necesaria la introducción del número 3. De igual manera si seleccionó 1 variable cuasará error que mostrará en pantalla: **Solo puede trabajar con una variable,** así de esta manera el operador si por algún error digita un dato incorrecto el programa se encargará de avisarle.

Una vez que el programa ha solicitado cual (es) son las variables a medir, continua de acuerdo a la secuencia de la Fig.2.6 con el tiempo que debe esperar la sonda antes de iniciar la adquisición de los datos en el pozo, de esta manera en la Fig.2.5(c) se muestra la ventana para esta información. Puede notarse que el operador está restringido a un tiempo entre 1 a 30 minutos, esto es así, pues los requerimientos que del sistema se exigen, cumplen con este parámetro. El operador puede digitar cualquier número fuera de este rango, pero esto causará dos pitillos por medio del parlante de la computadora que le indican que su dato está fuera del rango establecido y tiene que dar un número dentro del permitido.

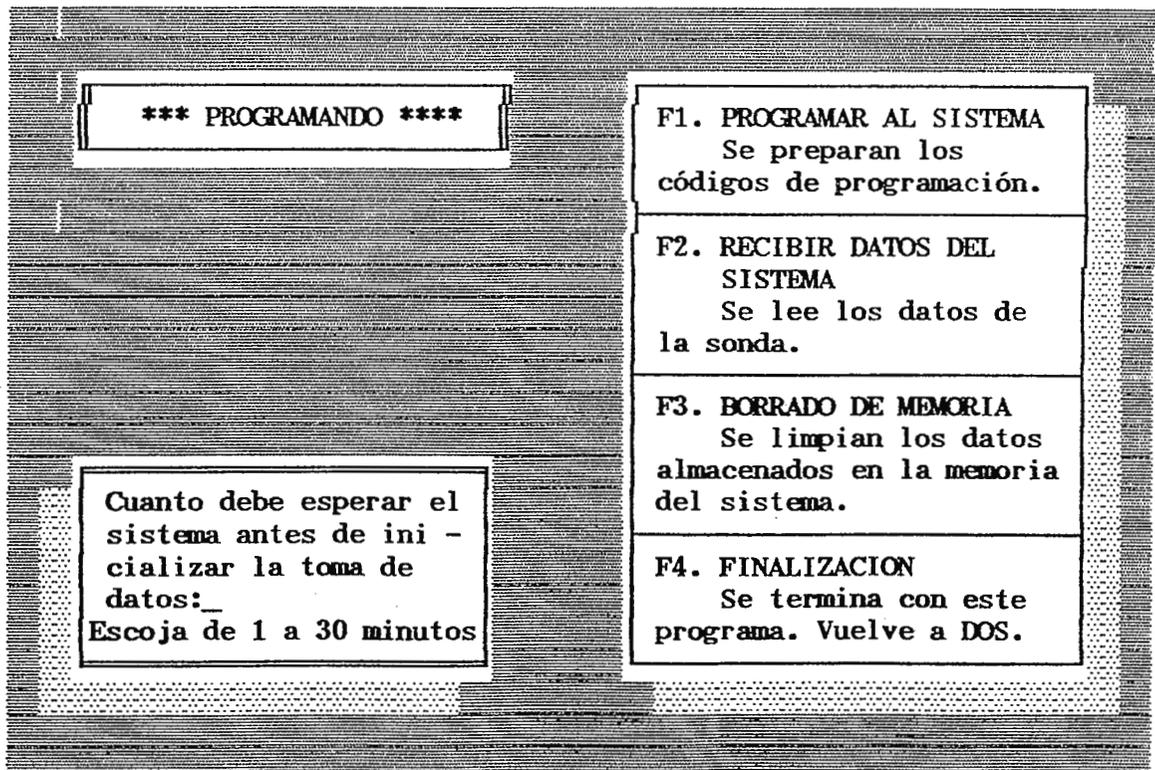


Fig.2.5(c).

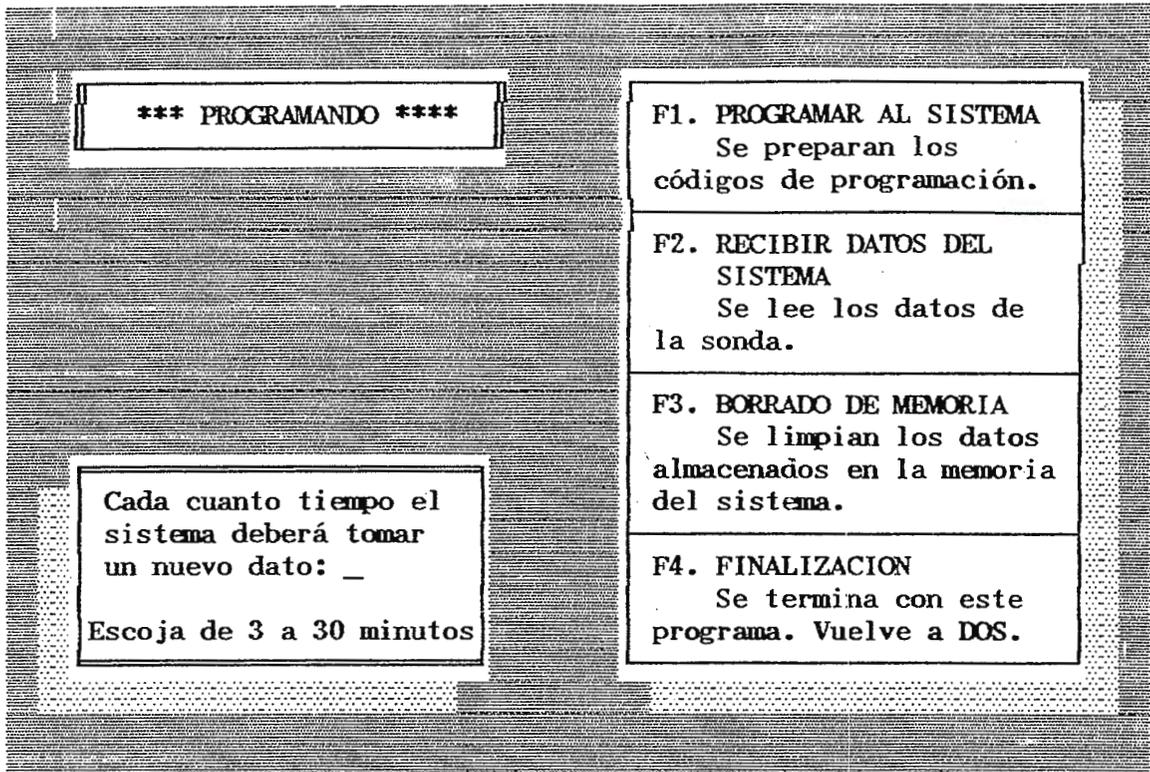


Fig.2.5(d).

Una vez establecido el tiempo de espera para que el sistema tome datos al ser desconectada del computador y llevada al pozo, es necesaria también la información del tiempo que debe tomar entre cada estación. Tal y como se muestra en la Fig.2.5(d), el tiempo está restringido de 3 a 30 minutos. El tiempo de 3 minutos puede parecer lento, pero esto es así, puesto que la velocidad del transductor de temperatura es tal que hay que esperar aproximadamente 2.5 minutos para tomar una lectura que sea estable en el transductor. Naturalmente este tiempo puede ser más corto pero no se asegura mediciones confiables.

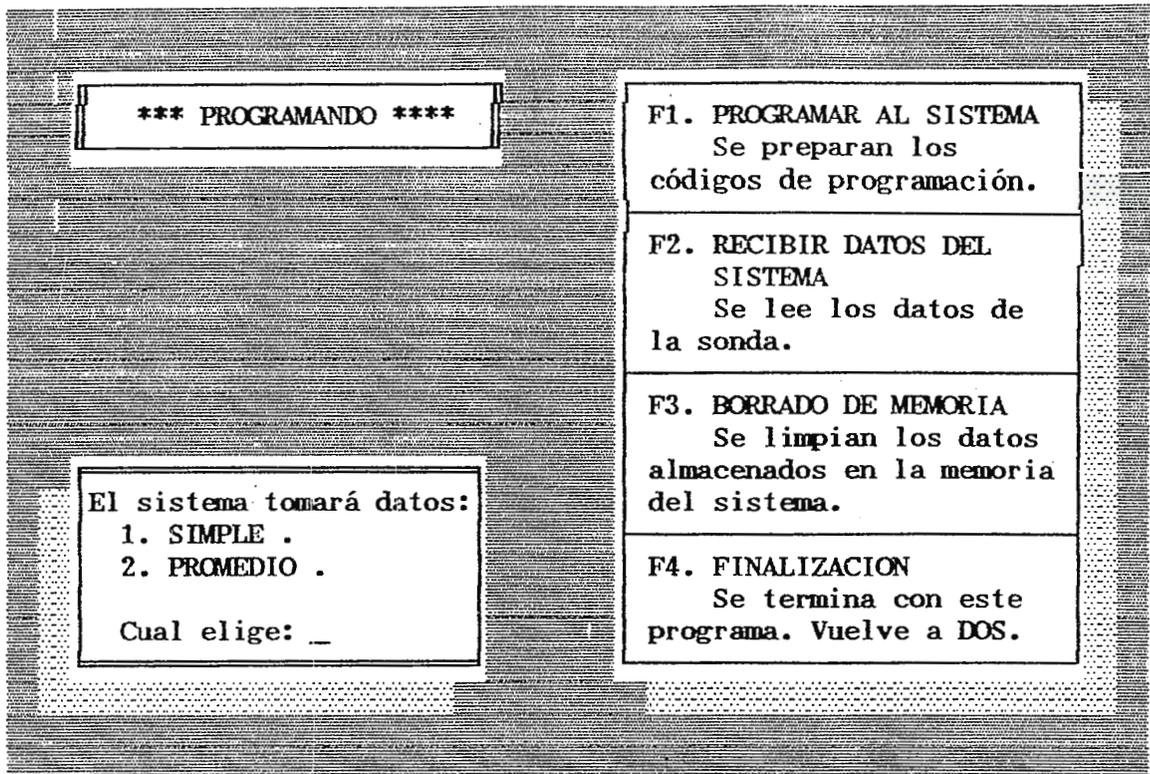


Fig.2-5(e).

El sistema de adquisición de datos tiene la versatilidad de realizar una toma simple o promedio de los datos. Esto es, que si se programa para un muestreo simple de datos, dato a dato es almacenado en la memoria del sistema y si es promedio, del total de datos sólo se obtiene uno que luego es almacenado en la memoria. Este último método implica un mejor aprovechamiento de la memoria cuando se está trabajando con una gran cantidad de información pero resulta desventajoso por el hecho de que un sólo dato no proporciona exactamente el comportamiento del pozo en ese punto. En la fig.2-5(e) el programa solicita esta información.

Ya sea que sea una toma simple o promedio de los datos, el sistema debe tener además el dato del número de muestras que tomará por variable en un punto determinado del pozo geotérmico. Para ello en la Fig.2-5(f) se muestra la ventana que el programa crea para obtener esa información. El número de muestras esta limitado a un máximo de 10, así que el operador puede seleccionar una cantidad dentro de ese rango, el programa no permitirá mayor número de muestras de las establecidas.

<p>*** PROGRAMANDO ****</p>	<p>F1. PROGRAMAR AL SISTEMA Se preparan los códigos de programación.</p>
	<p>F2. RECIBIR DATOS DEL SISTEMA Se lee los datos de la sonda.</p>
	<p>F3. BORRADO DE MEMORIA Se limpian los datos almacenados en la memoria del sistema.</p>
<p>Cuantas muestras quiere tomar: _ Puede tomar de 1 a 10.</p>	<p>F4. FINALIZACION Se termina con este programa. Vuelve a DOS.</p>

Fig.2.5(f).

El programa presenta además la cantidad exacta de datos que puede tomar en un determinado tiempo de trabajo, esto lo realiza de acuerdo a los parámetros anteriores que se ha establecido. Como se aprecia en la Fig.2.5(g) el programa solicita del operador el tiempo en horas y minutos a continuación de ello se le presenta en pantalla cuantos datos de cada variable podrá tomar y el número de estaciones que podrá ejecutar. Con esta información el operador tendrá a su mano la cantidad de paradas que la sonda puede ejecutar dentro del pozo para una determinada cantidad de datos. Se establece además que el operador pueda manejar mayor cantidad de datos, tan solo modificando los primeros parámetros de información o aumentando el tiempo en que permanecerá la sonda en el pozo.

*** PROGRAMANDO ****	
<p>Cuanto tiempo piensa dejar la sonda en el pozo: HORAS =2 MINUTOS =30 Con los datos que ha programado y con este tiempo de trabajo, usted puede tomar: 250 datos de temperatura 250 datos de presión Total : 500 datos Y un total de estaciones de: 50 Si desea obtener más datos se le recomienda aumentar el tiempo que permanecerá la sonda en el pozo.</p>	<p>F1. PROGRAMAR AL SISTEMA Se preparan los códigos de programación.</p> <p>F2. RECIBIR DATOS DEL SISTEMA Se lee los datos de la sonda.</p> <p>F3. BORRADO DE MEMORIA Se limpian los datos almacenados en la memoria del sistema.</p> <p>F4. FINALIZACION Se termina con este programa. Vuelve a DOS.</p>
PRESIONE ALGUANA TECLA PARA CONTINUAR	

Fig.2.5(g).

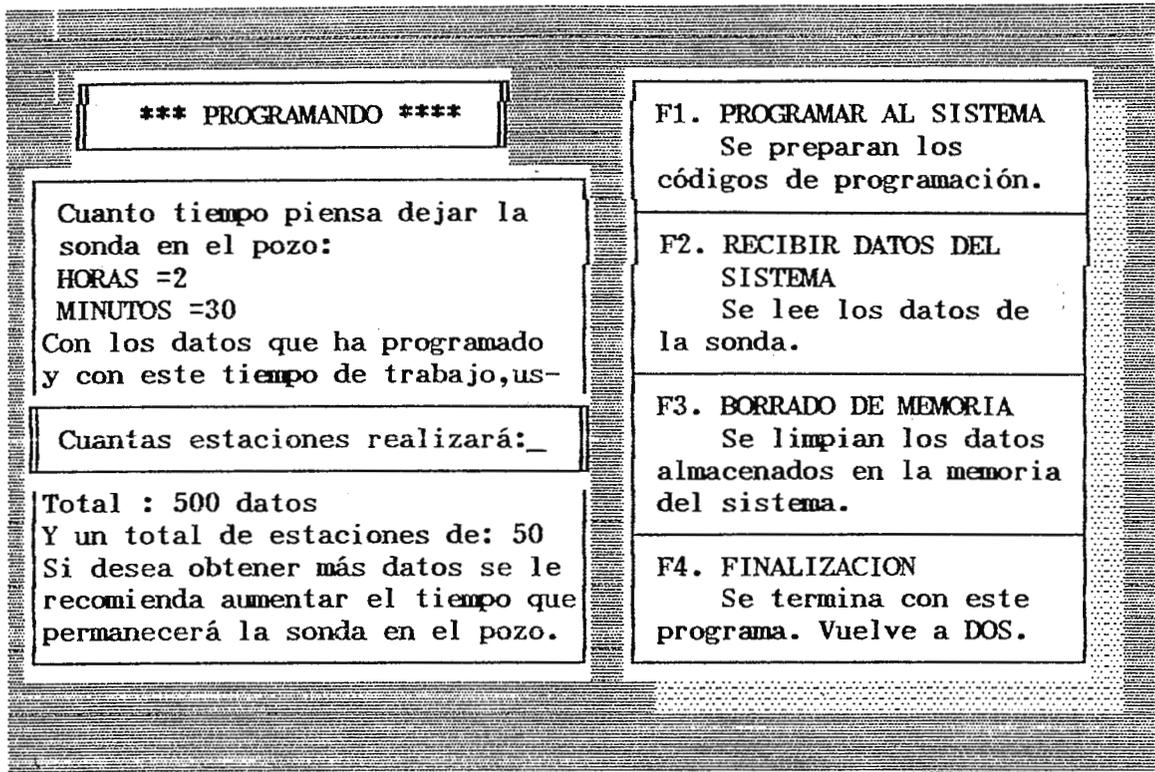


Fig.2.5(h).

La Fig.2.5(h) nos mostrará que el programa solicita el número de estaciones que se van a realizar dentro del pozo, como ejemplo el máximo que se permite es de 50, si el operador solicita más de esta cantidad el programa simplemente lo invalidará hasta encontrar una cantidad permitida dentro del rango.

Así, de esta manera, el proceso de programación de los datos necesarios para que la sonda trabaje ha concluido, solo basta para que sean enviados a ella y verificados para estar seguro de la información grabada en la memoria del sistema.

A continuación el programa realiza un resumen de todos los parámetros que el operador ha realizado (Fig.2.5(i)).

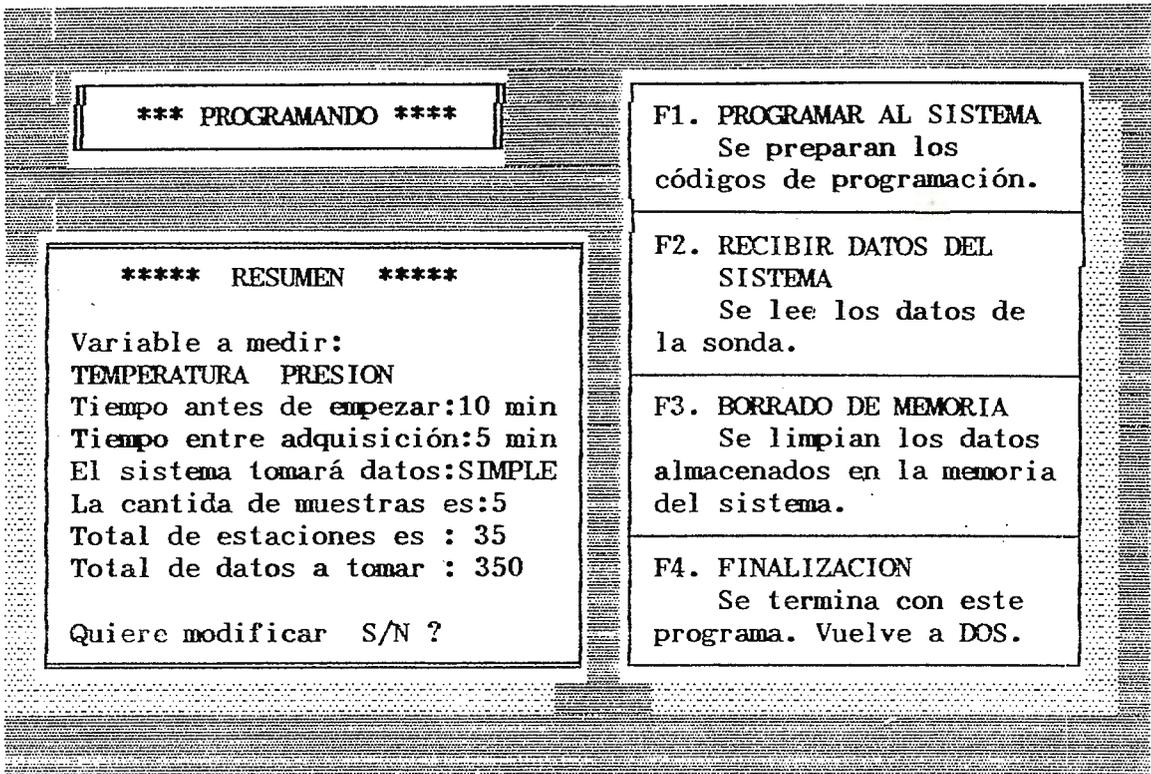


Fig.2.5(i).

En este momento el operador tiene la ventaja de recordar cada uno de los pasos que realizó anteriormente y con la oportunidad de modificar algunas de ellas si es que así lo desea. El programa pregunta si quiere realizar modificaciones, para ello bastará oprimir la tecla S e inmediatamente volverá a la Fig.2.5(b), y cambiará ya sea algunos de los parámetros o totalmente iniciará una nueva programación. El programa no ofrece la posibilidad de modificar solamente un dato, sino más bien de modificar todos, por tanto tenga presente esto si quiere modificar la programación.

Si está satisfecho con la programación realizada, entonces es el momento de transmitir cada uno de esos datos que se ven en la pantalla a la sonda, esto se ejecutará inmediatamente si presiona la tecla N, provocando que se realice la secuencia como se muestra en la Fig.2.7.

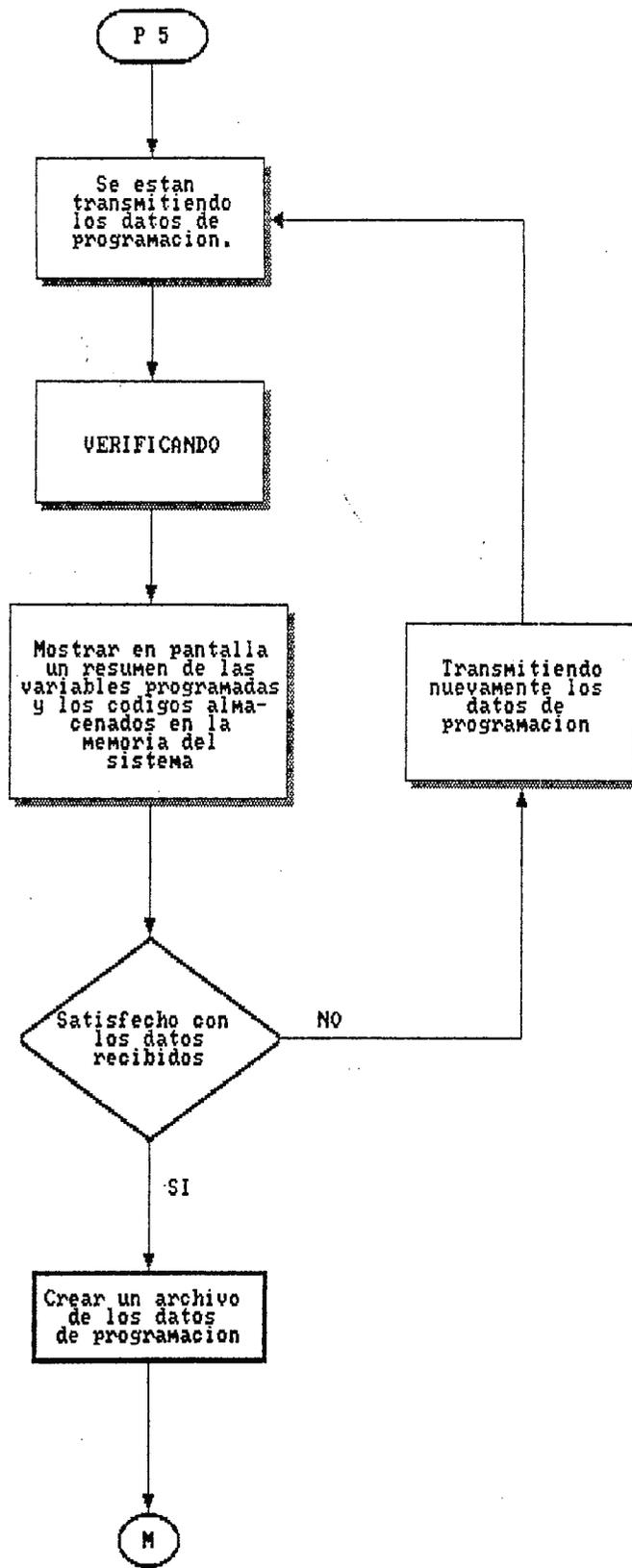


Fig.2.7 Transmisión y Verificación de los datos en la sonda.

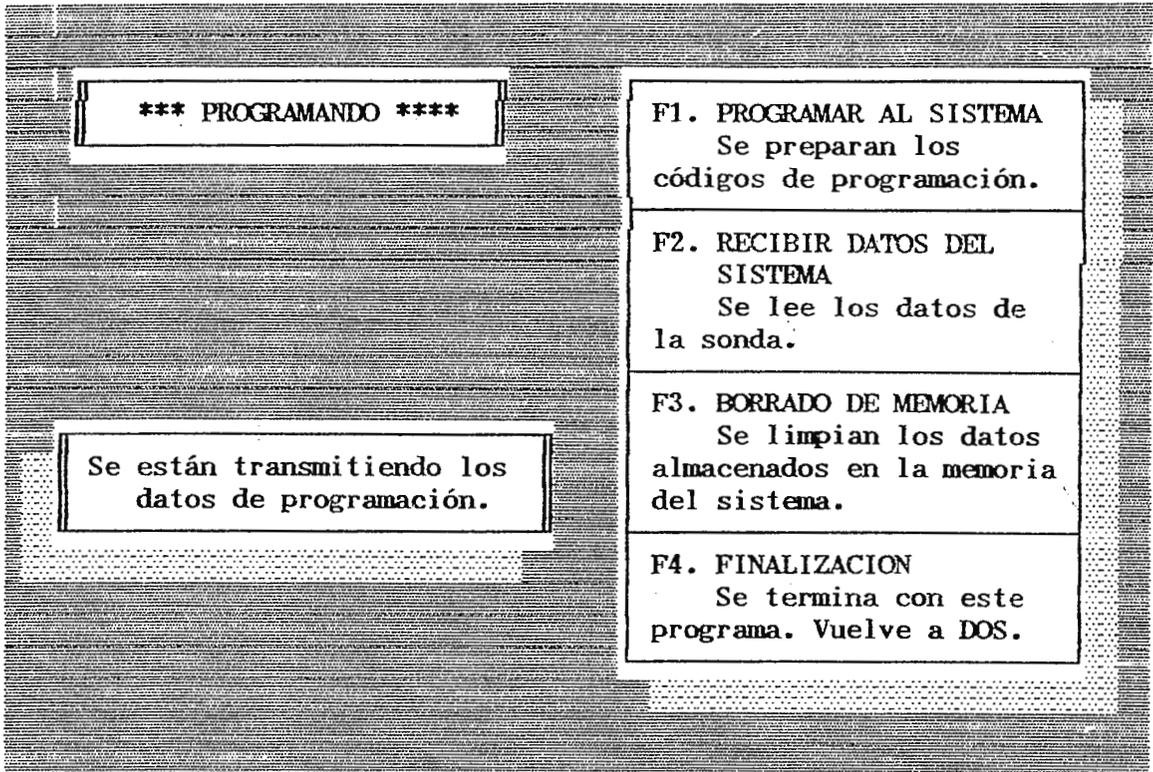


Fig.2.5(j).

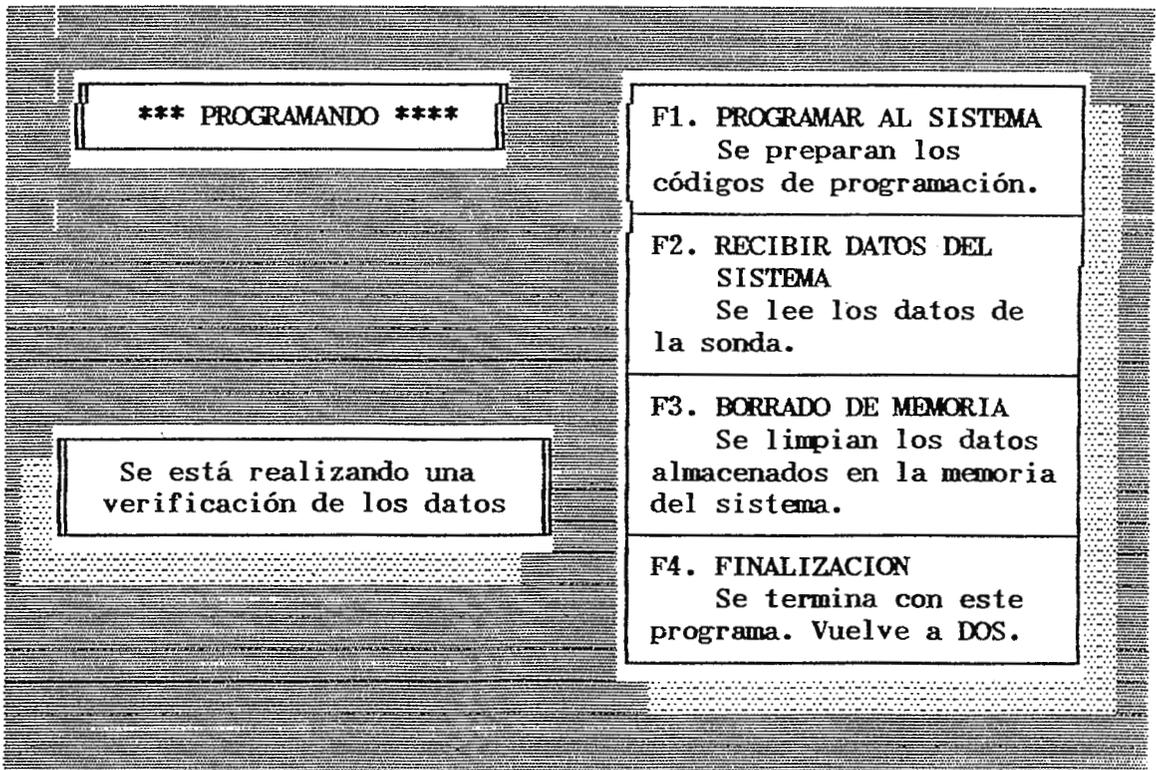


Fig.2.5(k).

Cuando el programa ha realizado la transmisión de los datos, éste además ejecuta otra rutina automáticamente que se llama Verificación, que tiene como objetivo leer los datos que se acaban de enviar a la sonda. El mensaje como se muestra en la Fig.2.5(k) se encuentra parpadeando hasta que finalice de recibir los datos de programación de la sonda. El operador no oprimirá alguna tecla hasta el momento, cualquier tecla que oprima será almacenada en el buffer y ejecutada después de finalizar las rutinas anteriores. Al finalizar la rutina de verificación, el operador verá en pantalla un resumen de los datos programados y los datos de la sonda (Veáse Fig.2.5(1)).

*** PROGRAMANDO ****	
Datos programados	SISTEMA
Variable a medir:	Se lee los datos de
TEMPERATURA PRESION.	Variable a medir:
Tiempo antes de empezar :10 min	TEMPERATURA PRESION
Tiempo entre adquisición: 5 min	Tiempo antes de empezar :10min
El sistema tomará datos :SIMPLE	Tiempo entre adquisición:5 min
Total de muestras a tomar: 5	El sistema tomará datos :SIMPLE
Total de estaciones son : 35	Total de muestras a tomar: 5
Total de datos a tomar :350	Total de estaciones son : 35
	Total de datos a tomar : 350
ESTA SATISFECHO CON LOS DATOS RECIBIDOS S/N ?	

Fig.2.5(1).

El resumen de la Fig.2.5(1) le indicará al operador que realmente los datos que programó se encuentran almacenados en el sistema(Hardware). Además si existe alguna disparidad de los datos programados y los datos en la sonda se podrá transmitir nuevamente los datos de programación para confirmar la igualdad en ellos, para ello en la parte inferior del recuadro se muestra el mensaje si está satisfecho con los datos recibidos. Al oprimir la tecla N se visualizará en pantalla la Fig.2.5(m) y con ello se tendrá que esperar hasta que termine la rutina de retransmisión.

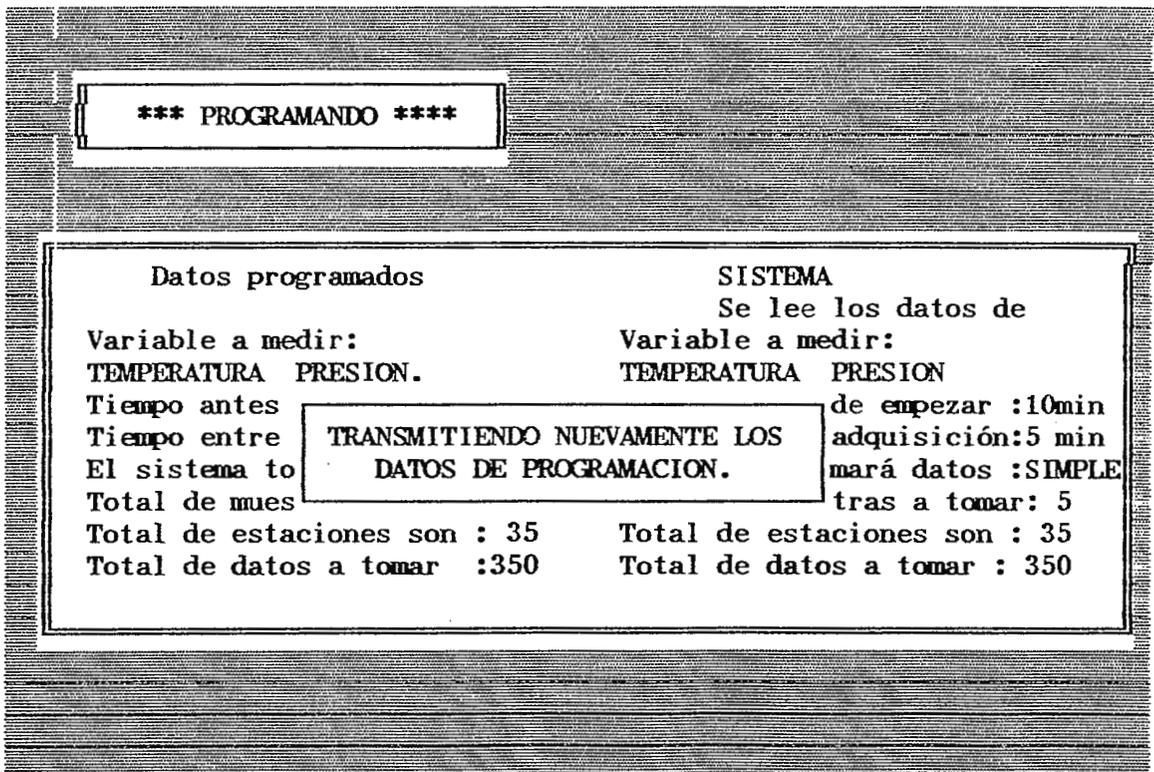


Fig.2.5(m).

Si el operador está satisfecho con los datos recibidos, al oprimir la tecla S retornará al menú principal de la Fig.2.2. En este momento se recomienda que el operador borre la memoria del sistema que está destinada para datos, para ello se oprime F3 y verá en pantalla el mensaje como se muestra en la Fig.2.5(n).

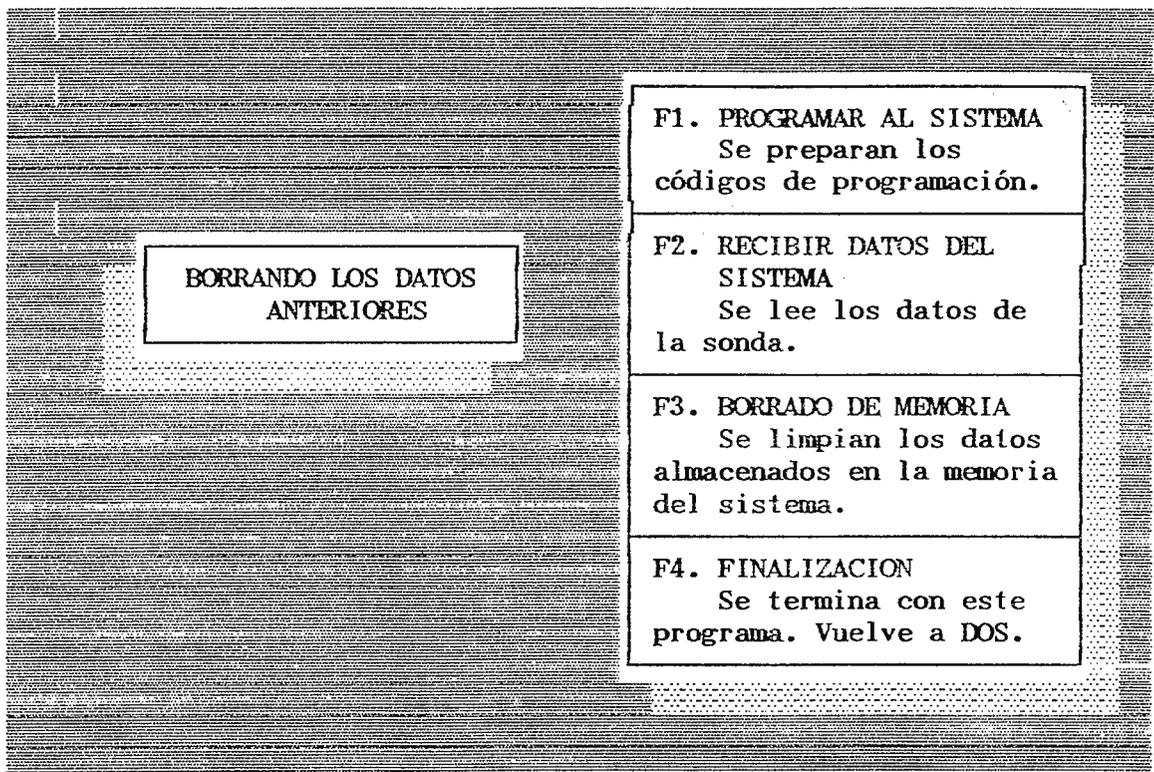


Fig.2.5(n).

- Fig.2.5 (a) Inicialización de la programación.
 (b) Selección de la variable a medir .
 (c) y (d) Ventana de programación de tiempo.
 (e) Selección de la toma de datos .
 (f) Preparar la cantidad de muestras .
 (g) Cálculo de la cantidad de datos a tomar.
 (h) Selección del número de estaciones.
 (i) Resumen de las variables de programación.
 (j) Transmisión de datos de programación.
 (k) Verificación .
 (l) Resumen de los datos programados y datos sonda.
 (m) Ventana de retransmisión de datos de programación.
 (n) Selección de borrado de datos del sistema (F3).

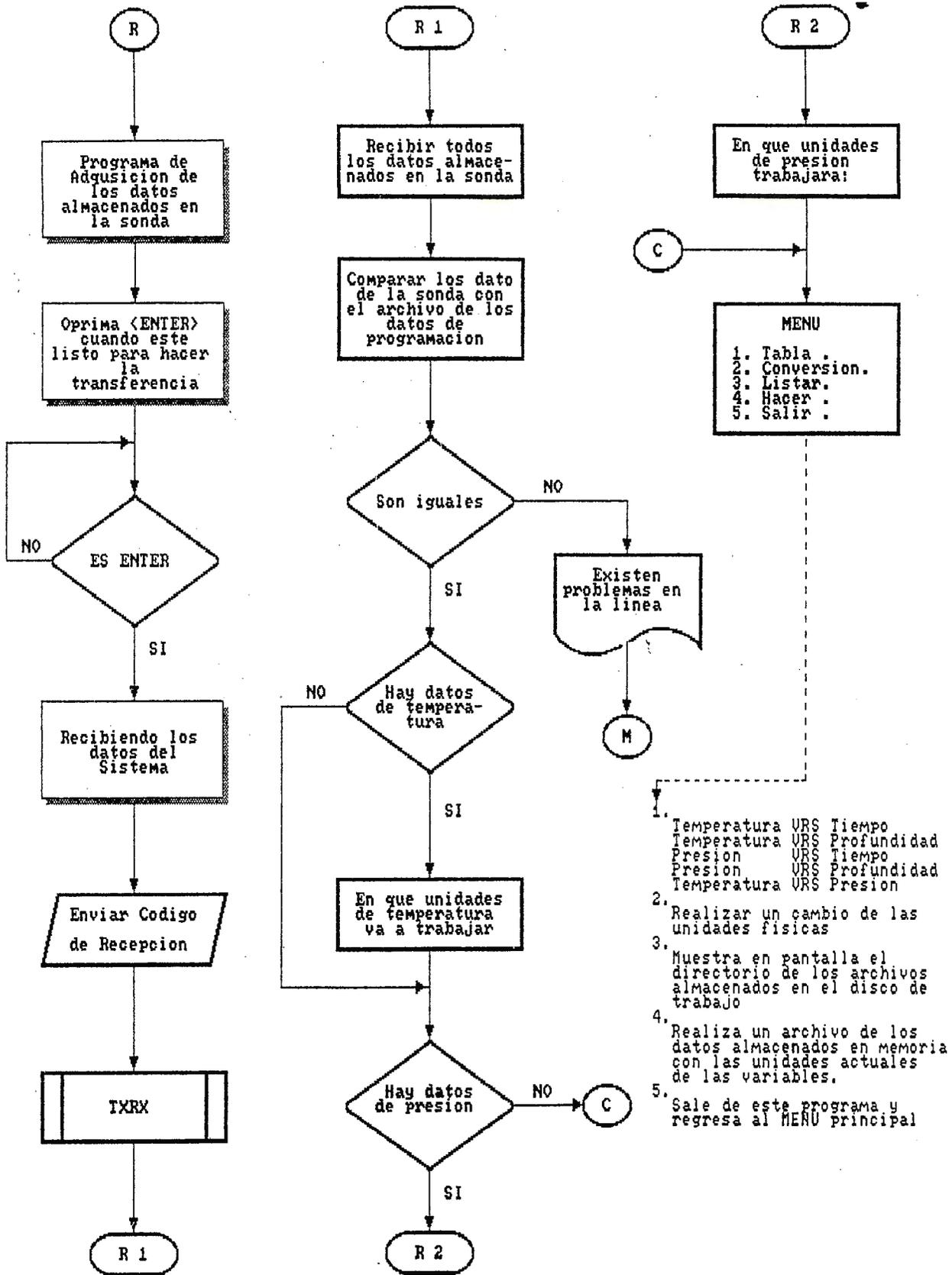


Fig.2.8 Menu de Lectura de datos y presentación en pantalla.

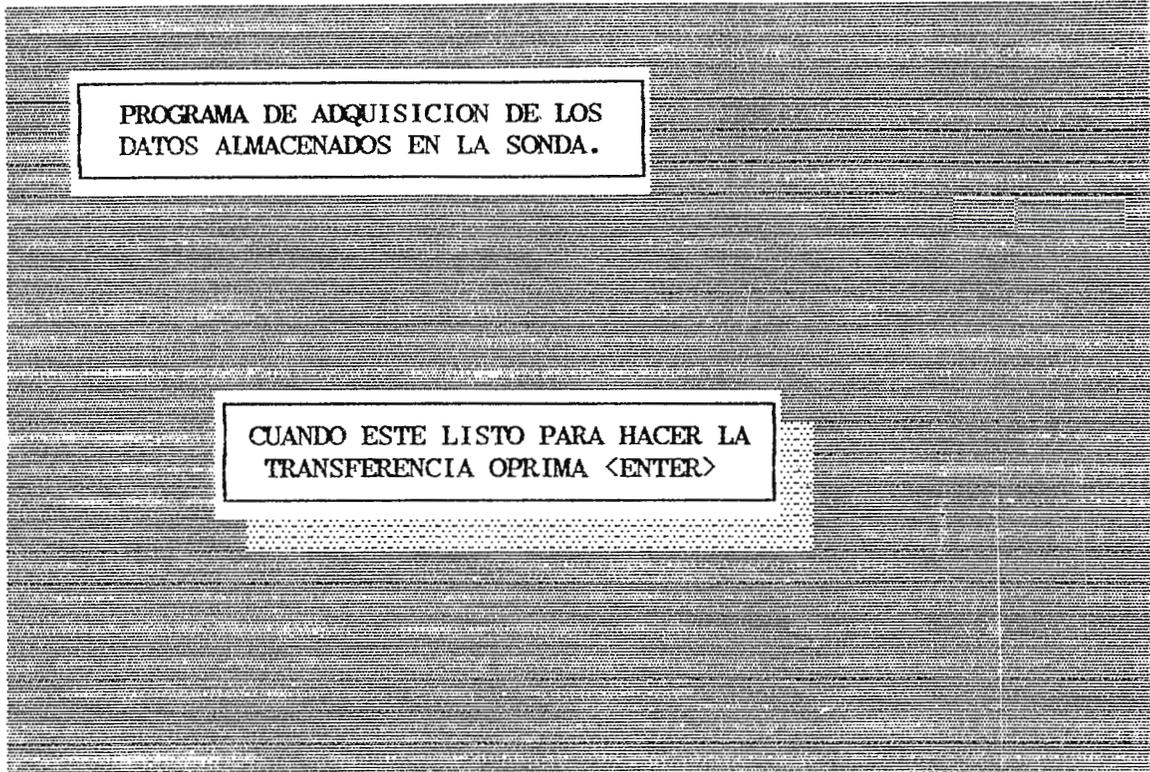


Fig.2.9(a).

Cuando el usuario oprime ENTER, se inicializa la rutina de petición de datos en la sonda, para ello se presenta en pantalla el mensaje de la Fig.2.9(b), mensaje que estará parpadeando mientras realiza la tarea de leer cada uno de los datos del sistema de adquisición de datos.

El programa se encarga primero de recibir los datos con que se programó al sistema, éstos datos los comparará con un archivo que fue realizado en la sección de programación para verificar exactamente que la sonda ha pasado por su proceso de recolección de información. Si existe diferencias entre los datos recibidos y los datos que fueron almacenados anteriormente en un archivo, el programa le presentará al usuario cualquiera de dos alternativas: El sistema no ha sido programado, por tanto los datos que se reciben son simplemente basura del sistema y el usuario retornará al menú principal.

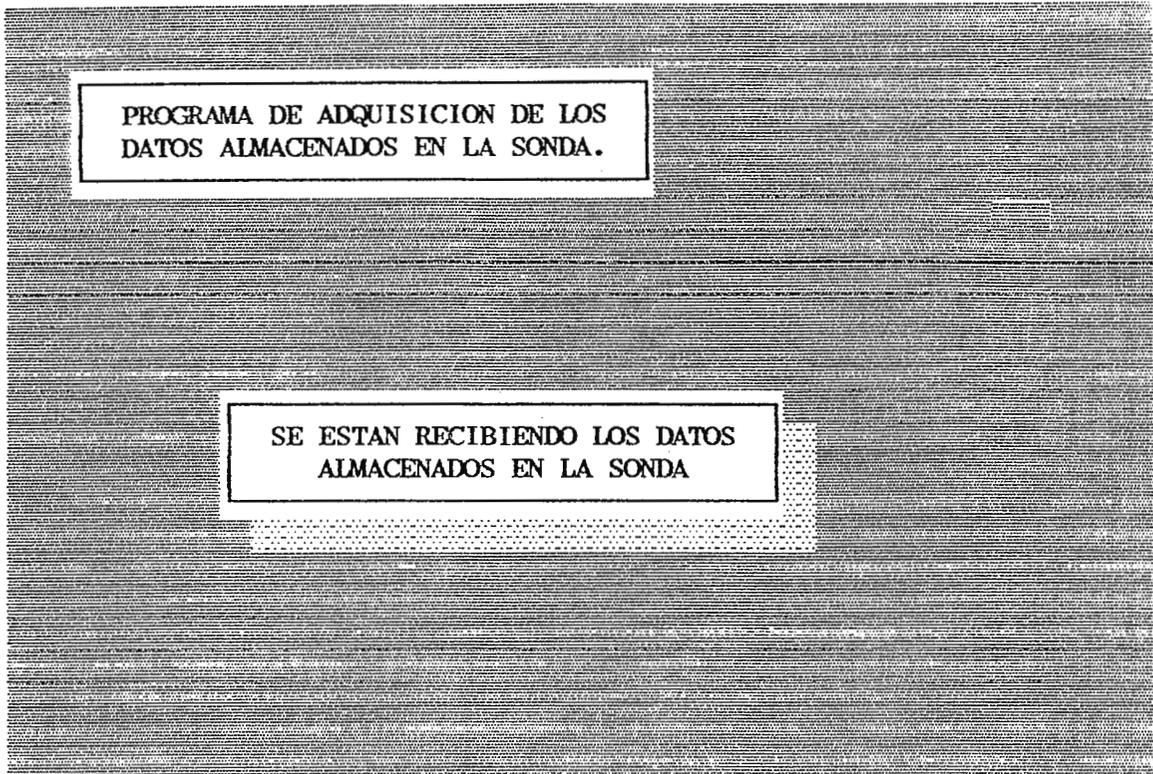


Fig.2.9(b).

Cuando la rutina de recepción de datos haya completado su tarea, comienza a monitorear cada uno de los datos de programación para establecerle al usuario cuales son las unidades físicas con que se va a trabajar, de ello se presenta en la Fig.2.9(c) y (d) las unidades con que se pueden trabajar ya sea la presión o la temperatura. Estas ventanas no siempre se presentarán al operador, sino que serán válidas si solamente han sido programadas previamente, esto es, que si solo se programó para tomar datos de temperatura la Fig.2.9(d) no se visualizará en pantalla.

En la fig.2.9(c) el programa espera reconocer con que variable física de temperatura el usuario desea trabajar. El programa solamente esperará algunos de los números 1 ó 2 para continuar con su ejecución y cualquier otra selección del parte del usuario será invalidada.

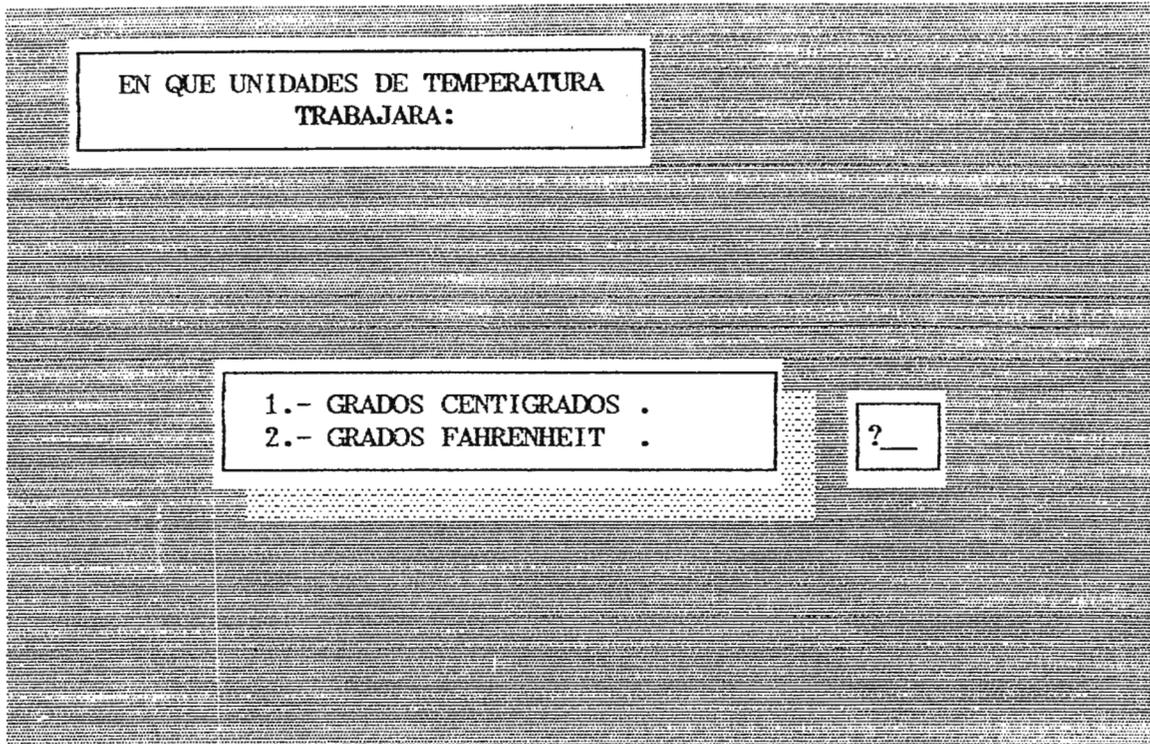


Fig.2.9(c).

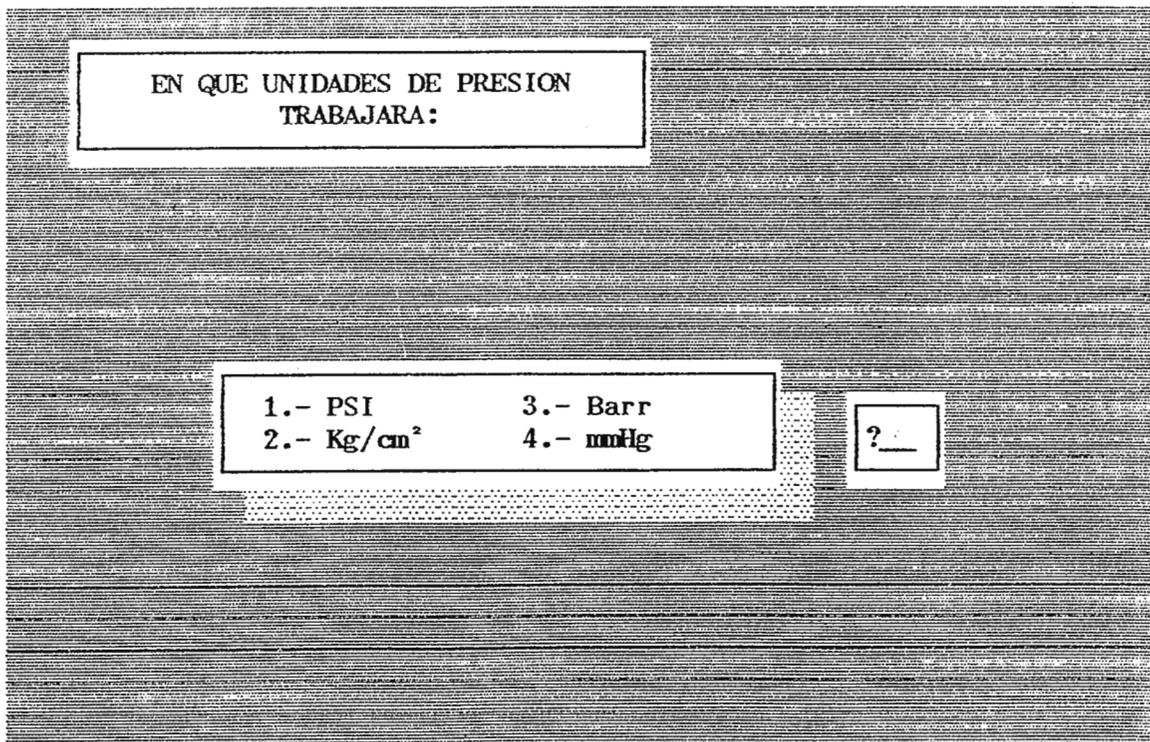


Fig.2.9(d).

Cuando se haya finalizado de preparar las unidades físicas con que va a trabajar las variables, corresponde el programa a presentar el menú donde se mostrará en pantalla una serie de opciones que el usuario pueda ejecutar para interpretar los resultados. En la Fig.2.9(e) se presenta un menú que contiene las siguientes opciones:

Tabla: tiene como objetivo realizar una tabla de datos en función del tiempo o la profundidad.

Conversión: realiza un cambio en las unidades en que se está trabajando.

Listado: presenta en pantalla cada uno de los archivos de trabajo almacenados en el disco. (Toma por omisión la unidad A)

Hacer: graba en el disco la información residente en memoria, asignándole un nombre para que después pueda ser llamado en otra oportunidad.

Salir: concluye con la ejecución del programa de lectura de datos.

◀====Tabla====Conversión====Listado====Hacer====Salir====▶

Como va a realizar la tabulación:		
Temperatura	VRS	Tiempo
Temperatura	VRS	Profundidad
Presión	VRS	Tiempo
Presión	VRS	Profundidad
Temperatura	VRS	Presión

Fig.2.9(ε).

◀ Tabla Conversion Listado Hacer Salir ▶

Como va a realizar la tabulación:			Hora a que se tomo el dato	Temperatura °C
Temperatura	VRS	Tiempo	9:45:35	89.23
Temperatura	VRS	Profundidad	:45:38	90.55
Presión	VRS	Tiempo	:45:41	95.67
Presión	VRS	Profundidad	:45:44	98.05
Temperatura	VRS	Presión	:45:47	99.45
			9:50:28	104.78
			:50:31	106.21
			:50:34	106.78
			:50:37	108.44
			:50:40	109.38
			9:55:04	111.09
			:55:07	112.47
			:55:10	113.05
			:55:13	115.86
			:55:16	116.00

PRESIONE ALGUNA TECLA PARA CONTINUAR

Fig.2.9(f).

- Fig.2.9 (a) Inicialización de lectura de los datos en la sonda.
 (b) Recibiendo los datos de la sonda.
 (c) Unidades para temperatura.
 (d) Unidades para presión.
 (e) Menú del programa de recepción .
 (f) Tabla de Temperatura VRS Tiempo.

Inicialmente el programa se ubica en TABLA y el usuario para acceder a alguna de las opciones debe oprimir la tecla que se mira más repintada en la pantalla. Como ejemplo se ilustra en la Fig.2.9(f) cuando el operador oprime la tecla T, correspondiente a una tabla de Temperatura VRS Tiempo. En este momento se visualizan cada uno de los datos que la sonda tomó mientras estaba en el pozo y a la hora que realizó las muestras. Si la cantidad de datos es tal que no se pueden mostrar todas de una vez en el cuadro de la Fig.2.9(f) se muestra en la parte inferior del recuadro, el mensaje: PRESIONE ALGUNA TECLA PARA CONTINUAR. Cuando se haya finalizado de mostrar los datos correspondientes, el usuario podrá

seleccionar alguna otra opción que se presenta en TABLA o las restantes funciones del menú. Puede posicionarse en cualquier sección de la pantalla para el menú con las teclas de cursor → ← ↑ ↓.

2.4 Direccionamiento del puerto serie de la PC. XXXXXXXXXXXXXXXXXXXX

Todas las XT/PC y AT/PS de hoy día, cuentan en su configuración de hardware con un IC de 40 pines (8250) conocido como Universal Asynchronous Receiver/Transmitter (UART), encargado de la transferencia de datos paralelos del microprocesador a datos serie con el mundo exterior.

Este puerto debe ser programado por medio de ciertos códigos a unas direcciones específicas; para el caso, se muestra a continuación la inicialización del puerto serie (COM1) en Turbo Basic:

```
10 OUT &H3FB,&H80
20 OUT &H3F9,0
30 OUT &H3F8,&H60
40 OUT &H3FB,&H1F
50 OUT &H3F9,0
60 OUT &H3FB,3
```

Para que la UART sea inicializado, 4 pasos son importantes:

- a. Establecer el bit rate.
- b. Definir el formato de la palabra (especificar paridad, número de stops bits y longitud del carácter).
- c. Enmascarar todas las interrupciones de comunicación.
- d. Habilitar las líneas DTR y RTS.

De acuerdo al ejemplo, en la línea 10 el octavo bit del registro 3FB es ajustado a 1 para preparar el bit rate de la comunicación, de las líneas 20 y 30 se carga en los registros 3F9 y 3F8 el número divisor para

la ecuación: $\text{bit rate} = 1,843,200 \div (16 * \text{divisor}) \text{ bps}$. El divisor lo conforman dos bytes en donde para 3F9 se localiza el MSB y en 3F8 el LSB, para el ejemplo el número es 0060H; luego el bit rate = 1200 bps, que es una de las velocidades estándar en comunicaciones. A continuación en la línea 40 se prepara el formato de la palabra (registro 3FB) con 8 bits de datos, 2 stops bits y paridad par. En la línea 50 las interrupciones son habilitadas y finalmente en la línea 60 las señales DTR y RTS son activas, preparando así los buffers para Transmitir y/o Recibir, de esta manera, con todas las instrucciones anteriores ya se ha preparado el puerto serie RS-232 de la PC para la comunicación con el exterior. Ya sea, que se quiera transmitir/recibir, es el registro 3F8 que se logra esto, y las instrucciones que lo permiten son:

OUT &H3FC,2	PROGRAMA COMO TRANSMISOR
OUT &H3F8,DATO	ENVIA UN DATO CONOCIDO
OUT &H3FC,1	PROGRAMA COMO RECEPTOR
Z=INP(&H3F8)	CARGA A Z EL DATO RECIBIDO

Tabla 1.1 Problemas de Transmision y Soluciones.

PROBLEMA	EXPLICACION	SOLUCION
I. El computador se quedaba bloqueado.	Cuando se realizaba las Pruebas de Transmision y Recepcion de mas de 12 datos	Se dimensionaron vectores de acuerdo a la cantidad de datos que se utilizan.
II. Se perdia la informacion.	Al transmitir o recibir algunos datos no eran los esperados, algunos presentaban errores como: Paridad, estructura y dato que le caia encima a otro dato, este ultimo conocido como Overrun.	<p>Para resolver el caso que un dato le caia a otro, que era problema de sincronismo, se utilizaron las banderas del puerto para asegurarse cuando el buffer estaba listo para transmitir o recibir un nuevo dato.</p> <p>Para mejorar las condiciones de transmision y recepcion y tener datos correctos, se transmite el mismo dato 10 veces y se recibe tambien 10 veces y luego es pasado todos los datos a una rutina la se encargaria de buscar en grupos de diez, cual es el dato que mas se repite, y ese dato sera el valido.</p>

CAPITULO 3

HARDWARE DEL SISTEMA.

3.1 Diseño del circuito del prototipo experimental ████████████████████

En esta sección se describirá en detalle lo que constituye el sistema de adquisición de datos en su versión de prototipo experimental.

Este se divide en dos partes principales:

- A. Sistema Digital y
- B. Sistema Analógico.

El Sistema Digital es el que comprende mayor cantidad de elementos y es la que realiza las funciones de control del sistema: procesamiento de datos, almacenamiento de datos y comunicación con la computadora. Esta parte se divide en las siguientes secciones:

- a) Sección de control.
- b) Sección de memorias y decodificador de direcciones.
- c) Sección de interfaces.
- d) Sección de control externo. Circuito para pruebas de control.
- e) Sección de visualización. Circuito para pruebas de visualización.

El Sistema Analógico tiene como función principal el acondicionamiento de las señales de salida de los transductores a fin de que dichas señales puedan ser procesadas por el sistema en su parte digital. Esta parte a su vez esta formada por las secciones siguientes:

- a) Acondicionamiento de la señal de temperatura.
- b) Acondicionamiento de la señal de presión.
- c) Voltajes de referencia.
- d) Nivel de batería.
- e) Amplificador común.

3.1.1 Sistema Digital del sistema de adquisición de datos.

3.1.1-1 Sección de control (Fig.3.10)

En esta primera sección se encuentra el cerebro del sistema: el microprocesador 8088(IC2), el cual cuenta como cualquier microprocesador con un bus de direcciones, un bus de datos, señales de control, líneas para interrupciones y líneas de suministro para su funcionamiento. Entre las características más sobresalientes de este integrado están su velocidad de procesamiento (5MHz) y su gran capacidad (1Mbyte) de memoria; la primera de ellas ha hecho necesario que el tren de pulsos o señal de reloj que marca el paso de funcionamiento del micro, y que debe ser aplicado desde el exterior deba tener características especiales tales como: un ciclo de trabajo (Duty cycle) entre 33 y 50%, y tiempos de transición menores o iguales a 10ns; es por ello que el fabricante produce además un circuito integrado adecuado a ese propósito, siendo éste el 8284 (IC1), el cual además le proporciona al microprocesador las señales de RESET y READY. Para producir la señal de reloj el 8284 se vale de un cristal de resonancia serie de 14.31818 MHz, frecuencia que dividida entre 3 internamente en el 8284 genera la señal deseada. Con respecto a la señal de reset, lo que IC1 hace es acondicionar y sincronizar con la señal de reloj, la señal que recibe por la entrada \overline{RES} , a ella está conectada una red RC cuya función es producir un retardo que haga que el pulso de reset que es aplicada al micro tenga una duración mayor a la mínima necesaria (50 μ s) para que éste lo interprete correctamente. La red de R4 y C4 permite que al encender el sistema éste automáticamente se inicialice a través de un estado de reset y además a través de S1 para efectos de prueba se le proporciona al sistema un RESET en funcionamiento.

En lo que se refiere a la señal READY, esta ha sido usada para

proporcionarle al sistema una modalidad de funcionamiento paso a paso desactivando el microprocesador luego de que ha ejecutado una instrucción. La explicación del circuito paso a paso es la siguiente:

Las compuertas IC12b e IC12d forman un eliminador de rebote para el pulsador S3, a fin de asegurar que no hayan transiciones falsas en el circuito. Cuando el interruptor S2 está en la posición RUN la línea RDY2 tiene un "alto" y el circuito trabaja normalmente a la velocidad de reloj del sistema. La línea RDY2 pertenece al IC8284 y ésta se refleja en la salida READY que se conecta al 8088 y que cuando tiene un nivel "bajo" éste deja de operar en espera a que se restablezca el nivel de dicha línea. Si el interruptor S2 está en la posición STEP, la línea RDY tiene normalmente un nivel "bajo" pero cuando se pulsa S3 ésta pasa a "alto" habilitando al 8088, pero aprovechando que durante cada ciclo de lectura o escritura la señal ALE tiene que habilitarse, se usa esta señal para restablecer al biestable IC13 y con ello a la línea RDY2, obteniéndose el efecto de que el sistema realiza un ciclo de lectura/escritura cada vez que se acciona S3.

Los circuitos integrados IC3, IC4 e IC5 son los manejadores de buses, de sus salidas se obtienen el bus de datos y el bus de direcciones del sistema. Los interruptores manejadores de buses se hacen necesarios porque el 8088 utiliza varios de sus pines para 2 funciones, por ejemplo: los pines del 9 al 16 proporcionan el bus de datos y también los 8 bits menos significativos del bus de direcciones. Para poder capturar los datos correctos se ocupan las señales ALE, \overline{DEN} y DT/ \overline{R} ; la primera se activa (en alto) cuando en los pines están presentes direcciones (ALE=Habilitador de enganche de direcciones) es por esto que se conectan al habilitador LE de los 74LS373 (IC3 e IC4). \overline{DEN} se activa (bajo) cuando el

bus es ocupado por datos ($\overline{\text{DEN}}$ =Habilitador de datos) y se conecta al habilitador de IC5. $\text{DT}/\overline{\text{R}}$ sirve para determinar la dirección de transferencia de los datos ($\text{DT}/\overline{\text{R}}$ =Transmisión/Recepción de datos), si está en "alto" el 8088 envía datos y si está en "bajo" espera recibirlos, por esto, esta señal se conecta a la entrada de dirección de IC5 que es un manejador bidireccional (Transceiver).

La señal PE que se conecta a todos los habilitadores de salida de los manejadores de bus normalmente está en "bajo" habilitándolos, pero se coloca en "alto" cuando se va a usar el control externo (Fig.3.13) y por ello las líneas provenientes del 8088 deben quedar aisladas y los manejadores de buses deben presentar alta impedancia.

Otras señales del 8088 utilizadas son: $\overline{\text{WR}}$, $\overline{\text{RD}}$, $\text{IO}/\overline{\text{M}}$ y $\overline{\text{TEST}}$. $\overline{\text{WR}}$ se activa (en bajo) cuando el 8088 escribe datos; $\overline{\text{RD}}$ se activa (en bajo) cuando el micro lee los datos; $\text{IO}/\overline{\text{M}}$ es utilizada para seleccionar la comunicación con periféricos (alto) o memorias (bajo). $\overline{\text{TEST}}$ es una entrada que cuando se activa por un nivel "alto" y en combinación con la instrucción WAIT (por software) deja al microprocesador en "estado de espera" hasta que dicho pin se active. Esto sirve para adaptar la velocidad de respuesta del convertidor analógico-digital al sistema del micro.

Los circuitos integrados IC23 e IC24 son contadores usados como divisores de frecuencia, los cuales toman la señal de salida PCLK del 8284 y de ella se obtienen :

$\text{CLK2} = \text{PCLK} = \text{CLK}/2 = 2.4\text{MHz}$ y $\text{CLK3} = \text{PCLK}/2000 = 1200\text{Hz}$. Ambas señales son necesarias para el funcionamiento del integrado usado para la comunicación serie via RS-232C (USART) con el computador de superficie.

3.1.1-2 Sección de memorias y decodificador de direcciones (Fig 3.11)

Esta sección comprende los circuitos integrados IC6 e IC25 que constituyen las memorias del sistema, el primero es la memoria de solo lectura (ROM) cuya capacidad es de 2K para un bus de datos de 8 bits. En ella esta contenido el programa que rige el funcionamiento del sistema el cual ocupa aproximadamente 800 bytes de esta memoria es decir menos del 50% de la misma. El segundo es la memoria de acceso aleatorio (RAM) de donde se pueden leer y también escribir datos, su capacidad es de 8K para un bus de datos de 8 bits.

El resto de compuertas lógicas mostradas en la Fig.3.11 forman el llamado decodificador de direcciones que es de donde se generan las señales de habilitación para las memorias y periféricos, esto lo hace respondiendo a la lógica que se expondrá a continuación:

El sistema es bastante sencillo y de la misma manera es su mapa de memorias y de periféricos los cuales son:

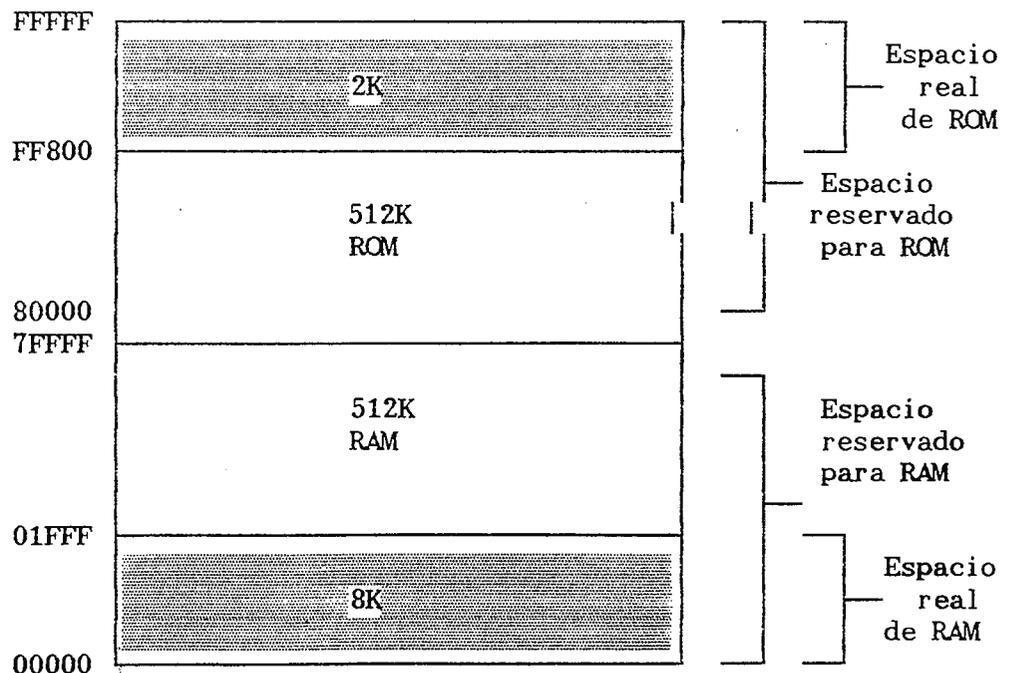


Fig.3.1 Mapa de memorias del sistema de adquisición de datos.

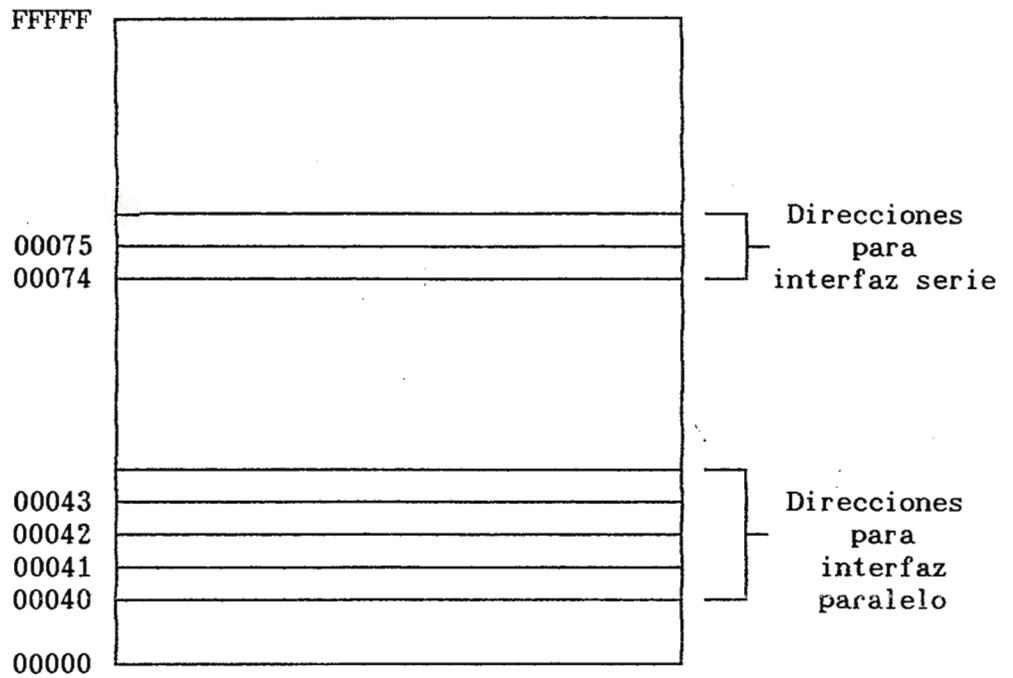


Fig.3.2 Mapa de periféricos del sistema de adquisición de datos.

Acerca del decodificador de direcciones, las señales de salida de éste pueden ser divididos en dos: los que habilitan a las memorias y las que habilitan a los periféricos; para que se pueda tener más claro el funcionamiento del mismo a continuación se presenta las tablas de verdad y ecuaciones lógicas de esta parte del circuito.

Tabla Nº 2.2.1 Tabla de verdad del decodificador de direcciones.
Parte de memorias. La línea de control $IO/\bar{M} = 0$

Entradas			Salidas			Comentario
A_{19}	\overline{RD}	\overline{WR}	\overline{RAMSEL}	\overline{RAMWR}	$\overline{ROMSELRD}$	
0	X	X	0	1	1	Se lee de la RAM
0	X	0	0	0	1	Se escribe en ROM
1	0	X	1	1	0	Se lee de la ROM

$$\begin{aligned}\overline{\text{RAMSEL}} &= \text{IO}/\overline{\text{M}} + \text{A19} \\ \overline{\text{RAMWR}} &= \overline{\text{WR}} + \text{A19} \\ \overline{\text{ROMSELRD}} &= \text{IO}/\overline{\text{M}} + \overline{\text{A19}} + \overline{\text{RD}}\end{aligned}$$

NOTA : Las líneas $\overline{\text{RAMSEL}}$ y $\overline{\text{RAMWR}}$ físicamente no existen en la Fig.3.11 debido a que la estructura interna de la RAM proporciona una función lógica con la función requerida.

Tabla No2.2.1 Tabla de verdad del decodificador de direcciones Parte de periféricos. La línea de control $\text{IO}/\overline{\text{M}}=1$

A_2	Entradas		Salidas				Comentario
	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{IORD}}$	$\overline{\text{IOWR}}$	$\overline{\text{CSPPI}}$	$\overline{\text{CSUSART}}$	
0	0	1	0	1	0	1	Se lee del PPI
0	1	0	1	0	0	1	Se escribe en PPI
1	0	1	0	1	1	0	Se lee de UASRT
1	1	0	1	0	1	0	Se escribe USART

$$\begin{aligned}\overline{\text{IORD}} &= \overline{\text{IO}/\overline{\text{M}}} + \overline{\text{RD}} \\ \overline{\text{IOWR}} &= \overline{\text{IO}/\overline{\text{M}}} + \overline{\text{WR}} \\ \overline{\text{CSPPI}} &= \text{A2} \\ \overline{\text{CSUSART}} &= \text{A2}\end{aligned}$$

Una última observación del decodificador de direcciones es la generación de la señal utilizada para dar inicio a un ciclo de conversión en el convertidor analógico-digital, ésta ha sido llamada CONV y su función lógica es la siguiente:

$$\text{CONV} = \overline{\text{A0}} + \overline{\text{A1}} + \overline{\text{A2}} + \overline{\text{IORD}}$$

En donde la línea CONV presentará un pulso en alto cuando A0 , A1 , A2 e $\overline{\text{IORD}}$ estén en "bajo", es decir, cuando se lee el puerto A del PPI.

3.1.1-3 Sección de interfaces (Fig.3.14)

La sección de interfaces está formada por los dispositivos necesarios para que el sistema digital se comunique con el mundo exterior. El 8255 (IC27) es un interfaz programable de periféricos (PPI) que puede comunicar al sistema (acoplar su bus de datos) con 3 puertos de entrada/

salida, en este sistema el puerto A se ocupa como salida para manejar las entradas de selección de los multiplexores MUX1 (IC34) y MUX2 (IC41) de la sección analógica del sistema, el puerto B lee los 8 bits menos significativos del ADC y el puerto C los 4 bits más significativos del mismo. El PPI ocupa cuatro direcciones de periféricos del sistema, 3 son para el accesamiento de los puertos y la cuarta es para acceder el registro de control del integrado.

El puerto de comunicación serie IC8251A conocido como USART (IC26) al igual que el PPI necesita ser programado a través del bus de datos y para ello su entrada C/\bar{D} debe estar en "alto" (esta entrada se ha conectado a la línea de dirección A0). Por otra parte cuando C/\bar{D} esta en "bajo" se transfieren los datos del bus hacia el exterior, saliendo en forma serie por el pin TXD y se reciben datos del exterior hacia el bus de datos en forma serie por el pin RXD.

El 8251A contiene un registro interno donde se encuentran bits indicadores o banderas que reflejan el estado de la transmisión, ejemplo de ello: errores en la comunicación, activación de funciones, registros disponibles, etc. Este registro es accesado por el 8088 cuando lee y la entrada C/\bar{D} está en "alto".

Los circuitos integrados IC24 e IC25 son los adaptadores de niveles TTL del sistema a niveles RS-232 propios de la comunicación serie.

3.1.1-4 Convertidor Analogo/Digital

Es el dispositivo que finalmente trasladará el valor de la variable medida presente en forma de voltaje, a un código digital que pueda ser almacenado en una memoria por el sistema. El ADC usado en el prototipo del sistema de adquisición de datos es el modelo ADC80AG-12 de la empresa Burr-Brown, cuyas características son las siguientes:

- Tipo = aproximaciones sucesivas, lo que lo hace ser bastante rápido. Empleando un ciclo de reloj por bit y usando su propio reloj interno de 500KHz, proporciona para 12 bits un tiempo de conversión de 25 μ s.

- Exactitud = los errores comunes como offset, ganancia, linealidad y sensibilidad a ruido de la fuente no exceden a $\pm\frac{1}{2}$ LSB y además cuenta con pines de ajuste de offset y ganancia que reducen aún más el error.

- Rango de señal de entrada = en este sentido el ADC es bastante versátil ya que por medio de sencillas conexiones externas que el fabricante indica es posible trabajar con cualquiera de los siguientes rangos de voltaje de entrada:

una polaridad = 0 a 5V y 0 a 10V

doble polaridad = -5 a +5V, -2.5 a +2.5V y -10 a +10V

para el caso el sistema usa la modalidad de 0 a 5V.

- Voltaje de alimentación = en este caso el ADC80AG-12 resulta desventajoso pues necesita de una alimentación de doble polaridad de ± 15 V y además una alimentación lógica de +5V. Esta característica lo hace totalmente inadecuado para sistemas portátiles como el que se pretende implementar y se ha usado solo para fines de pruebas en este prototipo.

- Señales de control = la señal CONV se usa para iniciar el ciclo de conversión del ADC y consiste en un pulso en "alto" (nivel TTL) como ancho mínimo de 100ns y máximo de 2 μ s. Asimismo el ADC proporciona la señal STATUS, que se mantiene en alto mientras el dispositivo se encuentra realizando el proceso de conversión.

- Salida digital = es de 12 bits con niveles TTL y ocupa el código complementario (lógica negativa), por el hecho de contar con 12 bits de salida digital, el sistema se vuelve bastante preciso ($1/2^{12} = .0244\%$ de fondo de escala) lo cual es muy importante para cualquier sistema, pero

sobre todo para este caso, donde se ha trabajado con un transductor de respuesta fuertemente no lineal y que para no complicar el circuito de entrada ha debido trabajarse en un solo rango (o escala) para cada variable y se ha optado por hacer linealización por software.

En la fig.3.14, el ADC (IC28) salen las conexiones de las señales de control: a la entrada CONV le llega precisamente la señal CONV ya definida en la sección anterior y de la salida STATUS se lleva una línea hacia la entrada $\overline{\text{TEST}}$ del 8088, obteniéndose de esta manera que el micro se detenga mientras se realice la conversión y se reactive cuando ésta termine. Los arreglos de resistencias que están conectados al ADC son los ajustes del offset y la ganancia.

3.1.1-5 Circuito para pruebas. Control externo (Fig.3.11)

El objetivo de esta sección fue permitir a los investigadores el acceso a las memorias y periféricos del sistema, para ello las líneas de control y los buses de dirección y de datos provenientes del circuito de control (Fig.3.10) debieron aislarse del sistema y de esta manera a través de mini-interruptores y pulsadores se simularon los estados lógicos de cada una de las líneas de los buses del sistema de acuerdo a las necesidades del momento.

Para hacer el cambio de control del microprocesador a control externo se uso una línea activada por S28 y llamada $\overline{\text{PE}}$ (programación externa) cuya función consistió en deshabilitar los manejadores de buses IC23, IC24 e IC25 y las líneas de control $\text{IO}/\overline{\text{M}}$, $\overline{\text{RD}}$ y $\overline{\text{WR}}$ a través de IC10 e IC11 y al mismo tiempo habilitar los latches IC14, IC15 e IC16 para que a través de los mini-interruptores S4 a S27 se controla el continuo de los buses y líneas de control.

3.1.1-6 Circuito para pruebas. Visualización (Fig.3.12)

Para la visualización del contenido de los buses de datos y de direcciones se usaron decodificadores de hexadecimal a 7 segmentos y presentadores de este tipo. Aparte de lo anterior el circuito de prueba cuenta con otros circuitos visualizadores específicos, tal es el caso del detector de pulsos a base del IC74LS123 (IC31) en el cual un LED se enciende cuando el ADC realizaba una conversión, es decir, cuando el sistema toma un dato, esto lo indica el estado de la línea STATUS proveniente del ADC.

Otro caso es la visualización del canal del cual en un momento dado el sistema toma datos, para la versión del prototipo experimental las posibilidades son cuatro: temperatura, presión, voltajes de referencia y nivel de batería. Esto se realiza por medio de un decodificador de 3 a 8 (IC32) que maneja 4 leds y los activa dependiendo del estado de las líneas de selección del MUX1 (IC34) que es el selector de canales del sistema.

3.1.2 Parte analógica del sistema de adquisición de datos.

3.1.2-1 Transductores.

Termistores: son elementos formados de materiales semiconductores sólidos y se caracterizan por tener un coeficiente de resistividad altamente negativo. En una temperatura dada, un termistor actúa como una resistencia. Si la temperatura cambia debido a su disipación interna o a una variación de la temperatura ambiente, la resistencia cambia repetitivamente como función de la temperatura, generalmente en forma exponencial.

Los termistores son de bajo costo y de todos los transductores de temperatura comunes son los que poseen la más alta sensibilidad. A 25°C un

termistor típico puede tener un cambio de resistencia de $-4.5\%/^{\circ}\text{C}$. Son fabricados en un rango de valores que van desde decenas de ohmios hasta megaohmios a 25°C . La curva de respuesta es no lineal pero predecible. Los sensores disponibles en el comercio proporcionan salidas útiles desde -50 hasta $+300^{\circ}\text{C}$. Debido a su alta sensibilidad, ellos son frecuentemente la mejor alternativa en aparatos de control y de medida de alta resolución. Otra característica típica de los termistores es que generalmente son bastante pequeños y poseen una respuesta bastante rápida, en el orden de los segundos e incluso los microsegundos.

La curva de un termistor dado puede ser aceptablemente obtenida a través de la ecuación de Steinhart-Hart:

$$1/T = A + B \ln R + C(\ln)^3$$

donde : T = temperatura en grados Kelvin.

R = resistencia del termistor.

A, B, C = constantes de la ecuación.

Características del termistor empleado en el prototipo:

Rango de -20 a $+150^{\circ}\text{C}$; resistencia a $25^{\circ}\text{C} = 2252\Omega$

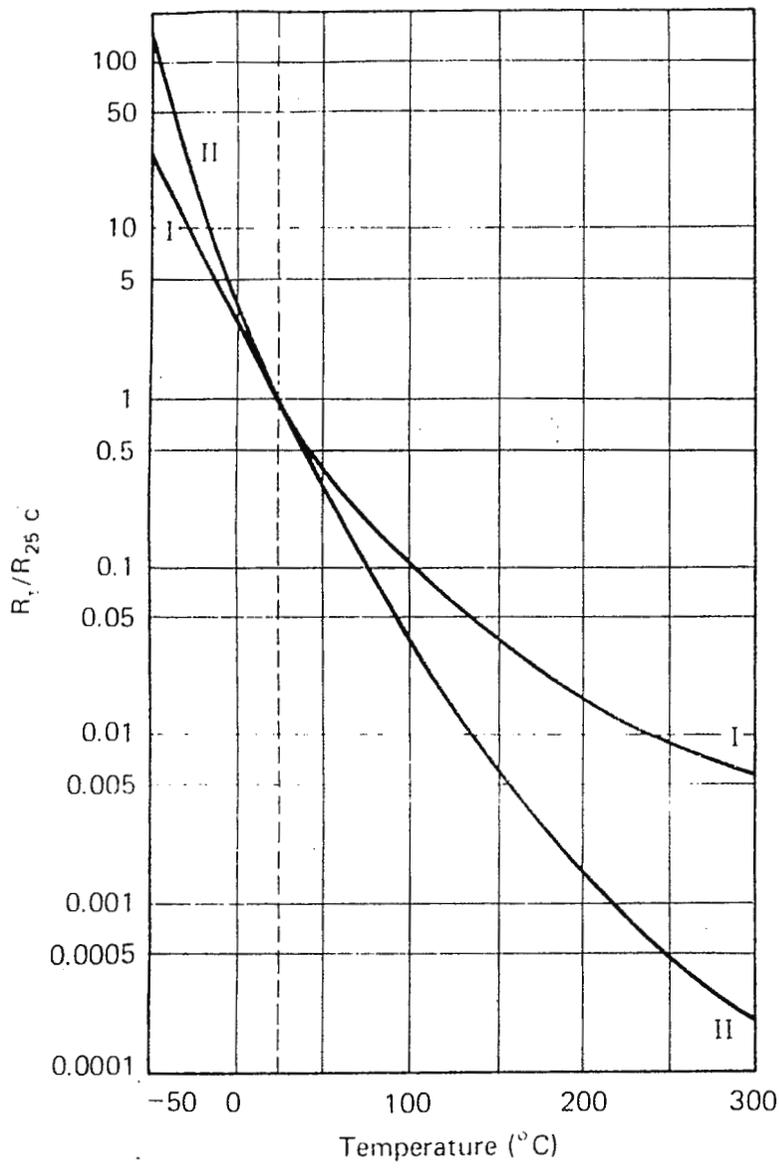
Transductor de presión : para la demostración de la medición de presión, en lugar de usar un transductor "formal" de presión (sea strain gage o semiconductor) se usará un medidor de flujo, es decir, que incluye un circuito de acondicionamiento de señal para producir una señal de corriente directamente para un presentador analógico y a continuación se detallan las características de este aparato:

Es un transmisor de presión diferencial, su aplicación principal es medir el flujo de vapor a través de una tubería, por medio de la caída de presión en un tramo de la misma, su rango de medición es de 0 a 50KPa (máximo). La señal de salida la proporciona en forma de corriente en un sistema de 2 hilos y en un rango de 4 a 20 mA y necesita una alimentación de 24VDC \pm 2.4V. El conexionado se muestra en la Fig.3.4.

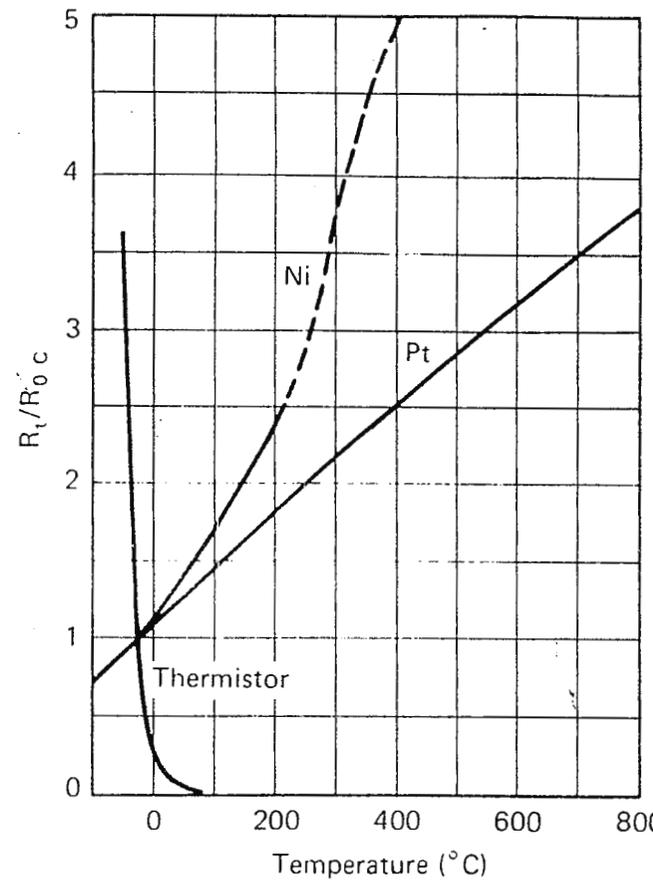
La resistencia de carga en serie a la línea para obtener salida de voltaje puede tener un valor entre 0 a 450 Ω . El transmisor además posee dos ajustes:

- Ajuste de cero. Se regula una salida de 4mA cuando la diferencia de presión es cero.
- Ajuste de salida máxima. Se regula una salida de 20mA cuando la diferencia de presión es el máximo deseado (este valor debe estar solamente en el rango de 5 a 50KPa)

El circuito básico de como se conecta el transductor de presión capacitiva es el puente de impedancias alimentado por AC (Fig.3.5). El transmisor como la mayoría de los diseños incorporan circuitos para demodular la señal de AC de salida en un voltaje DC y para proporcionar compensación por temperatura y ajustes de cero, de ganancia y linealidad. Algunos diseño emplean técnicas digitales para compensación y ajustes; y pueden proveer una salida en forma digital.



(a)



(b)

Fig.3.3 Curva característica de los Termistores. R vrs. T

Fig.3.4 Diagrama de conexiones para el transmisor y receptor.

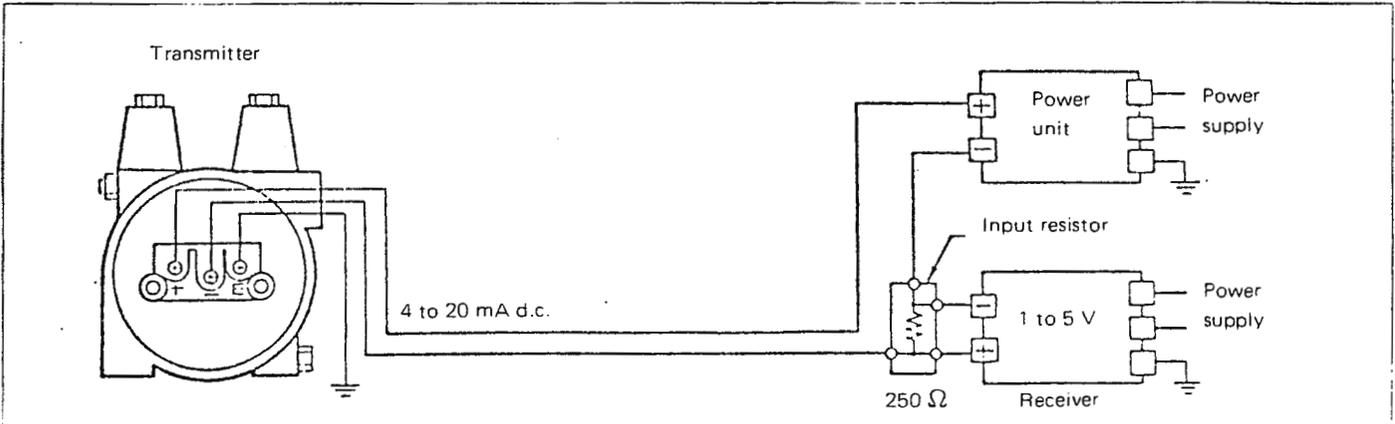


Fig. 11 Wiring connection for transmitter, receiver and power supply

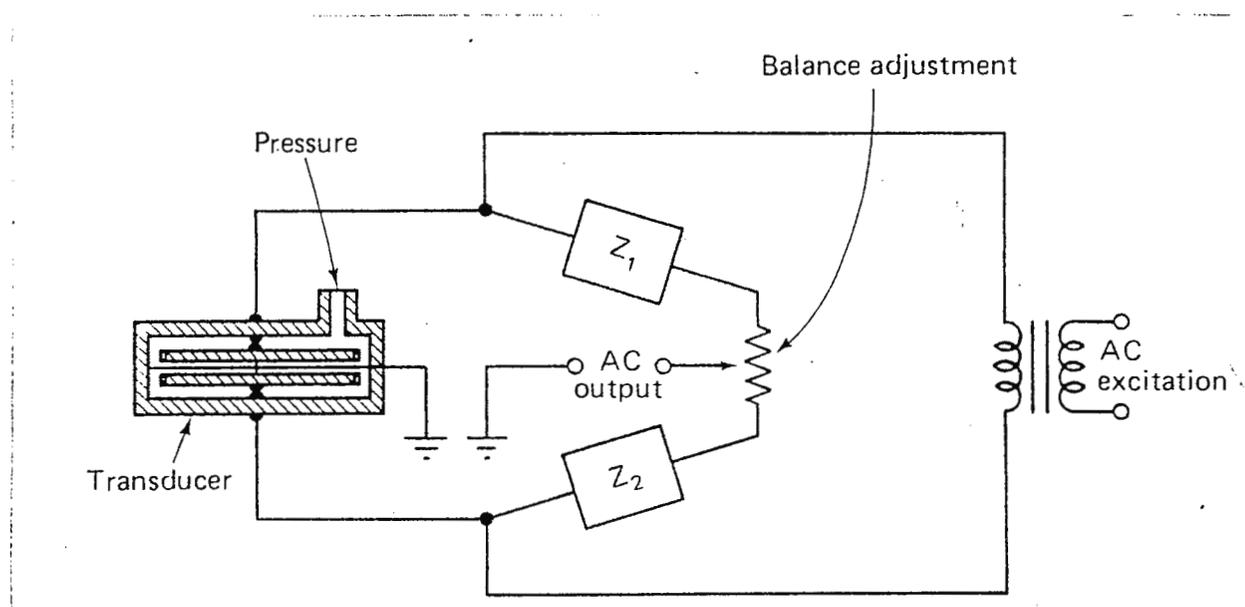


Fig.3.5 Puente de impedancias para un transductor de presión capacitivo.

Para finalizar lo que al transmisor se refiere resta únicamente por decir que ha sido ajustado para medir de 0 a 350mmHg con respecto a la presión atmosférica local (lado de baja presión al aire) y para obtener una salida de voltaje se conectará una resistencia de 10Ω y el voltaje de salida serán entonces de 40 a 200mV.

3.1.2-2 Circuito de la sección analógica

Esta sección se divide en 5 y a continuación se explican la operación de cada una:

a) Acondicionamiento de la señal de temperatura (Fig.3.6)

Por medio de una fuente de corriente constante, las variaciones de resistencia del termistor son convertidas a una señal de voltaje (V_1) y esta señal es amplificada por el Amp1. La señal V_2 es un voltaje constante y su valor es igual al que toma V_1 cuando la temperatura es

0°C, de esta manera ajustando adecuadamente V2 se consigue que V3 tome el valor de cero voltios. El control RV en la Fig. 3.6 representa el ajuste de ganancia del Amp1.

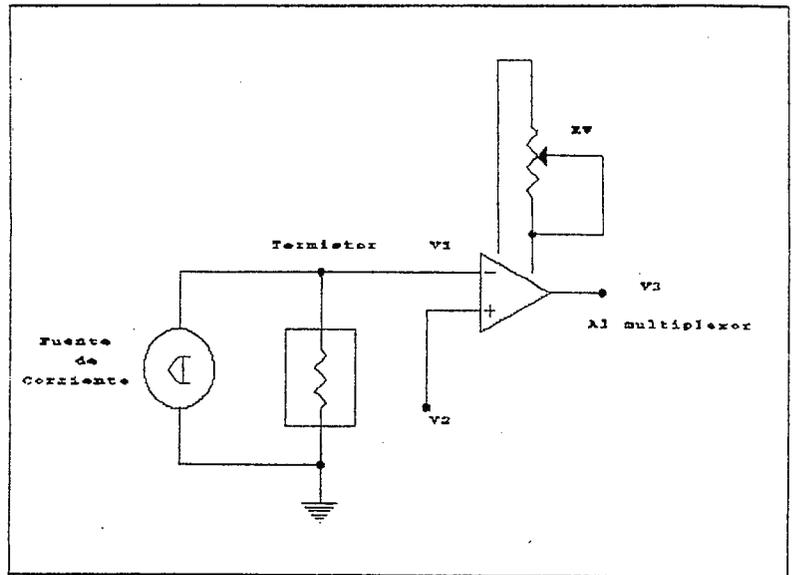


Fig.3.6 Circuito de acondicionamiento de la señal de temperatura.

b) Acondicionamiento de la señal de presión (Fig.3.7)

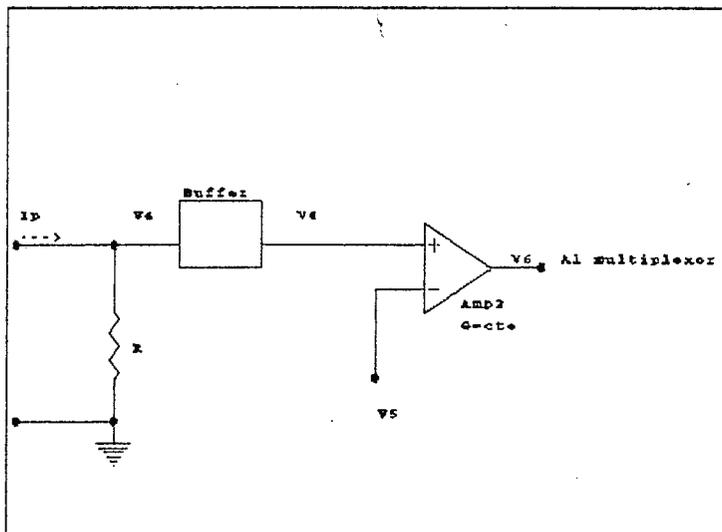


Fig.3.7 Circuito de acondicionamiento de la señal de presión.

El diagrama es parecido al de temperatura, pero la diferencia básica está a la entrada en donde la variable no es la resistencia sino la corriente I_p , ésta es la señal de salida proporcionada por el transmisor de presión diferencial. El buffer

aisla etapas para no cargar la caída en R, siendo V4 la señal de voltaje útil a amplificar por medio de Amp2, que posee para este caso una ganancia constante y al igual que el Amp1 para temperatura funciona como restador, porque la señal V5 tiene la función de anular el voltaje de salida V6 cuando la presión sea cero, cuando esto ocurra la corriente I_p

tendrá cierto valor que provocará caída en R y un cierto valor de V4 que deberá ser entonces igual a V5.

c) Sección de voltajes de referencia (Fig.3.8)

Los voltajes se obtienen creando una red divisora de tensión a partir de un voltaje constante y muy estable V7, luego a través de las líneas de selección se hace que el voltaje de salida V8 tome el valor de cualquiera de los voltajes creados a

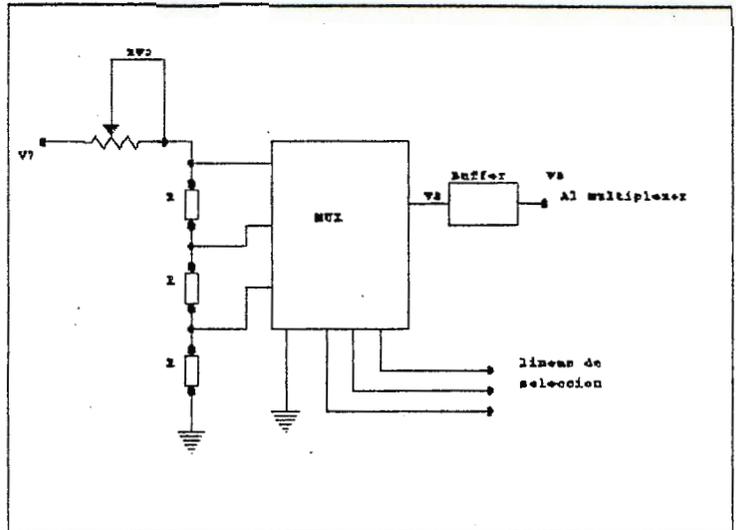


Fig.3.8 Esquema de la sección de los voltajes de referencia.

partir del divisor de tensión. Finalmente el buffer nuevamente aísla a los voltajes de referencia para que no se alteren por la carga del amplificador de salida.

d) Sección de nivel de la batería.

No es más que un divisor de tensión que proporciona un voltaje pequeño que pueda ser aplicado al amplificador de salida común del circuito y que además represente el nivel de la batería.

e) Sección del amplificador común de salida.

Al extremo común del multiplexor principal se encuentra el amplificador de salida, el cual proporciona la mayor amplificación a cada una de las señales del sistema y los lleva al rango de voltaje en que puedan ser convertidas a código digital por el ADC.

3.1.2-3 Proceso de diseño del circuito en su sistema analógico.

a) Corrección del voltaje offset

Debido al error producido por los voltajes offset en los circuitos operacionales de las diferentes etapas del circuito se optó por usar la red de corrección de offset de la Fig.3.9 en los operacionales IC33, IC35, IC37 e IC38 de las Fig.3.15 y 3.16

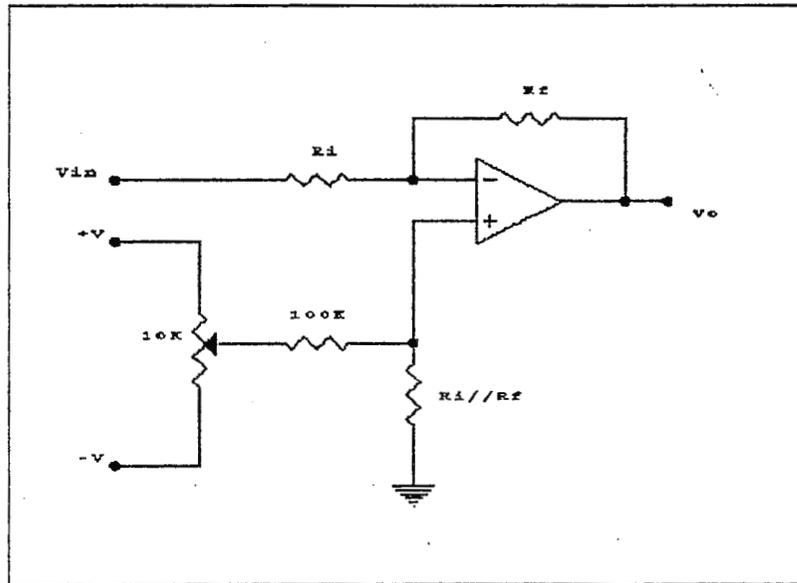


Fig 3.9 Red de corrección de offset en un amplificador inversor.

b) Amplificador común

La ganancia de este amplificador para ser empleado por igual por las distintas señales se tomó con ganancia de -16.6 producto de:

$$G = -R_f/R_i = -100K/5.1K = -19.6$$

c) Acondicionamiento de la señal de temperatura (Fig.3.6 y 3.15)

El IC42 proporciona un voltaje constante de +2.5V que aplicado a la resistencia de precisión de 15K proporciona la corriente constante de:

$$I = 2.5 / 15K = 166.66\mu A$$

La resistencia del termistor a 0°C es de : 6750Ω

Por lo que para que V3 sea 0V, V2 debe valor:

$$V2 = R(0^{\circ}\text{C}) * I$$

$$= 6750 * 166.66 = 1.125\text{V}$$

El potenciómetro de precisión RV6 es el encargado de que se logre este valor, por lo que viene a ser el ajuste de cero de temperatura. Para conocer la ganancia necesaria para Ampl se procede así:

La resistencia del termistor a 150°C es aproximadamente 50Ω entonces:

$$V1 = -IR_{(150^{\circ}\text{C})} = 166.66\mu\text{A} * 50\Omega = -8.33\text{mV}$$

si $t = 150^{\circ}$, $V_{in(ADC)} = 5\text{V}$, entonces $V3 = 5\text{V}/G_{out} = 5\text{V}/-19.6 = -255\text{mV}$

Ecuación del Ampl:

$$V3 = G_{(Ampl)} [V2 + V1]$$

$$G_{(Ampl)} = V3 / (V2 + V1) = -255\text{mV} / (1.125 - 8.33\text{mV}) = -0.228$$

$$G_{(Ampl)} = -0.228$$

si la R_i del Ampl valen 10K , R_f debe ser aproximadamente $= 2.28\text{K}$, ajustando RV4 se ajustará entonces la ganancia del Ampl.

La ecuación general de la sección de acondicionamiento de la señal de temperatura es:

$$V1 = -IR_{(t)}$$

$$V3 = G_{(Ampl)} * [V2 + V1]$$

$$V_o = G_{out} V3$$

$$V_o = G_{TOT} [V2 - IR_{(t)}]$$

$$V_o = 4.48 [1.125 - (166.66\mu\text{A}R_{(t)})]$$

d) Acondicionamiento de señal de presión (Fig.3.7 y 3.15)

$$I_{pmin} = 4\text{mA} \quad (\text{Presión} = 0\text{mmHg})$$

$$I_{pmax} = 20\text{mA} \quad (\text{Presión} = 350\text{mmHg})$$

$$R = 10\Omega \quad (R63)$$

$$V4_{min} = 40\text{mV}$$

$$V4_{max} = 200\text{mV}$$

Para que V6 sea cero voltios, cuando V4 sea mínimo, V5 debe valer -40mV, para ello se vale de IC43=2.5V constantes y con RV7 se logra el valor de V5 deseado, siendo RV7 el ajuste de cero de presión.

La ganancia necesaria para el Amp2 es:

$$V_{G(\max)} = 5V / -19.6 = -255mV$$

$$V_{G(\max)} = G_{(\text{Amp2})} (V4 + V5)_{\max}$$

$$G_{(\text{Amp2})} = V_{G(\max)} / V4_{\max} + V5, \quad V5 = -40mV$$

$$G_{(\text{Amp2})} = -255mV / (200mV - 40mV) = -1.594$$

En este caso se optó por usar ganancia de 1.5 con resistencia fijas de 15K Ω y 10K Ω para evitar ajustes y aunque de esta manera no se ocupe todo el rango de entrada del convertidor, la linealidad de la variable de entrada y la gran resolución del ADC permiten que la precisión no se vuelva inaceptable.

Ecuación de transferencia de la sección de acondicionamiento de la señal de presión:

$$V_o = G_{\text{total}} (V4 + V5) \quad V4 = RI_p, \quad I_p = 4mA + KP$$

$$V_o = 29.4 [10^{-2}(4 + 0.0457P) - 4 \cdot 10^{-2}] \quad P [=] \text{ mmHg}$$

$$V_o = 0.01343P \quad P [=] \text{ mmHg} \quad K = 0.0457mA/\text{mmHg}$$

e) Sección de voltajes de referencia (Fig.3.16)

En este caso se ajusta RV9 para que cuando se accese el voltaje de referencia de mayor valor a la salida de V8 se obtenga -0.255V que es la máxima escala en esta etapa. El resto de la red divisora está compuesta por 8 resistencias de precisión y de valor idéntico, los voltajes de referencia se obtendrán en pasos de -31.9mV desde 0 hasta -255mV.

f) Nivel de la batería.

Se seleccionó el divisor de tensión para obtener un voltaje de salida $V_o = 2.21V$ cuando el voltaje de la batería este correcto, y que conforme

este vaya bajando a V_o le sucederá lo mismo y el sistema podrá detectarlo.

3.1.2-4 Proceso de calibración

1. Colocando a tierra la entrada de cada amplificador se ajusta el offset a fin de reducirlo a lo menos posible.
2. Calibración de voltaje de referencia ajustando RV9.
3. Calibración parte de temperatura:
 - a) ajuste a cero para cero grados centígrados se ajusta RV6 de modo que V_o sea cero.
 - b) ajuste de ganancia. Para un valor conocido de la resistencia del termistor se ajusta RV4 a fin de obtener a la salida V_o el valor esperado que fue obtenido por medio de la ecuación de transferencia del sistema.
4. Calibración parte de presión:
 - a) ajuste a cero: para diferencia de presión de cero ($I_p=4mA$) se ajusta RV7 para que V_o se igual a cero.
5. Haciendo uso de la computadora y con todo el sistema funcionando comprueba la exactitud de las ecuaciones de transferencia, sobre todo en el caso de temperatura en donde, además de la ecuación obtenida en la etapa de diseño interviene la ecuación de transferencia T vrs R del termistor. El proceso es simplemente hacer trabajar al sistema tomando distintos datos de presión y temperatura y confrontarlos con las lecturas de presión y temperatura para cada caso tomados con medidores confiables y dependiendo de los resultados se debiere hacer un nuevo ajusta en el hardware o modificar la ecuación de transferencia en el software, sobre todo hay que recordar que la linealización lo hace a través de la ecuación de transferencia del transductor.

Tabla 3.3 Problemas y Soluciones

PROBLEMAS	EXPLICACION	SOLUCIONES
a. Offset.	<p>Producia alteracion en el comportamiento del Sistema. Los resultados de la etapa analogica no eran los esperados.</p>	<p>Implementar las redes de compensacion a todos los operacionales.</p>
b. Ruido.	<p>De este existieron dos tipos: la del propio circuito y el otro de ruidos parasitos que se introducian al sistema.</p>	<p>Tanto en las lineas de alimentacion como en los pines de los operacionales se colocaron capacitores. El valor de estos fueron del orden los microfaradios para las lineas de alimentacion y de 0.1 y 0.01 en los operacionales</p>
c. Lineas de tierra.	<p>Dependiendo de la distancia y debido a la caida en ellas(breadboard), producian diferencias de potencial que introducian error entre etapas cuando se trabajaba con pequenas senales, e inclusive en un momento el multiplexor de los voltajes de referencia no interpretaba el nivel logico cero.</p>	<p>Para reducir los errores de este tipo, fue crear el mayor numero de pasos posibles entre lineas de tierra para reducir las caidas de potencial.</p>

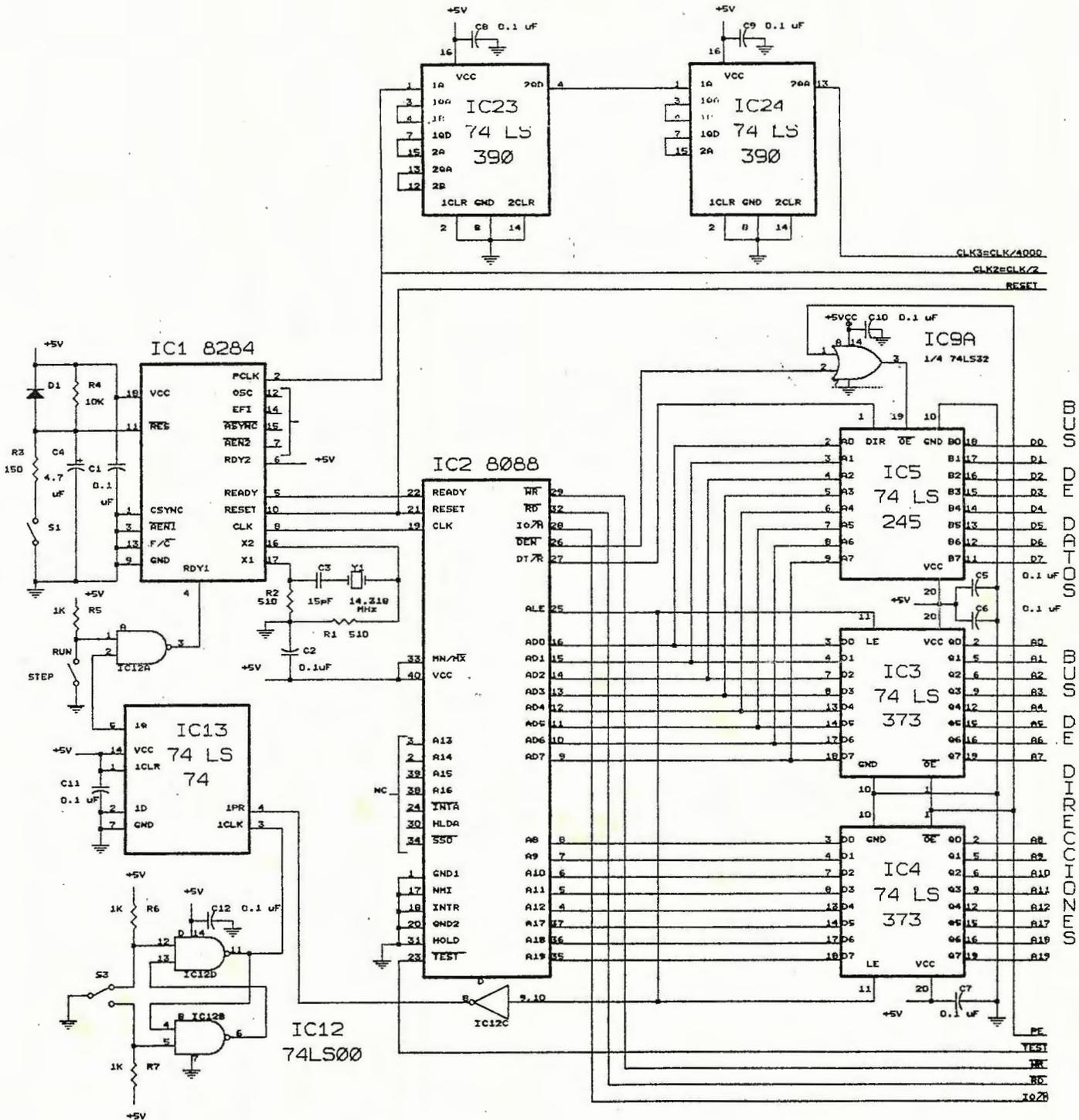


Fig. 3.10 SECCION DE CONTROL. PROTOTIPO

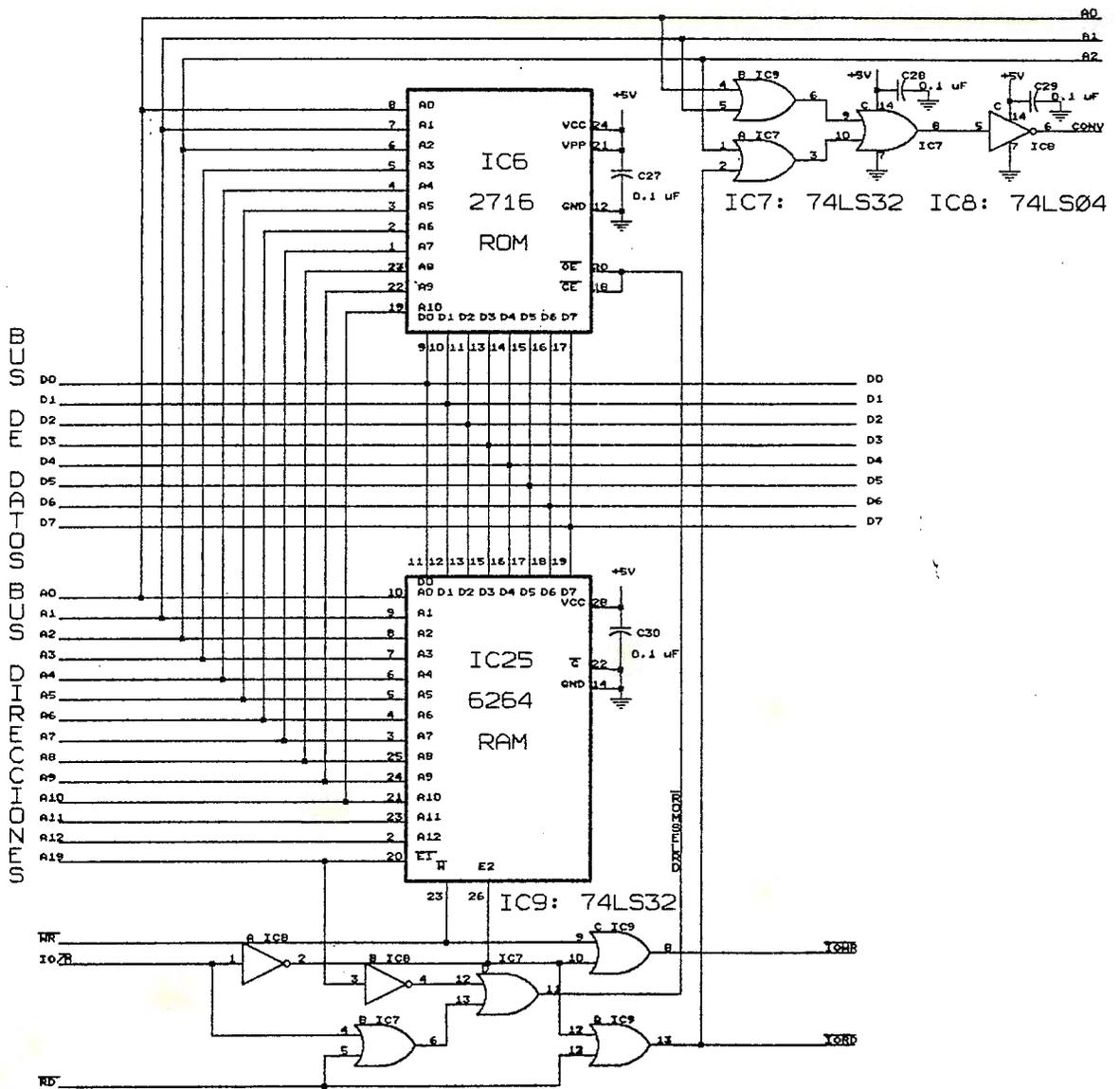


Fig. 3.11 SECCION DE MEMORIAS Y
DECODIFICADOR DE DIRECCIONES

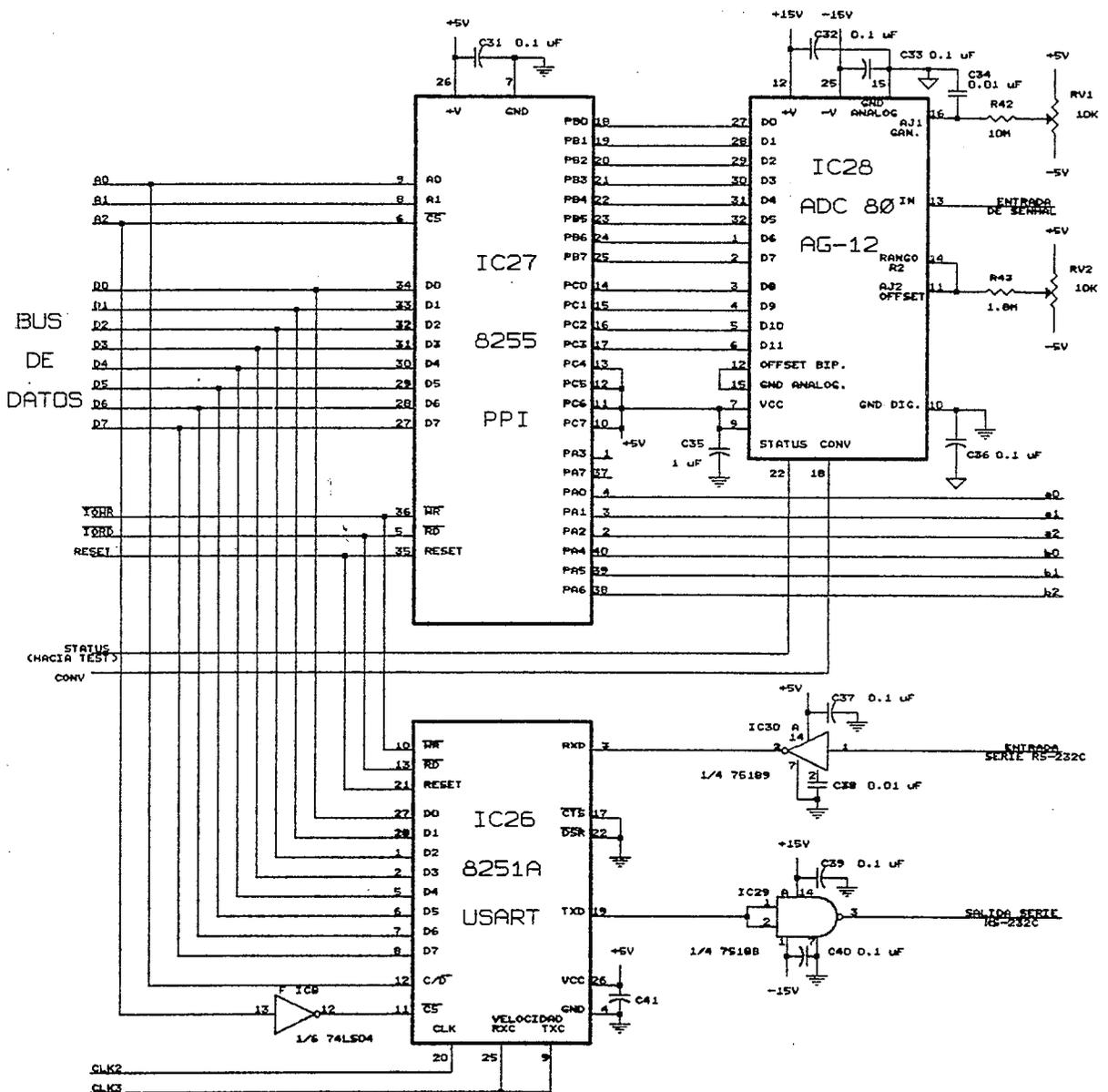


Fig. 3.12 SECCION DE INTERFACES. PROTOTIPO

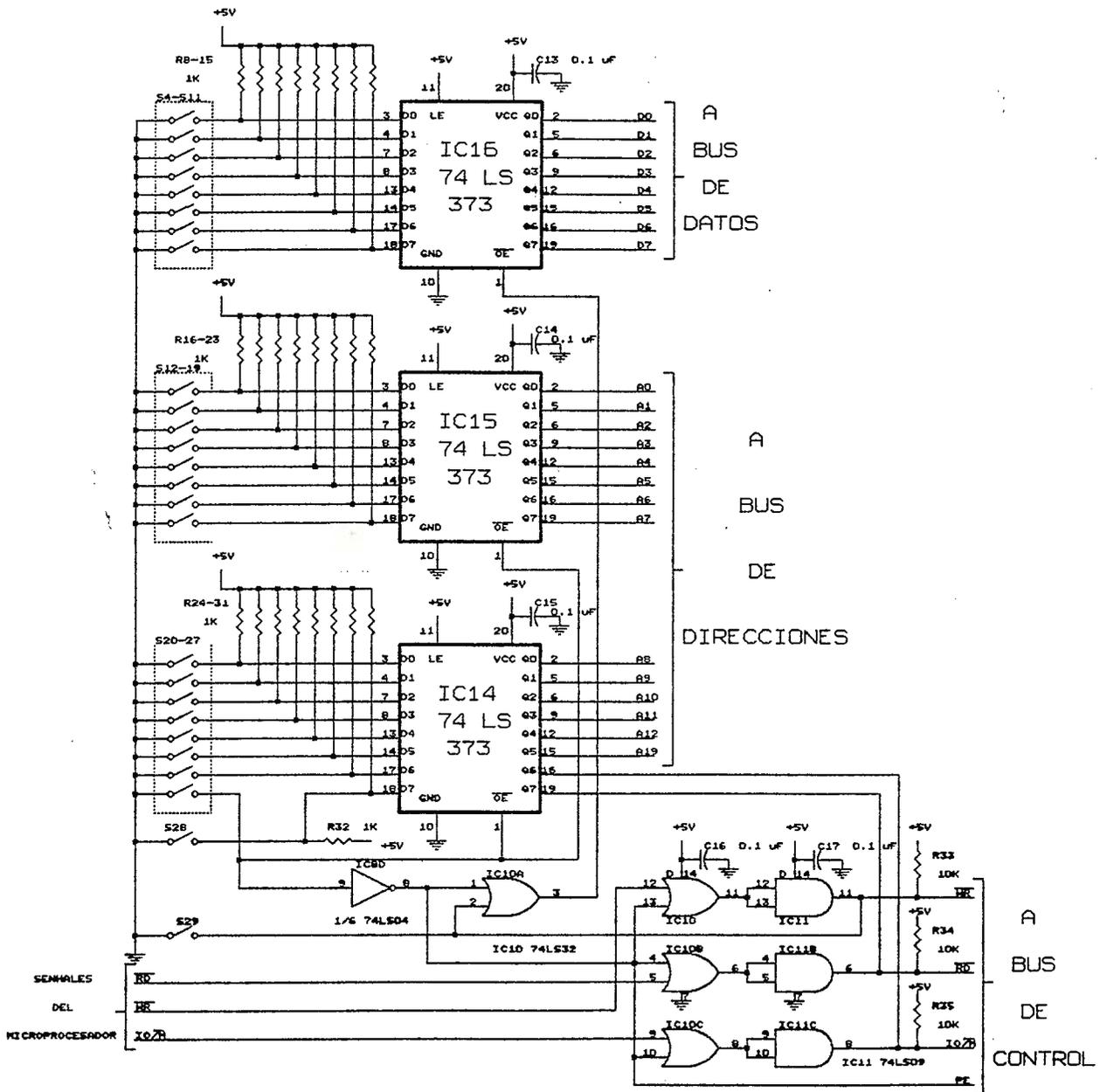


Fig. 3.13 SECCION DE CONTROL, CIRCUITO DE PRUEBA DEL PROTOTIPO

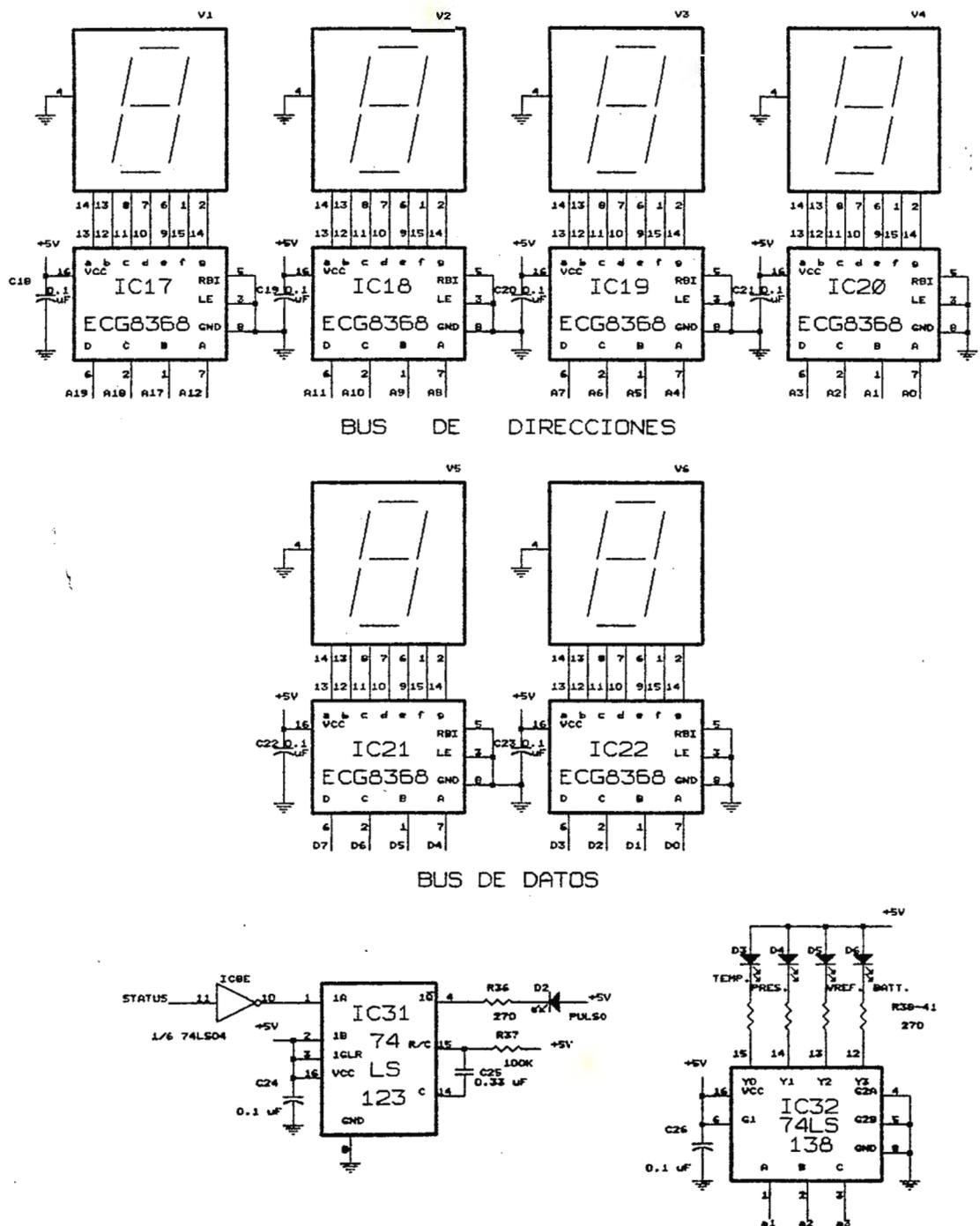


Fig. 3.14 SECCION DE VISUALIZACION. CIRCUITO DE PRUEBA DEL PROTOTIPO

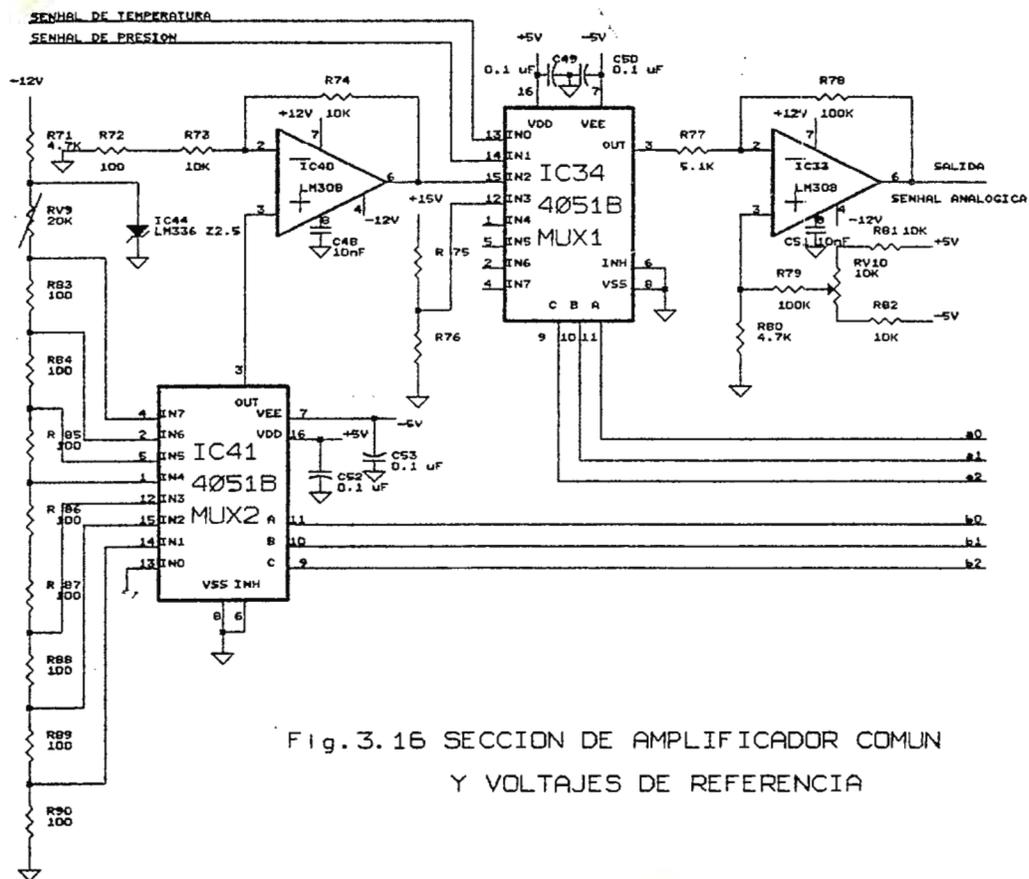


Fig. 3. 16 SECCION DE AMPLIFICADOR COMUN
Y VOLTAJES DE REFERENCIA

CAPITULO 4

SOFTWARE DEL SISTEMA PROGRAMA MONITOR.

velocidad de transmisión = 1200 bits por segundo

bit de paridad = paridad impar

número de bits de parada= 1bit

y la segunda se refiere a programar la USART, ya sea, como transmisor o como receptor, pero estas se realizan en las rutinas que lo requieran.

4.1.2 Estado1

El Estado1 programa la USART como receptor y es un estado de espera en el que se encontrará normalmente el programa cuando no se le haya asignado nada que hacer desde la computadora (Fig.4.1); en éste el sistema recibe el dato que desde aquella se le envía, dicho dato puede ser cualquiera de las opciones de operación que el sistema puede ejecutar .Si el código recibido es un código válido, dependiendo de cual sea; el programa pasa del estado1 a realizar la rutina que le corresponda.

Si el dato recibido no es código válido entonces envía de vuelta a la computadora dicho código para informarle de lo que se recibió (esto lo hace TX) y se vuelve a ejecutar el programa desde el inicio.

4.1.3 Vector de Subrutinas

Este es simplemente un espacio dentro del programa en donde se ubican las interrupciones de llamado de subrutinas y a la vuelta de estas se envía la secuencia del programa al inicio del mismo (Fig.4.2). Dependiendo del código recibido en el Estado1, por medio del vector de subrutinas se envía al micro a que ejecute la subrutina correspondiente.

4.1.4 Programar

Lo que hace esta rutina es recibir los códigos de programación que la computadora le envía y los almacena en las localidades de memoria de la RAM (Fig.4.2). El total de códigos de programación es de nueve bytes, que están desde la localidad 00000 hasta la 00008.

4.2.1 Transmisión.

Antes de describir en que consiste esta subrutina es necesario explicar que cuenta con dos partes que tiene por nombre: TRNS y TRANSII, las cuales son llamadas desde las subrutinas VERIFICAR y CALIBRACION respectivamente.

Como se muestra en la Fig.4.3 la subrutina TRANSMISION consta de dos secciones, en la primera se envían 10 veces c/u de los nueve datos de programación del sistema, y en la segunda se envían 10 veces c/u de los datos recogidos por el sistema.

Cuando se llama la rutina TRANSMISION (desde el principio) se esperan transmitir los datos de programación y los datos recogidos producto de la medición.

Cuando se llama a TRNS solo se transmiten los datos de programación. Cuando se llama a TRANSII solo se transmiten datos recogidos de la autocalibración. (en este caso voltajes de referencia).

4.2.2 Pruebal.

Pruebal es la subrutina de prueba de línea, la cual consiste en recibir los 100 datos enviados por la computadora y almacenarlos en la memoria RAM y después transmitirlos nuevamente hacia la computadora. Véase Fig.4.4.

4.2.3 Autocalibración.

Lo que esta subrutina hace es leer los ocho voltajes de referencia y el nivel de la batería, luego guarda estos datos en memoria y posteriormente los transmite hacia la computadora. Para el caso de los voltajes de referencia hará una sumatoria de 16 muestras de cada voltaje para que la PC obtenga el promedio de cada uno. (Figuras 4.5 y 4.6)

El proceso de toma de datos es el siguiente: Se coloca un cierto

código en el acumulador del microprocesador y se escribe en el puerto A del PPI para que habilite al multiplexor1 y así tomar datos de referencia (o de nivel de batería), y al multiplexor2 para seleccionar el voltaje deseado (se comenzará con el más bajo y se finalizará con el más alto). Luego se pierde un poco de tiempo y entonces se toma la muestra leyendo de los puertos B y C del PPI. Esto se hace 16 veces para cada voltaje de referencia y para ello la sumatoria se lleva en el registro interno del microprocesador llamado DX, y al final se guarda el total en un memoria de RAM, se vuelve a hacer lo mismo para los otros voltajes de referencia y para el nivel de batería donde no se lleva una sumatoria. Al terminar la toma de datos se llama la subrutina TRANSII (parte de transmisión) para que los datos se envíen a la computadora.

4.2.4 Adquisición.

En las Figuras 4.7, 4.8 y 4.9 se muestran los flujogramas correspondientes a esta rutina, pero antes de explicar es importante describir los códigos de programación que rigen el funcionamiento de la misma en la Tabla 4.1.

La subrutina de adquisición de datos es la más importante del sistema y consiste básicamente en:

- 1) Habilitar el multiplexor1 para seleccionar el canal adecuado, sea presión o temperatura, mandando un código a través del puerto A del PPI.
- 2) Dar al ADC el pulso de inicio de conversión.
- 3) Leer el dato a través de los puertos B y C.
- 4) Guardar el dato.

Tabla 4.1 Resumen de los códigos de programación

Símbolo	Localidad	Descripción
M(0)	00000	Código de la variable a medir 0 = Temperatura, 1 = Presión 2 = Ambos.
M(1) y M(2)	00001 y 00002	Código del tiempo antes de la primera toma de datos. M(1) es LSB y M(2) es MSB.
M(3) y M(4)	00003 y 00004	Código del tiempo entre estaciones de toma de datos. M(3) LSB y M(4) MSB.
M(5)	00005	Código de la obtención de datos: 0 = simple ; 1 = promedio
M(6)	00006	Número total de muestras
M(7) y M(8)	00007 y 00008	Número total de datos

Ahora bien, los códigos de programación intervienen de la siguiente manera:

- Si se va a obtener promedio de las muestras en una estación entonces el sistema realiza una sumatoria con las muestras, ocupando registros internos del microprocesador (CX para temperatura y BX para presiones), y al finalizar la toma de muestras en una estación se guardan los totales. En caso de ser toma simple, el sistema guarda dato por dato.
- Para la habilitación de los canales de presión y temperatura, otro registro interno del micro, es el encargado de llevar el control, éste es el DL, cuyo contenido (00=temperatura, 01=presión) se envía al puerto A del PPI para llevar a cabo la habilitación. Inicialmente el sistema toma un dato de temperatura, y para saber si debe tomar presión consulta con M(0), de resultar cierto cambia el código de DL y repite el

proceso hasta acabar la toma de muestras y sucesivamente hasta acabar la toma de datos.

- Siempre que se toma un dato se consulta a M(6) para saber si debe tomar más datos en una estación, o debe detenerse al haber tomado la última muestra requerida; lo mismo al terminar una estación el sistema consulta con M(7) y M(8) para finalizar el proceso y retornar si es que se han tomado todos los datos o repetir toda la rutina para una estación más si es que no es así.

- Los retardos de tiempo en la surutina son de tres tipos:

a) Antes de la toma de datos: es una pérdida de tiempo definida por el usuario y almacenada en M(1) y M(2).

b) Entre estaciones: también definida por el usuario, y se lleva a cabo después de que se han tomado todas las muestra en una estación.

c) Entre muestras: es con el fin de adaptar la velocidad del sistema a la velocidad de respuesta del transductor, o dicho de otra forma, es darle tiempo al transductor. Si se va a tomar solo temperatura es de 2seg. y se toma presión es entonces de 3seg.

4.2.5 TESTRAM.

TESTRAM (Fig.4.10) es un subrutina que prueba el estado de la memoria RAM del sistema, y lo ejecuta de la siguiente manera:

1. Lee y guarda el contenido de una localidad de memoria.
2. Escribe una palabra de prueba en esa localidad.
3. Si la localidad esta bien, la palabra escrita deberá conservarse y al leer de nuevo en esa localidad deberá estar la palabra de prueba.
4. Cuando no es así, se detiene el proceso y se le comunica a la computadora el resultado erroneo (por medio de TX).

5. Si para todas las localidades bajo prueba el resultado fue satisfactorio entonces se envía a la computadora la palabra de prueba (resultado correcto) a través de TX.

4.2.6 RESETRAM.

La función de la subrutina RESETRAM (Fig.4.11) es escribir el código 00 en todas las localidades destinadas a almacenar datos.

4.2.7 Verificar.

Tiene como objetivo transmitir hacia el computador, el contenido de las nueve primeras localidades de la RAM que corresponden a los códigos de programación; para ello se vale de la subrutina de transmisión y únicamente coloca por medio de DL (registro interno del micro) un indicador que solo efectuará la transmisión de los códigos de programación y luego retornará a inicio. (Fig.4.12)

4.2.8 TX.

Su función es únicamente transmitir el último dato dado por el computador 10 veces hacia la computadora. Con la excepción de cuando la llaman desde TESTRAM es ocupada para devolver el código que la computadora envía primero, con el fin de que la misma "entienda" que su dato fue bien recibido. (Fig.4.12)

4.2.9 Retardo.

Es una base de tiempo de 0.1seg; se realiza a base de un contador interno que ocupa un registro del micro, el cual se carga con cierto número que se va decreciendo hasta llegar a cero. Utiliza como entrada el registro BX cuyo contenido establece cuantas veces se repitirá el proceso. (Fig.4.13)

La ecuación que calcula el retardo es:

$$\text{Tiempo} = (0.1)\text{BX seg.}$$

Tabla 4.2 Problemas y Soluciones

PROBLEMAS	EXPLICACION	SOLUCION
<p>a. Datos erroneos en la comunicacion.</p>	<p>Exactamente no se sabe a que se deben, pero podrian ser: incompatibilidad de los relojes debido a que la frecuencia a la que la USART se hace trabajar es de 1193Hz y no de 1200Hz como debe de ser; pero esto no es seguro, otra causa es la del ruido en el medio ambiente. Como las condiciones del medio no son siempre las mismas en algunos casos puede que el ruido sea tal que distorsione aun mas los datos.</p>	<p>Para el caso de la recepcion los datos de programacion se reciben 10 veces cada dato, de esto se asegura que la mayor parte de ellos llegaran sin error, y una rutina de obtener el dato que mas se repite aseguraria el dato verdadero. Otra salida es realizar un programa que detecte cuando ha recibido un dato malo y decirle al computador que transmita solo ese dato hasta que este libre de errores.</p>
<p>b. Salida automatica de estado de espera.</p>	<p>Se refiere al caso en que el sistema espera recibir cierta cantidad de datos, y si por alguna razon no los recibe todos, el sistema permanecera en ese estado por tiempo indefinido.</p>	<p>Las rutinas que reciben datos (Programar y Pruebal) se incluye un contador para que si la espera de recepcion de un dato al otro excede cierto tiempo (aprox 1/4seg) el lazo de espera se acaba y se retorna al inicio.</p>
<p>c. Inicializacion del Apuntador de pila.</p>	<p>Al termino de una subrutina cuando se vuelve al estado de espera de codigo debe inicializarse el apuntador de pila; porque se dio el caso de que si esto no se hace el estado libre de la pila (contenido del indicador de pila) pudiera ser que se desplace hasta localidades de RAM que almacenen datos y una perdida de informacion.</p>	<p>Al final de cada subrutina se inicializa todos los registros internos del micro. Esto es vuelve a inicio.</p>
<p>d. Tiempo de retardo entre la colocacion del codigo de habilitacion de canales del MUX y del pulso de inicio de lectura del ADC.</p>	<p>Cuando ambas acciones se hacian en un solo paso, usando una sola instruccion el ADC tomaba datos erroneos debido al tiempo de respuesta del MUX.</p>	<p>Para el caso del sistema del prototipo se exagero dicho tiempo y se coloco en 1seg(para asegurar) pero en realidad debe ser mucho menor, unas 10 veces mas que lo que el fabricante asegura es el tiempo de respuesta del MUX.</p>

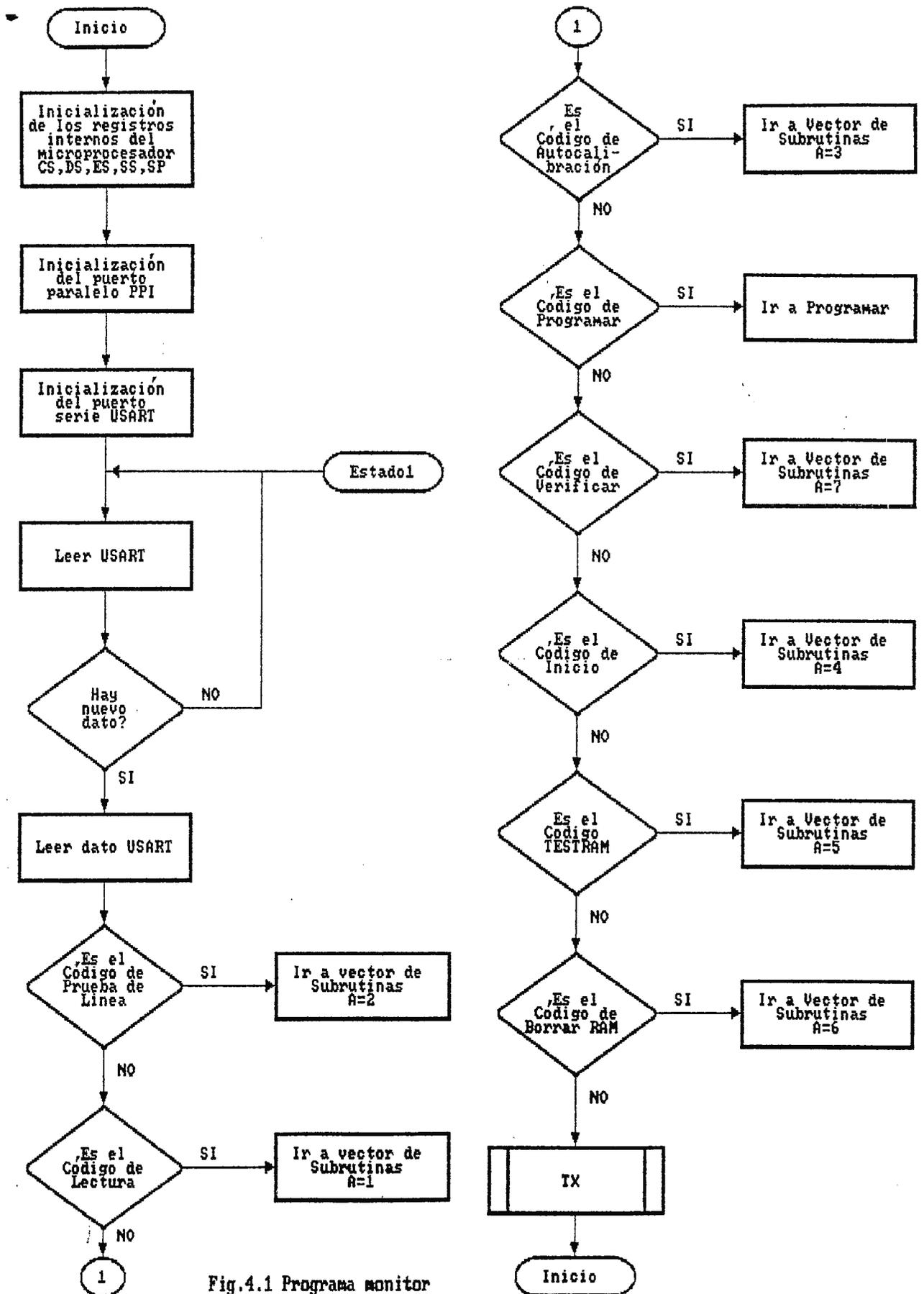


Fig.4.1 Programa monitor

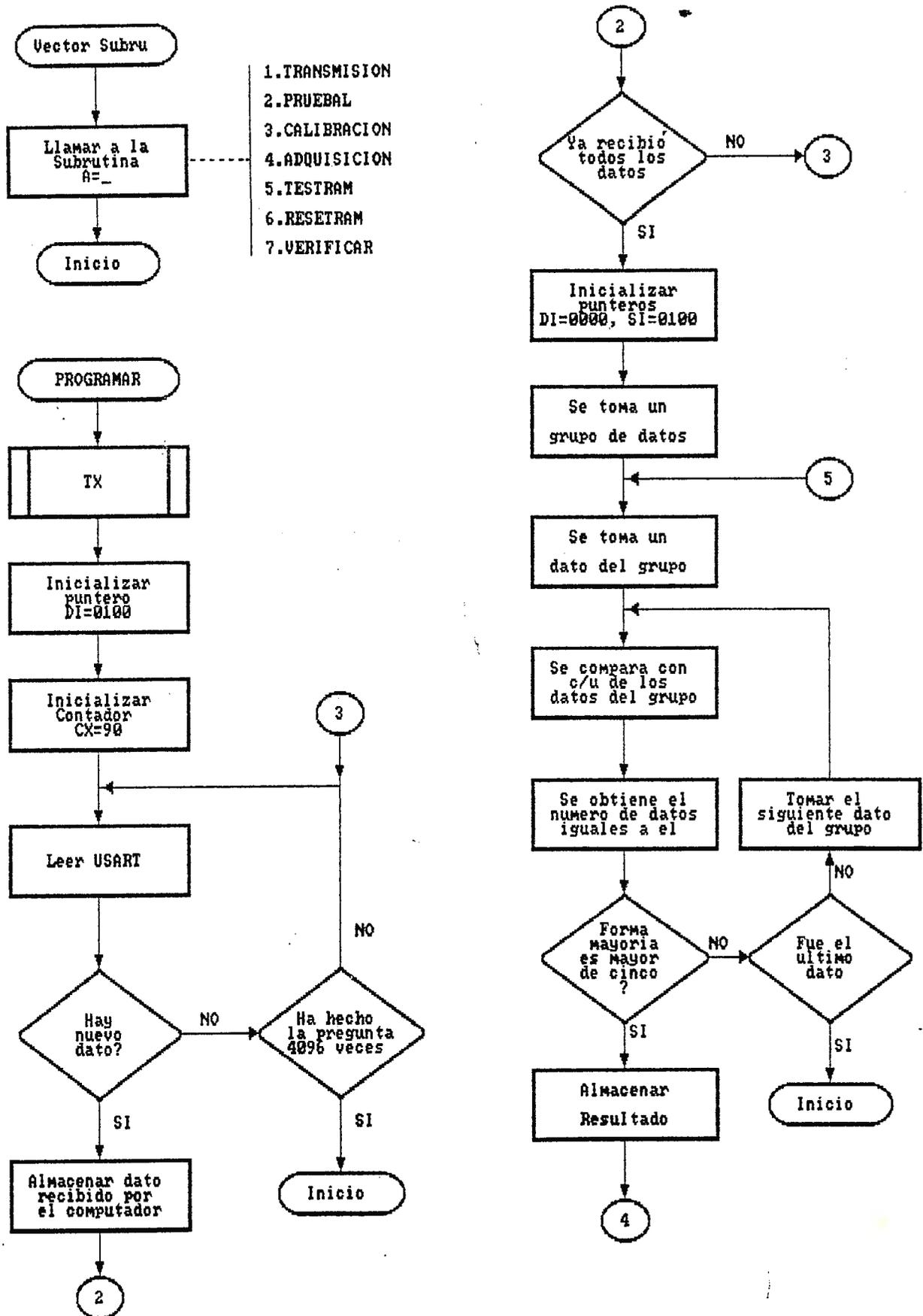


Fig.4.2 Rutina de Programación de la sonda.

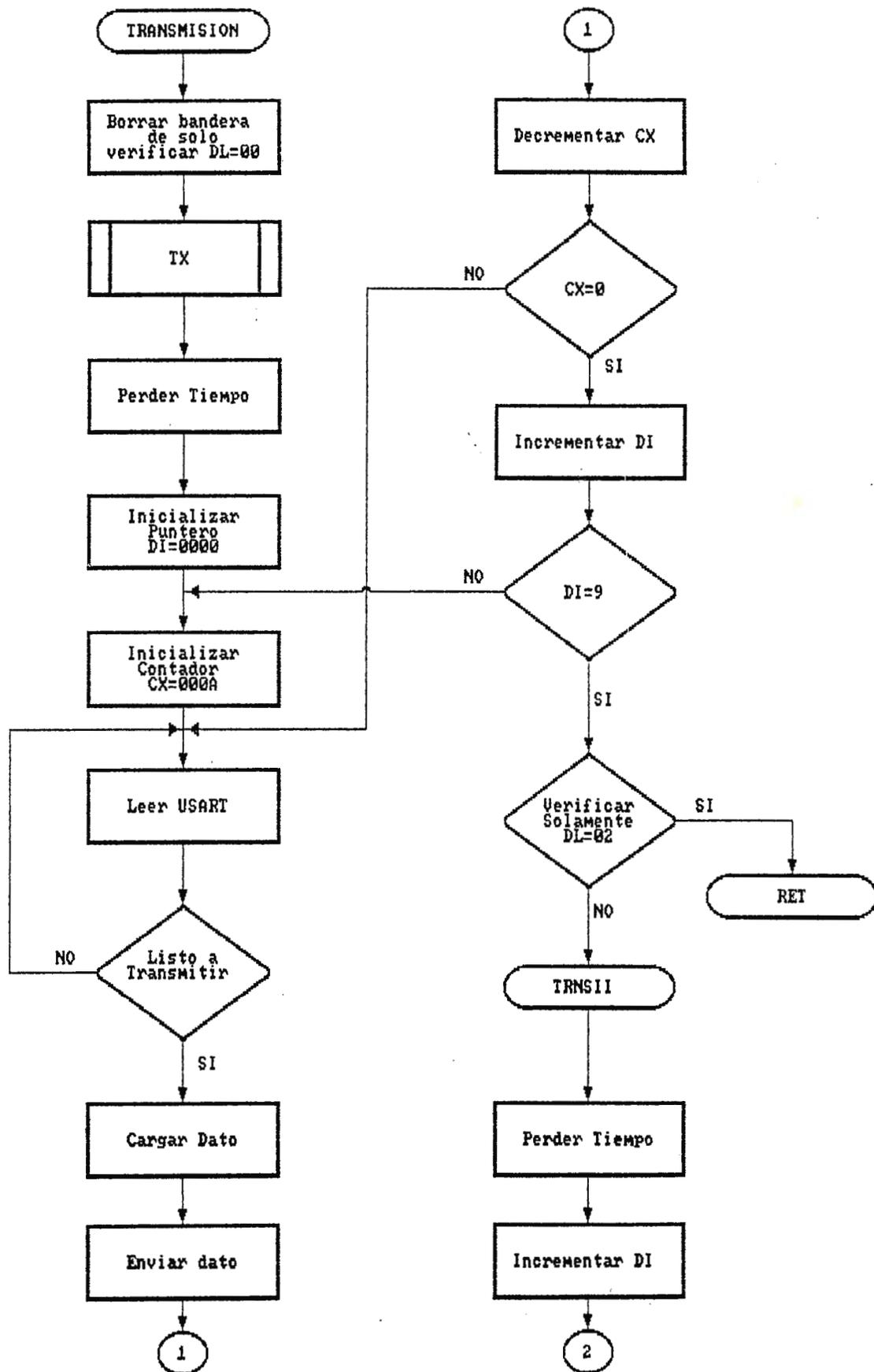
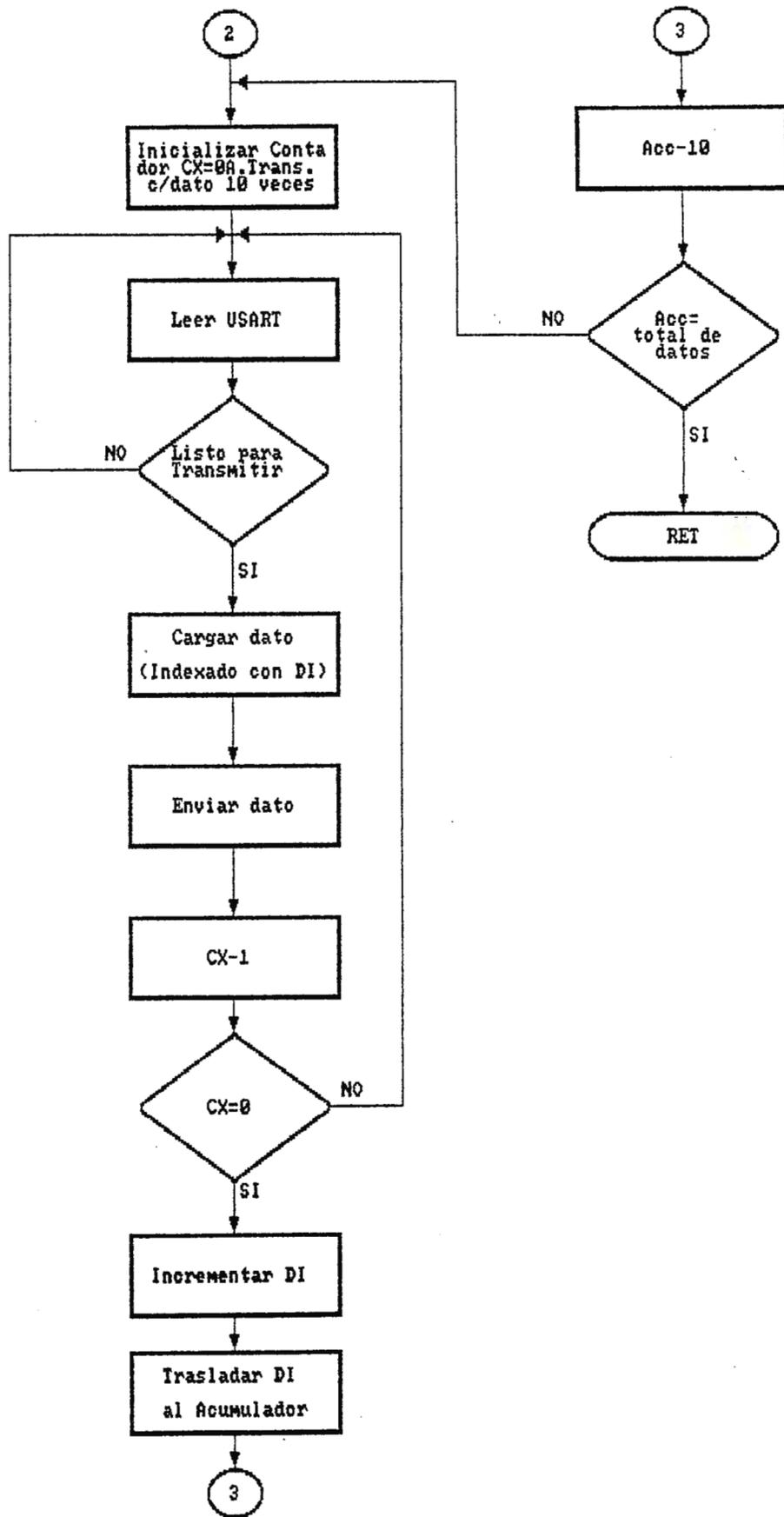


Fig.4.3 Subrutina de Transmisión I.



Continuación de la Fig.4.3

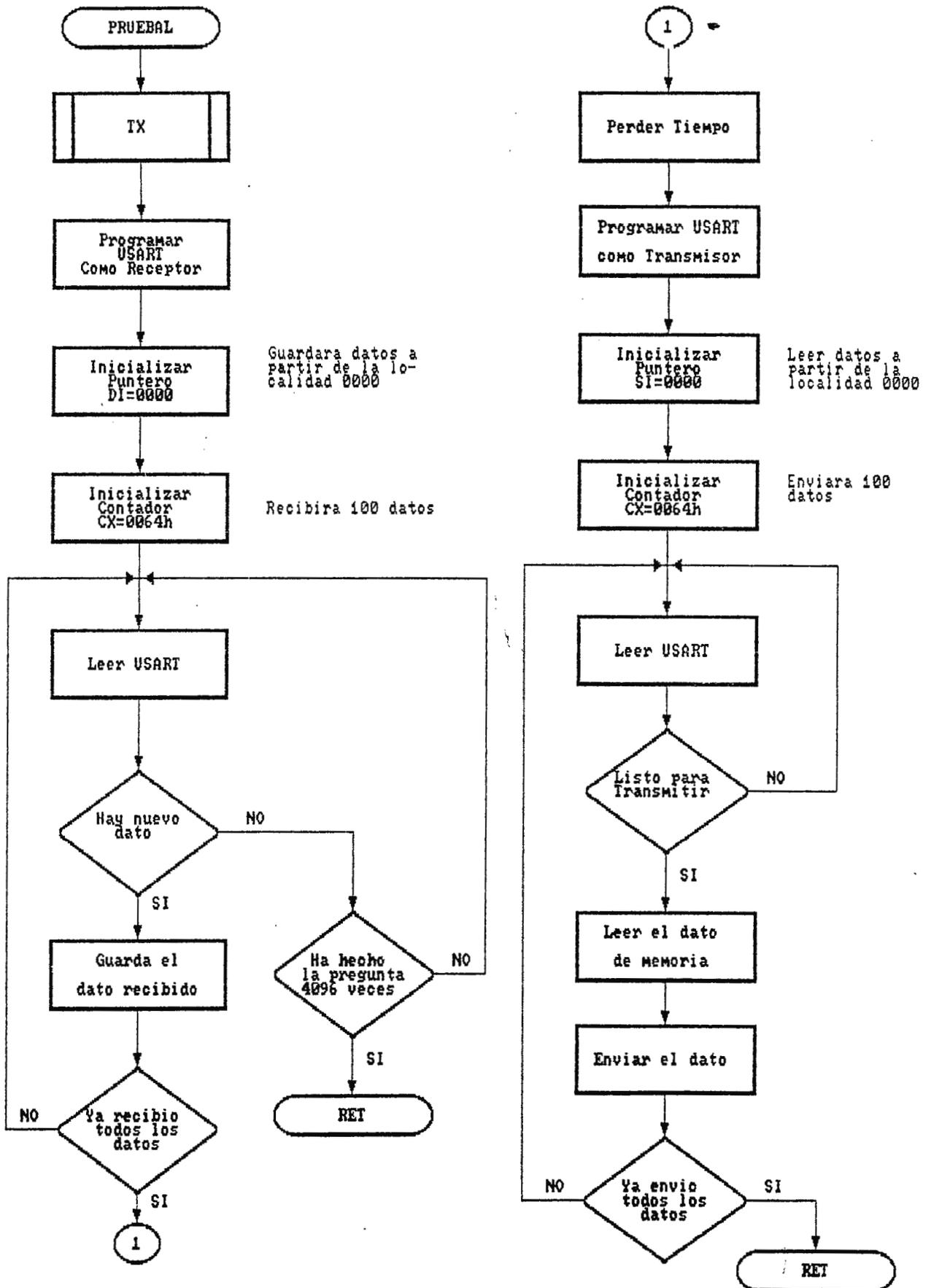


Fig.4.4 Subrutina de prueba de linea.

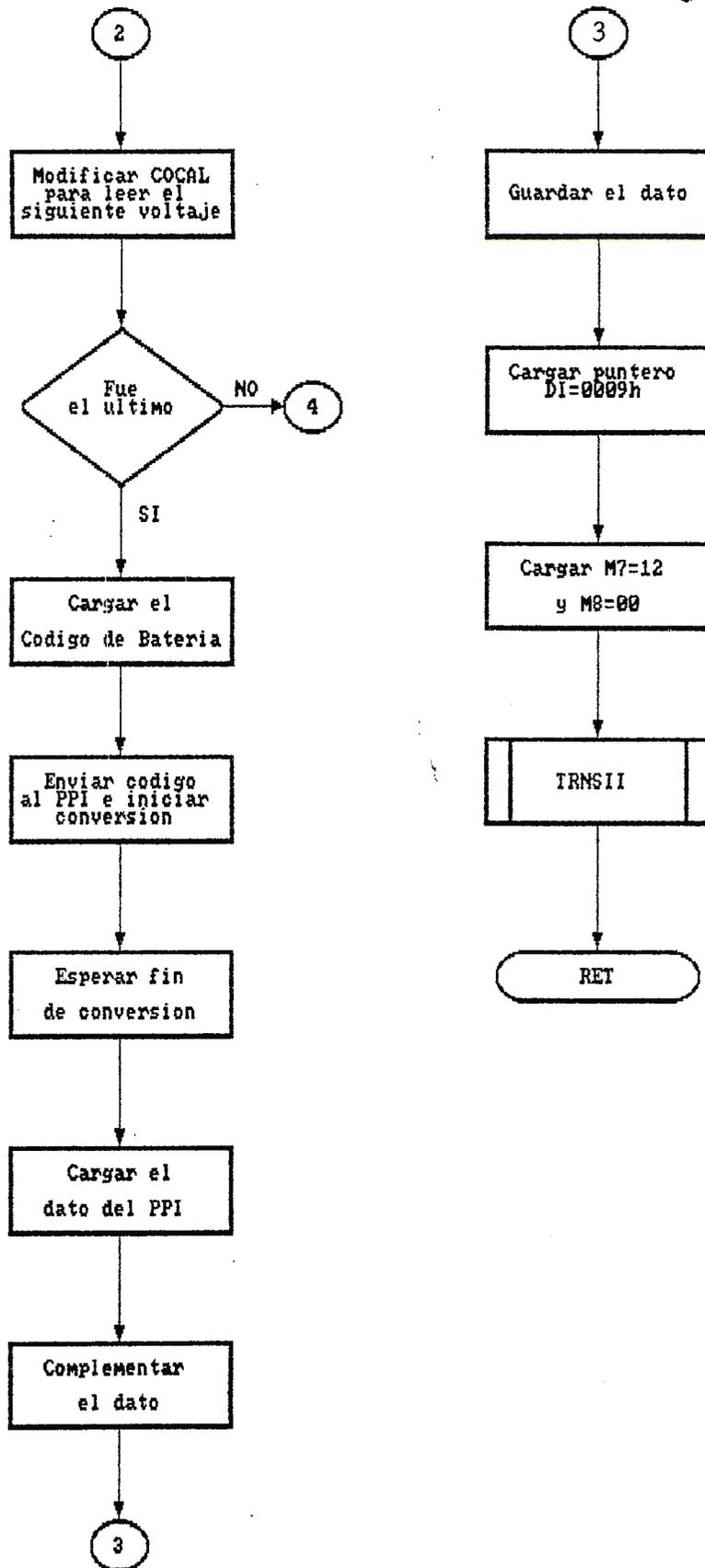


Fig.4.6 Subrutina de Calibración II.

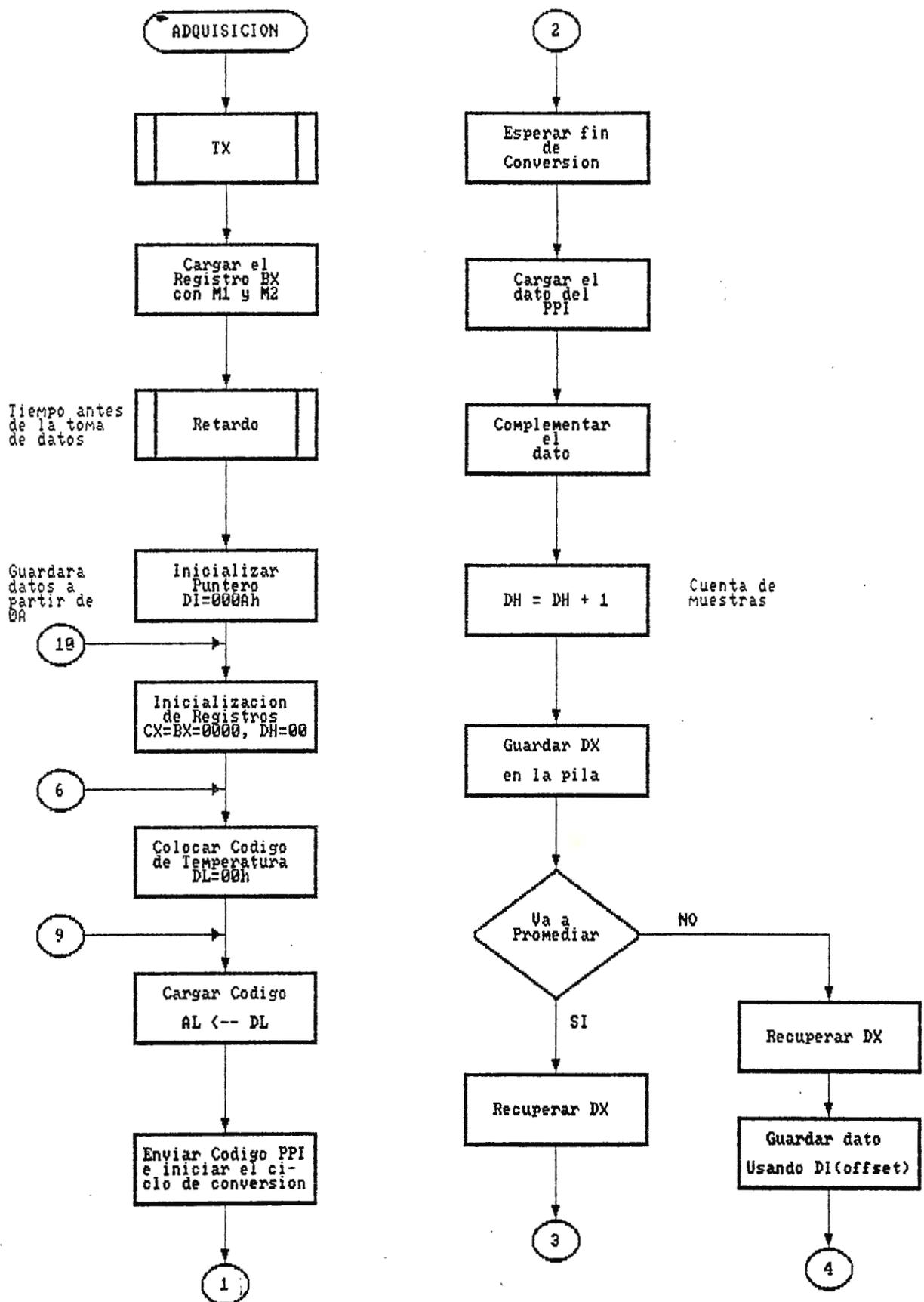


Fig.4.7 Subrutina de Adquisición de datos I.

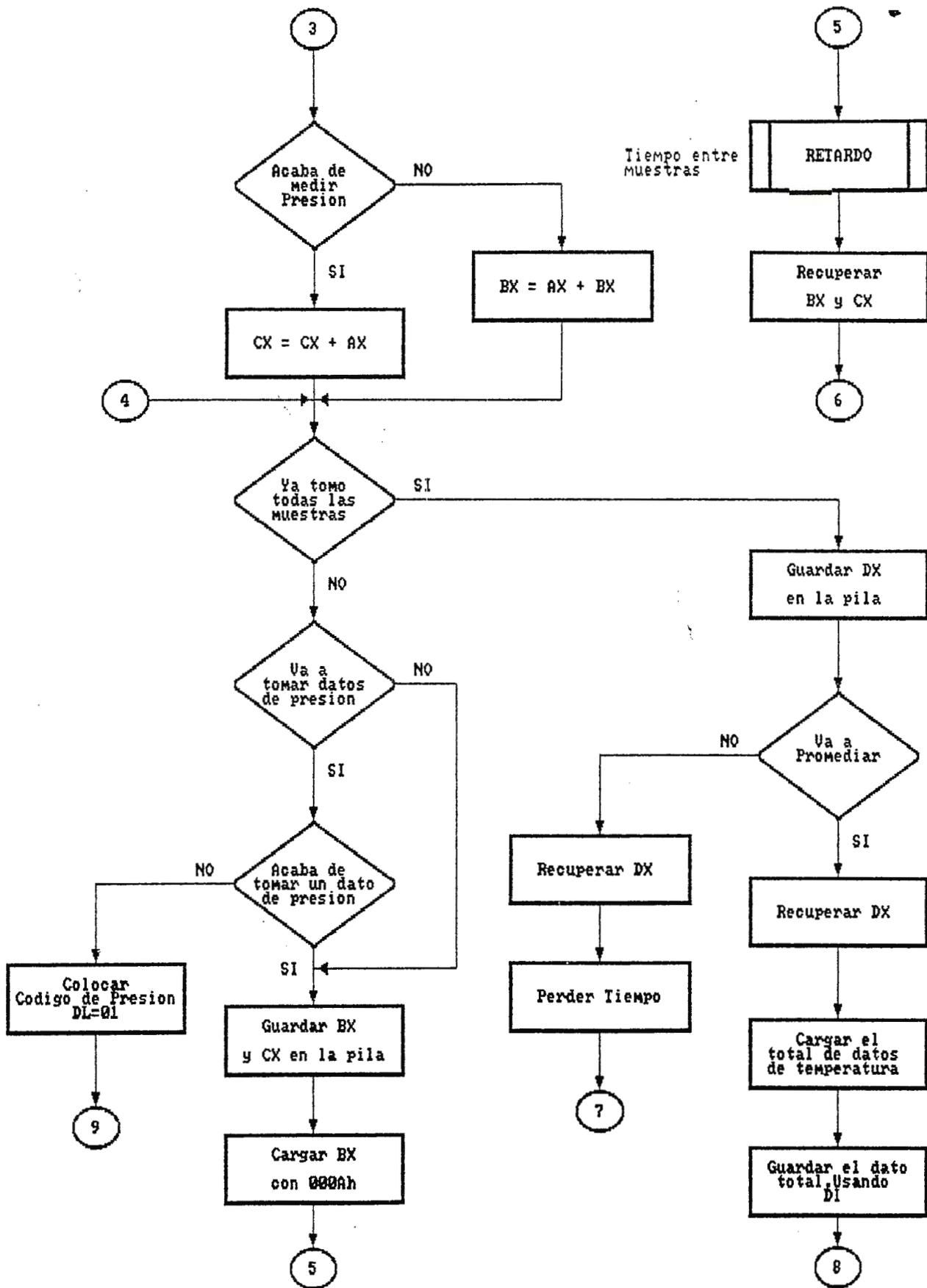


Fig.4.8 Subrutina de Adquisición de datos II.

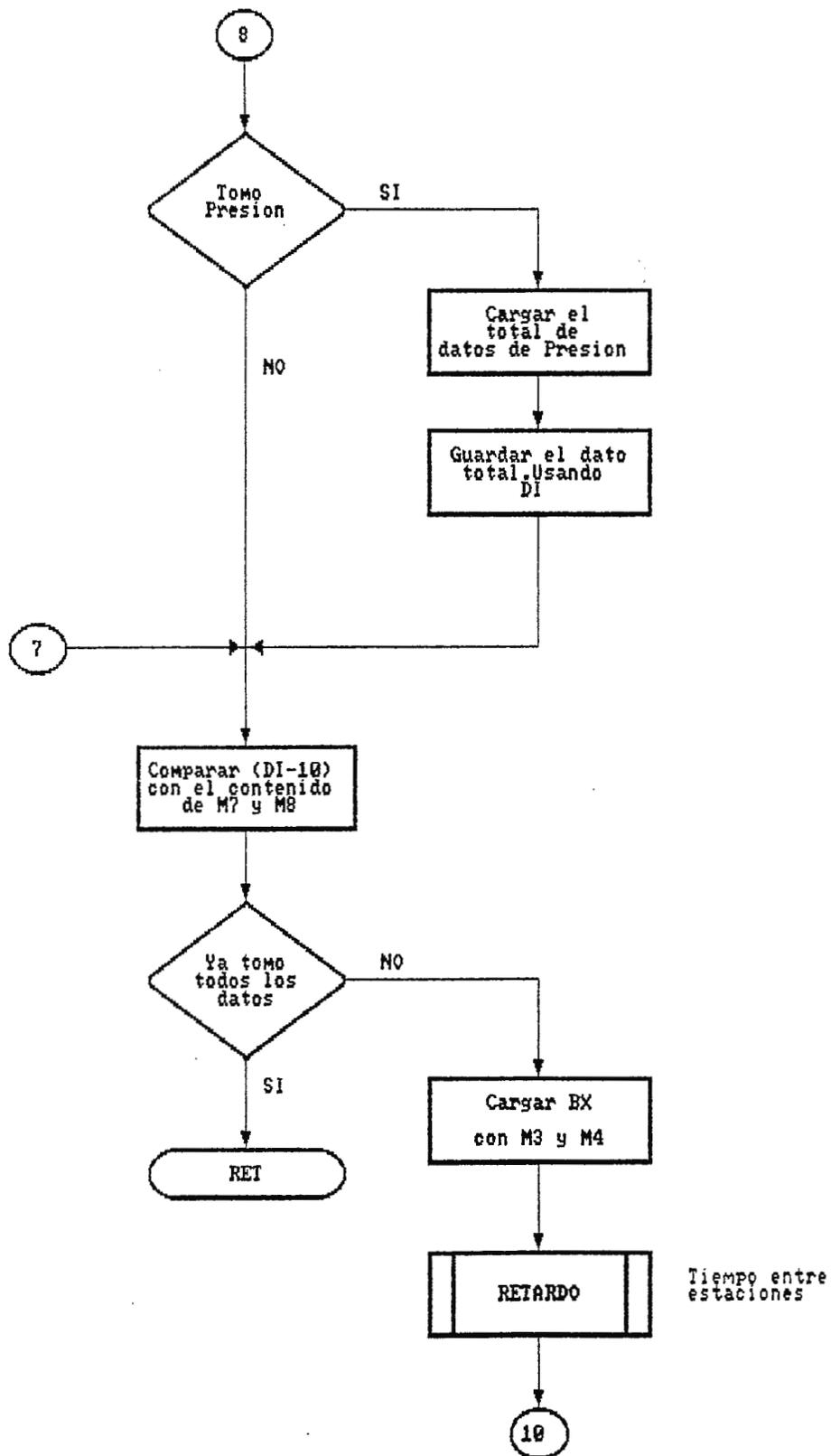


Fig.4.9 Subrutina de Adquisición de datos III.

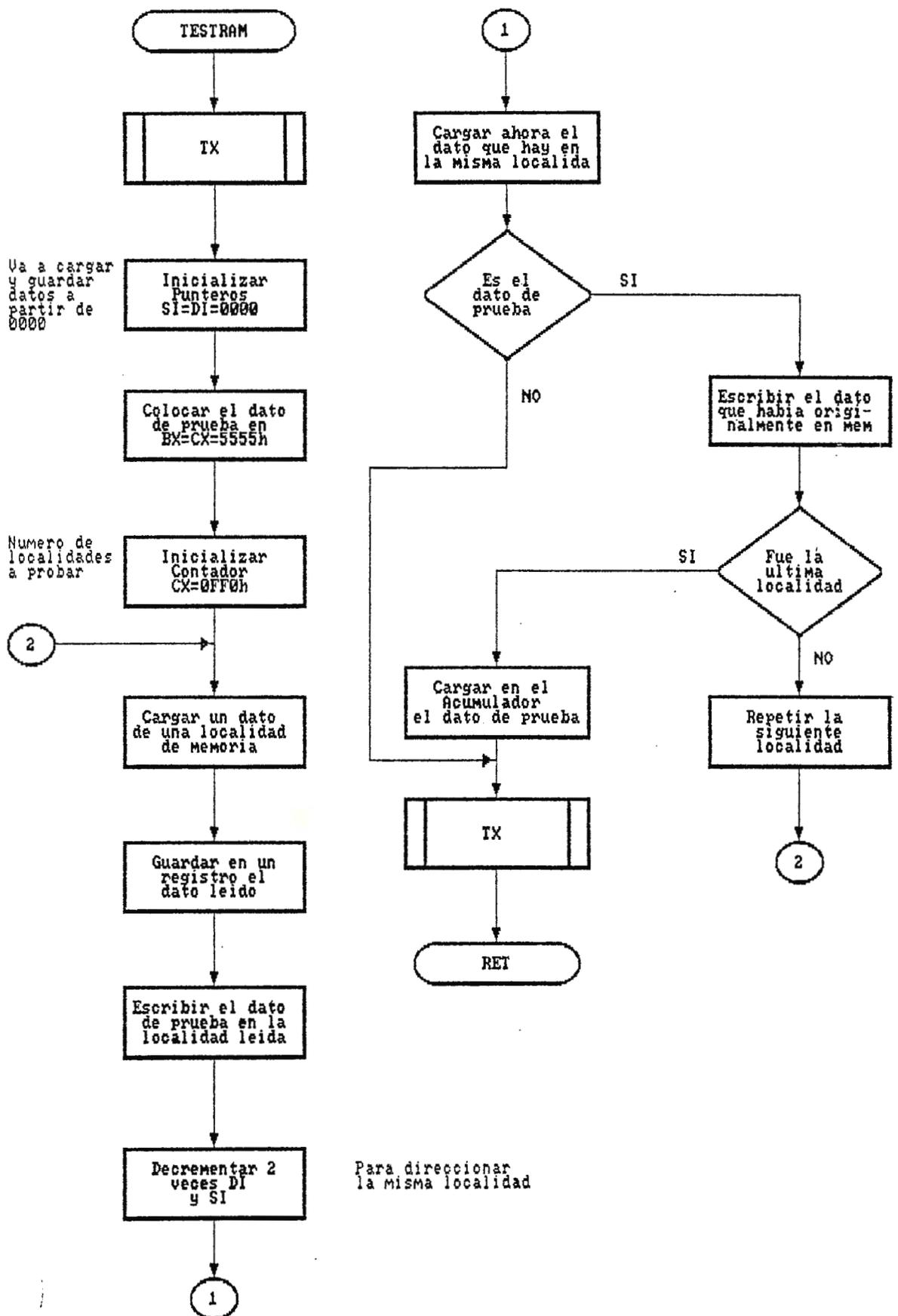


Fig.4.10 Subrutina de prueba de RAM (TESTRAM).

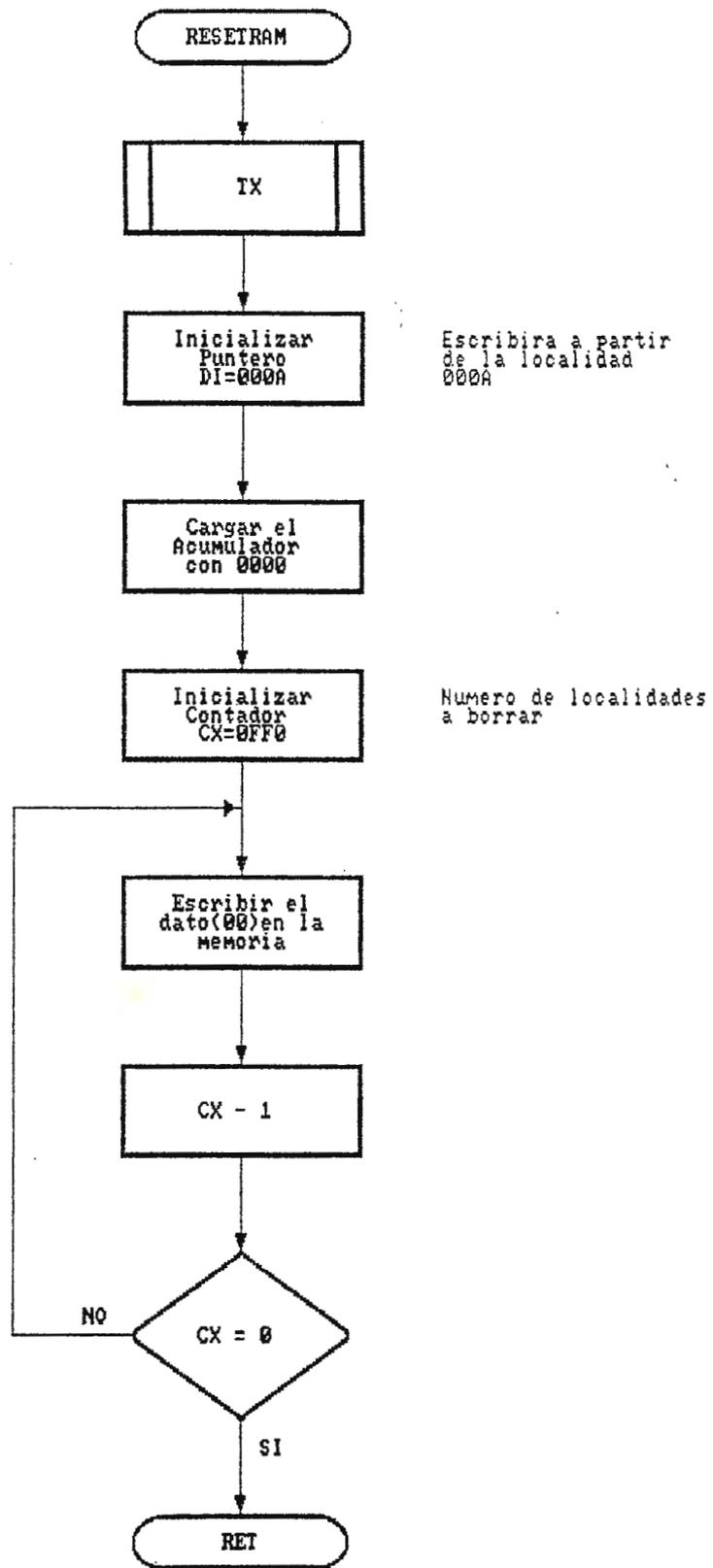


Fig.4.11 Subrutina de borrado de memoria RAM (RESETRAM).

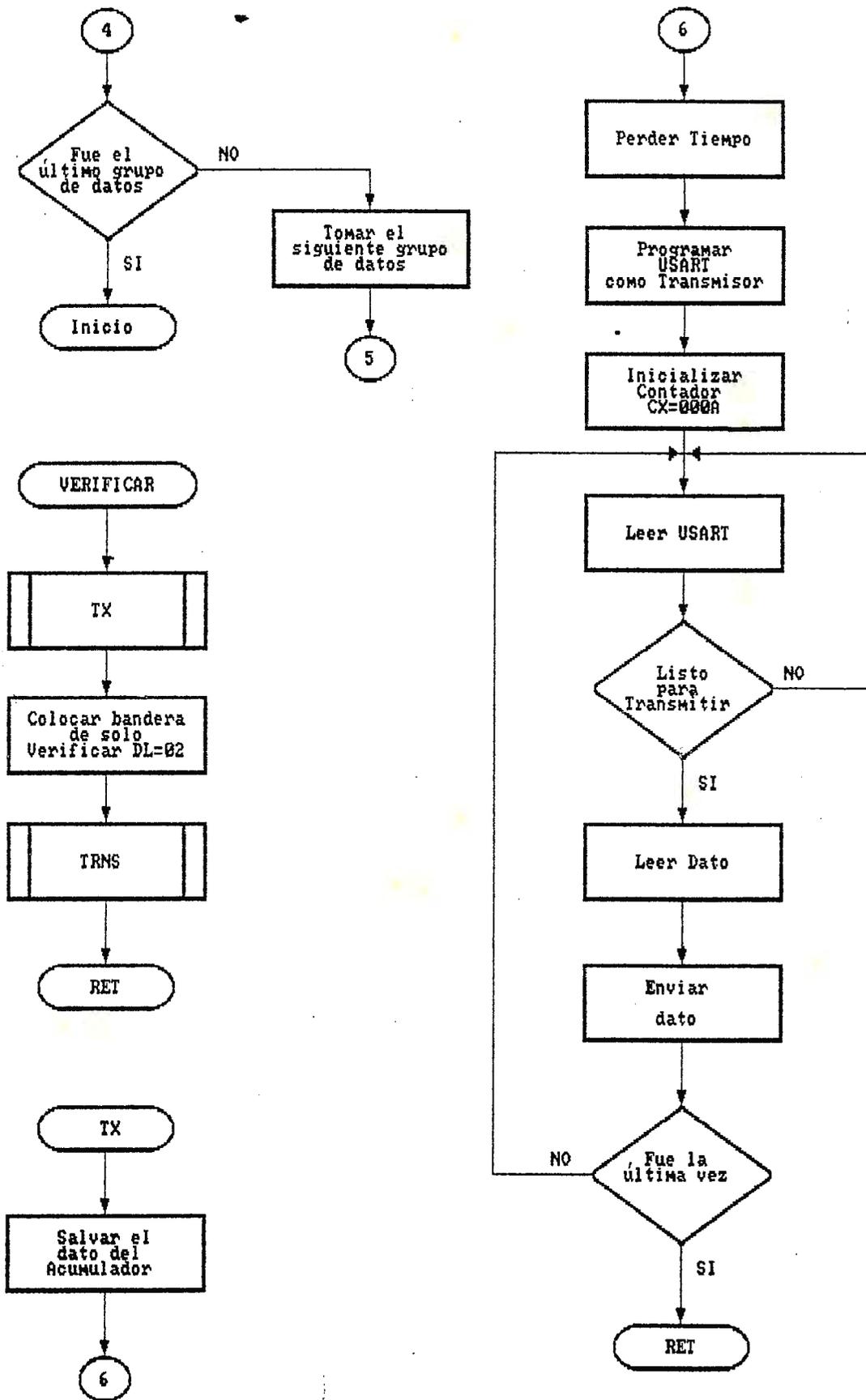


Fig.4.12 Subrutinas de Verificación y Transmisión.

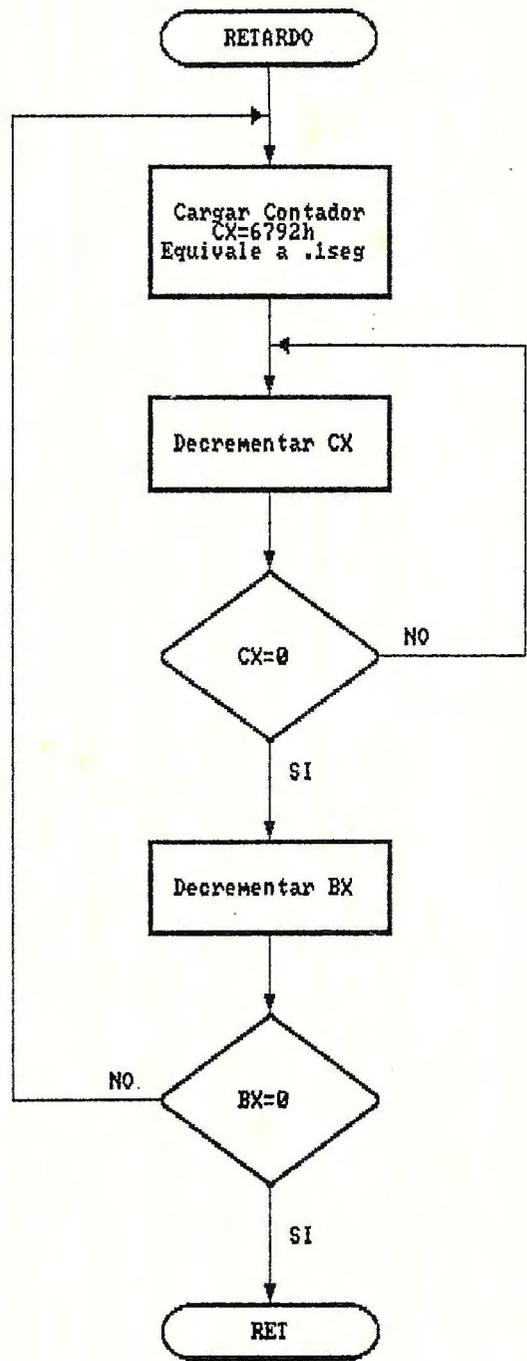


Fig.4.13 Subrutina de Retardo de Tiempo.

CAPITULO 5

DISEÑO DEL SISTEMA DE ADQUISICION DE DATOS DE TIEMPO DIFERIDO MODELO REAL.

Hasta el momento el diseño del prototipo experimental cumple con los requerimientos establecidos (Sección 1.2); pero para someterlo a las condiciones de un pozo geotérmico son necesarios realizar cambios tanto en programación como en el sistema analógico y digital, así que, en este capítulo se presentan los cambios necesarios en el: Programa de la PC, programa monitor y hardware del sistema. Hay que aclarar que tanto el software como el hardware a que se refiere no han sido implementados, de manera que los resultados están sujetos a cambios cuando se efectuen la integración de ellos.

5.1 Software de la PC

El Software de la PC es el que menos cambios produce, ya que todas las funciones anteriores son válidas, solo que ahora el número de variables que el sistema puede manejar se incrementa a seis, de las cuales: temperatura y presión se mantienen y se han agregado otras cuatro más llamadas ALFA, BETA, GAMA y EPSILON, por tanto, esto implica que el operador puede seleccionar cualquiera de estas variables cuando el programa le pregunte: **Cuales variables va a medir**. De esta manera se pueden seleccionar ahora más de dos variables que son igualmente válidas para el proceso. Todas las demás funciones que se encuentran en la sección del Software de la PC se mantienen sin cambios, esto es: tiempo antes de empezar, tiempo entre muestras, etc. Luego el programa que corresponde a la programación de los códigos solamente tendrá esta nueva opción y los restantes menús permanecen iguales. Además el número de códigos ahora aumenta, teniendo que la rutina que se encarga de transmitir los códigos tendrá que aumentar la cantidad de datos de nueve a once (estos códigos se explican con mayor detalle en las secciones siguientes).

El programa que corresponde a la lectura de los datos de la sonda se verá afectado en la cantidad de datos de programación que recibe y como los va a decodificar. Cuando se reciben los datos, éstos son decodificados para que el programa "entienda" de que variables trabajará y realizar las respectivas conversiones de unidades físicas. Como no se han definido cuales son las cuatro variables más que el sistema puede manejar no se explica por el momento en que factores de conversión se trabajarán, pero queda establecido que el sistema si podrá trabajar con mayor cantidad de variables físicas.

Básicamente cada uno de los menús de ambos programas permanecen sin experimentar grandes cambios, esto implica que el programa es dócil para manejar más variables de las establecidas en el prototipo, aunque la presentación de las ventanas que se verán en pantalla tengan que realizarse para las seis variables de trabajo. Es pues más presentación en pantalla que cálculos en el programa, así de esta manera los cambios en el programa de la PC están dirigidos especialmente a como se le presentarán al operador.

5.2 Hardware

5.2.1 Sistema digital.

En lo que se refiere a esta parte de la circuiteria, en donde se comprende directamente al sistema con microprocesador; los cambios a realizar en términos generales son dos:

a) El primero de ellos consiste en sustituir a todos los integrados del sistema por su equivalente en lógica CMOS, buscando de esta manera reducir el consumo de potencia del sistema y hacerlo así portatil por el uso de baterias para su alimentación. De la misma manera dichos integrados deberán estar capacitados de poder funcionar en el rango militar de

temperatura (-85 a +125°C) debido a las condiciones a que estarán sometidas. En cuanto a la estructura del sistema, esta es la misma y en general habrán pequeños cambios en las conexiones cuando la equivalencia entre componentes no es exacta, pero la función que desempeñan no cambia. En anexos se provee la lista de elementos de todo el sistema y se mencionan también en los diagramas de la Fig.5.11 a Fig.5.14.

b) El segundo es un cambio muy importante y se refiere al ADC, debido a que el convertidor analógico-digital empleado en el prototipo presenta dos desventajas difíciles de superar las cuales son: un elevado consumo (disipación) de potencia y la necesidad de 3 voltajes de alimentación +15, -15 y +5V. Así que, para el sistema real, se recomienda usar un nuevo ADC alimentado con +5V y con una baja disipación de potencia y cuyas características se describen en el siguiente numeral. Una característica importante que presenta este ADC es que para dar inicio al ciclo de lectura (o conversión) no requiere de una señal eléctrica sino de una "palabra digital" introducida a través de sus líneas de datos D0 a D5, lo que implica el ahorro de unas compuertas lógicas y la dificultad del programa de adquisición de datos, el cual será explicado en la sección 5.3

5.2.2 Convertidor Analógico-Digital.

El convertidor recomendado es el TLC 1225AM (Fig.5.13), el cual es fabricado por la Texas Instruments y cuyas características son las siguientes:

- **Tipo:** conversión por aproximaciones sucesivas, donde la conversión se toma 27 ciclos de reloj para realizarse, y como necesita de un reloj externo que el fabricante recomienda de 2.6MHz, así el tiempo de conversión viene a ser de 10 μ s.

- **Exactitud:** el error máximo esperado es de $\pm 1\text{LSB}$.
- **Rango de señal de entrada:** esta establecido entre 0 y V_{cc} cuando se alimenta con una sola polaridad, y entre $-V_{cc}$ y $+V_{cc}$ para doble polaridad.
- **Voltaje de alimentación:** 0 a +5V (unipolar); -5 a +5V (bipolar)
- **Salida digital:** 12bits y un bit de signo, señal compatible CMOS y TTL, los 12 bits permiten conservar una alta resolución (0.0244%).
- **Señales de control:** en este sentido difiere del ADC ocupada en el prototipo, porque basta una señal para que el dispositivo tome datos, sino que es necesario colocar una palabra código en 6 de sus líneas de datos para que lo haga. Para ello cuenta con líneas de control tal y como cualquier periférico del sistema (Fig.5.13) su señal de reloj es derivada de la misma que alimenta a la USART, sus líneas \overline{RD} y \overline{WR} son conectadas a \overline{IOR} e \overline{IOWR} de donde se controla como a un periférico y para seleccionar el dispositivo el \overline{CS} se conecta a una compuerta cuya función es: $\overline{CS} = A2 + A3$. El ADC también cuenta con dos señales de salida Ready Out e INT pero para esta aplicación no serán ocupadas.
- **Función de Autocalibración:** este convertidor ocupa una técnica de autocalibración por la cual se calibran automáticamente los capacitores internos que el dispositivo emplea en la conversión. La rutina de operación comienza colocando el código de calibración entre los pines del 15 al 20 del ADC, luego que ha terminado la calibración (ésta toma los ciclos de reloj) se coloca en los mismos pines el código de toma de datos y al final de ésta (toma 27 ciclos de reloj) se leen los 13bits que corresponden al código digital de salida.

5.2.3 Transductores.

Resistencias dependientes de la temperatura (RTD).

Los conductores generalmente exhiben un incremento de resistencias con la temperatura. El cambio en resistencias es dado por el coeficiente de resistencia de temperatura del conductor δ y la resistencia base de un sensor conductivo de temperatura (usualmente es una resistencia a 0°C y algunas veces en otro temperatura especificada) que esta dada por el área transversal (o calibre de un conductor) y la resistividad σ del material.

Para medir temperatura se necesita tener un conductor el cual tenga:

- (1) una razonablemente alta α , de modo que se pueda obtener un apreciable cambio de resistencia.
- (2) una α que permanezca constante sobre un amplio rango de temperatura, así que la característica de resistencia vrs temperatura se acerque lo más posible a una relación lineal; y
- (3) una razonablemente alta resistividad, para minimizar la cantidad de material requerido para fabricarlo.

Una relación aproximada para la característica de resistencia vrs temperatura de los conductores entre 0 y 100°C esta dada por la ecuación:

$$R_t = R_0(1 + \alpha t) \quad \text{donde: } R_t:\text{resistencia en temperatura } ^{\circ}\text{C}$$

R_0 :resistencia 0°C

α :coeficiente de resistencia

Las resistencias de platino se reconocen por su precisión en sensores de temperatura. Es posible encontrarlos para operar en rangos de -250°C hasta 850°C .

Las RTD de platino están disponibles en rangos de decenas de ohmios hasta kilohmios con un coeficiente de temperatura de aproximadamente

0.4%/oC del valor de resistencia en 25°C.

La relación R vrs T de un elemento de platino entre -183 y +630°C esta dado por la ecuación de Callendar-Van Dusen:

$$R_t = R_0 + R_0\alpha[t - \delta(0.01t - 1)(0.01t) - \beta(0.01t - 1)(0.01t)^3]$$

donde:

R_t : resistencia en temperatura t, °C.

R_0 : resistencia en 0°C.

α , β y δ son constantes; α es usualmente determinada midiendo la resistencia del elemento en 100°C y β de medir la resistencia bajo 0°C, usualmente a -182.96°C; δ es determinada midiendo bien arriba de los 100°C, tal como a 444.7°C. Valores típicos son los siguientes:

$$\alpha = 0,00392$$

$$\beta = 0 \text{ [si } t \text{ es positivo]} \text{ y } 0,11 \text{ [si } t \text{ es negativo]}$$

$$\delta = 1,49$$

El tipo de transductor recomendado es un RTD de platino de 100Ω de resistencia base y con $\alpha=0,00385$ tipo tres hilos.

Ventajas:

- Alta exactitud
- Poca no linealidad.
- Amplio rango de operación.
- No hay problemas con termovoltajes parásitos.

Transductores de presión tipo Strain-gage

La conversión de cambios de presión en cambios de resistencias, debido al estiramiento(Strain) en 2 o mucho más comúnmente, 4 brazos de un puente de Wheatstone ha sido usado en transductores de presión por muchos años. El sensor, bajo compresión decrece su resistencia mientras

que el sensor sometido a tensión incrementa su resistencia, y su conexión en el circuito puente es tal que los cambios de resistencias se suman, produciendo así un voltaje de salida más grande. Esta acción es típica para la mayoría de transductores "Strain gage" para varias medidas. El montaje usual de los "Strain gage" es que una pieza les aplica una fuerza cuando un diafragma al que éste está unido es sometido a una presión. En los transductores de presión absoluta la cavidad del diafragma debe ser evacuada y sellada. Algunos diseños tienen un encapsulado más grande para permitir alojar circuitos de acondicionamiento de señal y amplificación.

El transductor de presión seleccionado para el sistema de adquisición de datos está basado en el anterior principio, y sus características principales son:

- Rango de medida: 0 a 1000PSI (sobre presión atmosférica)
- Alimentación: 5Vdc y 15mA
- Salida máxima: 10mV (fondo de escala)
- Buenas características de repetitividad: $\pm 0.1\%$ de fondo de escala y aceptable presión ($\pm 1\%$)

5.2.4 Sistema Analógico. Diseño

La parte de acondicionamiento de señal del sistema de adquisición de datos es el que más cambios ha experimentado, debido que para este caso se ha diseñado en base a los transductores que corresponden realmente al sistema, así es que por eso se cubrirá esta sección nuevamente en forma completa como descripción y diseño. Para empezar, el circuito básico de acondicionamiento de señal que se empleará para las variables a medir, en este caso presión y temperatura, es el mostrado en la Fig.5.1, el cual se define como un amplificador de instrumentación que

consiste en montaje de 3 amplificadores operacionales que proporcionan una amplificación a la diferencia entre las señales de entrada V_1 y V_2 mediante la ecuación general:

$$V_o = G(V_1 - V_2)$$

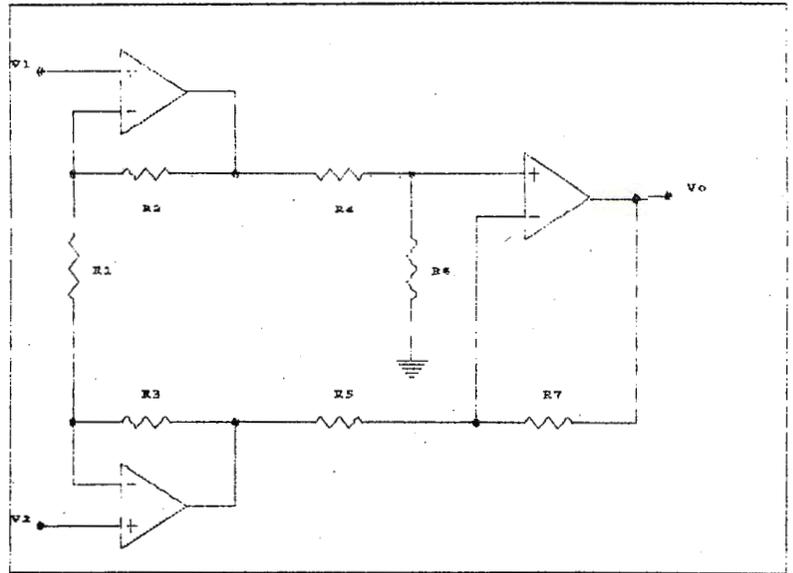


Fig.5.1 Amplificador de instrumentación.

Una cualidad importante de este montaje es que mejora las características de un montaje de amplificador diferencial básico (con un solo operacional), ya que, en este caso se presenta una impedancia de entrada muy alta para las dos señales de entrada y se puede conseguir una alta ganancia sin afectar demasiado la señal de salida debido a la corriente de offset. Por otra parte en cuanto a las causas de error típicas como: el voltaje offset, la variación por temperatura y otras, siempre están en función de la calidad del dispositivo. Es por eso que en la selección de los op amp se tomaron en cuenta estas características:

- (a) Pequeños voltajes y corrientes de offset.
- (b) Pequeña corriente de polarización de entrada.
- (c) Pequeños desplazamientos debido a la temperatura.
- (d) Bajo consumo de potencia.
- (e) Rango de temperatura tipo militar.

Para el circuito de la Fig.5.1, haciendo $R_2=R_3$ y $R_4=R_5=R_6=R_7$, la fórmula para la ganancia se reduce así:

$$G = 1 + 2R_2/R_1$$

Al respecto del circuito formal de la Fig.5.1, dos notas importantes de éste nivel general son:

- a. A diferencia del diseño para el prototipo en este no existe un amplificador de salida común a todos los canales, sino que el MUX esta directamente conectado al convertidor, por lo demás la función del MUX es la misma.
- b. Debido a que el sistema solo emplea un voltaje positivo (+5V) de alimentación, se ha cambiado la referencia del amplificador de instrumentación a +2.5V, que es la mitad del rango de aceptación del ADC y alrededor de 2.5 se moverán los valores de salida de los amplificadores, esto es, con el fin de evitar las no linealidades de la zona de saturación de los op amp (cerca de 0V).

Diseño sección de Temperatura

El montaje a utilizar para transferir a señal de voltaje los cambios en resistencia de la RTD, es del tipo puente de Wheatstone como se observa en la Fig.5.2:

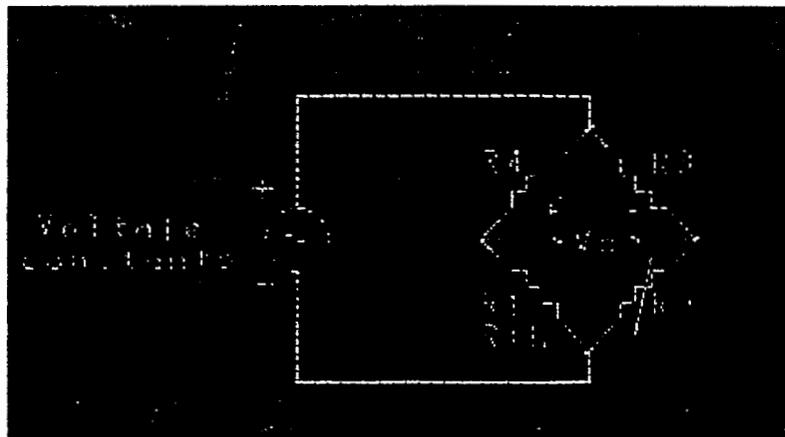


Fig.5.2 Montaje del transductor de temperatura RTD en el puente de Wheatstone.

a) tomando a $R_3=R_4$ y si I_t para 0°C toma el valor de 1mA

$$V1_{(0^\circ\text{C})} = 1\text{mA} * 100\Omega = 0.1\text{V}$$

$$\text{entonces: } R_4 = (2.5\text{V} - 0.1\text{V})/1\text{mA} = 2.4\text{K}\Omega$$

b) la ecuación de transferencia de puente es:

$$V_o = (R_1/R_1+R_4 - R_2/R_3+R_2)V_{in} \quad \text{si } R_1=R_2=100\Omega \text{ y } R_3=R_4=2.4\text{K}\Omega$$

$$V_o = V_{in}/250 (1-1/250 / 1+x/250)x$$

Esta función produce un rango de V_o de 0 a 133.3mV para un rango de 0 a 400°C . Pero desplazando el punto de equilibrio a la mitad del rango, es decir haciendo $V_o = \pm 66.66\text{mV}$. Para medio rango $V_2=166.66\text{mV}$, entonces se encuentra R_2 :

$$166.66\text{mV}/2.5\text{V} = R_2/2.4\text{K}\Omega+R_2$$

de donde $R_2=171.42\Omega$

De esta manera la nueva ecuación de transferencia viene a ser:

$$V_o = V_{in}Q(1-Q/1+Qx)x$$

donde Q es constante e igual a: $Q = R_2/(R_2+R_3) = 0.0666635$

y el rango de salida resultante es de:

$$V_o = V_1 - V_2 = \pm 66.66\text{mV}$$

c) amplificador de instrumentación:

Si $R_5=R_4=R_7=R_6$ (en la Fig.5.14) y $R_4=10\text{K}\Omega$ (para no cargar demasiado al op IC30) y también $R_2=R_3=10\text{K}\Omega$. Encontrar R_1 de la ecuación:

$$G = 1 + 2R_2/R_1$$

la ganancia que se necesita en el amplificador viene dado por el rango de voltaje de salida entre el rango del voltaje de entrada:

$$G = \delta V_o / \delta(V_1-V_2) = 3\text{V}/133.32\text{mV} = 22.5$$

y entonces $R_1=930.23\Omega$, esto nos conduce a utilizar un potenciómetro de precisión de $1\text{K}\Omega$. La etapa es complementada por condensadores en la alimentación del puente, en la entrada al amplificador y en los operacio-

nales para mantener al sistema libre de ruido. La ecuación de transferencia completa de la sección de temperatura resulta ser la siguiente:

$$V_o = 3.75 (0.933/1+0.066x)x \quad \text{donde } x=Rt^0/R_{(1880\text{C})}$$

Diseño sección de presión:

a) la característica eléctrica del transductor de presión es que proporciona una salida diferencial de voltaje de bajo nivel (0 a 10mV), la cual debe ser amplificada a través de un amplificador de instrumentación, tal como el descrito arriba y para ello se debió hacer un arreglo especial a la entrada (Fig.5.3).

La salida negativa del transductor se conecta a la referencia de rango medio del sistema (2.5V) de forma que en V1 se obtenga un voltaje de salida con respecto y por sobre la referencia. La red de R1, R2 y R3 forman

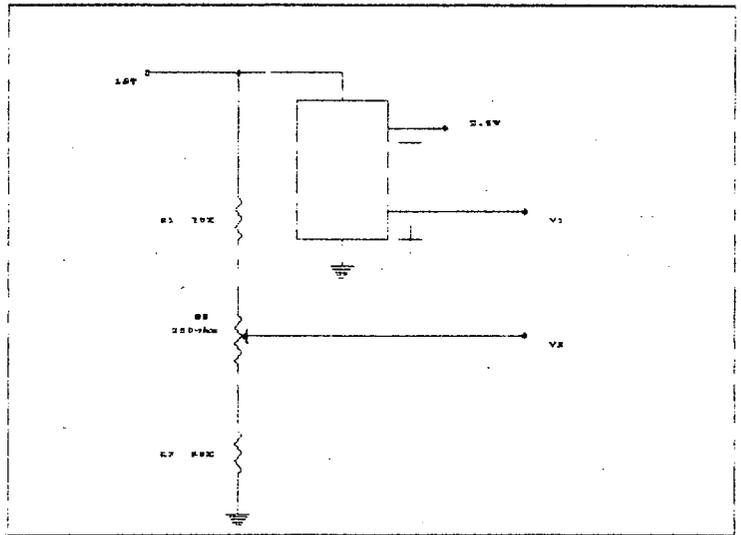


Fig.5.3 Arreglo de la etapa de entrada de la sección de presión.

el ajuste a "cero" del sistema, proporcionando el voltaje de entrada "negativo" al amplificador de instrumentación en su entrada inversora. El valor de las resistencias de la red son escogidas de manera de no cargar a la fuente y proporcionar un ajuste de V2 dentro del rango de variación de V1. En realidad el ajuste debe llamarse de "rango medio" porque lo que se quiere es que cuando la presión medida sea 500PSI, V1 deberá valer $2.5V + 5mV = 2.505V$; y entonces V2 debe tomar este mismo valor para que el voltaje diferencial de entrada al amplificador sea cero y la salida hacia el ADC

sea el voltaje de referencia=2.5V (valor medio del rango de aceptación del ADC) y de esta manera cuando la presión fuere menor a 500PSI, (V1-V2) tomará un valor negativo que producirá una salida menor a 2.5V, y por el contrario, si la presión fuere mayor a 500PSI, (V1-V2) será positivo y la salida hacia el ADC será mayor a 2.5V.

b) amplificador de instrumentación. Este en sus entradas "observará" una tensión diferencial en el rango de -5mV a +5mV, y en su salida deberá correspondientemente proporcionar un voltaje en el rango de 1V a 4V ó de otra manera (2.5V \pm 1.5V). Por ello la ganancia de la etapa deberá ser:

$$G = \delta V_o / \delta (V_1 - V_2) = 3V/10mV = 300$$

Al igual que para temperatura R4=R5=R6=R7=10K Ω . En el caso de R2=R3 debido a que la ganancia exigida para este caso el mayor tomarán el valor de 100K Ω y de nuevo el ajuste es através de R1. Usando la ecuación de ganancia tenemos:

$$G = 1 + 2R_2/R_1$$

$$R_1 = 2R_2/G - 1 = 2(100K\Omega)/300 - 1 = 668.9\Omega$$

para el caso se usará un potenciómetro de precisión de 1K Ω .

c) al igual que en la etapa de temperatura, se han colocado condensadores para eliminar el ruido.

d) la ecuación de transferencia de la sección de presión es la siguiente:

$$V_o = G(V_1 - V_2) + V_{ref} \quad \text{y si } V_1 - V_2 = 10^{-5}P - 5 \times 10^{-3}$$

$$V_o = 300(10^{-5}P - 5 \times 10^{-3}) + V_{ref} \quad P = \text{PSI}$$

$$V_o = 3 \times 10^{-3}P + 1$$

Sección de voltajes de referencia.

Como se puede observar en la Fig.5.14, esta parte solo ha sufrido pequeños cambios, tales como, que se han eliminado el buffer de salida porque la salida del MUX y del ADC cuya impedancia de entrada es muy

alta. Debido a que hoy no hay amplificación, los voltajes son directamente tomados de una referencia de 5V y la red de 8 resistencias hacen que se obtengan voltajes desde 0 hasta 5V en pasos de 0.625V.

Sección de prueba de la batería.

Esta simplemente consiste en un divisor de tensión formado por las resistencias R21 y R22 en la Fig.5.14. Cuando la batería tiene carga completa el voltaje de salida es:

$$V_o = (R/R+R)V_{batt} = (100/450)7.5 = 1.66V$$

A medida que se descargue la batería con el uso, el voltaje resultante será menor y el sistema decidirá el resultado a través del software.

5.2.5 Proceso de Calibración

(1) Parte de temperatura.

- a) Ajuste de media escala, aplicando una temperatura de 188°C, se ajusta RV1 para que a la salida de IC30 se obtengan 2.5V.
- b) Ajuste de ganancia. Para cero grados se ajusta RV2 hasta obtener a la salida del IC30 un valor de 1.0V.

(2) Parte de presión.

- a) Ajuste de media escala. Aplicando una presión de 500PSI se ajusta RV4 para que a la salida de IC33 se obtengan 2.5V.
- b) Ajuste de ganancia. A presión atmosférica se ajusta RV5 para que a la salida del IC33 se obtenga una lectura de 1.0V.

(3) Comprobación a través de una computadora.

Se trata de probar al sistema para varias presiones y temperaturas conocidas. Es en la computadora donde los datos digitales se operan a través de las ecuaciones de transferencia, tanto del circuito como del transductor. Es entonces de esta manera como se comprueban dichas

ecuaciones y si es necesario hacer algún cambio se hará en éstas precisamente.

(4) Obtener una relación de variación

Esta relación es con respecto a la temperatura de la lectura del transductor de presión y compensar por software.

5.3 Análisis del consumo de potencia.

En este apartado se presentará una tabla de datos (Tabla 5.1), en la que se presenta el consumo de potencia por dispositivo. El análisis se basa además en el tiempo de trabajo de cada dispositivo, es decir, la fracción de tiempo que opera el elemento (para algunos casos). Este tiempo de trabajo es obtenido en base a lo que el sistema exige: tal es el caso del microprocesador, que trabaja todo el tiempo a diferencia de los otros sistemas donde hay momentos que éste queda en estado de espera. De las memorias es la ROM la que trabaja mayor tiempo, ya que, todo el sistema sigue el programa grabado en ella; en cambio en la RAM solo de vez en cuando se lee y escriben datos en ella. De la misma manera de los periféricos el más usado es la USART debido a que en la rutina de "descanso" del sistema llamado ESTADO1 continuamente se accesa este dispositivo. En cambio el PPI y el ADC se activa mucho menos frecuentemente. El resto de los elementos consumen un promedio constante de potencia todo el tiempo.

Tabla 5.1 Consumo de Potencia de los elementos de la sonda.

ELEMENTO DEL CIRCUITO	CONSUMO DE CORRIENTE		CANTIDAD DE ELEMENTOS	TIEMPO DE TRABAJO	TOTAL (mA)
	ACTIVO mA	STANDBY			
MICROPROCESADOR 80C88	20	0.2	1	100%	20
GENERADOR DE RELOJ 82C84A	6	6	1	100%	6
TRANSCIVER (BUS DE DATOS) 82C86	0.5	0.004	1	100%	0.5
LATCHES (BUS DE DIRECCIONES) 74C373	1.2	---	2	100%	2.4
MEMORIA RAM 68L256A-76	3	3	2	2%	6
MEMORIA EPROM 27C16	8	0.04	1	30%	2.5
COMPUERTAS OR MC14871BAL	0.37	---	3	100%	1.11
COMPUERTA NOT MC14849BAL	2.18	---	1	100%	2.18
PUERTO PARALELO PPI 82C55	4	0.004	1	2%	0.004
PUERTO SERIE USART 82C51	1	0.004	1	15%	0.16

Continuacion de la Tabla 5.1

ELEMENTO DEL CIRCUITO	CONSUMO DE CORRIENTE		CANTIDAD DE ELEMENTOS	TIEMPO DE TRABAJO	TOTAL (mA)
	ACTIVO	mA STANDBY			
CONVERTIDOR ANALOG/DIGITAL TLC1225AM	13	6	1	2%	6.3
DIVISORES DE FRECUENCIA MC14160BAL	0.4325	----	4	100%	2.18
DECODIFICADOR MC14076BAL	0.145	----	1	100%	0.145
ALIMENTACION +5V LM2935	3	----	1	100%	0.29
MULTIPLEXORES ANALOGICOS MC14051BAL	0.145	----	2	100%	0.29
AMPLIFICADORES OPERACIONALES LM108AH	0.3	----	6	100%	1.8
TRANSDUCTOR DE PRESION	15	----	1	100%	15
REGULADOR +5V MC1504US	1.25	----	2	100%	2.5
REGULADOR +2.5V MC1503	2.2	----	1	100%	2.2
REGULADOR +2.5V LM336Z2.5	1.2	----	1	100%	1.2
CIRCUITERIA	3	----	1	100%	3
				TOTAL:	78.6 mA

Si el sistema se alimenta con 7.5V el consumo total de potencia resulta ser:

$$P = (7.5)(78.6\text{mA}) = 0.60\text{W}$$

5.4 Sugerencias

a. **Tierras** : en lo que se refiere al impreso, la parte analógica debe tener una red de líneas de tierra lo suficientemente amplia para reducir el ruido inducido y las caídas de potencial.

b. **Impreso Op-Amp** : el fabricante recomienda limpiar muy bien el impreso y protegerlo contra la contaminación, lo mismo para evitar corrientes de fuga provee los pines "no ocupados" como blindaje . La línea de blindaje se conecta a un potencial similar al de las entradas, en este caso sería el potencial de referencia +2.5V (catodo LM336 Z2.5).

c. **Lineas que conducen señales de bajo nivel**: se recomienda que sean lo más cortas posibles para evitar inducciones y corrientes de fuga.

d. Un detalle muy importante que no se había expuesto es con respecto a al transductor de presión seleccionado, el cual si bien es cierto que está especificado a trabajar en el rango de presión requerido de 0 a 1000PSI, su temperatura de operación por el contrario no se ajusta a aquellos que en realidad estará sometido, por lo que es absolutamente necesario hacer un acople por medio del cual el fluido en contacto con el transductor debe estar a una temperatura más baja que la del medio exterior, pero la presión debe ser transferida por completo.

e. Una sugerencia más con respecto a los transductores es acerca de su protección contra la corrosión del medio; ambos estan protegidos, pero dependiendo del grado de corrosión sería conveniente protegerlos adecuadamente de alguna manera sin que con ello se pierdan las características de conductividad térmica (para la RTD) y de presión. La cubierta original del RTD es de acero inoxidable 304.

f. Selección de las baterías :

Las baterías están divididas en dos clases básicas:

- a) Primarias, no recargables.
- b) Secundarias, recargables.

Dentro de las del primer tipo, las más aceptables son las Alkalinas con dióxido de magnesio (MnO_2), las cuales aunque poseen una limitada capacidad por tamaño tienen una amplia variedad de tamaños y capacidades, tiene una curva de descarga en pendiente lo que posibilita en cierta medida conocer cuantitativamente el grado de carga de la batería. Un modelo comercial de este tipo de batería garantiza 20Hrs. máximo de uso continuo del sistema.

Con las baterías primarias existe el problema de que no son recargables y una vez descargadas es necesario reemplazarlas, resulta aún peor cuando dado el requerimiento del sistema se tuvieran que obtener las baterías del extranjero resultaría un gasto de mantenimiento elevado; por ello lo que resulta más práctico es usar baterías recargables, de las cuales, las que son más recomendables son las de Níquel-Cadmio cuyas principales características son:

curva de descarga en flanco (se mantienen en un máximo el mayor tiempo posible y luego cae rápidamente el voltaje) y pueden trabajar a altas temperaturas. Presentan las desventajas de como la gran mayoría de baterías recargables tiene una baja densidad de energía, además el voltaje por celda es de 1.2V por lo que se necesitarán 6 unidades para sufrir la demanda del circuito dando de esta manera la suma de 7.2V.

g. Enfriamiento del sistema : como se mostró en el análisis de potencia del sistema, éste disipa un total de 0.6W, los cuales en un espacio cerrado producirían un lento pero constante aumento de la

temperatura del medio. Una solución puede ser que en vista que el sistema no será absolutamente sellado; puesto que debe destaparse para conectar el cable a la computadora, lo mismo que para reemplazar la batería, deberá tener conductos de ventilación para que cuando este en la superficie se evacue el calor producido por el sistema. Además la posibilidad de un enfriamiento interno más sofisticado; pero todo lo anterior está más allá del alcance de este documento.

5.5 Cambios en el programa que gobierna el sistema de adquisición de datos (Programa Monitor).

El esquema de operación utilizado en el prototipo se mantiene igual; constando de una rutina madre, 7 subrutinas principales y 2 subrutinas auxiliares. Los cambios se refieren a:

- a) la adquisición de datos debido al cambio del convertidor analogo-digital esto se da en las subrutinas de Adquisición y Calibración.
- b) expansión de la adquisición de datos, básicamente se faculta al sistema de poder hacer mediciones de 6 variables físicas (4 más aparte de los canales usados para presión y temperatura).
- c) Mejoras en las subrutinas de Prueba de RAM y borrado de la RAM.

Por lo anterior se proveen los flujogramas correspondientes a los cambios efectuados, así como una recopilación completa del programa a nivel de Assembler (Nemónicos), que son los siguientes Adquisición, Calibración, TESTRAM y RESETRAM.

ADQUISICION PROC NEAR

```
CALL TX           ;LLAMAR A TRANSMITIR
MOV BX,[0001]    ;CARGAR EL RETARDO
CALL RETARDO     ;
MOV DI,001AH     ;
CLD              ;
```

```

10:      MOV BX,00          ;LIMPIAR REGISTRO
        MOV CX,0EH        ;AJUSTAR CONTADOR
REPETIR: MOV [CX]+[000A],BX ;
        LOOP REPETIR
        MOV CX,BX         ;
        MOV DH,00        ;LIMPIAR CONTADOR
20:      MOV DL,00        ;COLOCAR COD.TEMPERATURA
        MOV AL,89H       ;PROGRAMAR PPI
        OUT 4BH,AL       ;PUERTO B SALIDA
        MOV AL,10H       ;CARGAR COD.CALIBRACION
        OUT 41H,AL       ;ENVIARLO POR PUERTO B
        MOV AL,DL        ;CARGAR COD. DEL CANAL
        OUT 48H,AL       ;ENVIARLO POR PUERTO A
        MOV BX,01H       ;PERDER TIEMPO
        CALL RETARDO     ;DE 0.1SEG
        MOV AL,2EH       ;CARGAR COD. CONVERSION
        OUT 41H,AL       ;ENVIARLO POR PUERTO B
        MOV CX,05        ;PERDER TIEMPO
TIEMPO:  LOOP TIEMPO     ;
        MOV AL,8BH       ;PROGRAMAR PPI
        OUT 4BH,AL       ;PUERTO B ENTRADA
        IN AX,41H        ;LEER EL DATO
        INC DH           ;INCREMENTAR DH
        MOV BL,[0005]    ;CARGAR COD. MUESTREO
        ADD [DL]+[000D],AX ;
        CMP DH,[0006]    ;YA TOMO TODAS LAS MUESTRAS
        JZ 60            ;
        MOV BX,0001      ;
        MOV CX,0005      ;
110:     INC BH          ;
        ROL BL           ;
        MOV AL,BL        ;VA A TOMAR
        AND AL,[0000]    ;DATOS DE
        JZ 90            ;PRESION
        CMP DH,DL        ;TOCA TOMAR
        JG 100           ;PRESION
90:      LOOP 110        ;
        MOV BX,000AH     ;
        CALL RETARDO     ;
        JMP 20           ;
100:     INC DL          ;
        JMP 30           ;
60:      MOV BL,[0005]   ;
        JZ 50            ;
        MOV DX,0000      ;
        MOV BX,0001      ;
        MOV CX,0006      ;
120:     MOV AL,BL       ;
        AND AL,[0000]    ;
        JZ 70            ;
        MOV AX,[DX]+[000D] ;
        STOSW           ;
        INC DX           ;

```



```

OUT 41H,AL      ;
IN AX,41H      ;
ADD DX,AX      ;
LOOP 17        ;
XCHG DX,AX     ;
STOSW         ;
ADD BH,10H     ;
MOV AL,BH      ;
CMP BH,82H     ;
JNZ 16         ;
MOV AL,89H     ;
OUT 48H,AL     ;
MOV AL,10H     ;
OUT 41H,AL     ;
MOV AL,07H     ;
OUT 48H,AL     ;
MOV BX,01H     ;
CALL RETARDO  ;
MOV AL,2EH     ;
OUT 41H,AL     ;
MOV CX,05H     ;
D:            ;
LOOP D         ;
MOV AL,8BH     ;
OUT 4BH,AL     ;
IN AX,41H     ;
STOSW         ;
MOV DI,0019H   ;
MOV AX,0012H   ;
MOV [0007],AX  ;
CALL TRNSII   ;
RET           ;

```

CALIBRACION ENDP

TESTRAM PROC NEAR

```

CALL TX      ;
MOV DI,00    ;
MOV BX,DI    ;
CLD         ;
MOV CX,FF80H ;
21:         ;
MOV AX,DI    ;
STOSB       ;
PUSH DI     ;
AND DI,1FFFH ;
CMP DI,1FF0H ;
JZ 20       ;
POP DI      ;
LOOP 21     ;
20:         ;
INC BH      ;
POP DI      ;
MOV DI,BX   ;

```

```

                CMP BX,8000H      ;
                JNZ 21            ;
                MOV SI,00        ; INICIALIZAR
                MOV DI,SI        ; PUNTEROS
                MOV BX,SI        ;
                CLD              ; AJUSTE AUTOMATICO
                MOV CX,FF80H     ; PREPARAR CONTADOR
                LODSB            ; CARGAR DE MEMORIA
25:             CMP AX,DI        ; DATO=DI
                JNZ 23          ; SON IGUALES?
                INC DI          ; DI=DI+1
                PUSH DI         ; SALVAR DI EN LA PILA
                AND DI,1FFFH    ; ES LA
                CMP DI,1FF0H    ; ULTIMA LOCALIDAD
                JZ 24           ; DE RAM
                POP DI          ; RECUPERAR DI
                LOOP 25         ;
24:             POP DI          ; RECUPERAR DI
                INC DH          ; INCREMENTAR DH
                MOV SI,BX       ; HABILITAR
                MOV DI,BX       ; SIGUIENTE RAM
                CMP BX,8000H    ; FUE LA ULTIMA
                JNZ 25          ; RAM
                MOV AX,CFF5H    ; CARGAR COD. OK
23:             CALL TX         ; TRANSMITIR LSB
                MOV AL,AH       ; PREPARAR MSB
                CALL TX         ; TRANSMITIR MSB
                RET             ; RETORNO

```

TESTRAM ENDP

RESETRAM PROC NEAR

```

                CALL TX         ;
                MOV DI,F000H    ; INICIALIZAR PUNTERO
                CLD             ;
                MOV CX,08H     ; CARGAR NUMERO DE RAMS
27:             MOV SI,DI       ; BUSCAR EN
                MOV AL,SS      ; RAM MAS ALTA
                STOSW         ;
                LODSB         ;
                CMP AL,55     ; EXISTE RAM
                JZ 26         ;
                SUB DI,2002H   ; ES LA ULTIMA
                LOOP 27       ; RAM
                RET           ; RETORNE
26:             ADD DI,0FE0H   ; PREPARAR CONTADOR
                MOV CX,DI     ; CARGAR CONTADOR
                MOV DI,001AH   ; PREPARAR REGISTRO
                MOV AL,00H     ; LIMPIAR ACUMULADOR
28:             STOSB         ; ESCRIBIR EN RAM
                LOOP 28       ; CONTINUAR HASTA TERMINAR

```

RET

;RETORNE

RESETRAM ENDP

NOTAS: Cambios en la programación: # de datos a recibir es de 110=CEH --> CX
 Número de veces a sacar mayoría: 11=OB
 Cambios en la transmisión: datos a enviar por TRNSII es de 11, que van de 00000 a 0000A
 En TRNSII en vez de INC DI, cargar DI con 001A.
 Al final comparar DI-001A con M(7) y M(8).

5.5.1 Rutina de adquisición de datos.

En la Tabla 5.2 se presenta en donde se colocan las memorias que el programa emplea tanto para guardar los códigos de programación como para realizar operaciones internas, y en las figuras 5.4 a 5.7 los flujogramas correspondientes.

Tabla 5.2 Resumen de las memorias utilizadas en el sistema.

Símbolo	Localidad	Descripción								
M(0)	00000	Código de la variable a medir : <table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> bit si es "1" el bit es válido bit 0: medir temperatura bit 1: medir presión bit 2: medir δ bit 3: medir σ bit 4: medir ω bit 5: medir Φ	7	6	5	4	3	2	1	0
7	6	5	4	3	2	1	0			
M(1) y M(2)	00001 y 00002	Código del tiempo antes de la 1ª toma de datos. M(1)LSB M(2)MSB								
M(3) y M(4)	00003 y 00004	Código del tiempo entre estaciones M(3) LSB, M(4) MSB								
M(5)	00005	Código de la obtención de datos Si es "0" es simple Si es "1" es promedio								
M(6)	00006	Número total de muestras								
M(7) y M(8)	00007 y 00008	Número total de datos								

Continuación de la Tabla 5.2

M(9) y M(10)	00009 y 0000A	Número de estaciones después de las cuales es necesaria una compensación en el retardo de tiempo entre estaciones.
M(11) y M(12)	0000B y 0000C	Conteo de las estaciones para determinar cuando debe hacerse la compensación en el retardo de tiempo entre estaciones.
M(13) y M(14)	0000D y 0000E	Suma de datos de temperatura.
M(15) y M(16)	0000F y 00010	Suma de datos de presión.
M(17) y M(18)	00011 y 00012	Suma de datos de δ .
M(19) y M(20)	00013 y 00014	Suma de datos de σ .
M(21) y M(22)	00015 y 00016	Suma de datos de ω .
M(23) y M(24)	00017 y 00018	Suma de datos de Φ .

a) Parte de adquisición de datos.

- 1) Se programa al puerto B del PPI como salida.
- 2) Se envía el código de autocalibración al ADC por medio del puerto B del PPI.
- 3) Se pierde tiempo en espera a que el ADC termine de autocalibrarse.
- 4) Se envía el código de inicio de conversión al ADC a través del puerto B del PPI.
- 5) Se pierde tiempo en espera a que el ADC termine de convertir el dato.
- 6) Se programa el puerto B del PPI como entrada.

7) Se lee el dato digital de salida del ADC a través de los puertos B y C del PPI.

b) Códigos de programación.

El único de ellos que ha cambiado es el de las variables a medir, ya que en el prototipo solo se podían leer dos variables, mientras que ahora se ha capacitado al sistema para leer hasta 6 variables. En aquella ocasión para determinar si se leería una u otra variable, bastaba una simple decisión, mientras que en este nuevo diseño como se puede ver en el flujograma, son necesarios casi tantas decisiones como variables a medir el sistema. Así pues durante el proceso de datos en cada estación se toman sucesivamente los datos programados, llevándose el control de los mismos a través del registro DL del microprocesador. Siempre que antes de hacer la toma de datos y en el momento de almacenar los resultados (cuando se ha hecho sumatoria) se consulta la memoria M(0) para decidir si el sistema debe tomar o si tomo los datos de cada una de las variables.

c) Compensación de tiempos.

Debido al tiempo que toma ejecutar el programa de adquisición, en el sistema se producen retardos indeseables que repetidos una gran cantidad de veces producirían graves errores en los tiempo de la toma de datos. Es por ello que se tuvo la necesidad de hacer una compensación en el retardo de tiempo entre datos del sistema. El criterio funciona de la siguiente manera: Después de cierto número de datos tomados, el error del sistema (desplazamiento en el tiempo) será igual a 0.1seg, cuando esto ocurra el tiempo programado para el retardo entre estaciones se compensa decremen-
tándolo en 0.1seg; corrigiendo de esta manera el error introducido durante la operación. El procedimiento es determinar el número de

estaciones luego de las cuales el sistema alcanzará el error de 0.1seg, esto lo hace la computadora y el resultado se envía como un nuevo código de programación del sistema M(9) y M(10); luego en operación el sistema lleva un conteo de estaciones en M(11) y M(12) y cuando alcanza el valor programado en M(9) y M(10) entonces compensa y borra el conteo, y éste comienza de nuevo y se realiza cuantas veces se necesite durante toda la operación.

5.5.2 Autocalibración.

Lo que se modifica es la toma de datos (Debido al cambio del ADC) y se realiza igual a como se hace en adquisición. Por lo demás tanto calibración, como transmisión y programación sufren pequeños cambios en el dimensionamiento de datos a recibir y transmitir (Ahora son 11 datos de programación) y los datos válidos son almacenados a partir de otra memoria (001A).

5.5.3 TESTRAM.

La nueva TESTRAM (Fig.5.8 y 5.9) al igual que la primera es una subrutina que prueba la memoria RAM del sistema con la diferencia que para hacerlo el procedimiento cambia y ahora es el siguiente:

1. Se escribe en casi toda la memoria RAM un número correlativo desde 0 hasta un cierto valor (que van de 1 en 1).
2. Se inspeccionan todas las localidades en donde se escribió y se esperará que se encuentren en ellas exactamente los datos escritos.
3. Si en una localidad no se encuentra el dato esperado, la rutina sale del lazo y finaliza transmitiendo a la PC la dirección donde estuvo la falla (usando la subrutina TX) y vuelve al inicio.
4. Si toda la memoria esta bien, al final se envía a la computadora el código de "todo bien" y se regresa al inicio.

5.5.4 RESETRAM

Al igual que en la subrutina RESETRAM (Fig.5.10) del prototipo, la finalidad es colocar el dato 00 en todas las memorias correspondientes a guardar datos de información. En esta subrutina previamente se incluye un procedimiento de búsqueda de la parte "más alta de la RAM", dicho en otras palabras en donde termina la RAM. Esto es porque el sistema puede trabajar como máximo en este caso con 64K de RAM, pero es posible hacerlo a menos capacidad 8K, 16K y 32K; y para todos los casos, las memorias reservadas para la pila estarán en la última página de la RAM. Si se colocara 00 en las localidades de la pila, el sistema se descontrolaría, es por eso, que una vez encontrado donde termina la RAM (donde comienza la pila) en una localidad más abajo debe terminar el programa de escribir 00.

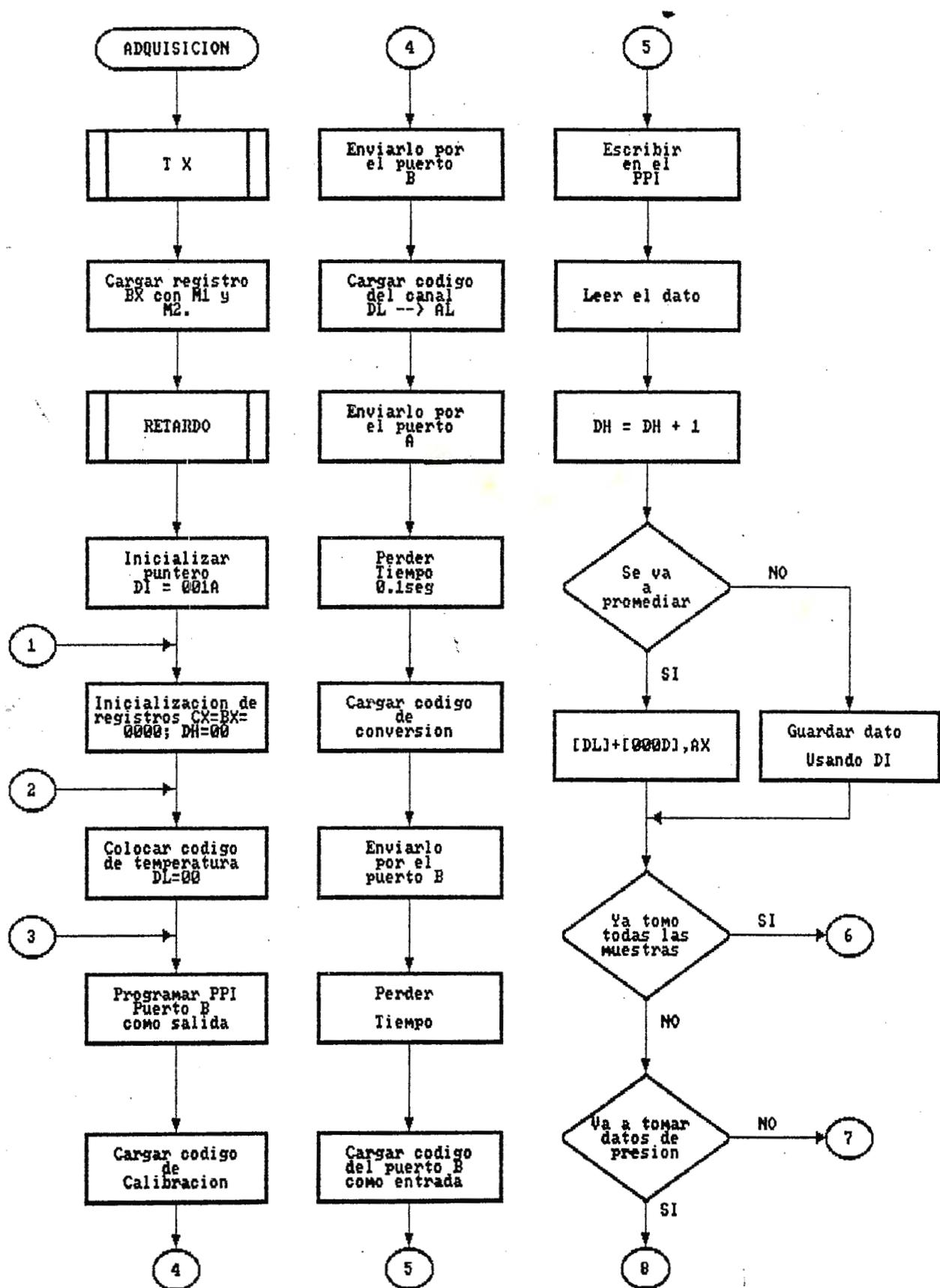


Fig.5.4 Flujo de Adquisicion de datos I.

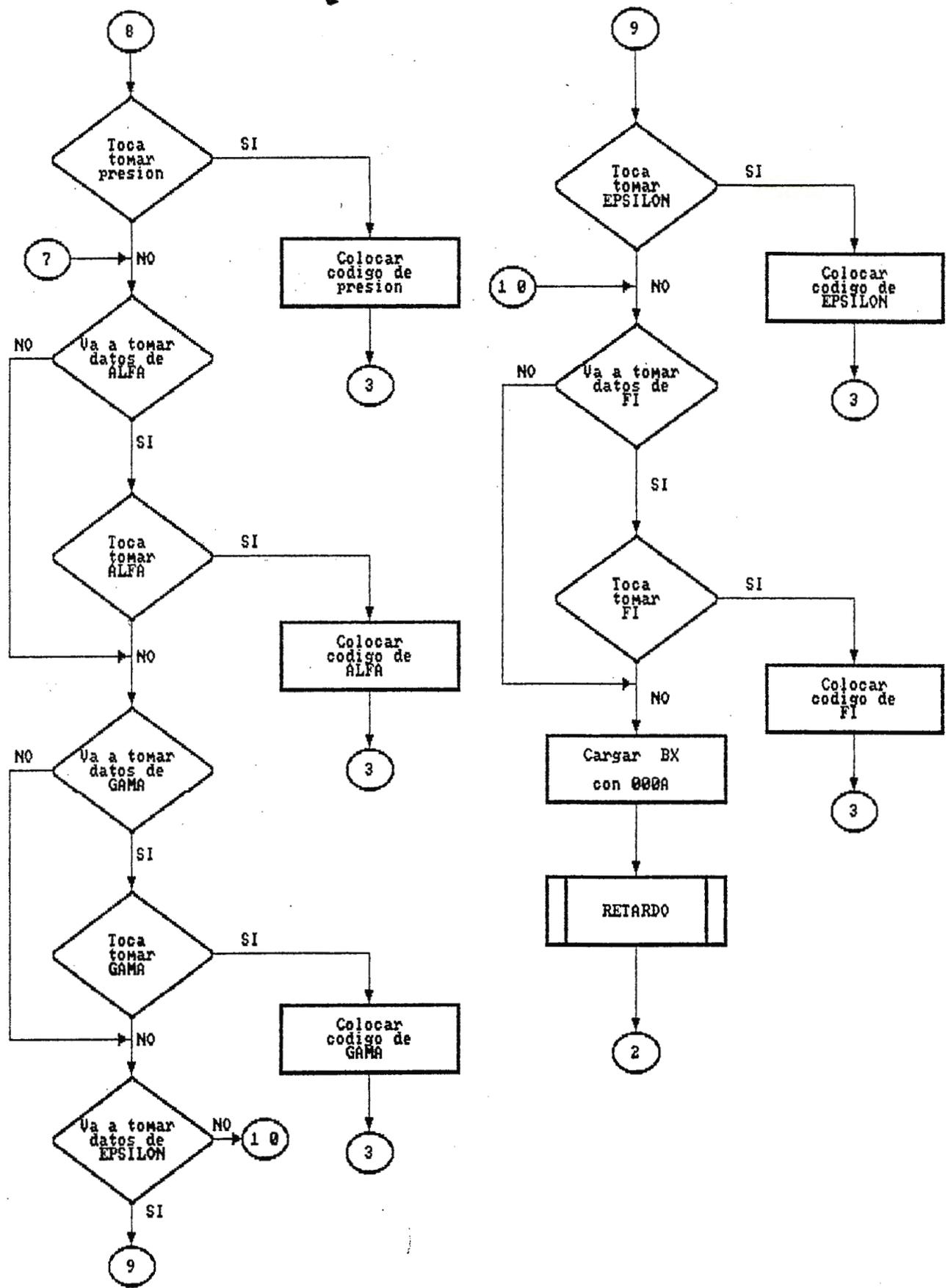


Fig.5.5 Adquisicion de datos II.

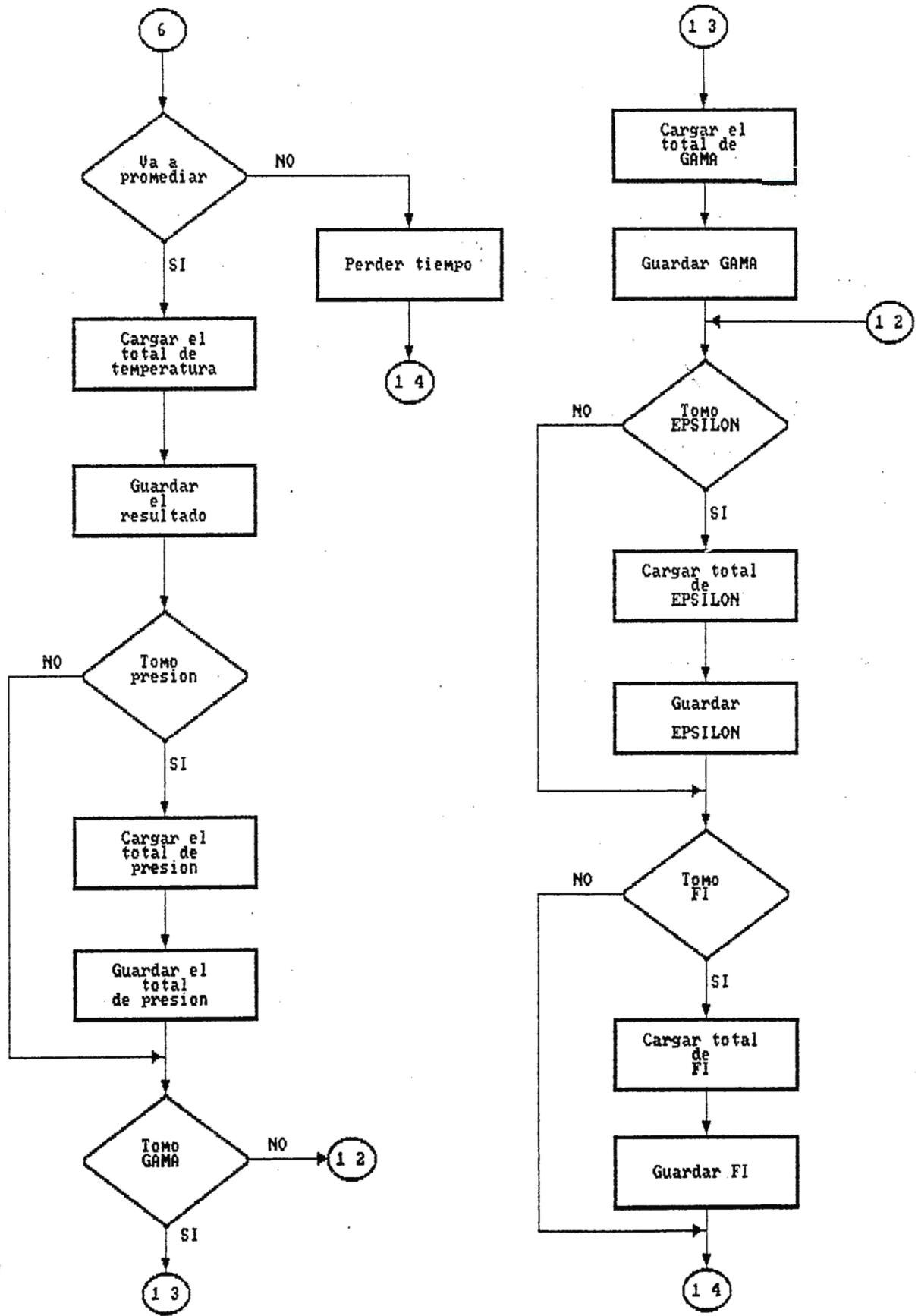


Fig.5.6 Adquisición de datos III.

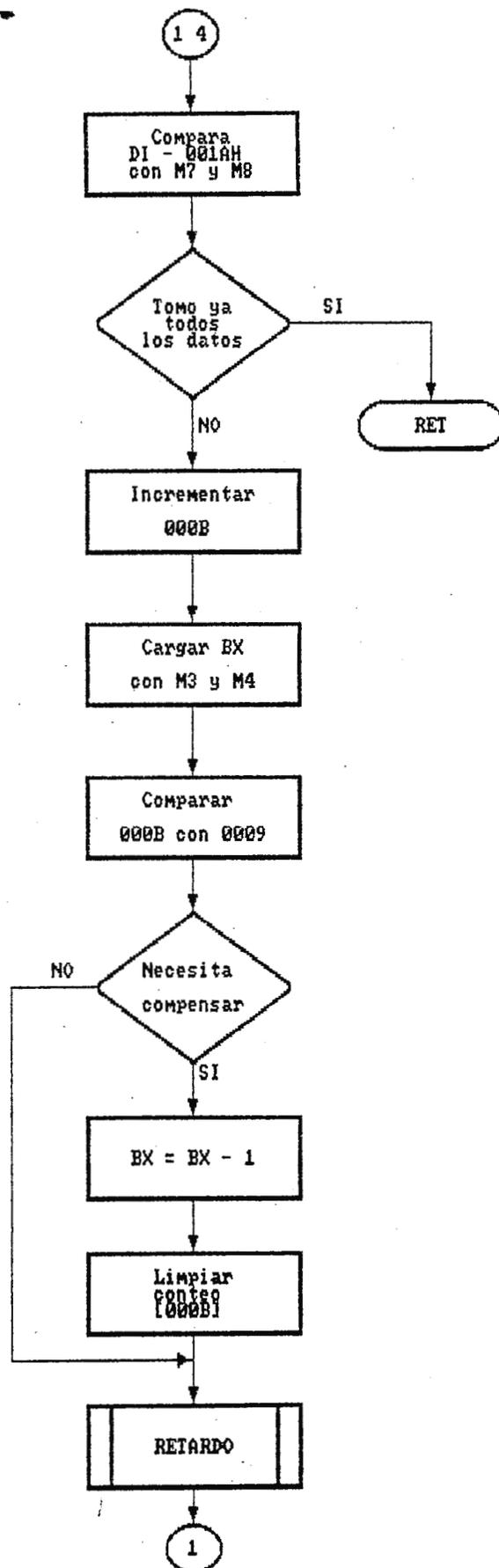


Fig. 5.7 Adquisición de datos IV.

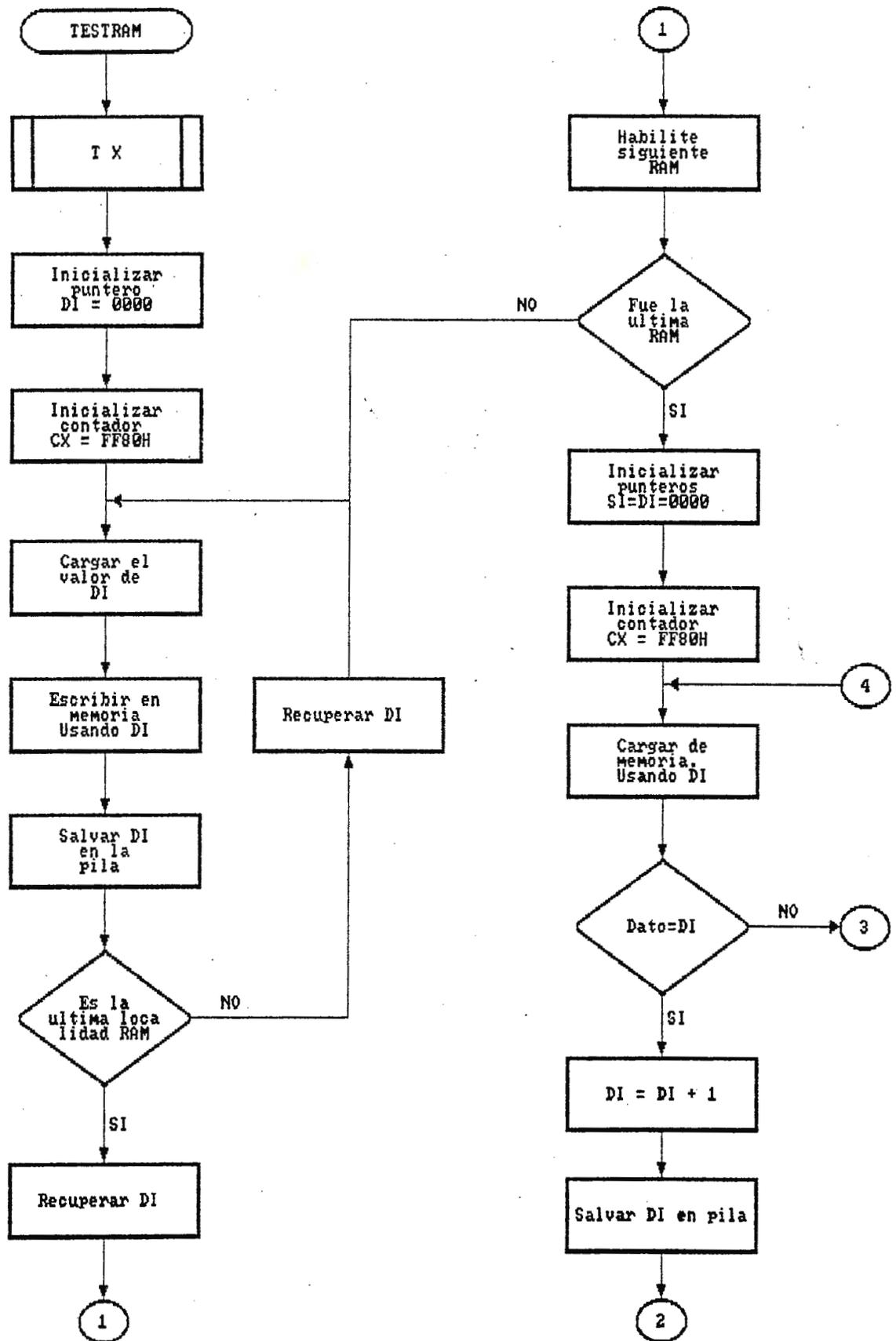


Fig.5.8 Subrutina de TESTRAM I.

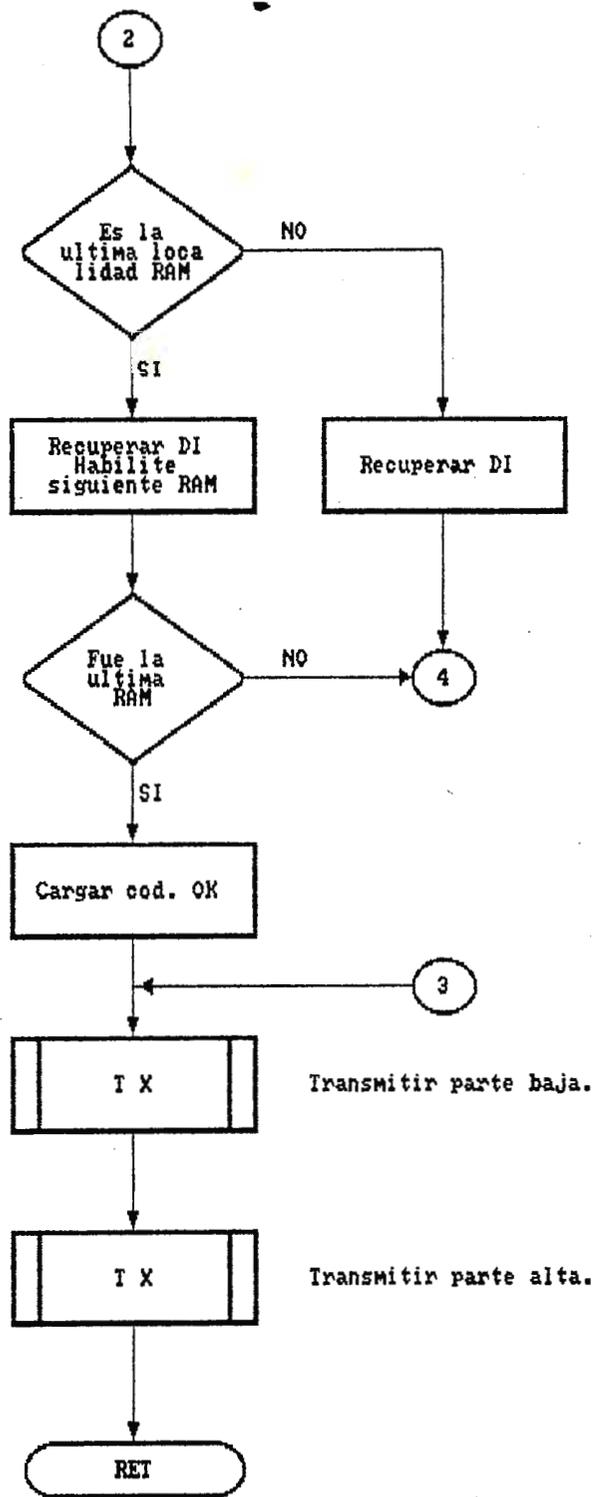


Fig.5.9 Subrutina de TESTRAM II.

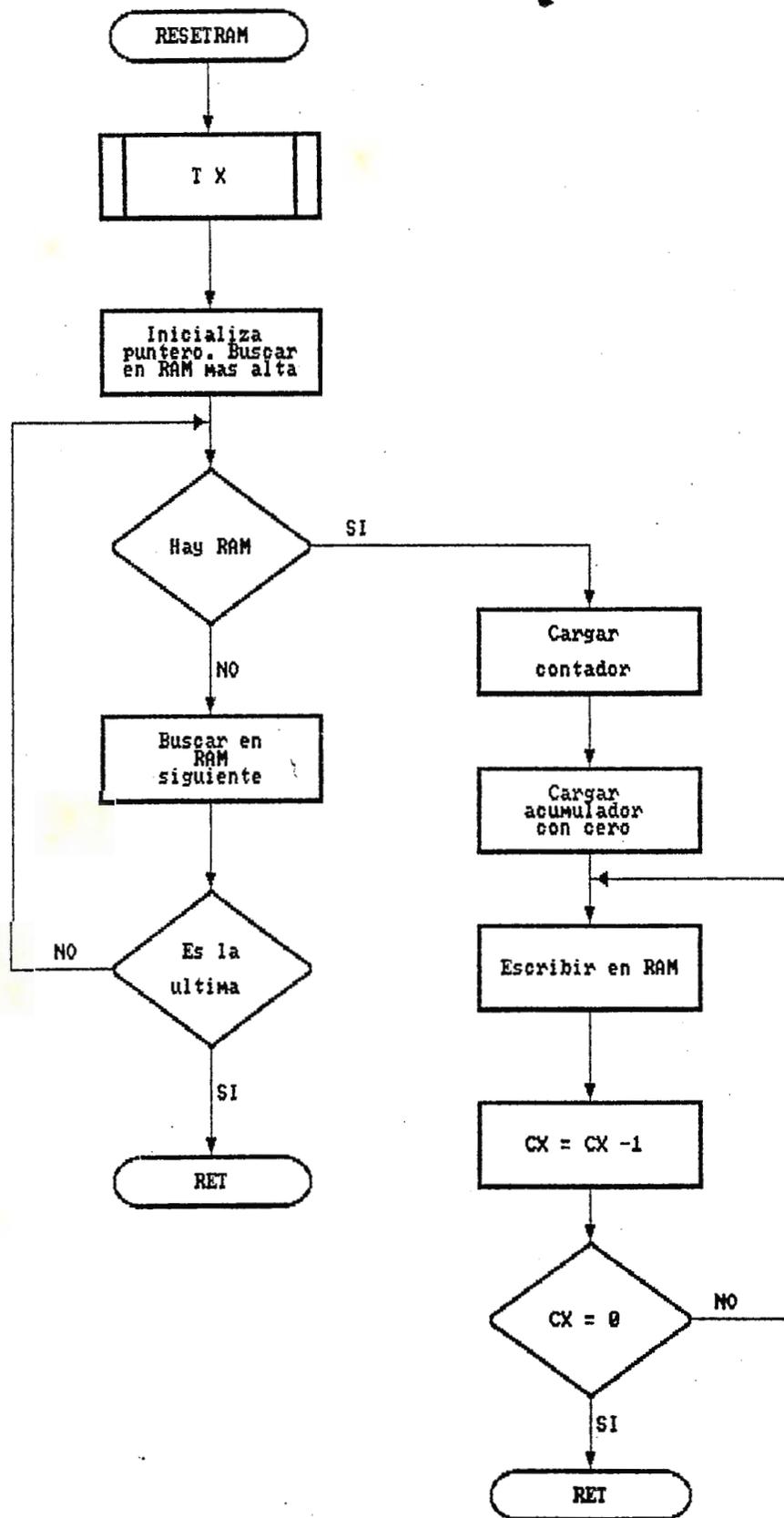


Fig.5.10 Subrutina de RESETRAM.

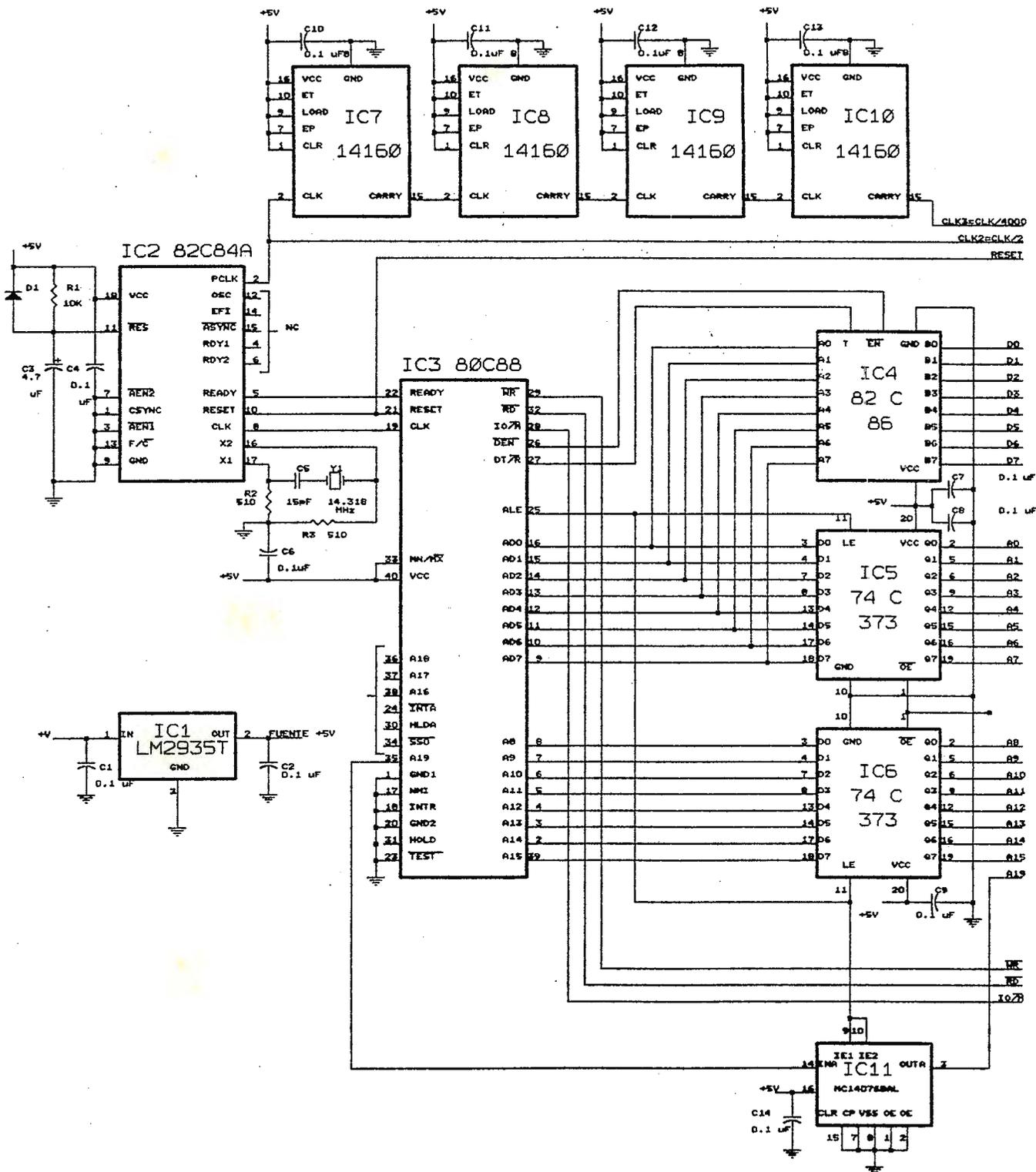


Fig. 5.11 SECCION DE CONTROL. SONDA

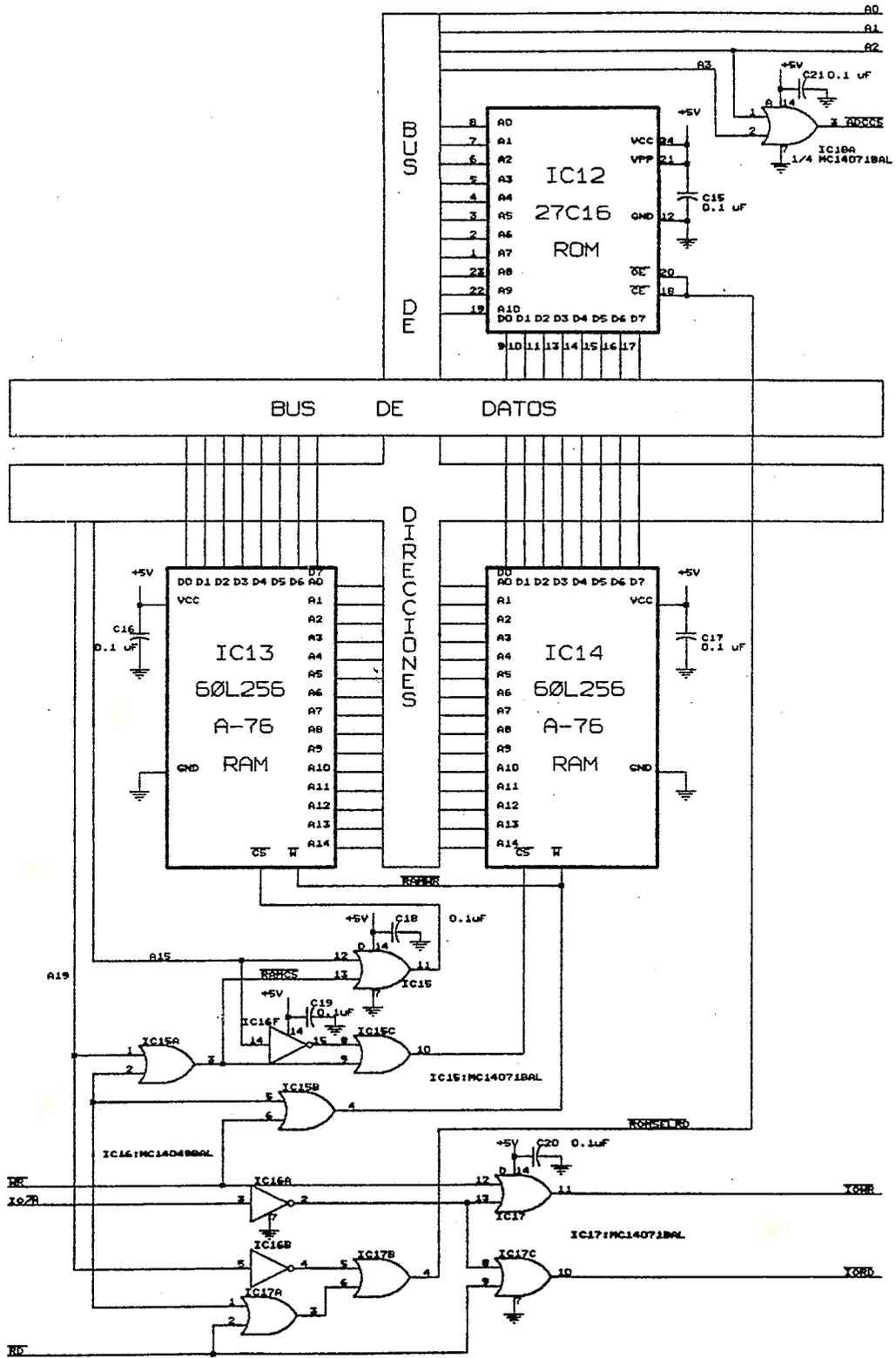


Fig. 5.12 SECCION DE MEMORIAS Y DECODIFICADOR DE DIRECCIONES SONDA.

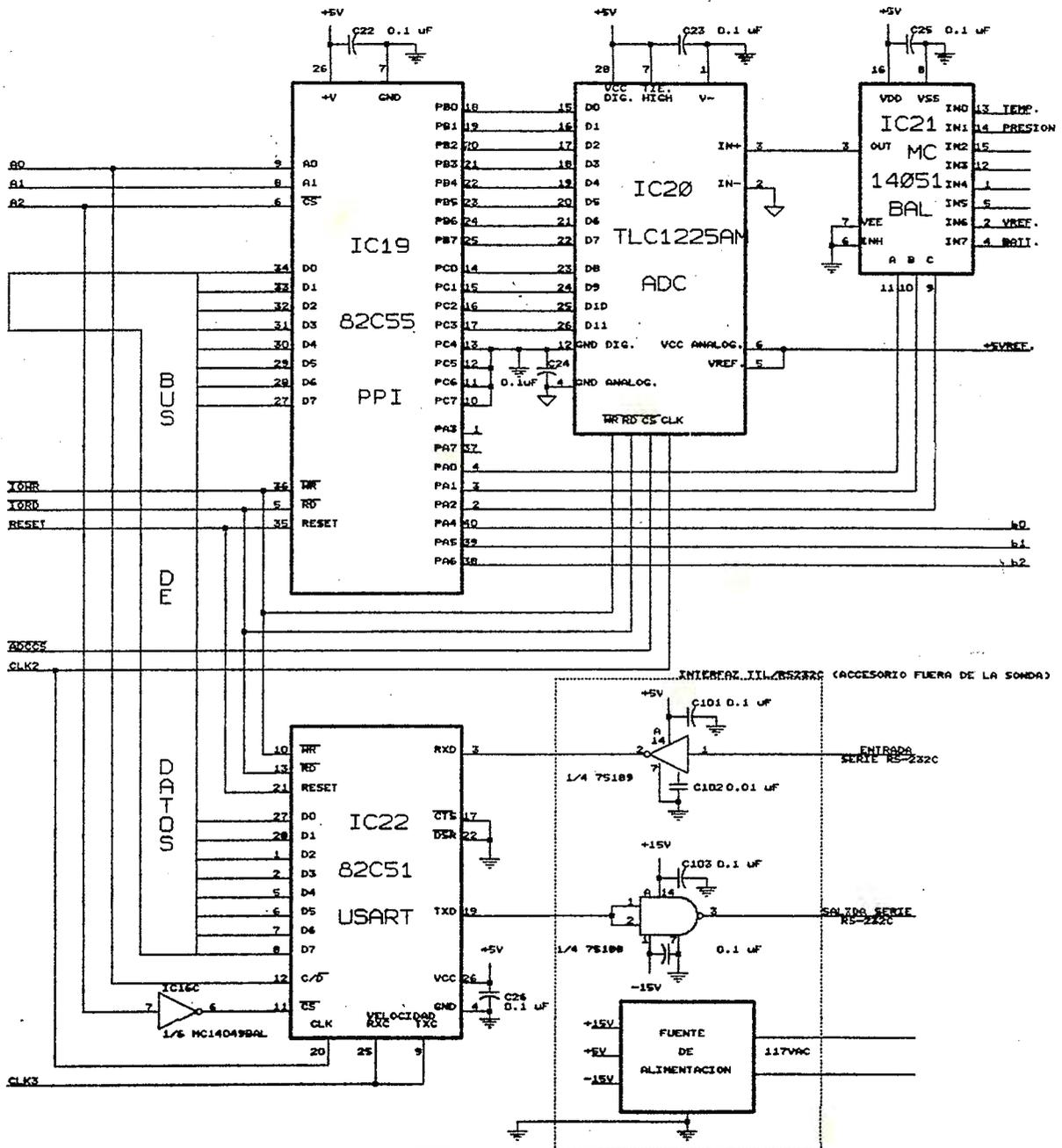


Fig.5.13 SECCION DE INTERFACES. Sonda

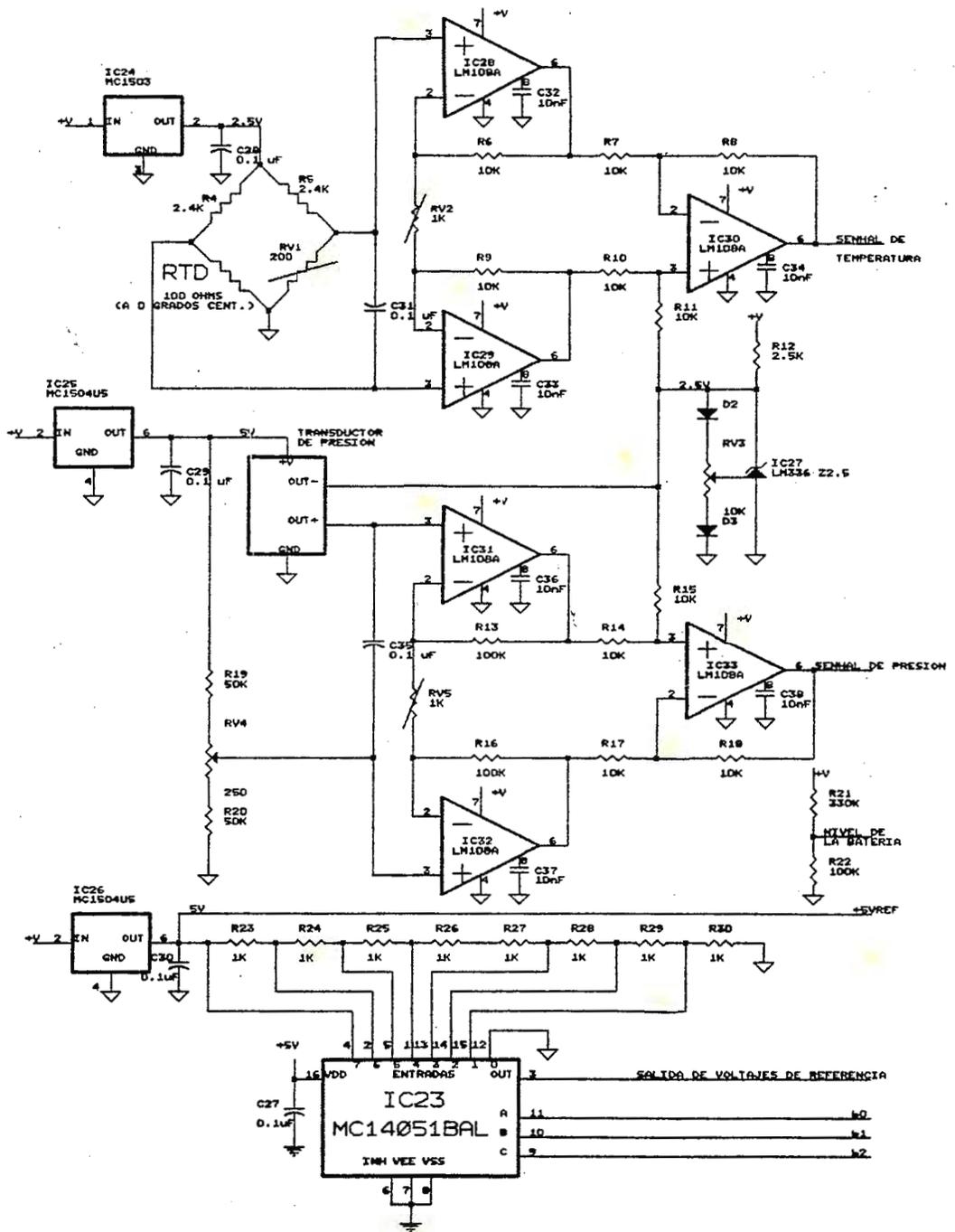


Fig. 5.14 SECCION DE ACONDICIONAMIENTO DE SEÑAL. Sonda.

CAPITULO 6

APLICACIONES ALTERNATIVAS DEL SISTEMA DE ADQUISICION DE DATOS

6.1 Descripción de la opción.

El sistema descrito en los capítulos anteriores contará con esta nueva opción que es útil en el siguiente caso:

Se trata que el sistema, siendo portátil y con capacidad de tomar y almacenar datos, pueda ser empleado en situaciones en que una persona debe tomar varios datos en distintos puntos, pero que dicha persona pueda transportarlo, colocarlo y tomar el dato cuando así lo requiera a través de un pulsador, y después que quiera observar los datos lo lleve y conecte a una computadora y pueda serle presentados los datos tomados.

6.2 Cambios en el programa de la PC.

Debe contar con dos funciones aparte de las ya descritas:

- Programar en opción manual: establece las variables a medir, el número de muestras a tomar y si el sistema obtendría el promedio o no. Los tiempos y el número total de datos en esta ocasión no interesan. Luego se envía el código de programación con los datos escritos y al final se envía el código de toma de datos manual. Por supuesto siempre se cuenta con las funciones de calibración, prueba de línea, prueba de ram, borrado y verificación.

- La opción de lectura es igualmente válida tanto para datos tomados automáticamente con retardos de tiempo o manualmente.

6.3 Cambios en el hardware.

- Realizar un circuito de reset con un pulsador tal y como se muestra en la Fig.6.1 en donde el pulsador forma parte del circuito del 82C84 que es el generador de las señales de reloj, reset y ready del micro.

- Hacer un circuito como el mostrado en la Fig.6.1 , en donde por medio de un pulsador y un eliminador de rebote se controla la entrada

TEST del micro.

- En caso de pretender ocupar el sistema de adquisición de datos en la aplicación original dentro de pozos geotérmicos y que además tenga la opción manual que naturalmente al diseñar la

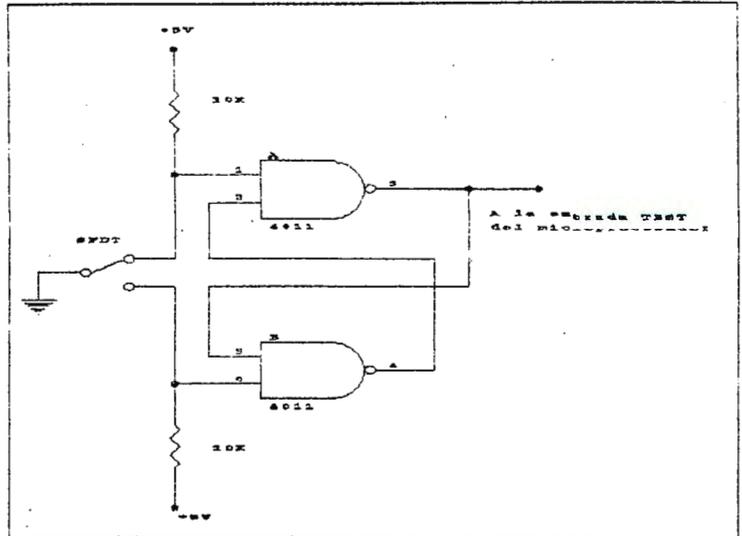


Fig.6.1 Eliminador de rebote

cubierta debe tenerse en

cuenta que cuando se ocupe el sistema para la aplicación dentro de los pozos, los pulsadores deben ser protegidos dentro de la sonda.

6.4 Cambios en el programa monitor.

Para la opción manual el programa debe tener una subrutina exclusiva para ella y en el ESTADO1 debe colocarse una decisión por medio de la cual al recibirse el código de la nueva opción, el programa pueda "saltar" y ejecutar la subrutina. A esta subrutina se le ha llamado OPMAN y su flujograma se muestra en la Fig.6.2.

Lo que la subrutina hace es poner al sistema en estado de espera con la instrucción WAIT, por medio de la cual éste pasa a monitorear el estado de la señal TEST, si ésta está en alto el micro obtiene hasta que la señal TEST pasa a "bajo". Esto sucederá cuando se oprima el pulsador SPDT (Fig.6.1); es entonces cuando el operador quiere tomar un dato y es por esto que la subrutina OPMAN se llama a la subrutina de adquisición en donde según los datos de programación toma las muestras de las variables programadas correspondientes a una estación y retorna a OPMAN en donde vuelve a esperar por otro pulso, que deberá activar el usuario cuando en

otro punto quiera tomar datos.

El resto de las instrucciones de la subrutina son detalles de operación del sistema para que puedan usarse las subrutinas previamente hechas y presentados en los capítulos anteriores. Tal es el caso de la inicialización del puntero DI en donde se establece de donde se empezaron a almacenar los datos(memoria 001A). La escritura de DI en M(7) y M(8) al volver de adquisición es para que cuando la computadora pida los datos, la subrutina de transmisión encuentra en M(7) y M(8) la cantidad de datos total tomados por el usuario.

Un par de notas respecto a la subrutina son:

- Cuando se llama a la subrutina de adquisición, esta no se ejecuta desde el principio sino desde la inicialización de los registros (punto 1, flujograma de adquisición)
- La subrutina OPMAN permanecerá para siempre dentro del lazo de espera ya que la característica fundamental de la opción es que el usuario determina todo, es decir, él tomará datos accionando S2 cuando el quiera (o cuando este listo) y cuantos quiera, es por ello que no se ha colocado una variable por la cual la subrutina termine. De ahí que se hace necesario la colocación de un pulsador de RESET que inicializa el sistema y pone fin a la operación.

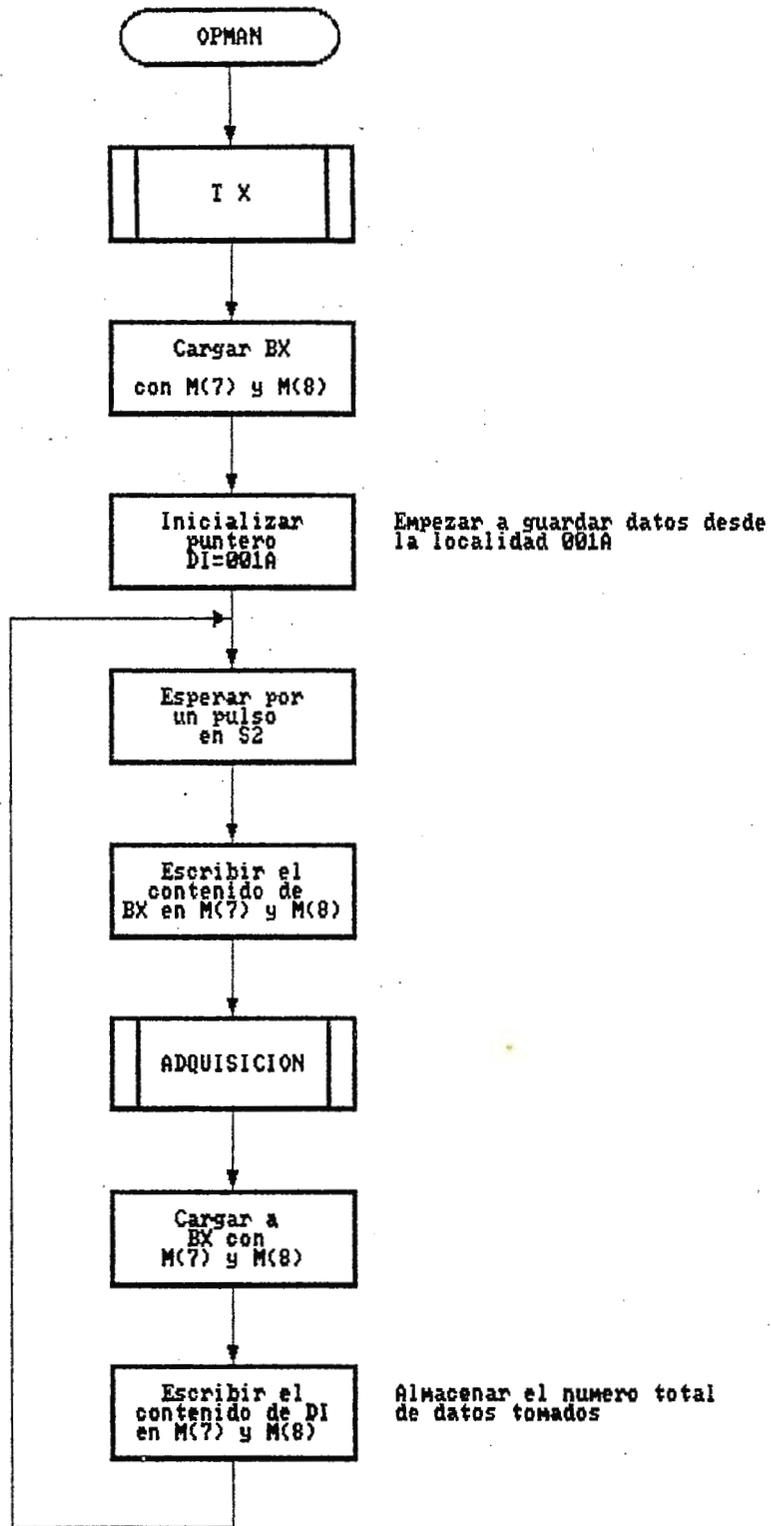


Fig.6.2 Subrutina de la opcion manual.

APENDICE A
LISTADO DEL PROGRAMA MONITOR

SYSTEM EQUATES

```

PORTA      EQU 40H      ;DIRECCION PUERTO A DEL PPI
PORTB      EQU 41H      ;DIRECCION PUERTO B DEL PPI
PORTC      EQU 42H      ;DIRECCION PUERTO C DEL PPI
COM        EQU 74H      ;PUERTO USART
COMCON     EQU 75H      ;PUERTO CONTROL USART
COD_PRUBEA EQU 20H      ;CODIGO DE PRUEBA DE LINEA
COD_TX     EQU 21H      ;CODIGO DE TRANSMISION(LECTURA)
COD_CALIB  EQU 0AH      ;CODIGO DE AUTOCALIBRACION
COD_PROG   EQU 22H      ;CODIGO DE PROGRAMACION
COD_VERF   EQU 23H      ;CIDOGO DE VERIFICACION
COD_TESTR  EQU 24H      ;CODIGO DE TESTRAM
COD_RESETR EQU 25H      ;CODIGO DE RESETRAM
COD_FIN    EQU 04H      ;FIN DE TRANSMISION
OFFSET    EQU 00H      ;OFFSET DE MEMORIA

```

```

*****
* Programa principal del sistema. *
* Aqui los registros internos del microprocesador *
* son inicializados y se preparan los puertos *
* serie y paralelo. Se espera recibir código al- *
* guno de la computadora. *
*****

```

SONDA SEGMENT PARA PUBLIC 'CODE'

ASSUME CS:SONDA,DS:SONDA,ES:SONDA,SS:SONDA

```

INICIO:      MOV AX,0000      ;INICIALIZA LOS REGISTROS:
              MOV DS,AX      ;DATA SEGMENT
              MOV ES,AX      ;EXTRA SEGMENT
              MOV SS,AX      ;STACK SEGMENT
              MOV AX,FFFEH    ;INICIALIZA
              MOV SP,AX      ;STACK POINTER
              MOV AL,8BH     ;INICIALIZA
              OUT 43H,AL     ;PPI
              MOV AL,00      ;PREPARAR EL PUERTO
              OUT COMCON,AL   ;SERIE PARA SU
              MOV CX,0002     ;PROGRAMACION
DELAY1:      LOOP DELAY1     ;PERDIDAS DE TIEMPO
              OUT COMCON,AL   ;NECESARIAS PARA QUE
              MOV CX,0002     ;LA USART RECIBA EL
DELAY2:      LOOP DELAY2     ;CODIGO CORRECTO Y
              OUT COMCON,AL   ;ASEGURARSE QUE LO
              MOV CX,0002     ;RECIBIO. POR ESTO
DELAY3:      LOOP DELAY3     ;SE REPITE 3 VECES
              MOV AL,40H     ;COMMANDO RESET
              OUT COMCON,AL   ;USART
              MOV CX,0002     ;
DELAY4:      LOOP DELAY4     ;
              MOV AL,7DH     ;PROGRAMAR FORMATO
              OUT COMCON,AL   ;1 STOP BIT, 8 DATA BITS

```

```

MOV CX,0002 ;
DELAY5: LOOP DELAY5 ;
ESTADO1: MOV AL,36H ;RECEPTOR HABILITADO
OUT COMCON,AL ;
EXISTE: IN AL,COMCON ;LEE EL REGISTRO DE ESTADO
AND AL,02 ;PRUEBA SI HAY DATO
JZ EXISTE ;EN LA LINEA
IN AL,COM ;CARGA DATO RECIBIDO
CMP AL,COD_PRUEBA ;ES EL CODIGO DE
JZ PRUEBAL ;PRUEBA DE LINEA
CMP AL,COD_TX ;ES EL CODIGO
JZ LECTURA ;DE LECTURA
CMP AL,COD_CALIB ;ES EL CODIGO DE
JZ CALIBRAR ;AUTOCALIBRACION
CMP AL,COD_PROG ;ES EL CODIGO DE
JZ PROGRAMAR ;PROGRAMACION
CMP AL,COD_VERF ;ES EL CODIGO DE
JZ VERIFICAR ;VERIFICACION
CMP AL,COD_FIN ;ES EL CODIGO DE
JZ SALIR ;FINALIZACION
CMP AL,COD_TESTR ;ES EL CODIGO DE
JZ TESTMEMORIA ;PRUEBA DE MEMORIA
CMP AL,COD_RESTR ;ES EL CODIGO DE
JZ RESETMEMORIA ;LIMPIAR MEMORIA
CALL RETRANSMITIR ;TRANSMITE CODIGO RECIBIDO
JMP ESTADO1 ;VUELVA AL ESTADO DE ESPERA
LECTURA: CALL TXDATOS ;
JMP SALTO ;
PRUEBAL: CALL PRUEBALINEA ;
JMP SALTO ;
CALIBRAR: CALL AUTOCALIBRACION ;
JMP SALTO ;
SALIR: CALL ADQUISICION ;
JMP SALTO ;
TESTMEMORIA: CALL TESTRAM ;
JMP SALTO ;
RESETMEMORIA: CALL RESETRAM ;
JMP SALTO ;
VERIFICAR: CALL VERIFICACION ;
SALTO: JMP INICIO ;

```

```

*****
* Este es el programa donde se reciben los datos *
* datos de programación, y son pasados por mayo- *
* ría para obtener los datos correctos y guardar *
* los en memoria. *
*****

```

```

PROGRAMAR:      CALL RETRASMITIR ;TRANSMITIR DATO RECIBIDO
                MOV DI,0100H ;INICIALIZAR OFFSET
                CLD ;
                MOV CX,005AH ;PREPARAR CONTADOR
                MOV AL,36H ;PROGRAMAR USART
                OUT COMCON,AL ;COMO RECEPTOR
SALVAR:         PUSH CX ;SALVAR CX EN LA PILA
                MOV CX,FFFFH ;PREPARA CONTADOR RETARDO
LISTA?:        IN COMCON,AL ;EXISTE DATO
                AND AL,02 ;EN LA LINEA
                JNZ RECUPERE ;
                LOOP LISTA? ;
                JMP SALTO1 ;
RECUPERE:      POP CX ;CARGUE NUEVO CX
                IN AL,COM ;LEER DATO USART
                STOSB ;GUARDAR DATO
                LOOP SALVAR ;IR A SALVAR NUEVO CX
                MOV SI,0100H ;PREPARAR REGISTROS
                MOV DI,OFFSET ;
MAYORIA:      PUSH SI ;SALVAR SI
                MOV BX,SI ;PASAR SI A BX
LIMPIAR:      MOV AH,00 ;LIMPIAR LOS
                MOV DH,00 ;CONTADORES
                MOV DL,[BX] ;CARGAR UN DATO DE MEMORIA
EXTRAER:      LODSB ;EXTRAER DE MEMORIA HACIA AL
                CMP DL,AL ;SON IGUALES?
                JNZ INCREMENTE ;VALLA CONTADOR DE DATOS
                INC DH ;INC. CONTADOR DE REPETIDOS
INCREMENTE:   IND AH ;INC. CONTADOR DE DATOS
                CMP AH,0AH ;YA SE TOMARON
                JNZ EXTRAER ;LOS DIEZ DATOS
                CMP DH,05H ;HAY MAS DE CINCO
                JG GRANDE ;DATOS REPETIDOS
                POP SI ;DEVOLVER SI
                PUSH SI ;SALVAR SI
                INC BX ;INCREMENTAR OFFSET
                PUSH BX ;SALVAR OFFSET
                SUB BX,SI ;RESTE BX-SI
                CMP BX,0AH ;YA SE TERMINARON
                JZ SALTO1 ;LOS DIEZ DATOS
                POP BX ;RECUPERAR BX
                JMP LIMPIAR ;
GRANDE:      MOV AL,DL ;GUARDAR EL RESULTADO
                STOSB ;VALIDO DE MAYORIA
                CMP DI,09 ;SE TRABAJARON LOS 9
                JNZ MAYORIA ;DATOS DE PROGRAMACION
SALTO1:      JMP SALTO ;

```

```

*****
*      Esta subrutina solo es la encargada de solo      *
*      transmitir los códigos de verificación.          *
*****

```

VERIFICACION PROC NEAR

```

CALL RETRANSMITIR ;
MOV DL,02          ;CODIGO DE VERIFICAR
MOV CX,FFFFH      ;RETARDO DE
DELAY6:           LOOP DELAY6      ;TIEMPO
CALL TRNS         ;LLAMAR A TRNS
RET               ;RETORNO DE SUBRUTINA

```

VERIFICACION ENDP

```

*****
*      Esta subrutina es la que trasmite 10 veces      *
*      el último dato enviado a la computadora.        *
*****

```

RETRANSMITIR PROC NEAR

```

PUSH AX           ;SALVA DATO EN LA PILA
MOV CX,0020H     ;PERDER
DELAY7:          LOOP DELAY7      ;TIEMPO
MOV AL,33H       ;PROGRAMAR USART
OUT COMCON,AL    ;COMO TRANSMISOR
POP AX           ;RECUPERA EL DATO
MOV BL,AL        ;COPIA DATO EN BL
MOV CX,000AH     ;CX PARA TRANSMITIR 10 VECES
VER:             IN AL,COMCON      ;ESTA LISTO PARA
AND AL,01H       ;TRANSMITIR EL
JZ VER           ;DATO
MOV AL,BL        ;RECUPERA AL SU DATO
OUT COM,AL       ;TRANSMITE EL DATO
LOOP VER         ;EJECUTE EL LAZO
RET              ;RETORNO

```

RETRANSMITIR ENDP

```

*****
*   La subrutina de Retardo solo son instrucciones   *
*   de pérdida de tiempo.                           *
*****

```

RETARDO PROC NEAR

```

DELAY:          MOV CX,6792H          ;CARGA EL CONTADOR PARA TIEMPO
DELAY8:         LOOP DELAY8          ;SE EJECUTA 6792H VECES
                DEC BX              ;DECREMENTA BX
                JNZ DELAY           ;SI NO ES CERO CONTINUE
                RET                 ;RETORNO

```

RETARDO ENDP

```

*****
*   Estas son las rutinas de transmisión de datos   *
*****

```

TXDATOS PROC NEAR

```

                MOV DL,00            ;CODIGO DE TRANSMISION TOTAL
                CALL RETRANSMITIR    ;
                MOV CX,FFFFH        ;RETARDO PEQUEÑO
DELAY9:         LOOP DELAY9         ;DE TIEMPO
TRNS:          MOV DI,OFFSET        ;PREPARAR OFFSET
                CLD                  ;AJUSTE AUTOMATICO DE DI
CONTAR:        MOV CX,000AH         ;PREPARAR CONTADOR
ENVIODATO:     IN AL,COMCON         ;LISTO PARA
                AND AL,01           ;TRANSMITIR
                JZ ENVIODATO        ;OTRO DATO
                MOV AL,[DI]         ;BUSCA DATO DE MEMORIA
                OUT COM,AL          ;ENVIA EL DATO
                LOOP ENVIODATO      ;REPETIRLO 9 VECES
                INC DI              ;
                CMP DI,+09H         ;YA SE TRANSMITIERON
                JNZ CONTAR          ;LOS 9 DATOS DE PROG.
                CMP DL,02H         ;VA A TRANSMITIR SOLO LOS
                JZ RETORNE          ;9 DATOS DE PROGRAMACION
TRANSII:       MOV CX,FFFFH        ;PERDER TIEMPO
DELAY10:       LOOP DELAY10        ;ANTES DE CONTINUAR
                INC DI              ;PREPARAR DI
CUENTAS:       MOV CX,000AH         ;AJUSTAR CONTADOR
TXOTRO?:       IN AL,COMCON         ;LISTO PARA
                AND AL,01           ;TRANSMITIR
                JZ TXOTRO?         ;OTRO DATO
                MOV AX,[DI]         ;CARGAR DATO DE MEMORIA
                OUT COM,AL          ;ENVIAR LSB
                LOOP TXOTRO?        ;REPETIRLO 9 VECES
                INC DI              ;
                MOV CX,000AH         ;CARGAR CONTADOR A 10
TXOTRO1?:      IN AL,COMCON         ;LISTO PARA
                AND AL,01           ;TRANSMITIR
                JZ TXOTRO1?        ;OTRO DATO

```

```

MOV AL, AH ;COPIAR EL MSB A AL
OUT COM, AL ;ENVIAR MSB
LOOP TXOTRO1? ;REPETIRLO 9 VECES
INC DI ;
MOV AX, DI ;COPIE DI A AX
SUB AX, 000C ;RESTE AX 12DEC
CMP AX, [0007] ;YA SE TRANSMITIERON
JNZ CUENTAS ;TODOS LOS DATOS
RET ;RETORNO

RETORNE:

TXDATOS ENDP
TRNS ENDP
TRANS II ENDP

```

```

*****
* Subrutina de Prueba de Línea. *
*****

```

```

PRUEBALINEA PROC NEAR

```

```

CALL RETRANSMITIR ;
MOV AL, 36H ;PROGRAMAR USART
OUT COMCON, AL ;COMO RECEPTOR
MOV DI, OFFSET ;AJUSTAR OFFSET
CLD ;
MOV CX, 0064H ;CX PARA 100 DATOS
CONTADOR: PUSH CX ;
MOV CX, FFFFH ;
HAYDATO: IN AL, COMCON ;EXISTE DATO
AND AL, 02 ;EN LA LINEA
JNZ ACTUALIZE ;
LOOP HAYDATO ;
JMP RETORNO1 ;SALIR SI NO HAY DATO EN LINEA
ACTUALIZE: POP CX ;RECUPERAR CONTADOR
IN AL, COM ;RECIBA EL DATO
STOSB ;GUARDE EL DATO
LOOP CONTADOR ;
MOV CX, FFFFH ;PERDER
DELAY11: LOOP DELAY11 ;TIEMPO ANTES DE SEGUIR
MOV AL, 33H ;PROGRAMAR USART
OUT COMCON, AL ;COMO TRANSMISOR
MOV CX, 0064H ;CX PARA 100 DATOS
MOV SI, OFFSET ;AJUSTAR OFFSET
BANDERA: IN AL, COMCON ;LISTO PARA
AND AL, 01 ;TRANSMITIR LOS
JZ BANDERA ;DATOS
LODSB ;EXTRAER DATO DE MEMORIA
OUT COM, AL ;ENVIAR DATO
LOOP BANDERA ;HAGALO 99 VECES
RETORNO1: RET ;

```

```

PRUEBALINEA ENDP

```

```

*****
*           Subrutina de Autocalibración.           *
*****

```

```

AUTOCALIBRACION PROC NEAR

```

```

                                CALL RETRASNMITIR ;
                                MOV DI,000AH      ;AJUSTAR DI A 10
                                CLD                ;
                                MOV AL,02         ;CARGUE AL CON DOS
                                MOV BH,AL        ;COPIE AL EN BL
CALIB:                          MOV DX,0000     ;PREPARAR REGISTRO SUMAS
                                MOV SI,AX       ;
                                MOV CX,0010H    ;16 MUESTRAS DE C/U
HABILITE:                       MOV AX,SI      ;
                                PUSH BX        ;
                                MOV BX,0AH     ;ESPERA 1 SEG
                                CALL RETARDO   ;ANTES DE CALIBRAR E INICIAR
                                POP BX        ;
                                OUT PORTA,AL   ;EL CICLO DE CONVERSION
                                WAIT          ;ESPERAR FIN DE CONVERSION
                                IN AX,PORTB   ;CARGAR DATO DEL ADC
                                NOT AX       ;COMPLEMENTAR
                                ADD DX,AX     ;SUMAR LOS DATOS
                                LOOP HABILITE  ;REPETIRLO 15 VECES
                                XCHG DX,AX    ;AX <--> DX
                                STOSW        ;ALMACENE LA PALABRA
                                ADD BH,10H   ;PREPARE EL SIGTE.CANAL
                                MOV AL,BH    ;COPIE A AL
                                CMP BH,82H   ;YA SE HABILITARON LOS
                                JNZ CALIB    ;OCHO CANALES DEL MUX
                                MOV AL,03    ;SENSAR BATERIA
                                MOV BX,0AH   ;PERDER TIEMPO
                                CALL RETARDO ;ANTES DE CONTINUAR
                                OUT PORTA,AL ;INICIAR CONVERSION
                                WAIT        ;ESPERAR FIN CONVERSION ADC
                                IN AL,PORTC  ;CARGAR MSB
                                MOV AH,AL   ;COPIARLO EN AH
                                IN AL,PORTB  ;CARGAR LSB
                                NOT AX     ;COMPLEMENTAR
                                MOV [001A],AX ;
                                MOV DI,0009H ;
                                MOV AX,0012H ;
                                MOV [0007],AX ;
                                CALL TRANSII ;
                                RET         ;

```

```

AUTOCALIBRACION ENDP

```

```

*****
* Este es el programa principal de la sonda. Aquí *
* es donde los datos operados por el monitor son *
* almacenados en su banco de memoria correspon - *
* pondientes a las variables de temperatura y *
* presión. *
*****

```

ADQUISICION PROC NEAR

```

CALL RETRANSMITIR ;
MOV BX,[0001] ;CARGAR A BX CON EL
CALL RETARDO ;RETARDO ANTES DE INICIAR
MOV DI,000AH ;EMPEZAR A GUARDAR EN 000A
CLD ;
AGAIN: MOV CX,0000 ;
MOV BX,0000 ;
MOV DH,00 ;
OTRATOMA: MOV DL,00 ;
CHMUX: MOV AL,DL ;
OUT PORTA,AL ;
PUSH BX ;
MOV BX,0AH ;ESPERAR 1 SEGUNDO
CALL RETARDO ;ANTES DE REALIZAR
OUT PORTA,AL ;LA CONVERSION
POP BX ;
WAIT ;ESPERAR FIN CONVERSION
IN AX,PORTB ;CARGA EL DATO DEL ADC
NOT AX ;COMPLEMENTAR
INC DH ;
PUSH DX ;
MOV DL,[0005] ;CARGUE EL CODIGO DE MUESTREO
CMP DL,00 ;VA A REALIZAR UNA
JZ SIMPLE ;TOMA SIMPLE
POP DX ;PROMEDIANDO
CMP DL,01 ;EN ESTE MOMENTO HA LEIDO
JZ PRESION ;UN DATO DE PRESION
ADD CX,AX ;SUMA TEMPERATURA
CUANTOS: CMP DH,[0006] ;YA TOMO TODAS
JZ MUESTRAS ;LAS MUESTRAS
MOV AL,[0000] ;CARGA CODIGO DE LA VARIABLE
CMP AL,00 ;VA A MEDIR
JZ CODIGO ;PRESION
CMP DL,01 ;ESTA MIDIENDO
JZ CODIGO ;PRESION?
MOV DL,01 ;SI NO, ENTONCES
JMP CHMUX ;MIDA PRESION
CODIGO: PUSH CX ;SALVAR CX
PUSH BX ;SALVAR BX
MOV BX,000A ;CARGAR A BX PARA
CALL RETARDO ;UN RETARDO DE 1 SEG
POP BX ;RECUPERA BX
POP CX ;Y CX
JMP OTRATOMA ;VALLA A TOMAR OTRO DATO

```

SIMPLE:	POP DX	;ES UNA TOMA SIMPLE
	STOSW	;GUARDE CADA DATO
	JMP CUANTOS	;CUANTAS MUESTRAS VAN
PRESION:	ADD BX,AX	;PROMEDIANDO DATOS
	JMP CUANTOS	;DE PRESION
MUESTRAS:	PUSH DX	;SALVE DX
	MOV DL,[0005]	;TOMA CODIGO DE MUESTREO
	CMP DL,00	;ESTA PROMEDIANDO
	JZ SALTO3	;LOS DATOS
	POP DX	;RECUPERE DX
	XCHG CX,AX	;INTERCAMBIAR REGISTROS
	STOSW	;GUARDANDO SUMATORIAS
	CMP DL,01	;ESTA TOMANDO
	JNZ TOTAL	;PRESION
	XCHG BX,AX	;INTERCAMBIAR REGISTROS
	STOSW	;GUARDAR SUMATORIAS TEMPERATURA
TOTAL:	MOV AX,DI	;COPIAR CONTADOR DATOS
	SUB AX,000AH	;YA SE
	CMP AX,[0007]	;TOMARON TODOS
	JNZ RETARDO2	;LOS DATOS
	RET	;RETORNO
RETARDO2:	MOV BX,[0003]	;CARGAR A BX CON EL
	CALL RETARDO	;RETARDO ENTRE MUESTRAS
	JMP AGAIN	;SALTE A OTRA ADQUISICION
SALTO3:	POP DX	;
	MOV CX,0001	;PIERDA TIEMPO
DELAY13:	LOOP DELAY13	;PARA UNA TOMA SIMPLE Y
	JMP TOTAL	;VERIFIQUE SI TERMINO

ADQUISICION ENDP

```

*****
*   Rutina de Prueba de la memoria   *
*****

```

TESTRAM PROC NEAR

	CALL RETRANSMITIR	;
	MOV DI,0000	;
	MOV SI,DI	;
	CLD	;
	MOV BX,5555H	;CARGAR CODIGO DE PRUEBA
	MOV DX,BX	;GUARDAR CODIGO DE PRUEBA
	MOV CX,0FF0H	;# DE LOCALIDADES A PROBAR
PRUBEARAM:	LODSW	;
	XCHG BX,AX	;
	STOSW	;
	DEC DI	;
	DEC DI	;
	DEC SI	;
	DEC SI	;
	LODSW	;
	CMP DX,AX	;

```

                                JNZ TESTENVIO      ;
                                XCHG BX,AX          ;
                                STOSW              ;
                                LOOP PRUEBARAM     ;
                                XCHG BX,AX        ;
TESTENVIO:                      CALL RETRANSMITIR ;
                                RET                ;

```

TESTRAMM ENDP

```

*****
*   Rutina para limpiar la memoria del sistema   *
*****

```

RESETRAM PROC NEAR

```

                                CALL RETRANSMITIR ;
                                MOV DI,000AH      ;
                                CLD                ;
                                MOV AX,0000       ;LIMPIE AX
                                MOV CX,0FF8H     ;# DE LOCALIDADES A BORRAR
LIMPIARRAM:                      STOSW          ;LIMPIE LA LOCALIDAD
                                LOOP LIMPIARRAM  ;REPITA HASTA TERMINAR
                                RET                ;RETORNE

```

RESETRAM ENDP

APENDICE B
LISTADO DEL PROGRAMA EN
TURBO BASIC

```

        'PROGRAMA PRINCIPAL DE LA PC
SCREEN 1,1
LINE (10,20)-(300,50),2,BF,&HA0A0
A$="SISTEMA DE ADQUISICION DE DATOS":I=LEN(A$)
FOR C=1 TO I
B$=LEFT$(A$,C):X=50+(C*10)
LOCATE 5,5:PRINT B$;:DELAY 0.2:SOUND X,4:NEXT
LOCATE 22,2:PRINT "Presione alguna tecla para continuar"
X$=INPUT$(1)
LOCATE 9,16:PRINT"VERSION 1.0
LOCATE 10,11:PRINT"Copyright (UDB) 1991"
V$=STRING$(30,"=")
LOCATE 12,5:PRINT CHR$(201);V$;CHR$(187)
FOR I=13 TO 17
LOCATE I,5:PRINT CHR$(186);SPC(30);CHR$(186):NEXT
LOCATE 14,10:PRINT"ELABORADO POR:"
LOCATE 15,10:PRINT"OSCAR WENCESLAO RIVAS"
LOCATE 16,10:PRINT"CARLOS ROBERTO ZAPATA"
LOCATE 18,5:PRINT CHR$(200);V$;CHR$(188)
LOCATE 22,2:PRINT "Presione alguna tecla para continuar"
x$=input$(1):screen 0,0:width 80
        ' INICIALIZACION DEL SISTEMA
        DIM A(10),CODIGO(4000),D(5000),N(5000),VDIG(5000)
        DIM T(10000),VOLT(100),M(5000)
        ' A(10),D(100),CODIGO(256),N(4000),M(10),VOLT(20)
MENU:
CLS
        ' **** PRESENTACION DEL MENU ****

CUADRO:
color 15,0
V$=STRING$(79," ")
FOR I=1 TO 24
PRINT V$:NEXT
A$=STRING$(27,"-")
B$=STRING$(27," ")
FOR C=2 TO 20
LOCATE C,48:PRINT B$:COLOR 15,1:NEXT
FOR C=3 TO 19
LOCATE C,48:PRINT "|":LOCATE C,75:PRINT "|"
NEXT
LOCATE 2,49:PRINT A$
LOCATE 2,48:PRINT CHR$(218):LOCATE 2,75:PRINT CHR$(191)
LOCATE 20,48:PRINT CHR$(192):LOCATE 20,49:PRINT A$
LOCATE 20,75:PRINT CHR$(217)
AA$=STRING$(26,"-")
LOCATE 6,48:PRINT CHR$(195):LOCATE 6,49:PRINT AA$:LOCATE
6,75:PRINT CHR$(180)
LOCATE 11,48:PRINT "|":LOCATE 11,49:PRINT AA$:LOCATE
11,75:PRINT "|"
LOCATE 16,48:PRINT "|":LOCATE 16,49:PRINT AA$:LOCATE
16,75:PRINT "|"
C$=STRING$(27," ")
color 15,0:LOCATE 21,51:PRINT C$
D$=STRING$(2," ")

```

```

FOR I=3 TO 21
LOCATE 1,76:PRINT D$:NEXT:color 15,1
LOCATE 3,50:PRINT "F1. PROGRAMAR AL SISTEMA"
LOCATE 4,54:PRINT "Se preparan los":LOCATE 5,50:PRINT
"códigos de programación"
LOCATE 7,50:PRINT "F2. RECIBIR DATOS DEL":LOCATE 8,54:PRINT
"SISTEMA"
LOCATE 9,54:PRINT "Se leen los datos":LOCATE 10,50:PRINT "de
la sonda."
LOCATE 12,50:PRINT "F3. BORRADO DE MEMORIA"
LOCATE 13,54:PRINT "Se limpian los datos"
LOCATE 14,50:PRINT "almacenados en la memoria"
LOCATE 15,50:PRINT "del sistema."
LOCATE 17,50:PRINT "F4. FINALIZACION"
LOCATE 18,54:PRINT "Se termina con este"
LOCATE 19,50:PRINT "programa. Vuelve a DOS"
PREGUNTE:
    KB$=INKEY$:IF KB$="" THEN PREGUNTE
    IF LEN(KB$)=2 THEN KB$=RIGHT$(KB$,1)
    IF KB$=CHR$(59) THEN PROGRAMAR
    IF KB$=CHR$(60) THEN END
    IF KB$=CHR$(61) THEN gosub BORRADO
    IF KB$=CHR$(62) THEN SALIDAS ELSE PREGUNTE
PROGRAMAR:
AD$=STRING$(26,"-"):BB$=STRING$(3,"*"):color 4,0
LOCATE 2,10:PRINT "▯";AD$;"▯":LOCATE 3,10:PRINT
"▯";SPC(26);"▯"
LOCATE 4,10:PRINT "▯";AD$;"▯"
color 14,0:LOCATE 3,14:PRINT BB$;" PROGRAMANDO ";BB$
a:
A$=STRING$(25,"="):color 15,1
LOCATE 14,10:PRINT "▯";A$;"▯"
FOR I=15 TO 17
LOCATE 1,10:PRINT "▯";SPC(25);"▯":NEXT
LOCATE 18,10:PRINT"▯";A$;"▯":color 31,0
locate 20,14:print "ESPERE UN MOMENTO"
b:
color 10,1
LOCATE 15,17:PRINT "Revisando la"
LOCATE 16,17:PRINT "memoria del"
LOCATE 17,19:PRINT "Sistema":goto variables
  **LLAMAR AL PROGRAMA DE PRUEBA DE MEMORIA**
  CONTADOR=0
  BOTAFOGO:
  EDER=0
  B=36:GOSUB SOCRATES
  IF EDER=0 THEN TONINHO
  LINEAMALA:
  IF CONTADOR>6 THEN WRONG
  GOTO BOTAFOGO
  WRONG:
  LOCATE 15,17:PRINT" EXISTEN "
  LOCATE 16,17:PRINT"PROBLEMAS EN"
  LOCATE 17,17:PRINT" LA LINEA "

```

```

                LOCATE      20,14:PRINT"REVISE      CONEXIONES":GOTO
LIMPIEPANTALLA
                WRONG1:
                LOCATE 16,17:PRINT "OLVIDESE. LA"
                LOCATE 17,17:PRINT "MEMORIA SE FREGO"
                XS=INPUT$(1):GOTO LIMPIEPANTALLA
                TONINHO:
                IF H<>85 THEN WRONG1
                DELAY 1
LOCATE 15,17:PRINT " Realizando ":locate 16,17:print "una
prueba de"
locate 17,19:print " línea "
                '**LLAMAR AL PROGRAMA DE PRUEBA DE LINEA**
                CONTADOR=0
                PROGRAMAR2:
                EDER=0
                B=32:GOSUB NELINHO
                IF EDER=0 THEN VALDO
                IF CONTADOR>6 THEN LINEAMALA
                GOTO PROGRAMAR2
                VALDO:
                IF DATOSMALOS>10 THEN PRINT "NO ES RECOMENDABLE
PROGRAMAR"
                :GOTO limpiepantalla
                DELAY 1
20:
LOCATE 15,17:PRINT " Ejecutando":locate 16,17:print "
1a "
locate 17,17:print "Autocalibración"
                '**LLAMAR AL PROGRAMA DE CALIBRACION**
                CONTADOR=0
                PALMEIRAS:
                EDER=0
                B=10:GOSUB NELINHO
                IF EDER=0 THEN CEREZO
                IF CONTADOR>6 THEN LINEAMALA
                GOTO PALMEIRAS
                CEREZO:
                IF VOLT(17)<1434 THEN PRINT "LOW BATT":GOTO
LIMPIEPANTALLA
                '**REALIZAR UN ARCHIVO DE LOS DATOS DE CALIBRACION
                J=1
                FOR I=1 TO 16 STEP 2
                VOLT(J)=VOLT(I)/16
                J=J+1
                NEXT
                ' AJUSTAR LOS DATOS DE CALIBRACION
                A=0-VOLT(1)
                B=&H200-VOLT(2)
                C=&H400-VOLT(3)
                D=&H600-VOLT(4)
                E=&HA00-VOLT(5)
                F=&HC00-VOLT(6)
                G=&HE00-VOLT(7)

```

```

H=&HFFF-VOLT(8)
IF CONTADOR1>48 THEN ZESERGIO
IF A>32 OR A<-32 THEN CONTADOR1=CONTADOR1+1:GOTO
PALMEIRAS
IF CONTADOR1>48 THEN ZESERGIO
IF B>32 OR B<-32 THEN CONTADOR1=CONTADOR1+1:GOTO
PALMEIRAS
IF CONTADOR1>48 THEN ZESERGIO
IF C>32 OR C<-32 THEN CONTADOR1=CONTADOR1+1:GOTO
PALMEIRAS
IF CONTADOR1>48 THEN ZESERGIO
IF D>32 OR D<-32 THEN CONTADOR1=CONTADOR1+1:GOTO
PALMEIRAS
IF CONTADOR1>48 THEN ZESERGIO
IF E>32 OR E<-32 THEN CONTADOR1=CONTADOR1+1:GOTO
PALMEIRAS
IF CONTADOR1>48 THEN ZESERGIO
IF F>32 OR F<-32 THEN CONTADOR1=CONTADOR1+1:GOTO
PALMEIRAS
IF CONTADOR1>48 THEN ZESERGIO
IF G>32 OR G<-32 THEN CONTADOR1=CONTADOR1+1:GOTO
PALMEIRAS
IF CONTADOR1>48 THEN ZESERGIO
IF H>32 OR H<-32 THEN CONTADOR1=CONTADOR1+1:GOTO
PALMEIRAS
PASO:
' ABRIR ARCHIVO PARA ALMACENAR LOS CODIGOS DE
CALIBRACION
OPEN "CAL.91" FOR OUTPUT AS #1
WRITE #1,A,B,C,D,E,F,G,H
CLOSE:GOTO VARIABLES
ZESERGIO:
PRINT "LA PARTE ANALOGICA DEL SISTEMA inservible":GOTO
limpiepantalla
VARIABLES:
COLOR 15,0:Z$=STRING$(20,"█"):LOCATE 20,14:PRINT Z$
CLEAR
A$=STRING$(25,"="):COLOR 15,1
LOCATE 14,10:PRINT "┌";A$;"┐"
FOR I=15 TO 17
LOCATE I,10:PRINT "│";SPC(25);"│":NEXT
LOCATE 18,10:PRINT"└";A$;"┘":color 10,1
LOCATE 15,14:PRINT "Cuantas variables va":locate 17,17:print
"
"
LOCATE 16,17:INPUT "a medir:",VAR
IF (VAR>2) OR (VAR=0) THEN VARIABLES
color 15,1
LOCATE 11,10:PRINT "┌";A$;"┐"
FOR I=12 TO 16
LOCATE I,10:PRINT "│";SPC(25);"│":NEXT
LOCATE 17,10:PRINT "└";A$;"┘"
B$=STRING$(2,"█"):COLOR 15,0
FOR I=12 TO 18
LOCATE I,8:PRINT B$:NEXT

```

```

B1$=STRING$(25,"")
LOCATE 18,10:PRINT B1$:locate 18,35:color 15,0:print ""
B2:
color 10,1
LOCATE 12,14:PRINT "Variable a medir:"
LOCATE 13,14:PRINT "1. TEMPERATURA."
LOCATE 14,14:PRINT "2. PRESION."
LOCATE 15,14:PRINT "3. AMBAS."
LOCATE 16,14:INPUT "Cual elige:",I
IF (I=0) OR (I>3) THEN SOUND 200,2:GOTO B2
IF (VAR=1) AND (I=3) THEN A1
IF (VAR=2) AND ((I=1) OR (I=2)) THEN A2
IF I=1 THEN M(1)=0:T$="TEMPERATURA": GOTO B1
IF I=2 THEN M(1)=1:P$="PRESION": GOTO B1
IF I=3 THEN M(1)=2:T$="TEMPERATURA":P$="PRESION":GOTO B1:ELSE B2
A1:
LOCATE 20,8:PRINT"SOLO PUEDE ELEGIR UNA VARIABLE":GOTO
B2
A2:
LOCATE 20,8:PRINT"USTED HA PROGRAMADO 2 VARIABLES":GOTO
B2
B1:
COLOR 15,0:LOCATE 20,8:PRINT STRING$(31,""):COLOR 10,1
LOCATE 12,14:PRINT "Cuanto debe esperar"
locate 13,14:print "el sistema antes de"
locate 14,14:print "inicializar la toma"
locate 16,11:print "Escoja de 1 a 30 minutos"
locate 15,14:input "de datos:",TIEMP01
if (TIEMP01>30) OR (TIEMP01=0) then SOUND 200,2:GOTO
B1
D1=(TIEMP01*600)-10
DATO=D1/256:M(3)=INT(DATO):DATO1=(DATO-M(3))*256:M(2)=INT(D
ATO1)
RETARDOS:
c:
locate 12,14:print "Cada cuanto el siste-"
locate 13,14:print "ma tomará un nuevo"
locate 16,11:print "Escoja de 3 a 30 minutos"
locate 15,14:print " "
locate 14,14:print " ":locate 14,14:input
"dato:",TIEMPO2
IF (TIEMPO2<3) OR (TIEMPO2>30) THEN SOUND 200,2:GOTO
c
d:
locate 12,12:print "El sistema tomará datos:"
locate 13,14:print "1. SIMPLE. "
locate 14,14:print "2. PROMEDIO. "
locate 16,11:print " "
locate 15,14:input "Cual elige:",I
IF (I=0) OR (I>2) THEN SOUND 200,2:GOTO d
IF I=1 THEN M(6)=0:S$="SIMPLE": GOTO e
IF I=2 THEN M(6)=1:PR$="PROMEDIO": ELSE d

```

```

e:
C$=STRING$(30,"█"):color 15,0
LOCATE 11,10:PRINT C$:LOCATE 12,8:PRINT "█":color 15,1
LOCATE 12,10:PRINT "┘";A$;"┘"
FOR I=13 TO 15
LOCATE I,10:PRINT "|";SPC(25);"|":NEXT
LOCATE 16,10:PRINT "└";A$;"└"
FOR I=13 TO 17
color 15,0:LOCATE I,8:PRINT B$:NEXT
LOCATE 17,10:PRINT B$:locate 17,35:color 15,0:print "██████████"
LOCATE 18,8:PRINT C$
MUESTRAS:
color 10,1
if M(1)=0 then m=20 else m=30
n=1/m*(TIEMPO2*600-10)+1:n=int(n)
LOCATE 13,12:PRINT "Cuántas muestras quiere"
locate 15,12:print "Puede tomar de 1 a";n
locate 14,12:input "tomar:",d:memoria=4000
IF (d=0) OR (d>n) THEN SOUND 1200,2:GOTO MUESTRAS
IF d>=10 THEN d=10
IF (M(6)=1) AND (M(1)=0) THEN ZZ=1:M(7)=d
IF (M(6)=1 AND (M(1)=1 OR M(1)=2)) THEN
ZZ=2:M(7)=d*2
IF M(1)=0 THEN LUISINHO
D2=(TIEMPO2*600)-30*(d-1)-10:GOTO DIDI
LUISINHO:
D2=(TIEMPO2*600)-20*(d-1)-10
DIDI:
DATO=D2/256:M(5)=INT(DATO):DATO1=(DATO-M(5))*256:M(4)=INT(D
ATO1)
IF (M(6)=0) AND (M(1)=0) THEN M(7)=d:ZZ=M(7):GOTO f
IF (M(6)=0 AND (M(1)=1 OR M(1)=2)) THEN
M(7)=d*2:ZZ=M(7)
f:
A$=STRING$(38,"-"):color 15,1
LOCATE 5,4:PRINT "┘";A$;"┘"
FOR I=6 TO 20
color 15,1:LOCATE I,4:PRINT "|";SPC(38);"|":
color 15,0:LOCATE I,2:PRINT B$:NEXT
color 15,0
LOCATE 21,2:PRINT "█";:color 15,1:print "└";A$;"└"
color 15,0
B1$=STRING$(40,"█"):LOCATE 22,2:PRINT B1$
color 10,1:ri=0
locate 6,7:print "Cuanto tiempo piensa dejar la sonda"
locate 7,7:print "en el pozo:"
locate 8,7:input "HORAS=",HH
locate 9,7:input "MINUTOS=",MM:HH=HH*60:TT=HH+MM
DATOS=(TT/TIEMPO2)*ZZ:DATOS=INT(DATOS):DATOS1=DATOS/2
ESTACIONES=(TT/TIEMPO2):ESTACIONES=INT(ESTACIONES)
locate 10,6:print "Con los datos que ha programado con"
locate 11,6:print "anterioridad, y con este tiempo de"
locate 12,6:print "trabajo de la sonda, usted puede"
locate 13,6:print "tomar:"

```

```

        IF VAR=2 THEN DATOS=INT(DATOS1):GOTO XX
locate 14,6:print DATOS;" DATOS DE ";T$;P$
locate 16,6:print "TOTAL=";DATOS
DD:
locate 17,6:PRINT"El Número de Estaciones es:";ESTACIONES
locate 18,6:print "Si desea obtener más datos se le re-"
locate 19,6:print "comienda aumentar el tiempo que per-"
locate 20,6:print "manecerá la sonda en el pozo."
DD1:
LOCATE 23,6:PRINT "PRESIONE ALGUNA TECLA PARA
CONTINUAR":x$=input$(1)
if ri=1 then f
GOTO g
XX:
locate 14,6:PRINT DATOS;" DATOS DE ";T$;" Y"
locate 15,6:PRINT DATOS;" DATOS DE ";P$
locate 16,6:PRINT "TOTAL: ";DATOS*2;" DATOS"
locate 17,6:print "El Número de Estaciones es:";ESTACIONES
IF (DATOS*2)>(memoria/VAR) THEN FULL else DD
FULL:
locate 18,6:PRINT"No existe capacidad suficiente"
locate 19,6:PRINT"en la memoria del sistema para"
locate 20,6:PRINT"almacenar esa cantidad de datos.":ri=1:goto
DD1
g:
U$=STRING$(44," ")
COLOR 15,0:LOCATE 23,6:PRINT U$
color 15,1
locate 12,4:print "[";A$;"]"
locate 13,4:print "[";spc(38);"]"
LOCATE 14,4:PRINT "[";A$;"]"
color 10,1
locate 13,7:input "Cuantas estaciones va a realizar:",EST
IF EST=0 THEN g
X=EST*M(7)
IF M(6)=1 THEN IF M(1)=0 THEN X=EST ELSE X=EST*2
IF (X>memoria/VAR) OR (X>DATOS*VAR) THEN memoriafull
IF X=0 THEN g ELSE h
memoriafull:
LOCATE 20,6:PRINT STRING$(32," ")
COLOR 17,0:LOCATE 18,6:PRINT"NO PUEDO ACEPTAR ESA CANTIDAD"
COLOR 17,0:LOCATE 19,6:PRINT"DE DATOS. INTENTELO DE NUEVO."
DELAY 3
COLOR 14,1:LOCATE 18,6:PRINT STRING$(36," ")
LOCATE 19,6:PRINT STRING$(36," "):goto g
h:
Z$=STRING$(44," "):color 15,0
LOCATE 5,4:PRINT Z$:LOCATE 6,2:PRINT Z$:LOCATE 7,2:PRINT " "
color 15,1
LOCATE 7,4:PRINT "┌";STRING$(40,"=");┌"
LOCATE 20,4:PRINT "└";STRING$(40,"=");└"
FOR I=8 TO 19
LOCATE I,4:PRINT "|";SPC(40);"|":NEXT
COLOR 15,0:LOCATE 21,2:PRINT STRING$(41," ")

```

```

LOCATE 21,43:PRINT "█":LOCATE 22,2:PRINT Z$
' =====RESUMEN=====
' =====
DATO=X/256:M(9)=INT(DATO):DATO1=(DATO-M(9))*256:M(8)=INT(DA
TOL)
U$=STRING$(44,"█")
COLOR 15,0:LOCATE 23,6:PRINT U$:color 10,1
locate 8,14:print "**** RESUMEN ****"
locate 10,6:print "Variable a medir:"
color 14,1:locate 11,6:print T$;" ";P$:color 10,1
locate 12,6:print "Tiempo antes de empezar :";:color 14,1:?
TIEMPO1;" Minutos"
color 10,1:locate 13,6:print "Tiempo entre adquisición:";
:color 14,1:? TIEMPO2;" Minutos"
color 10,1:locate 14,6:print "El Sistema tomará
datos:";:color 14,1:? S$;PR$
color 10,1:locate 15,6:print "La cantidad de muestras a tomar
es:";:color 14,1:? d
color 10,1:locate 16,6:print "Total de estaciones
es:";:color 14,1:? EST
color 10,1:locate 17,6:print "Total de datos a
tomar:";:color 14,1:? X
color 10,1:locate 19,6:print "Quiere modificar la
programación S/N?"
MODI:
Q$=INKEY$:IF Q$="" THEN MODI
IF (Q$="S") OR (Q$="s") THEN AA
IF (Q$="N") OR (Q$="n") THEN i ELSE MODI
AA:
A$=STRING$(44,"█")
FOR I=5 TO 22
COLOR 15,0:LOCATE I,2:PRINT A$:NEXT
GOTO VARIABLES
i:
color 15,0
FOR I=7 TO 11
LOCATE I,2:PRINT Z$:NEXT
FOR I=16 TO 21
LOCATE I,2:PRINT Z$:NEXT
LOCATE 12,2:PRINT "█"
color 15,1:LOCATE 12,4:PRINT "||";A$;"||":locate 12,44:color
15,0:print "█"
FOR I=13 TO 14
color 15,1
LOCATE I,4:PRINT "||";SPC(38);"||"
locate I,44:color 15,0:print "█":NEXT
color 15,1:LOCATE 15,4:PRINT "||";A$;"||":locate 15,44:color
15,0:print "█"
color 15,0:LOCATE 16,2:PRINT B1$
color 10,1:locate 13,11:print "Se están transmitiendo los"
locate 14,13:print "datos de programación.":DELAY 1
DIRCEU:
CONTADOR=0
MANE:

```

```

EDER=0
B=34:GOSUB NELINHO
IF EDER=0 THEN j
IF CONTADOR>6 THEN LINEAMALA
GOTO MANE

j:
locate 13,11:print "Quiere verificar los datos"
locate 14,12:print " de programación S/N ?"
VERIFICAR:
Q$=INKEY$:IF Q$="" THEN VERIFICAR
IF (Q$="S") OR (Q$="s") THEN k
IF (Q$="N") OR (Q$="n") THEN JJ ELSE VERIFICAR
JJ:
Y$=STRING$(42," ")
FOR I=1 TO 17
COLOR 15,0:LOCATE I,2:PRINT Y$:NEXT
GOTO PREGUNTE
k:
GILMAR:
CONTADOR=0
JAIRZINHO:
EDER=0
B=35:GOSUB NELINHO
IF EDER=0 THEN DECODIFICACION1
IF CONTADOR>6 THEN LINEAMALA
GOTO JAIRZINHO
DECODIFICACION1:
' *** DECODIFICACION DE LOS CODIGOS DE LA SONDA ***
IF N(1)=0 THEN V1$=T$
IF N(1)=1 THEN V2$=P$
IF N(1)=2 THEN V1$=T$:V2$=P$
IF N(6)=0 THEN V3$=S$
IF N(6)=1 THEN V3$=PR$
IF N(1)=0 THEN VAVA=N(7) ELSE VAVA=N(7)/2
IF N(1)=0 THEN CARECA=(N(7)-1)*20 ELSE
CARECA=(N(7)/2-1)*30
DC1=N(2)+N(3)*256:TEMPOC1=(DC1+10)/600
DC2=(N(4)+N(5)*256)+CARECA+10
TEMPOC2=DC2/600
TOTAL=N(8)+N(9)*256:TOTAL=INT(TOTAL+.1)
SS:
' PRESENTACION EN PANTALLA DE LOS CODIGOS PROGRAMADOS
' Y LOS CODIGOS ALMACENADOS EN LA SONDA

30:
color 15,0
FOR I=13 TO 16
LOCATE I,2:PRINT " ":NEXT
FOR I=2 TO 6
LOCATE I,48:PRINT STRING$(30," "):NEXT
color 15,1:LOCATE 7,3:PRINT "||";STRING$(73,"=");"|"
FOR I=8 TO 18
LOCATE I,3:PRINT "||";SPC(73);"||":NEXT
LOCATE 19,3:PRINT "||";STRING$(73,"=");"|"
color 15,0:LOCATE 20,48:PRINT STRING$(30," ")

```

```

LOCATE 21,48:PRINT STRING$(30," ")
color 14,1
locate 8,9:print "Datos programados":locate 8,48:print "Datos
de la sonda"
locate 10,5:print "Variable a medir:"
color 11,1:locate 11,5:print T$;" ";P$:color 14,1
locate 12,5:print "Tiempo antes de empezar :";:color 11,1:?
TIEMPO1;"Min"
color 14,1:locate 13,5:print "Tiempo entre adquisición:";_
:color 11,1:? TIEMPO2;"Min":color 14,1
locate 14,5:print "El Sistema tomará datos :";:color 11,1:?
S$;PR$
color 14,1:locate 15,5:print "Total de muestras a
tomar:";:color 11,1:? d
color 14,1:locate 16,5:print "Total de estaciones son
:";:color 11,1:? EST
color 14,1:locate 17,5:print "Total de datos a tomar
:";:color 11,1:? X
color 14,1:locate 10,43:print "Variable a medir:"
color 15,1:locate 11,43:print V1$;" ";V2$:color 14,1
locate 12,43:print "Tiempo antes de empezar :";:color 15,1:?
TEMPOC1;"Min"
color 14,1:locate 13,43:print "Tiempo entre adquisición:";_
:color 15,1:? TEMPOC2;"Min"
color 14,1:locate 14,43:print "El Sistema tomará datos
:";:color 15,1:? V3$
color 14,1:locate 15,43:print "Total de muestras a
tomar:";:color 15,1:? VAVA
color 14,1:locate 16,43:print "Total de estaciones son
:";:color 15,1:? EST
color 14,1:locate 17,43:print "Total de datos a tomar
:";:color 15,1:? TOTAL
color 14,1:LOCATE 21,17:PRINT "ESTA SATISFECHO CON LOS DATOS
RECIBIDOS S/N ?"
SATIS:
Q$=INKEY$:IF Q$="" THEN SATIS
IF Q$="S" OR Q$="s" THEN CUADRO
IF Q$="N" OR Q$="n" THEN VENT ELSE SATIS
VENT:
COLOR 15,0:LOCATE 21,17:PRINT STRING$(45," ")
COLOR 15,1:LOCATE 12,17:PRINT "┌";STRING$(41,"-");"┐"
FOR I=13 TO 14
LOCATE I,17:PRINT "│";SPC(41);"│":NEXT
LOCATE 15,17:PRINT "└";STRING$(41,"-");"┘":color 14,1
locate 13,23:print "TRANSMITIENDO NUEVAMENTE LOS"
locate 14,26:print "DATOS DE PROGRAMACION."
DIRCEU1:
CONTADOR=0
MANE1:
EDER=0
B=34:GOSUB NELINHO
IF EDER=0 THEN k
IF CONTADOR>6 THEN LINEAMALA
GOTO MANE1

```

```

BORRADO:
color 15,1
A$=STRING$(30,"-"):B$=STRING$(2,"██")
LOCATE 9,10:PRINT "┌";A$;"┐"
FOR I=10 TO 12
color 15,0:LOCATE I,8:PRINT B$;:color 15,1:?
"└";SPC(30);"┘":NEXT
color 15,0:LOCATE 13,8:PRINT B$;:color 15,1:? "┌";A$;"┐"
color 15,0:LOCATE 14,8:PRINT STRING$(31,"██"):COLOR 30,1
LOCATE 10,17:PRINT "BORRANDO LOS DATOS"
LOCATE 12,21:PRINT "ANTERIORES.":DELAY 3
Y$=STRING$(36,"██"):COLOR 15,0
FOR I=8 TO 14
LOCATE I,8:PRINT Y$:NEXT
CONTADOR=0
ROMARIO:
EDER=0
B=37:GOSUB NELINHO
IF EDER=0 THEN RETURN
IF CONTADOR>6 THEN LINEAMALA
GOTO ROMARIO

```

SALIDAS:

```

CONTADOR=0
JUNIOR:
B=4:GOSUB NELINHO
IF EDER=0 THEN JORGINHO
IF CONTADOR>6 THEN PRINT "NO PUDE COMENZAR":GOTO
LIMPIEPANTALLA
GOTO JUNIOR
JORGINHO:
CLS
LOCATE 8,8:PRINT "EL SISTEMA TOMARA DATOS A
LAS: ";TIMES
END
SOCRATES:
OUT &H3FB,&H80
OUT &H3F9,0
OUT &H3F8,&H60
OUT &H3FB,&H1F
OUT &H3F9,0
NELINHO:
TAFFAREL=10
IF B=32 THEN PRUEBAL
IF B=34 THEN PROGRAMACAO
IF B=35 THEN VERIFICACAO
IF B=10 THEN CALIBRACAO
IF B=37 THEN RESETRAM
IF B=4 THEN FIN
IF B=13 THEN FINE
IF B=36 THEN TESTRAM
FINE:
END

```

TX:

```

OUT &H3FC,2

```

```

OUT &H3F8,B
GOTO MAYORIAL
RX:
OUT &H3FC,1
FOR C=1 TO 1000
Z=INP(&H3FD):T=Z AND 1
IF T=1 THEN ENTRADA
NEXT
ENTRADA:
W=INP(&H3F8)
RETURN

```

MAYORIAL:

```

FOR I=1 TO TAFFAREL
GOSUB RX
D(I)=W
NEXT
XUXA=1
MAZINHO:
SUN=0
FOR I=1 TO TAFFAREL
IF D(XUXA)<>D(I) THEN VOGUE
SUN=SUN+1
VOGUE:
NEXT
IF SUN > 5 THEN SIG
XUXA=XUXA + 1
IF XUXA < 7 THEN MAZINHO
PRINT "NO HUBO MAYORIA":RETURN
SIG:
H=D(XUXA)
RETURN

```

MAYORIA2:

```

FOR I=1 TO TOPO
CODIGO(I)=D(I):NEXT
VARIABLE=1:OK=1:CONTEO=0
LAZO1:
BULLS=VARIABLE
LAZO:
DATO=CODIGO(VARIABLE):REPETIDO=0
FOR I=1 TO 10
IF DATO=CODIGO(CONTEO) THEN REPETIDO=REPETIDO+1
CONTEO=CONTEO+1
NEXT
IF REPETIDO>5 THEN RESULTADO
VARIABLE=VARIABLE+1
DATO=CODIGO(VARIABLE):CONTEO=BULLS
GOTO LAZO
RESULTADO:
N(OK)=DATO
OK=OK+1
IF OK>ZICO THEN DECODIFICACION
VARIABLE=BULLS+10:CONTEO=VARIABLE
GOTO LAZO1
DECODIFICACION:

```

```

RETURN
PRUEBAL:
TAFFAREL=10:GOSUB TX
IF H <> B THEN CONTADOR=CONTADOR+1:EDER=1:RETURN
OUT &H3FC,2
FOR I=1 TO 100
RIO:
Z=INP(&H3FD):T=Z AND 32
IF T<>32 THEN RIO
OUT &H3F8,I
NEXT
FOR I=1 TO 101
C=0
DO UNTIL C=1000
Y=INP(&H3FD):T=Y AND 1
IF T=1 THEN BRANCO
C=C+1
LOOP
CONTADOR=CONTADOR+1:EDER=1
RETURN
BRANCO:
W= INP (&H3F8)
D(I)=W
NEXT
DATOSMALOS=0
FOR I=1 TO 101
IF D(I)<>I-1 THEN DATOSMALOS=DATOSMALOS+1
NEXT
RETURN
PROGRAMACAO:
FOR I=1 TO 9
A(I)=M(I):NEXT
GOSUB TX
IF H <> B THEN CONTADOR=CONTADOR+1:EDER=1:RETURN
OUT &H3FC,2
FOR I=1 TO 9
FOR J=1 TO 10
BRASIL:
Z=INP(&H3FD):T=Z AND 32
IF T<>32 THEN BRASIL
OUT &H3F8,A(I)
NEXT J:NEXT I
RETURN
CALIBRACAO:
GOSUB TX
IF H<>B THEN CONTADOR=CONTADOR+1:EDER=1:RETURN
TOPO=180:delay 128
GOSUB GOIAS
FOR I=1 TO ZICO STEP 2
VOLT(I) = N(I) + 256*N(I+1)
NEXT
RETURN
GOIAS:
FOR I=1 TO TOPO

```

```

C=0
DO UNTIL C=1000
Z=INP(&H3FD):T=Z AND 1
IF T=1 THEN FALCAO
C=C + 1
LOOP
FALCAO:
D(I)=INP(&H3F8)
NEXT
ZICO=TOPO/10
GOSUB MAYORIA2
RETURN
VERIFICACAO:
GOSUB TX
IF H <> B THEN CONTADOR=CONTADOR+1:EDER=1:RETURN
TOPO=90:ZICO=TOPO/10
GOSUB GOIAS
RETURN
FIN:
GOSUB TX
IF H <> B THEN CONTADOR=CONTADOR+1:EDER=1:RETURN
CLS:PRINT "LA SONDA EMPIEZA A TRABAJAR EN ESTE
MOMENTO"
RETURN
RESETRAM:
GOSUB TX
IF H<>B THEN CONTADOR=CONTADOR+1:EDER=1:RETURN
RETURN
TESTRAM:
GOSUB TX
IF H<>B THEN CONTADOR=CONTADOR+1:EDER=1:RETURN
FOR I=1 TO 1000
Z=INP(&h3FD):T=Z AND 1
IF T=1 THEN PAULINHO
NEXT
CONTADOR=CONTADOR+1:EDER=1
RETURN
PAULINHO:
GOSUB MAYORIA1
RETURN
LIMPIEPANTALLA:
W$=STRING$(35," "):COLOR 15,0
DELAY 3
FOR I=14 TO 20
LOCATE I,1:PRINT W$:NEXT
GOTO PREGUNTE

```

APENDICE C
HOJAS TECNICAS

8088 8-BIT HMOS MICROPROCESSOR 8088/8088-2

- 8-Bit Data Bus Interface
- 6-Bit Internal Architecture
- Direct Addressing Capability to 1 Mbyte of Memory
- Direct Software Compatibility with 8086 CPU
- 4-Word by 16-Bit Register Set with Symmetrical Operations
- 4 Operand Addressing Modes

- Byte, Word, and Block Operations
- 8-Bit and 16-Bit Signed and Unsigned Arithmetic in Binary or Decimal, Including Multiply and Divide
- Two Clock Rates:
 - 5 MHz for 8088
 - 8 MHz for 8088-2
- Available in EXPRESS
 - Standard Temperature Range
 - Extended Temperature Range

Intel® 8088 is a high performance microprocessor implemented in N-channel, depletion load, silicon gate technology (HMOS), and packaged in a 40-pin CERDIP package. The processor has attributes of both 8- and 16-bit microprocessors. It is directly compatible with 8086 software and 8080/8085 hardware and peripherals.

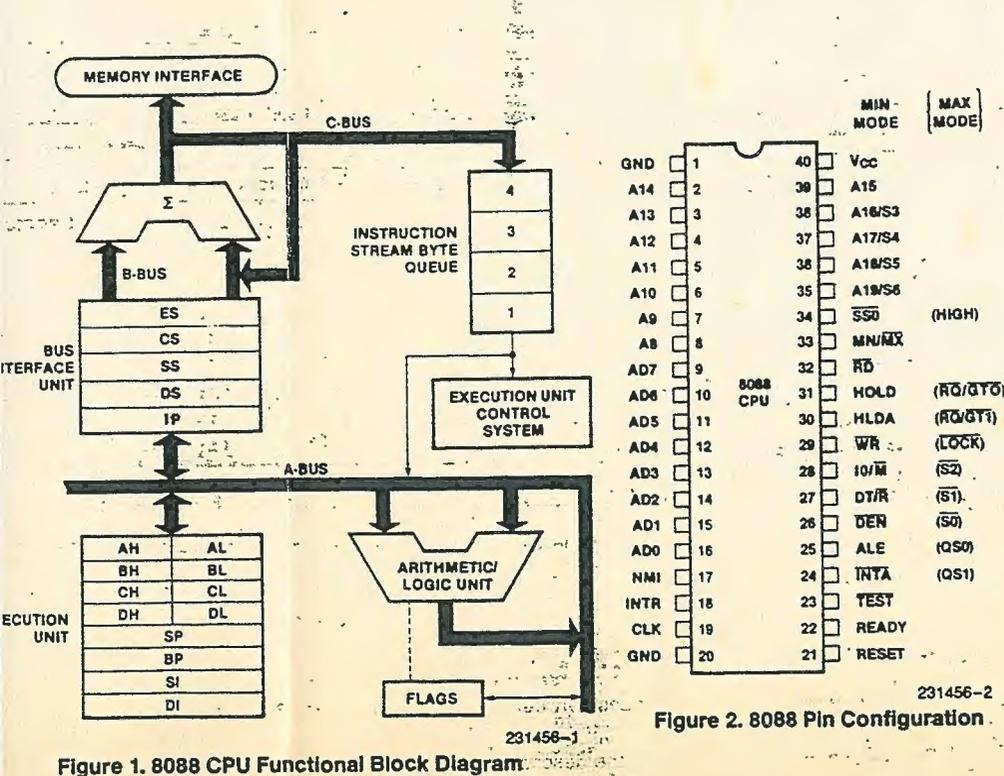


Table 1. Pin Description

The following pin function descriptions are for 8088 systems in either minimum or maximum mode. The "bus" in these descriptions is the direct multiplexed bus interface connection to the 8088 (without regard to additional bus buffers).

Symbol	Pin No.	Type	Name and Function															
AD7-AD0	9-16	I/O	ADDRESS DATA BUS: These lines constitute the time multiplexed memory/I/O address (T1) and data (T2, T3, Tw, T4) bus. These lines are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge".															
A15-A8	2-8, 39	O	ADDRESS BUS: These lines provide address bits 8 through 15 for the entire bus cycle (T1-T4). These lines do not have to be latched by the processor to remain valid. A15-A8 are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge".															
A19/S6, A18/S5, A17/S4, A16/S3	35-38	O	ADDRESS/STATUS: During T1, these are the four most significant address lines for memory operations. During I/O operations, these lines are LOW. During memory and I/O operations, status information is available on these lines during T2, T3, Tw, and T4. S6 is always low. The status of the interrupt enable flag bit (S5) is updated at the beginning of each clock cycle. S4 and S3 are encoded as shown. This information indicates which segment register is presently being used for data accessing. These lines float to 3-state OFF during local bus "hold acknowledge".															
			<table border="1" style="width: 100%;"> <thead> <tr> <th>S4</th> <th>S3</th> <th>Characteristics</th> </tr> </thead> <tbody> <tr> <td>0 (LOW)</td> <td>0</td> <td>Alternate Data</td> </tr> <tr> <td>0</td> <td>1</td> <td>Stack</td> </tr> <tr> <td>1 (HIGH)</td> <td>0</td> <td>Code or None</td> </tr> <tr> <td>1</td> <td>1</td> <td>Data</td> </tr> </tbody> </table>	S4	S3	Characteristics	0 (LOW)	0	Alternate Data	0	1	Stack	1 (HIGH)	0	Code or None	1	1	Data
S4	S3	Characteristics																
0 (LOW)	0	Alternate Data																
0	1	Stack																
1 (HIGH)	0	Code or None																
1	1	Data																
RD	32	O	READ: Read strobe indicates that the processor is performing a memory or I/O read cycle, depending on the state of the IO/M pin (pin 28). This signal is used to read devices which reside on the 8088 local bus. RD is active LOW during T2, T3 and Tw of any read cycle, and guaranteed to remain HIGH in T2 until the 8088 local bus has floated. This signal floats to 3-state OFF in "hold acknowledge".															
READY	22	I	READY: is the acknowledgement from the addressed memory or I/O device that it will complete the data transfer. The RDY signal from memory or I/O is synchronized by the 8284 clock generator to form the processor's READY. This signal is active HIGH. The 8088 READY input is not synchronized. Correct operation is not guaranteed if the set up and hold times are not met.															
INTR	18	I	INTERRUPT REQUEST: is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.															
TEST	23	I	TEST: input is examined by the "wait for test" instruction. If the TEST input is LOW, execution continues, otherwise the processor waits in the "idle" state. This input is synchronized internally during each clock cycle on the leading edge of CLK.															

Table 1. Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
NMI	17	I	NON-MASKABLE INTERRUPT: is an edge triggered input which causes a type 2 interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. This input is internally synchronized.
RESET	21	I	RESET: causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. It restarts execution, as described in the instruction set description, when RESET returns LOW. RESET is internally synchronized.
CLK	19	I	CLOCK: provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
V _{CC}	40		V_{CC}: is the +5V ± 10% power supply pin.
GND	1, 20		GND: are the ground pins.
MN/M \bar{X}	33	I	MINIMUM/MAXIMUM: indicates what mode the processor is to operate in. The two modes are discussed in the following sections.

The following pin function descriptions are for the 8088 minimum mode (i.e., MN/M \bar{X} = V_{CC}). Only the pin functions which are unique to minimum mode are described; all other pin functions are as described above.

Symbol	Pin No.	Type	Name and Function
IO/M	28	O	STATUS LINE: is an inverted maximum mode $\bar{S}2$. It is used to distinguish a memory access from an I/O access. IO/M becomes valid in the T4 preceding a bus cycle and remains valid until the final T4 of the cycle (I/O = HIGH, M = LOW). IO/M floats to 3-state OFF in local bus "hold acknowledge".
WR	29	O	WRITE: strobe indicates that the processor is performing a write memory or write I/O cycle, depending on the state of the IO/M signal. WR is active for T2, T3, and Tw of any write cycle. It is active LOW, and floats to 3-state OFF in local bus "hold acknowledge".
INTA	24	O	INTA: is used as a read strobe for interrupt acknowledge cycles. It is active LOW during T2, T3, and Tw of each interrupt acknowledge cycle.
ALE	25	O	ADDRESS LATCH ENABLE: is provided by the processor to latch the address into an address latch. It is a HIGH pulse active during clock low of T1 of any bus cycle. Note that ALE is never floated.
DT/R	27	O	DATA TRANSMIT/RECEIVE: is needed in a minimum system that desires to use a data bus transceiver. It is used to control the direction of data flow through the transceiver. Logically, DT/R is equivalent to $\bar{S}1$ in the maximum mode, and its timing is the same as for IO/M (T = HIGH, R = LOW). This signal floats to 3-state OFF in local "hold acknowledge".
DEN	26	O	DATA ENABLE: is provided as an output enable for the data bus transceiver in a minimum system which uses the transceiver. DEN is active LOW during each memory and I/O access, and for INTA cycles. For a read or INTA cycle, it is active from the middle of T2 until the middle of T4, while for a write cycle, it is active from the beginning of T2 until the middle of T4. DEN floats to 3-state OFF during local bus "hold acknowledge".

Table 1. Pin Description (Continued)

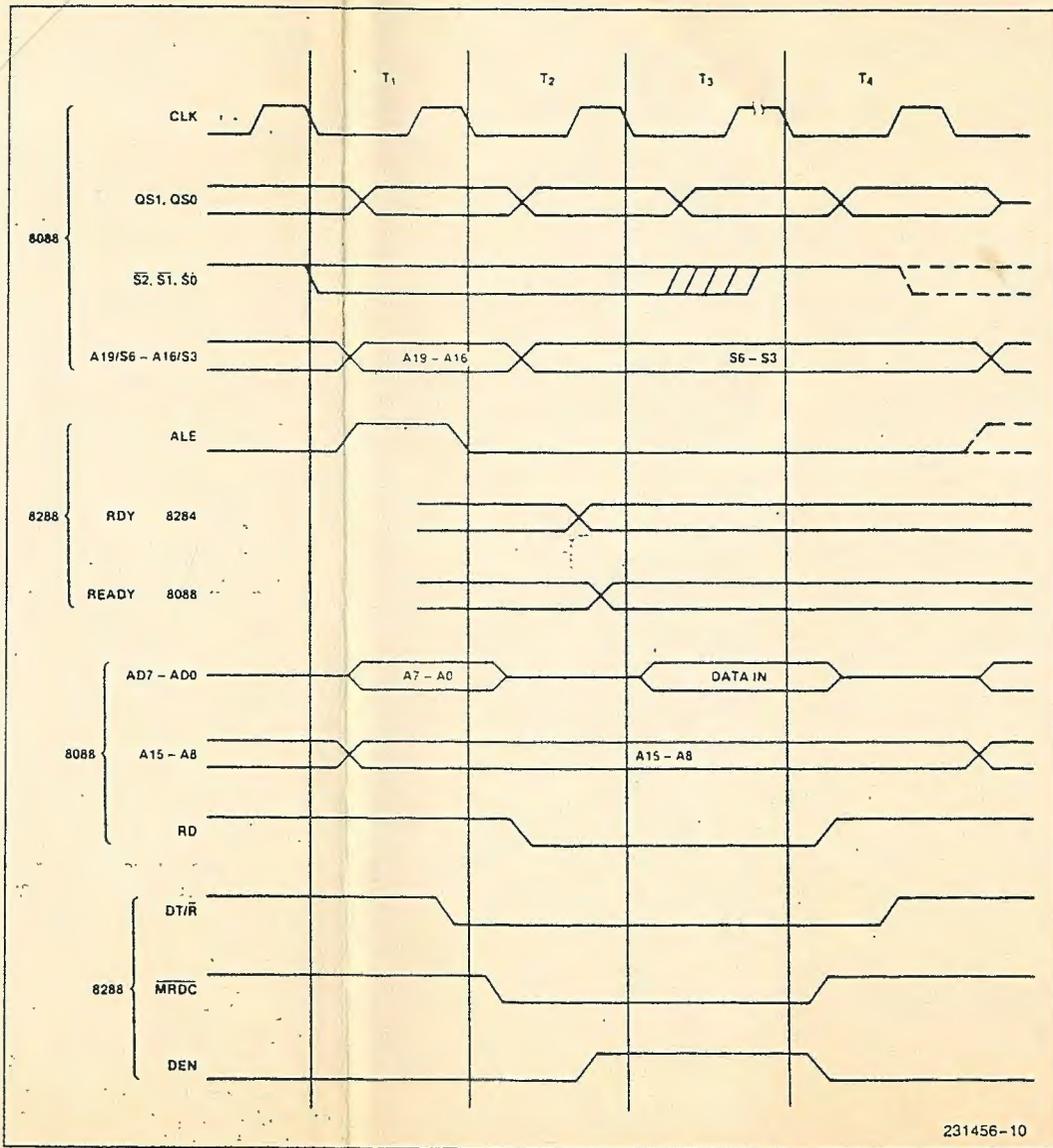
Symbol	Pin No.	Type	Name and Function																																				
HOLD, HLDA	31, 30	I, O	HOLD: indicates that another master is requesting a local bus "hold acknowledged, HOLD must be active HIGH. The processor receiving request will issue HLDA (HIGH) as an acknowledgement, in the middle of the next clock cycle. Simultaneous with the issuance of HLDA the processor drives the local bus and control lines. After HOLD is detected as being LOW, the processor lowers HLDA, and when the processor needs to run another instruction will again drive the local bus and control lines. HOLD and HLDA have pull-up resistors. Hold is not an asynchronous input. External synchronization should be used so that the system cannot otherwise guarantee the set up time.																																				
SSO	34	O	STATUS LINE: is logically equivalent to $\bar{S}0$ in the maximum mode. The combination of SSO, IO/M and DT/R allows the system to complete current bus cycle status.																																				
			<table border="1"> <thead> <tr> <th>IO/M</th> <th>DT/R</th> <th>SSO</th> <th>Characteristics</th> </tr> </thead> <tbody> <tr> <td>1(HIGH)</td> <td>0</td> <td>0</td> <td>Interrupt Acknowledge</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Read I/O Port</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Write I/O Port</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Halt</td> </tr> <tr> <td>0(LOW)</td> <td>0</td> <td>0</td> <td>Code Access</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Read Memory</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Write Memory</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Passive</td> </tr> </tbody> </table>	IO/M	DT/R	SSO	Characteristics	1(HIGH)	0	0	Interrupt Acknowledge	1	0	1	Read I/O Port	1	1	0	Write I/O Port	1	1	1	Halt	0(LOW)	0	0	Code Access	0	0	1	Read Memory	0	1	0	Write Memory	0	1	1	Passive
IO/M	DT/R	SSO	Characteristics																																				
1(HIGH)	0	0	Interrupt Acknowledge																																				
1	0	1	Read I/O Port																																				
1	1	0	Write I/O Port																																				
1	1	1	Halt																																				
0(LOW)	0	0	Code Access																																				
0	0	1	Read Memory																																				
0	1	0	Write Memory																																				
0	1	1	Passive																																				

The following pin function descriptions are for the 8088/8288 system in maximum mode (i.e., MN/M \bar{X} = GND). Only the pin functions which are unique to maximum mode are described; all other pin functions are as described above.

Symbol	Pin No.	Type	Name and Function																																				
$\bar{S}2, \bar{S}1, \bar{S}0$	26-28	O	STATUS: is active during clock high of T4, T1, and T2, and is returned to a passive state (1, 1, 1) during T3 or during Tw when READY is HIGH. These signals are used by the 8288 bus controller to generate all memory and I/O access signals. Any change by $\bar{S}2, \bar{S}1,$ or $\bar{S}0$ during T4 is used to indicate the end of a bus cycle, and the return to the passive state in T3 and Tw is used to indicate the end of a bus cycle. These signals float to 3-state OFF during "hold acknowledge". During the next clock cycle after RESET becomes active, these signals are active HIGH. This first clock, they float to 3-state OFF.																																				
			<table border="1"> <thead> <tr> <th>$\bar{S}2$</th> <th>$\bar{S}1$</th> <th>$\bar{S}0$</th> <th>Characteristics</th> </tr> </thead> <tbody> <tr> <td>0(LOW)</td> <td>0</td> <td>0</td> <td>Interrupt Acknowledge</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Read I/O Port</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Write I/O Port</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Halt</td> </tr> <tr> <td>1(HIGH)</td> <td>0</td> <td>0</td> <td>Code Access</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Read Memory</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Write Memory</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Passive</td> </tr> </tbody> </table>	$\bar{S}2$	$\bar{S}1$	$\bar{S}0$	Characteristics	0(LOW)	0	0	Interrupt Acknowledge	0	0	1	Read I/O Port	0	1	0	Write I/O Port	0	1	1	Halt	1(HIGH)	0	0	Code Access	1	0	1	Read Memory	1	1	0	Write Memory	1	1	1	Passive
$\bar{S}2$	$\bar{S}1$	$\bar{S}0$	Characteristics																																				
0(LOW)	0	0	Interrupt Acknowledge																																				
0	0	1	Read I/O Port																																				
0	1	0	Write I/O Port																																				
0	1	1	Halt																																				
1(HIGH)	0	0	Code Access																																				
1	0	1	Read Memory																																				
1	1	0	Write Memory																																				
1	1	1	Passive																																				

• \overline{SSO} provides the $\overline{S0}$ status information in the minimum mode. This output occurs on pin 34 in minimum mode only. $\overline{DT/R}$, $\overline{IO/\overline{M}}$, and \overline{SSO} provide the complete bus status in minimum mode.

- $\overline{IO/\overline{M}}$ has been inverted to be compatible with the MCS-85 bus structure.
- ALE is delayed by one clock cycle in the minimum mode when entering HALT, to allow the status to be latched with ALE.



231456-10

Figure 10. Medium Complexity System Timing

ABSOLUTE MAXIMUM RATINGS*

- Ambient Temperature Under Bias 0°C to +70°C
- Case Temperature (Plastic) 0°C to +95°C
- Case Temperature (CERDIP) 0°C to +75°C
- Storage Temperature -65°C to +150°C
- Voltage on Any Pin with Respect to Ground -1.0 to +7V
- Power Dissipation 2.5 Watt

*Notice: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating for functional operation of the device at other conditions above those indicated in the additional sections of this specification is not intended to imply that the device is intended for use in extended periods of exposure to absolute maximum rating.

D.C. CHARACTERISTICS

($T_A = 0^\circ\text{C}$ to 70°C , T_{CASE} (Plastic) = 0°C to 95°C , T_{CASE} (CERDIP) = 0°C to 75°C , $T_A = 0^\circ\text{C}$ to 55°C and $T_{\text{CASE}} = 0^\circ\text{C}$ to 75°C for P8088-2 only
 T_A is guaranteed as long as T_{CASE} is not exceeded)

($V_{CC} = 5V \pm 10\%$ for 8088, $V_{CC} = 5V \pm 5\%$ for 8088-2 and Extended Temperature EXPR)

Symbol	Parameter	Min	Max	Units	Test Co
V_{IL}	Input Low Voltage	-0.5	+0.8	V	(Note 1)
V_{IH}	Input High Voltage	2.0	$V_{CC} + 0.5$	V	(Notes 1, 2)
V_{OL}	Output Low Voltage		0.45	V	$I_{OL} = 2.0 \text{ mA}$
V_{OH}	Output High Voltage	2.4		V	$I_{OH} = -40 \mu\text{A}$
I_{CC}	Power Supply Current: 8088 8088-2 P8088		340 350 250	mA	$T_A = 25^\circ\text{C}$
I_{LI}	Input Leakage Current		± 10	μA	$0V \leq V_{IN} \leq V_{CC}$
I_{LO}	Output and I/O Leakage Current		± 10	μA	$0.45V \leq V_{O} \leq V_{CC}$
V_{CL}	Clock Input Low Voltage	-0.5	+0.6	V	
V_{CH}	Clock Input High Voltage	3.9	$V_{CC} + 1.0$	V	
C_{IN}	Capacitance If Input Buffer (All Input Except \overline{AD}_0 - \overline{AD}_7 , $\overline{RQ/GT}$)		15	pF	$f_c = 1 \text{ MHz}$
C_{IO}	Capacitance of I/O Buffer (\overline{AD}_0 - \overline{AD}_7 , $\overline{RQ/GT}$)		15	pF	$f_c = 1 \text{ MHz}$

NOTES:

1. V_{IL} tested with $\overline{MN/\overline{MX}}$ Pin = 0V
 V_{IH} tested with $\overline{MN/\overline{MX}}$ Pin = 5V
 $\overline{MN/\overline{MX}}$ Pin is a strap Pin
2. Not applicable to $\overline{RQ/GT}_0$ and $\overline{RQ/GT}_1$ Pins (Pins 30 and 31)
3. HOLD and HLDA I_{LI} Min = 30 μA , Max = 500 μA

CHARACTERISTICS

= 0°C to 70°C, T_{CASE} (Plastic) = 0°C to 95°C, T_{CASE} (CERDIP) = 0°C to 75°C,
 0°C to 55°C and T_{CASE} = 0°C to 80°C for P8088-2 only
 guaranteed as long as T_{CASE} is not exceeded)
 = 5V ± 10% for 8088, V_{CC} = 5V ± 5% for 8088-2 and Extended Temperature EXPRESS)

MUM COMPLEXITY SYSTEM TIMING REQUIREMENTS

Symbol	Parameter	8088		8088-2		Units	Test Conditions
		Min	Max	Min	Max		
CL	CLK Cycle Period	200	500	125	500	ns	
CLCH	CLK Low Time	118		68		ns	
CLCH	CLK High Time	69		44		ns	
1CH2	CLK Rise Time		10		10	ns	From 1.0V to 3.5V
2CL2	CLK Fall Time		10		10	ns	From 3.5V to 1.0V
CL	Data in Setup Time	30		20		ns	
CLDX	Data in Hold Time	10		10		ns	
CLVCL	RDY Setup Time into 8284 (Notes 1, 2)	35		35		ns	
CLR1X	RDY Hold Time into 8284 (Notes 1, 2)	0		0		ns	
CLHCH	READY Setup Time into 8088	118		68		ns	
CLRYX	READY Hold Time into 8088	30		20		ns	
CLLCL	READY Inactive to CLK (Note 3)	-8		-8		ns	
CLCH	HOLD Setup Time	35		20		ns	
CLCH	INTR, NMI, TEST Setup Time (Note 2)	30		15		ns	
CLH	Input Rise Time (Except CLK)		20		20	ns	From 0.8V to 2.0V
CLL	Input Fall Time (Except CLK)		12		12	ns	From 2.0V to 0.8V

A.C. CHARACTERISTICS (Continued)

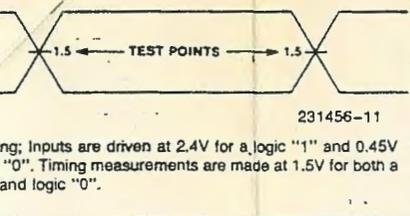
TIMING RESPONSES

Symbol	Parameter	8088		8088-2		Units	Test Conditions
		Min	Max	Min	Max		
TCLAV	Address Valid Delay	10	110	10	60	ns	
TCLAX	Address Hold Time	10		10		ns	
TCLAZ	Address Float Delay	TCLAX	80	TCLAX	50	ns	
TLHLL	ALE Width	TCLCH - 20		TCLCH - 10		ns	
TCLLH	ALE Active Delay		80		50	ns	
TCHLL	ALE Inactive Delay		85		55	ns	
TLLAX	Address Hold Time to ALE Inactive	TCHCL - 10		TCHCL - 10		ns	
TCLDV	Data Valid Delay	10	110	10	60	ns	
TCHDX	Data Hold Time	10		10		ns	
TWHDX	Data Hold Time after \overline{WR}	TCLCH - 30		TCLCH - 30		ns	
TCVCTV	Control Active Delay 1	10	110	10	70	ns	
TCHCTV	Control Active Delay 2	10	110	10	60	ns	
TCVCTX	Control Inactive Delay	10	110	10	70	ns	
TAZRL	Address Float to READ Active	0		0		ns	
TCLRL	\overline{RD} Active Delay	10	165	10	100	ns	
TCLRH	\overline{RD} Inactive Delay	10	150	10	80	ns	
TRHAV	\overline{RD} Inactive to Next Address Active	TCLCL - 45		TCLCL - 40		ns	
TCLHAV	HLDA Valid Delay	10	160	10	100	ns	
TRLRH	\overline{RD} Width	2TCLCL - 75		2TCLCL - 50		ns	
TWLWH	\overline{WR} Width	2TCLCL - 60		2TCLCL - 40		ns	
TAVAL	Address Valid to ALE Low	TCLCH - 60		TCLCH - 40		ns	
TOLOH	Output Rise Time		20		20	ns	From 0.8V to 2.0V
TOHOL	Output Fall Time		12		12	ns	From 2.0V to 0.8V

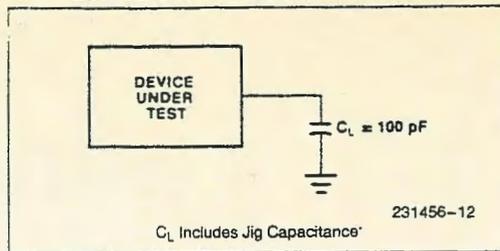
NOTES:

- Signal at 8284A shown for reference only. See 8284A data sheet for the most recent specifications.
- Set up requirement for asynchronous signal only to guarantee recognition at next CLK.
- Applies only to T2 state (8 ns into T3 state).

TESTING INPUT, OUTPUT WAVEFORM

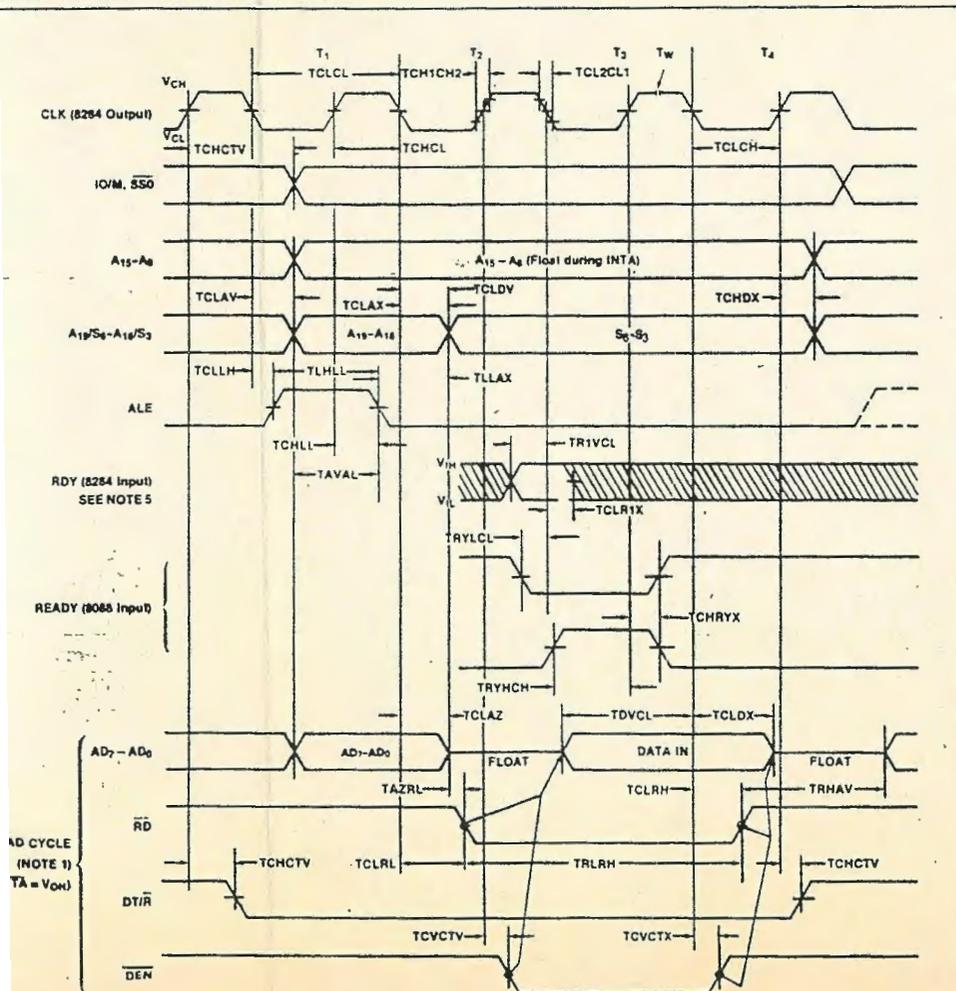


A.C. TESTING LOAD CIRCUIT



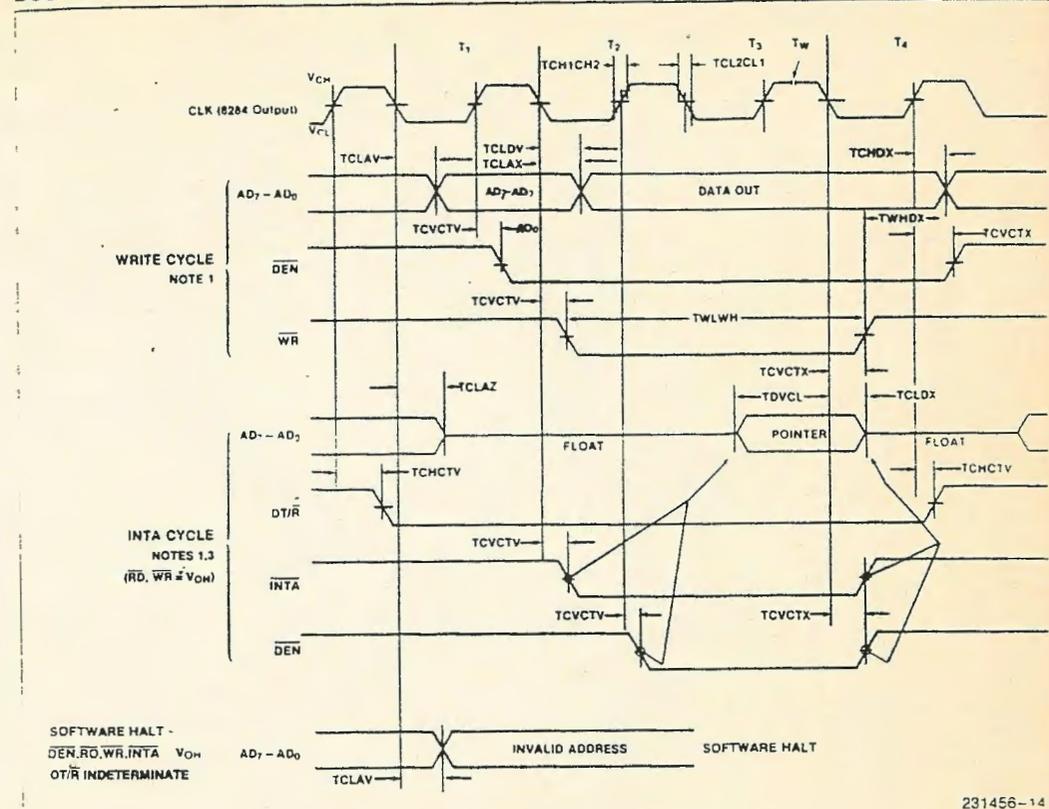
WAVEFORMS

MINIMUM MODE SYSTEM



WAVEFORMS (Continued)

BUS TIMING—MINIMUM MODE SYSTEM (Continued)



NOTES:

1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
2. RDY is sampled near the end of T_2 , T_3 , T_w to determine if T_w machine states are to be inserted.
3. Two INTA cycles run back-to-back. The 8088 local ADDR/DATA bus is floating during both INTA cycles. Control signals are shown for the second INTA cycle.
4. Signals at 8284 are shown for reference only.
5. All timing measurements are made at 1.5V unless otherwise noted.



8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
 - 24 Programmable I/O Pins
 - Completely TTL Compatible
 - Fully Compatible with Intel Microprocessor Families
 - Improved Timing Characteristics
 - Direct Bit Set/Reset Capability Easing Control Application Interface
 - Reduces System Package Count
 - Improved DC Driving Capability
 - Available in EXPRESS
 - Standard Temperature Range
 - Extended Temperature Range
 - 40 Pin DIP Package or 44 Lead PLCC
- (See Intel Packaging; Order Number: 231369)

The Intel 8255A is a general purpose programmable I/O device designed for use with Intel microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.

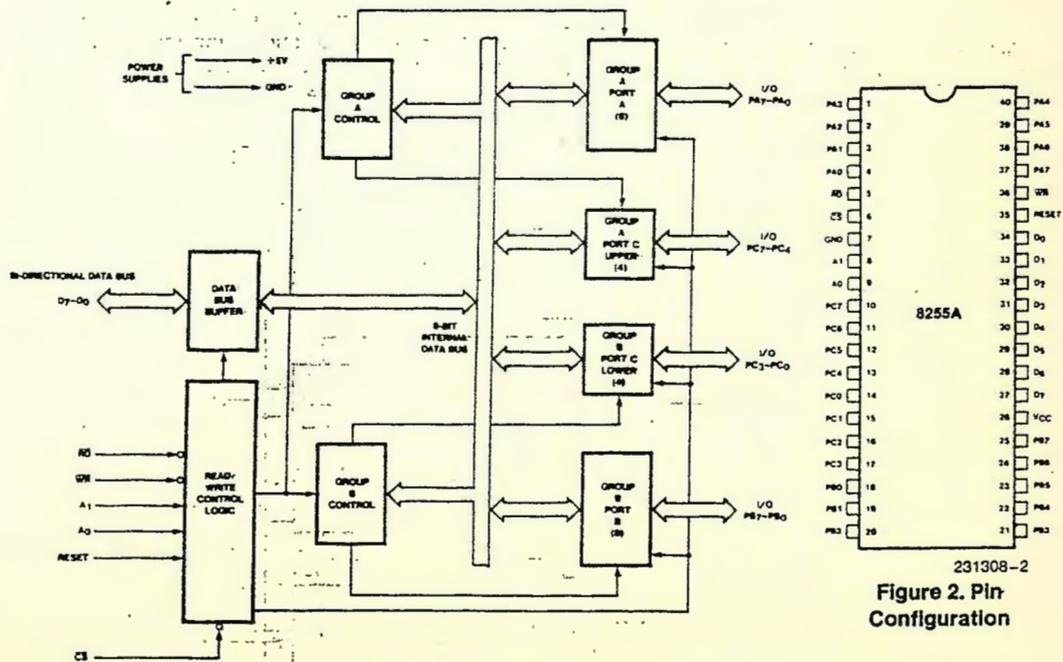


Figure 1. 8255A Block Diagram

231308-1

Figure 2. Pin Configuration

8255A FUNCTIONAL DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel microcomputer systems. Its function is that of a general-purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the

CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

(\overline{CS})

Chip Select. A "low" on this input pin enables the communication between the 8255A and the CPU.

(\overline{RD})

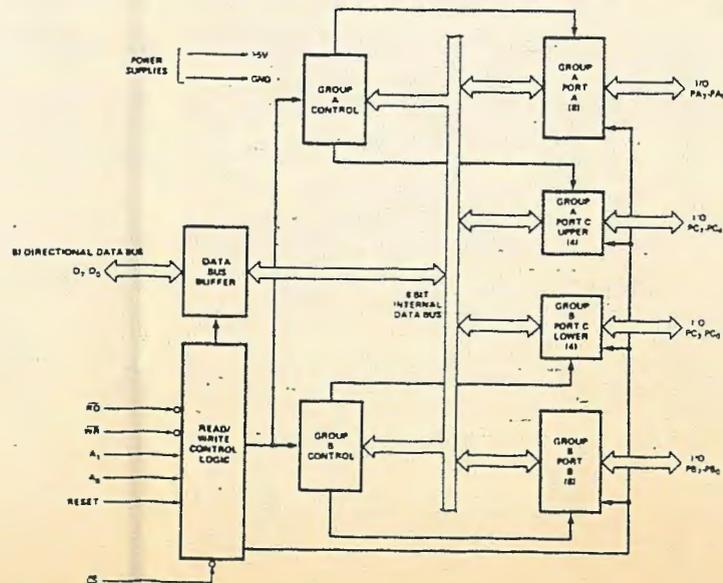
Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

(\overline{WR})

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).



8255A BASIC OPERATION

A_1	A_0	\overline{RD}	\overline{WR}	\overline{CS}	Input Operation (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
					Output Operation (WRITE)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					Disable Function
X	X	X	X	1	Data Bus → 3-State
1	1	0	1	0	Illegal Condition
X	X	1	1	0	Data Bus → 3-State

(RESET)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A—Port A and Port C upper (C_7-C_4)

Control Group B—Port B and Port C lower (C_3-C_0)

The Control Word Register can **Only** be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.

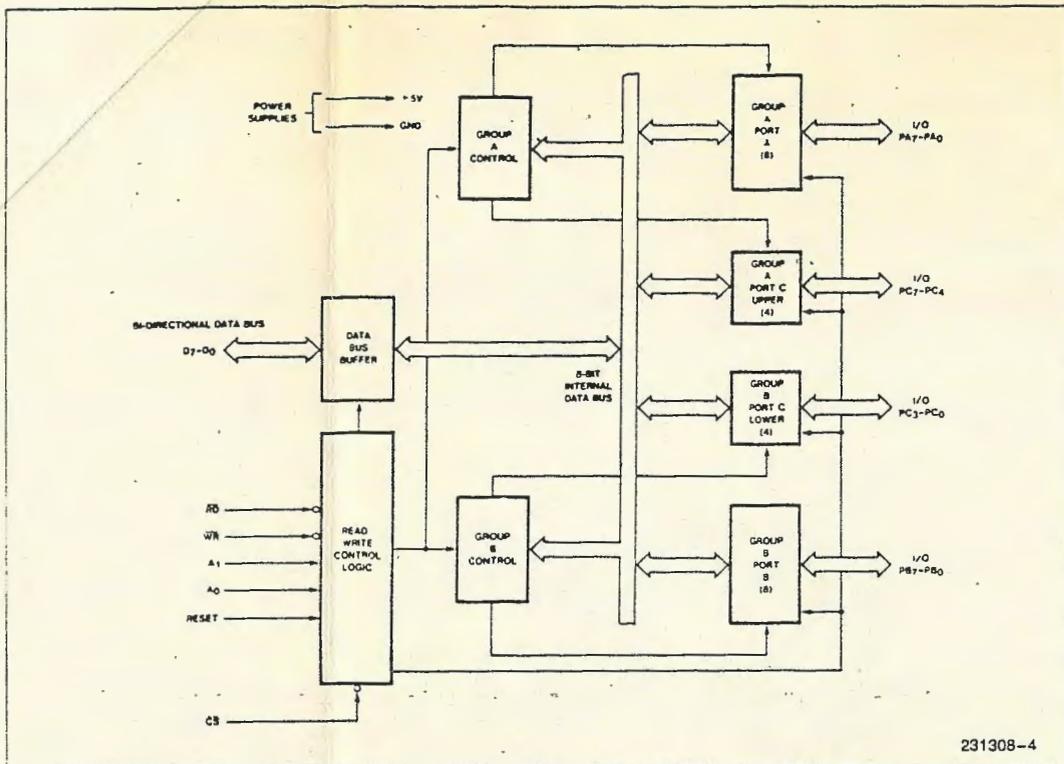
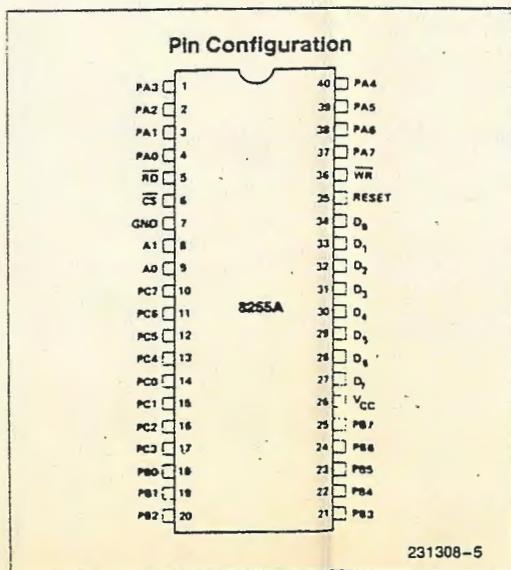


Figure 4. 8255A Block Diagram Showing Group A and Group B Control Functions



231308-5

Pin Names	
D ₇ -D ₀	Data Bus (Bi-Directional)
RESET	Reset Input
CS	Chip Select
RD	Read Input
WR	Write Input
A ₀ , A ₁	Port Address
PA ₇ -PA ₀	Port A (BIT)
PB ₇ -PB ₀	Port B (BIT)
PC ₇ -PC ₀	Port C (BIT)
V _{CC}	+ 5 Volts
GND	0 Volts

8255A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can

Mode 0—Basic Input/Output

Mode 1—Strobed Input/Output

Mode 2—Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two-portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

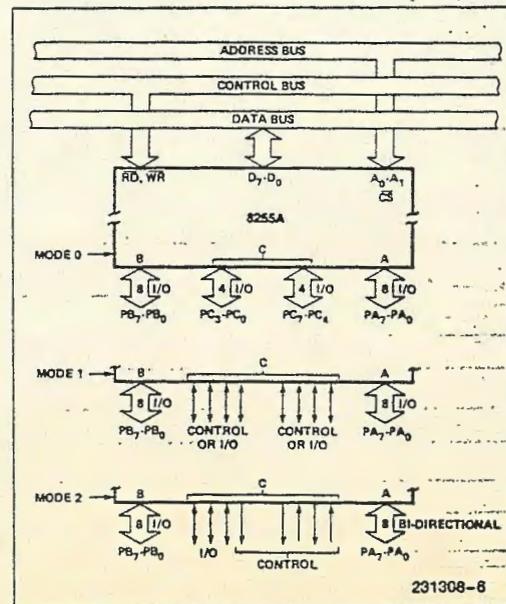


Figure 5. Basic Mode Definitions and Bus Interface

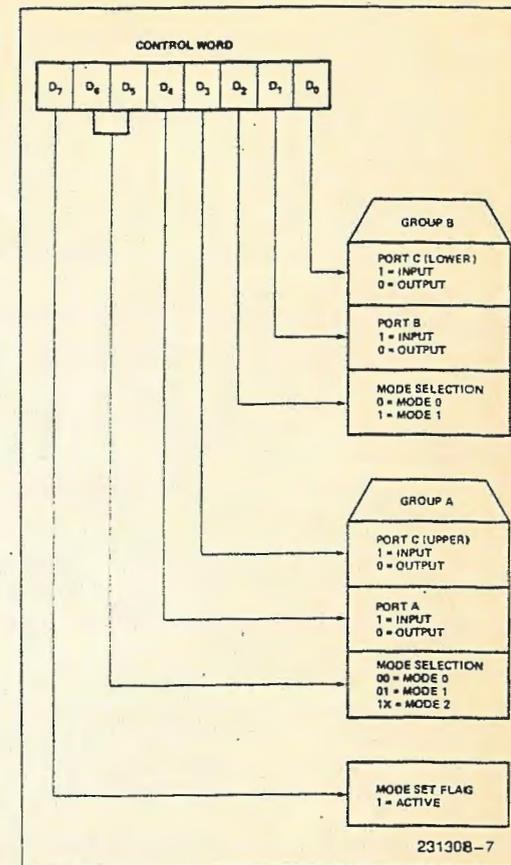


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

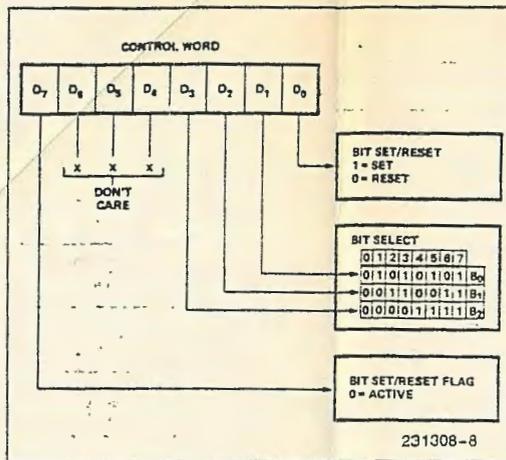


Figure 7. Bit Set/Reset Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET)—INTE is set—Interrupt enable

(BIT-RESET)—INTE is RESET—Interrupt disable

NOTE:

All Mask flip-flops are automatically reset during mode selection and device Reset.

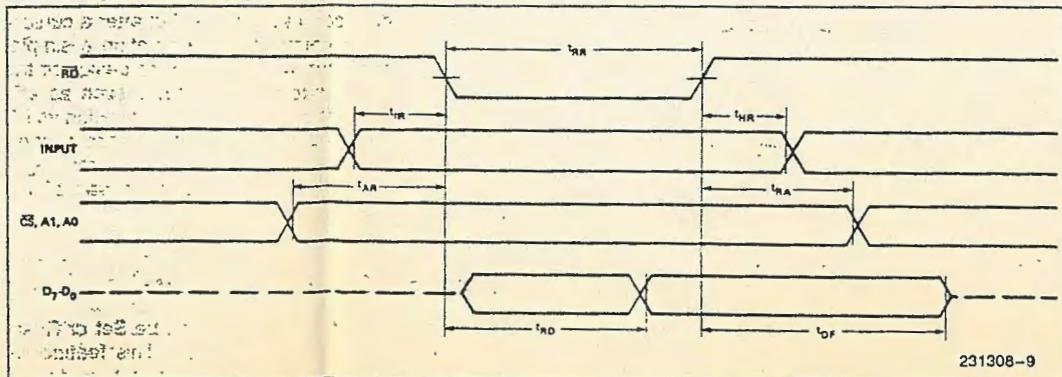
Operating Modes

MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

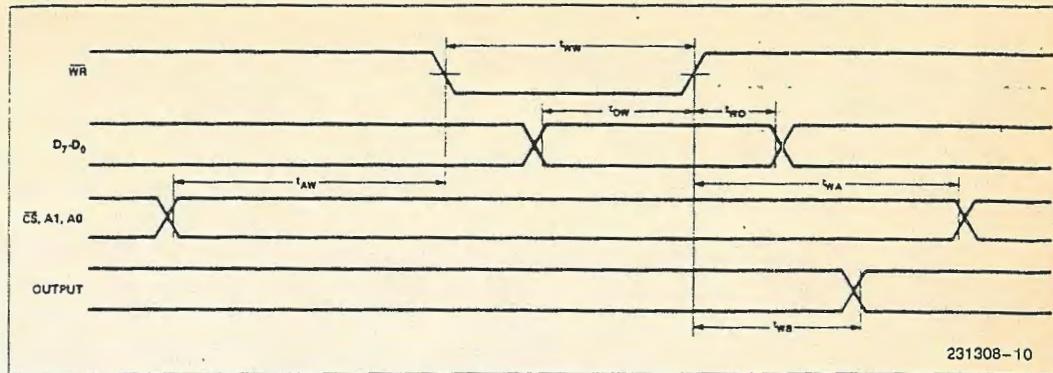
Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.

MODE 0 (BASIC INPUT)

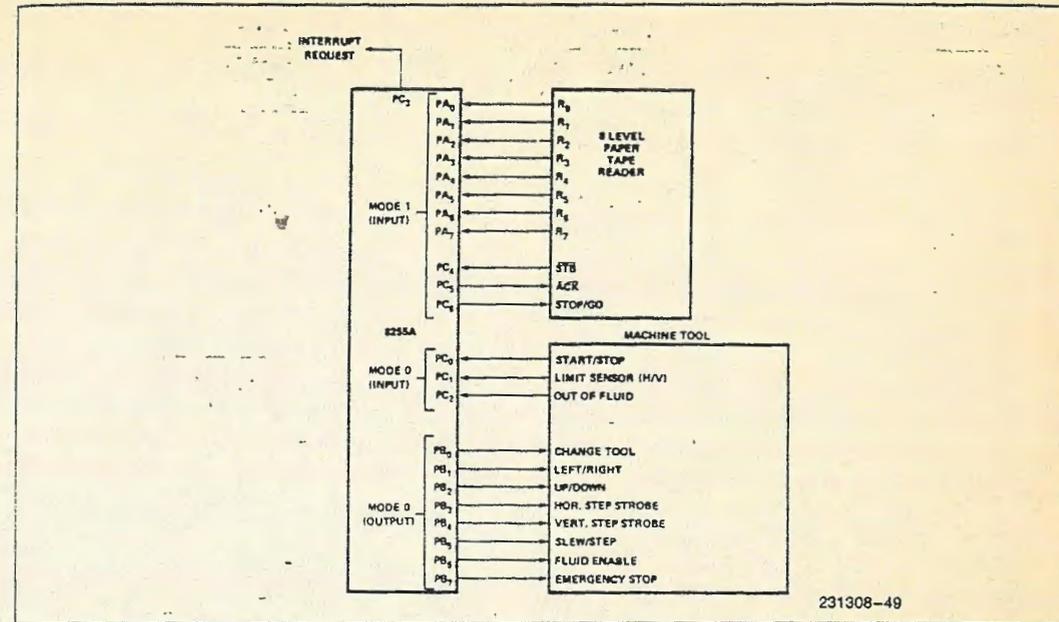


MODE 0 (BASIC OUTPUT)



MODE 0 PORT DEFINITION

A		B		Group A			Group B	
D ₄	D ₃	D ₁	D ₀	Port A	Port C (Upper)	#	Port B	Port C (Lower)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT



231308-49

Figure 25. Machine Tool Controller Interface

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias 0°C to 70°C
 Storage Temperature -65°C to +150°C
 Voltage on Any Pin
 with Respect to Ground -0.5V to +7V
 Power Dissipation 1 Watt

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS $T_A = 0^\circ\text{C to } 70^\circ\text{C}$, $V_{CC} = +5V \pm 10\%$, $GND = 0V^*$

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
$V_{OL} (DB)$	Output Low Voltage (Data Bus)		0.45*	V	$I_{OL} = 2.5 \text{ mA}$
$V_{OL} (PER)$	Output Low Voltage (Peripheral Port)		0.45*	V	$I_{OL} = 1.7 \text{ mA}$
$V_{OH} (DB)$	Output High Voltage (Data Bus)	2.4		V	$I_{OH} = -400 \mu\text{A}$
$V_{OH} (PER)$	Output High Voltage (Peripheral Port)	2.4		V	$I_{OH} = -200 \mu\text{A}$
$I_{DAR}^{(1)}$	Darlington Drive Current	-1.0	-4.0	mA	$R_{EXT} = 750\Omega$; $V_{EXT} = 1.5V$
I_{CC}	Power Supply Current		120	mA	
I_{IL}	Input Load Current		± 10	μA	$V_{IN} = V_{CC} \text{ to } 0V$
I_{OFL}	Output Float Leakage		± 10	μA	$V_{OUT} = V_{CC} \text{ to } 0.45V$

NOTE:

CAPACITANCE $T_A = 25^\circ\text{C}$, $V_{CC} = \text{GND} = 0\text{V}$

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	$f_c = 1\text{ MHz}^{(4)}$
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to GND ⁽⁴⁾

A.C. CHARACTERISTICS $T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = +5\text{V} \pm 10\%$, $\text{GND} = 0\text{V}^*$

Bus Parameters

READ

Symbol	Parameter	8255A		8255A-5		Unit
		Min	Max	Min	Max	
t_{AR}	Address Stable before READ	0		0		ns
t_{RA}	Address Stable after READ	0		0		ns
t_{RR}	READ Pulse Width	300		300		ns
t_{RD}	Data Valid from READ ⁽¹⁾		250		200	ns
t_{DF}	Data Float after READ	10	150	10	100	ns
t_{RW}	Time between READs and/or WRITEs	850		850		ns

WRITE

Symbol	Parameter	8255A		8255A-5		Unit
		Min	Max	Min	Max	
t_{AW}	Address Stable before WRITE	0		0		ns
t_{WA}	Address Stable after WRITE	20		20		ns
t_{WW}	WRITE Pulse Width	400		300		ns
t_{DW}	Data Valid to WRITE (T.E.)	100		100		ns
t_{WD}	Data Valid after WRITE	30		30		ns

OTHER TIMINGS

Symbol	Parameter	8255A		8255A-5		Unit
		Min	Max	Min	Max	
t_{WB}	WR = 1 to Output ⁽¹⁾		350		350	ns
t_{PR}	Peripheral Data before RD	0		0		ns
t_{PR}	Peripheral Data after RD	0		0		ns
t_{AK}	ACK Pulse Width	300		300		ns
t_{ST}	STB Pulse Width	500		500		ns
t_{PS}	Per. Data before T.E. of STB	0		0		ns
t_{PH}	Per. Data after T.E. of STB	180		180		ns
t_{AD}	ACK = 0 to Output ⁽¹⁾		300		300	ns
t_{KD}	ACK = 1 to Output Float	20	250	20	250	ns

A.C. CHARACTERISTICS (Continued)

OTHER TIMINGS (Continued)

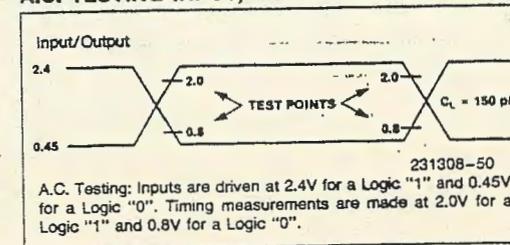
Symbol	Parameter	8255A		8255A-5		Unit
		Min	Max	Min	Max	
t_{WOB}	WR = 1 to OBF = 0 ⁽¹⁾		650		650	ns
t_{AOB}	ACK = 0 to OBF = 1 ⁽¹⁾		350		350	ns
t_{SIB}	STB = 0 to IBF = 1 ⁽¹⁾		300		300	ns
t_{RIB}	RD = 1 to IBF = 0 ⁽¹⁾		300		300	ns
t_{RIT}	RD = 0 to INTR = 0 ⁽¹⁾		400		400	ns
t_{SIT}	STB = 1 to INTR = 1 ⁽¹⁾		300		300	ns
t_{AIT}	ACK = 1 to INTR = 1 ⁽¹⁾		350		350	ns
t_{WIT}	WR = 0 to INTR = 0 ^(1,3)		850		850	ns

NOTES:

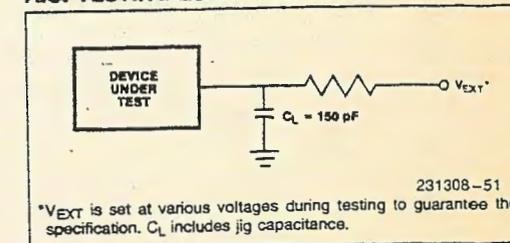
- Test Conditions: $C_L = 150\text{ pF}$.
- Period of Reset pulse must be at least $50\text{ }\mu\text{s}$ during or after power on. Subsequent Reset pulse can be 500 ns min.
- INTR \uparrow may occur as early as WR \downarrow .
- Sampled, not 100% tested.

*For Extended Temperature EXPRESS, use M8255A electrical parameters.

A.C. TESTING INPUT, OUTPUT WAVEFORM

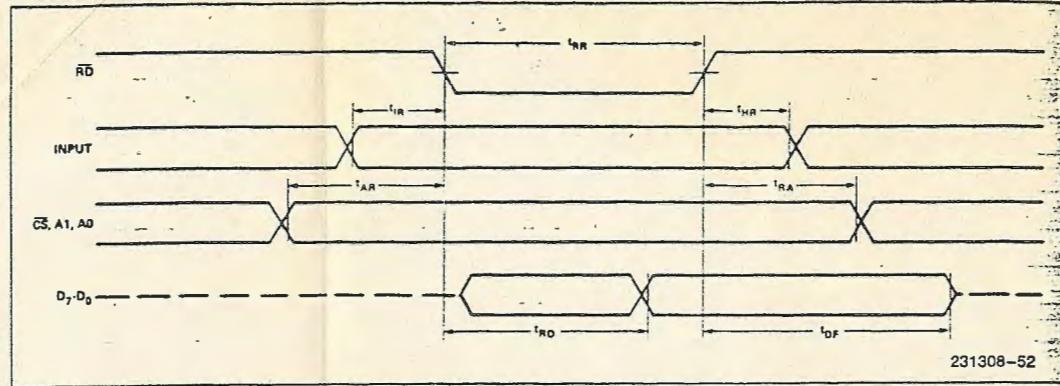


A.C. TESTING LOAD CIRCUIT

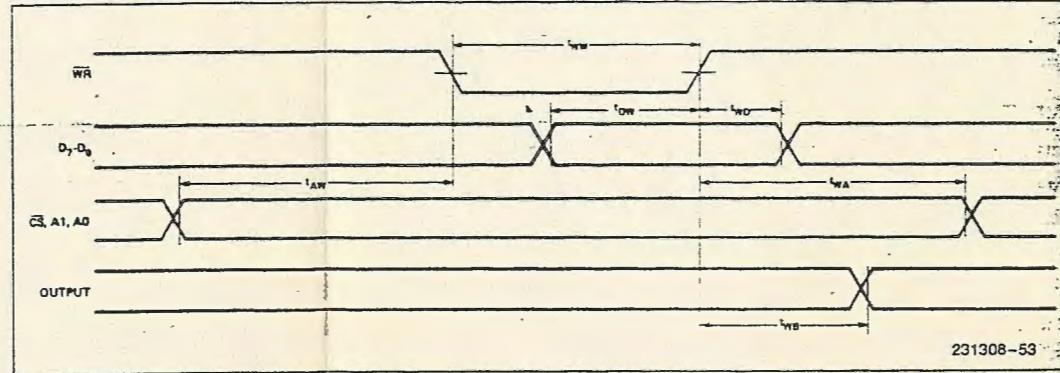


WAVEFORMS

MODE 0 (BASIC INPUT)

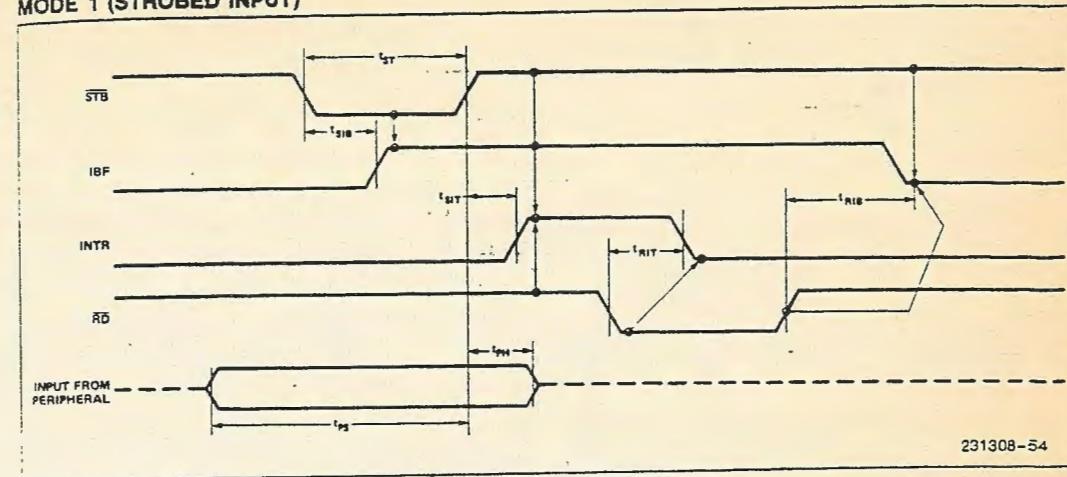


MODE 0 (BASIC OUTPUT)

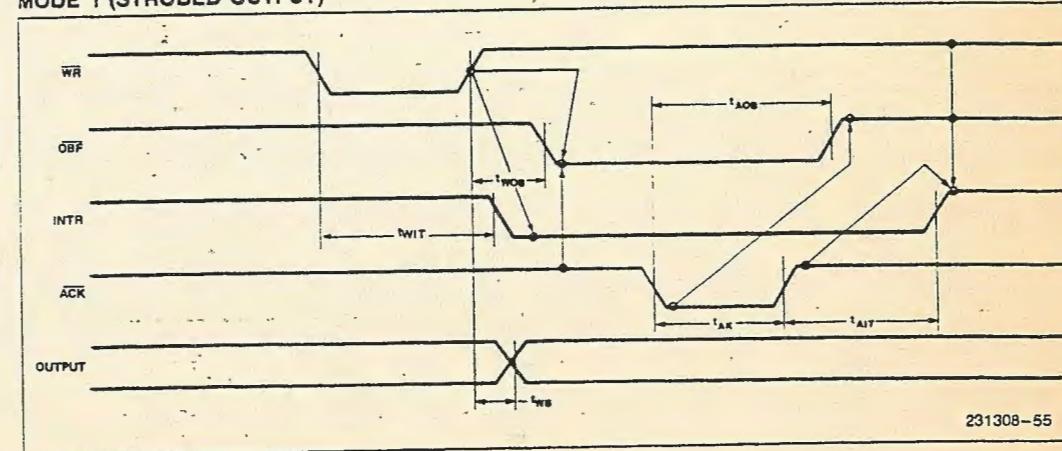


WAVEFORMS (Continued)

MODE 1 (STROBED INPUT)



MODE 1 (STROBED OUTPUT)

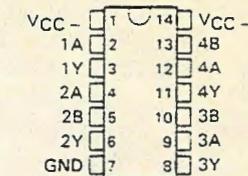


SN55188, SN75188 QUADRUPLE LINE DRIVERS

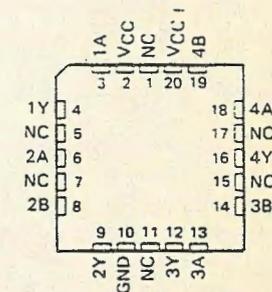
01323, SEPTEMBER 1983—REVISED SEPTEMBER 1986

- Meets Specifications of EIA RS-232-C
- Designed to be Interchangeable with Motorola MC1488
- Current-Limited Output: 10 mA Typ
- Power-Off Output Impedance: 300 Ω Min
- Slew Rate Control by Load Capacitor
- Flexible Supply Voltage Range
- Input Compatible with Most TTL Circuits

SN55188 . . . J PACKAGE
SN75188 . . . D OR J PACKAGE
(TOP VIEW)



SN55188 . . . FK
CHIP CARRIER PACKAGE
(TOP VIEW)



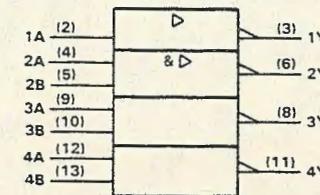
NC—No internal connection

description

The SN55188 and SN75188 are monolithic quadruple line drivers designed to interface data terminal equipment with data communications equipment in conformance with EIA Standard RS-232-C using a diode in series with each supply-voltage terminal as shown under typical applications.

The SN55188 is characterized for operation over the full military temperature range of -55°C to 125°C . The SN75188 is characterized for operation from 0°C to 70°C .

logic symbol†



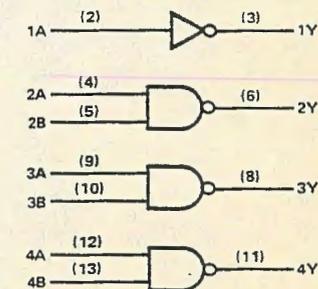
† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

FUNCTION TABLE
(DRIVERS 2 THRU 4)

A	B	Y
H	H	L
L	X	H
X	L	H

H = high level,
L = low level,
X = irrelevant

logic diagram (positive logic)



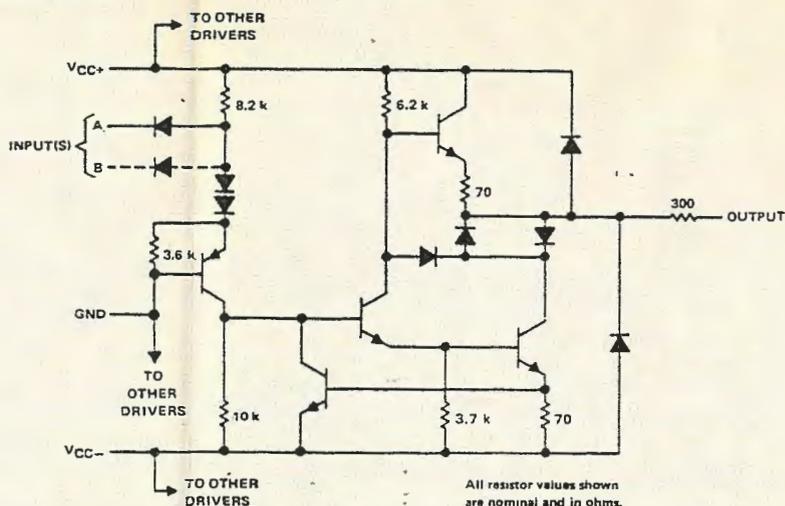
Positive logic

$$Y = \bar{A} \text{ (driver 1)}$$

$$Y = \bar{A}\bar{B} \text{ or } \bar{A} + \bar{B} \text{ (drivers 2 thru 4)}$$

Pin numbers shown are for D, J, and N packages.

schematic (each driver)



All resistor values shown are nominal and in ohms.

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

	SN55188	SN75188	UNIT
Supply voltage V_{CC+} at (or below) 25°C free-air temperature (see Notes 1 and 2)	15	15	V
Supply voltage V_{CC-} at (or below) 25°C free-air temperature (see Notes 1 and 2)	-15	-15	V
Input voltage range	-15 to 7	-15 to 7	V
Output voltage range	-15 to 15	-15 to 15	V
Continuous total dissipation at (or below) 25°C free-air temperature (see Note 2)	D package	950	mW
	FK package	1375	
	J package	1375	
	N package	1150	
Operating free-air temperature range	-55 to 125	0 to 70	°C
Storage temperature range	-65 to 150	-65 to 150	°C
Case temperature for 60 seconds	FK package	260	°C
Lead temperature 1,6 mm (1/16 inch) from case for 60 seconds	J package	300	°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds: D or N package		260	°C*

- NOTES: 1. All voltage values are with respect to the network ground terminal.
2. For operation above 25°C free-air temperature, refer to the Maximum Supply Voltage Curve, Figure 6, and the Dissipation Derating Curves in Appendix A. In the J package, SN55188 chips are alloy mounted and SN75188 chips are glass mounted.

recommended operating conditions

	SN55188			SN75188			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC+}	7.5	9	15	7.5	9	15	V
Supply voltage, V_{CC-}	-7.5	-9	-15	-7.5	-9	-15	V
High-level input voltage, V_{IH}	1.9			1.9			V
Low-level input voltage, V_{IL}			0.8			0.8	V
Operating free-air temperature, T_A	-55		125	0		70	°C

electrical characteristics over operating free-air temperature range, $V_{CC+} = 9\text{ V}$, $V_{CC-} = -9\text{ V}$ (unless otherwise noted)

PARAMETER	TEST CONDITIONS	SN55188		SN75188		UNIT	
		MIN	TYP† MAX	MIN	TYP† MAX		
V_{OH} High-level output voltage	$V_{IL} = 0.8\text{ V}$, $R_L = 3\text{ k}\Omega$	$V_{CC+} = 9\text{ V}$, $V_{CC-} = -9\text{ V}$	6	7	6	7	V
		$V_{CC+} = 13.2\text{ V}$, $V_{CC-} = -13.2\text{ V}$	9	10.5	9	10.5	
V_{OL} Low-level output voltage	$V_{IH} = 1.9\text{ V}$, $R_L = 3\text{ k}\Omega$	$V_{CC+} = 9\text{ V}$, $V_{CC-} = -9\text{ V}$	-7	-6	-7	-6	V
		$V_{CC+} = 13.2\text{ V}$, $V_{CC-} = -13.2\text{ V}$	-10.5	-9	-10.5	-9	
I_{IH} High-level input current	$V_I = 5\text{ V}$		10		10	μA	
I_{IL} Low-level input current	$V_I = 0$		-1 -1.6		-1 -1.6	mA	
$I_{OS(H)}$ Short-circuit output current at high level‡	$V_I = 0.8\text{ V}$, $V_O = 0$	-4.6	-9 -13.5	-6	-9 -12	mA	
$I_{OS(L)}$ Short-circuit output current at low level‡	$V_I = 1.9\text{ V}$, $V_O = 0$	4.6	9 13.5	6	9 12	mA	
r_o Output resistance, power off	$V_{CC+} = 0$, $V_{CC-} = 0$, $V_O = -2\text{ V to }2\text{ V}$	300		300		Ω	
I_{CC+} Supply current from V_{CC+}	No load	$V_{CC+} = 9\text{ V}$, All inputs at 1.9 V	15	20	15	20	mA
		$V_{CC+} = 9\text{ V}$, All inputs at 0.8 V	4.5	6	4.5	6	
		$V_{CC+} = 12\text{ V}$, All inputs at 1.9 V	19	25	19	25	
		$V_{CC+} = 12\text{ V}$, All inputs at 0.8 V	5.5	7	5.5	7	
I_{CC-} Supply current from V_{CC-}	No load	$V_{CC+} = 9\text{ V}$, All inputs at 1.9 V	-13	-17	-13	-17	mA
		$V_{CC+} = 9\text{ V}$, All inputs at 0.8 V	-0.5	-0.015	-0.5	-0.015	
		$V_{CC-} = -12\text{ V}$, All inputs at 1.9 V	-18	-23	-18	-23	
		$V_{CC-} = -12\text{ V}$, All inputs at 0.8 V	-0.5	-0.015	-0.5	-0.015	
		$V_{CC-} = -15\text{ V}$, All inputs at 1.9 V	-34	-34	-34	-34	
		$V_{CC-} = -15\text{ V}$, All inputs at 0.8 V	-2.5	-2.5	-2.5	-2.5	
P_D Total power dissipation	No load	$V_{CC+} = 9\text{ V}$, $V_{CC-} = -9\text{ V}$	333		333	mW	
		$V_{CC+} = 12\text{ V}$, $V_{CC-} = -12\text{ V}$	576		576		

- † All typical values are at $T_A = 25^\circ\text{C}$.
‡ Not more than one output should be shorted at a time.
NOTE 3: The algebraic convention in which the less positive (more negative) limit is designated as minimum, is used in this data sheet for logic voltage levels only, e.g., if -6 V is a maximum, the typical value is a more negative voltage.

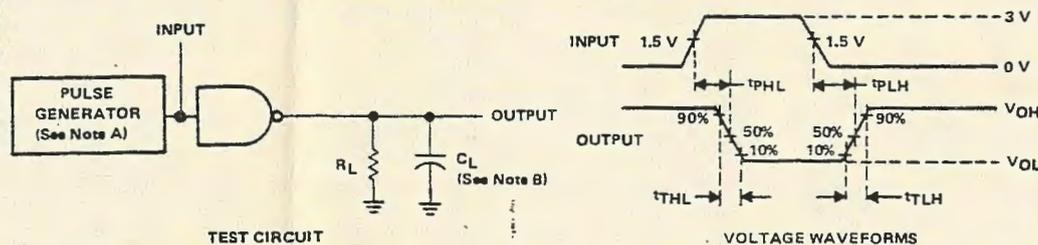
switching characteristics, $V_{CC+} = 9\text{ V}$, $V_{CC-} = -9\text{ V}$, $T_A = 25^\circ\text{C}$

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Propagation delay time, low-to-high-level output		220	350	ns
t_{PHL}	Propagation delay time, high-to-low-level output	$R_L = 3\text{ k}\Omega$, $C_L = 15\text{ pF}$,	100	175	ns
t_{TLH}	Transition time, low-to-high-level output [†]	See Figure 1	55	100	ns
t_{THL}	Transition time, high-to-low-level output [†]		45	75	ns
t_{TLH}^{\ddagger}	Transition time, low-to-high-level output [‡]	$R_L = 3\text{ k}\Omega$ to $7\text{ k}\Omega$, $C_L = 2500\text{ pF}$,	2.5		μs
t_{THL}^{\ddagger}	Transition time, high-to-low-level output [‡]	See Figure 1	3.0		μs

[†] Measured between 10% and 90% points of output waveform.

[‡] Measured between +3 V and -3 V points on the output waveform (EIA RS-232-C conditions)

PARAMETER MEASUREMENT INFORMATION



NOTES: A. The pulse generator has the following characteristics: $t_w = 0.5\ \mu\text{s}$, $\text{PRR} \leq 1\text{ MHz}$, $Z_0 = 50\ \Omega$.
 B. C_L includes probe and jig capacitance.

FIGURE 1. PROPAGATION AND TRANSITION TIMES

4 Line Drivers/Receivers

TYPICAL CHARACTERISTICS[†]

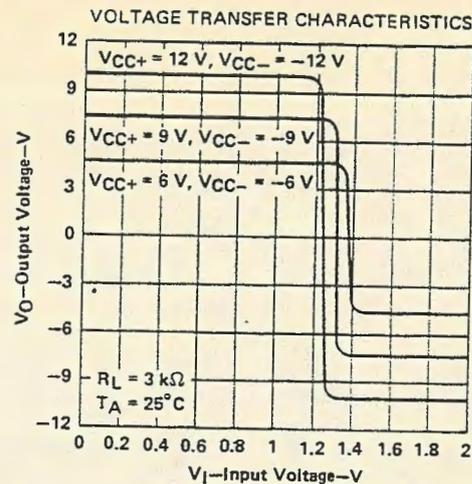


FIGURE 2

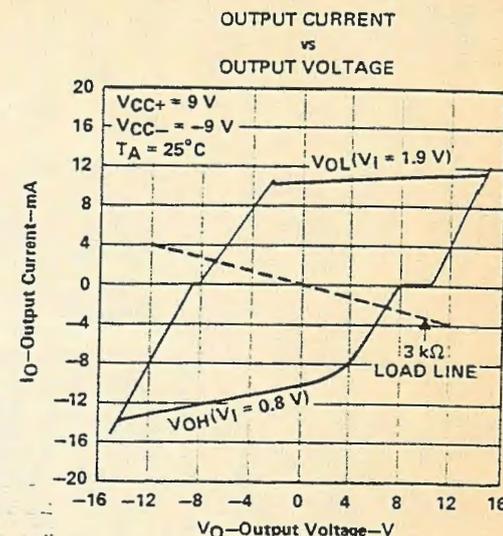


FIGURE 3

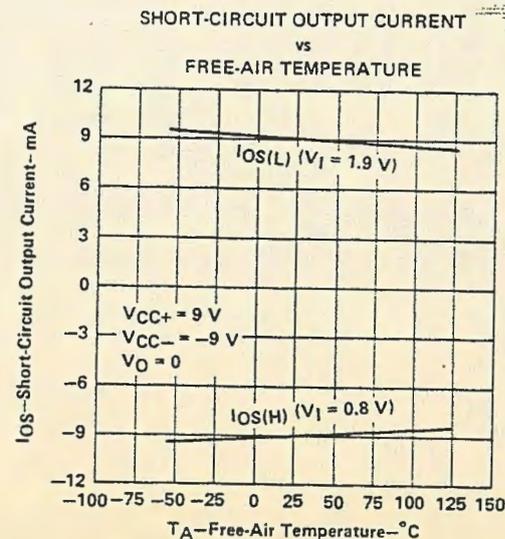


FIGURE 4

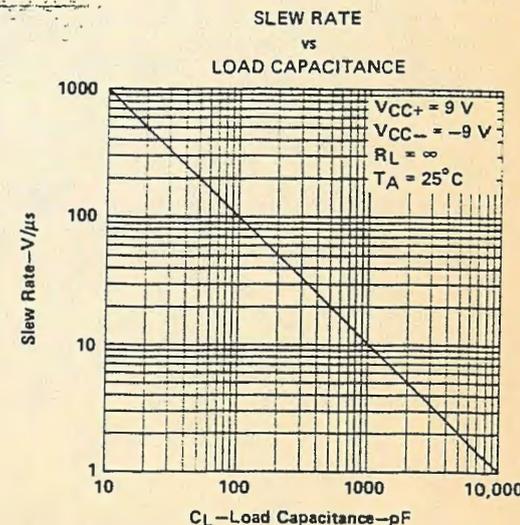


FIGURE 5

[†]Data for temperatures below 0°C and above 70°C are applicable to SN55188 circuit only.

THERMAL INFORMATION†

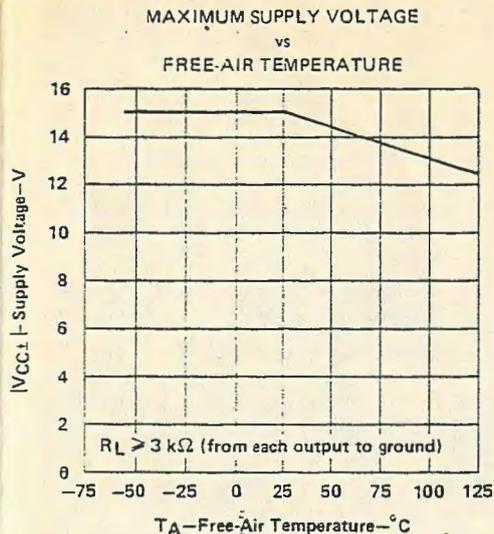


FIGURE 6

†Data for temperatures below 0°C and above 70°C are applicable to SN55188 circuit only.

TYPICAL APPLICATION DATA

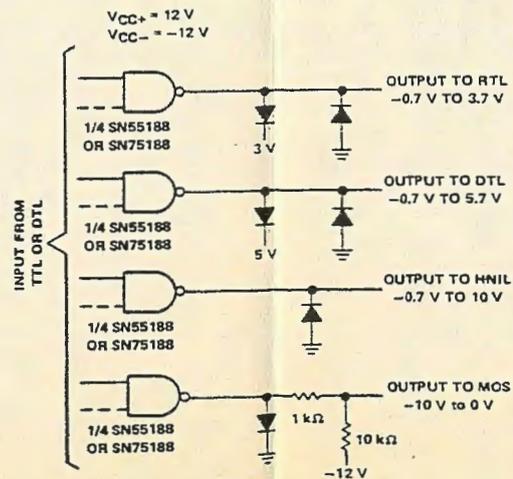


FIGURE 7. LOGIC TRANSLATOR APPLICATIONS

Diodes placed in series with the V_{CC+} and V_{CC-} leads will protect the SN55188/SN75188 in the fault condition in which the device outputs are shorted to ± 15 V and the power supplies are at low voltage and provide low-impedance paths to ground.

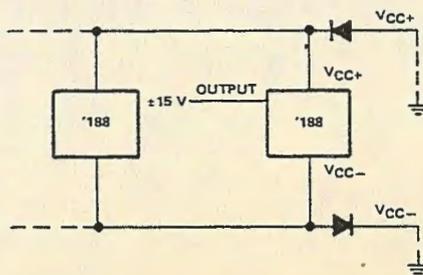


FIGURE 8. POWER SUPPLY PROTECTION TO MEET POWER-OFF FAULT CONDITIONS OF EIA STANDARD RS-232-C

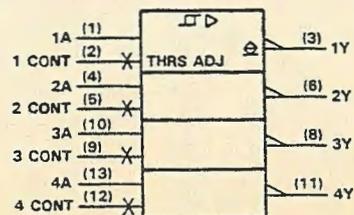
- Input Resistance . . . 3 kΩ to 7 kΩ
- Input Signal Range . . . ± 30 V
- Operates from Single 5-V Supply
- Built-in Input Hysteresis (Double Thresholds)
- Response Control Provides:
Input Threshold Shifting
Input Noise Filtering
- Satisfies Requirements of EIA RS-232-C
- Fully Interchangeable with Motorola MC1489, MC1489A

description

These devices are monolithic Low-power Schottky quadruple line receivers designed to satisfy the requirements of the standard interface between data terminal equipment and data communication equipment as defined by EIA Standard RS-232-C. A separate response control terminal is provided for each receiver. A resistor or a resistor and bias voltage source can be connected between this terminal and ground to shift the input threshold levels. An external capacitor can be connected between this terminal and ground to provide input noise filtering.

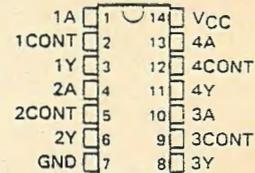
The SN55189 and SN55189A are characterized for operation over the full military temperature range of -55 °C to 125 °C. The SN75189 and SN75189A are characterized for operation from 0 °C to 70 °C.

logic symbol†

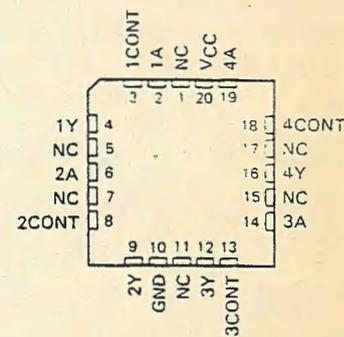


†This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12. Pin numbers shown are for D, J, and N packages.

SN55189, SN55189A . . . J PACKAGE
SN75189, SN75189A . . . D, J, OR N PACKAGE
(TOP VIEW)

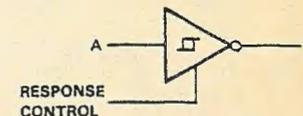


SN55189, SN55189A . . . FK PACKAGE
(TOP VIEW)

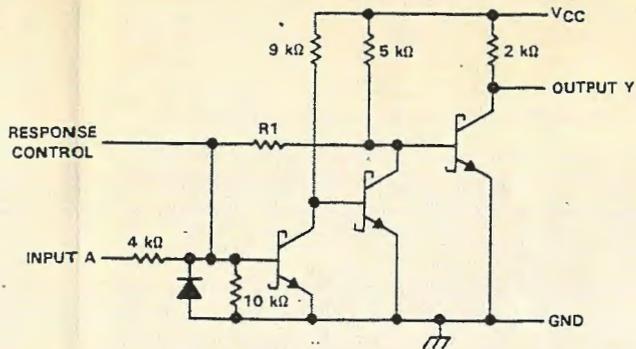


NC—No internal connection

logic diagram (each receiver)



schematic (each receiver)



	SN55189	SN55189A
	SN75189	SN75189A
R1	10 kΩ	2 kΩ

Resistor values shown are nominal.

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

	SN55189 SN55189A	SN75189 SN75189A	UNIT
Supply voltage, V_{CC} (see Note 1)	10	10	V
Input voltage	± 30	± 30	V
Output current	20	20	mA
Continuous total dissipation at (or below) 25°C free-air temperature (see Note 2)	D package	950	mW
	FK or J package	1375	
	J package	1025	
	N package	1150	
Operating temperature range	-55 to 125	0 to 70	°C
Storage temperature range	-65 to 150	-65 to 150	°C
Case temperature for 60 seconds: FK package	260	*	°C
Lead temperature 1,6 mm (1/16 inch) from case for 60 seconds: J package	300	300	°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds: D or N package		260	°C

NOTES: 1. All voltage values are with respect to network ground terminals.
 2. For operation above 25°C free-air temperature, refer to the Dissipation Derating Curves in Appendix A. In the J package, SN55189 and SN55189A chips are alloy mounted and SN75189 and SN75189A chips are glass mounted. In the N package, use the 9.2-mW/°C curve for these devices.

electrical characteristics over operating free-air temperature range, $V_{CC} = 5 V \pm 1\%$, (unless otherwise noted)

PARAMETER	TEST FIGURE	TEST CONDITIONS†	SN55189 SN55189A			SN75189 SN75189A			UNIT	
			MIN	TYP‡	MAX	MIN	TYP‡	MAX		
V_{T+} Positive-going threshold voltage	1	'189	$T_A = 25^\circ\text{C}$	1	1.3	1.5	1	1.3	1.5	V
			$T_A = 0^\circ\text{C to } 70^\circ\text{C}$				0.9		1.6	
		'189A	$T_A = 25^\circ\text{C}$	1.75	1.9	2.25	1.75	1.9	2.25	
			$T_A = 0^\circ\text{C to } 70^\circ\text{C}$				1.55		2.25	
V_{T-} Negative-going threshold voltage	1	'189	$T_A = 25^\circ\text{C}$	0.75	1.0	1.25	0.75	1.0	1.25	V
			$T_A = 0^\circ\text{C to } 70^\circ\text{C}$				0.65		1.25	
		'189A	$T_A = 25^\circ\text{C}$	0.35		1.6				
V_{OH} High-level output voltage	1	Input open.	$V_I = 0.75\text{ V}, I_{OH} = -0.5\text{ mA}$	2.6	4	5	2.6	4	5	V
			$I_{OH} = -0.5\text{ mA}$	2.6	4	5	2.6	4	5	
V_{OL} Low-level output voltage	1	$V_I = 3\text{ V}, I_{OL} = 10\text{ mA}$	0.2	0.45		0.2	0.45	V		
I_{IH} High-level input current	2	$V_I = 25\text{ V}$	3.6	8.3		3.6	8.3	mA		
		$V_I = 3\text{ V}$	0.43			0.43				
I_{IL} Low-level input current	2	$V_I = -25\text{ V}$	-3.6	-8.3		-3.6	-8.3	mA		
		$V_I = -3\text{ V}$	-0.43			-0.43				
I_{OS} Short-circuit output current	3		-3			-3		mA		
I_{CC} Supply current	2	$V_I = 5\text{ V},$ Outputs open	20	26		20	26	mA		

† All characteristics are measured with the response control terminal open.
 ‡ All typical values are at $V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$.

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

PARAMETER	TEST FIGURE	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH} Propagation delay time, low-to-high-level output	4	$C_L = 15\text{ pF}, R_L = 3.9\text{ k}\Omega$		25	95	ns
t_{PHL} Propagation delay time, high-to-low-level output		$C_L = 15\text{ pF}, R_L = 390\ \Omega$		25	50	
t_{TLH} Transition time, low-to-high-level output		$C_L = 15\text{ pF}, R_L = 3.9\text{ k}\Omega$	120	175	ns	
t_{THL} Transition time, high-to-low-level output		$C_L = 15\text{ pF}, R_L = 390\ \Omega$	10	20		

PARAMETER MEASUREMENT INFORMATION†

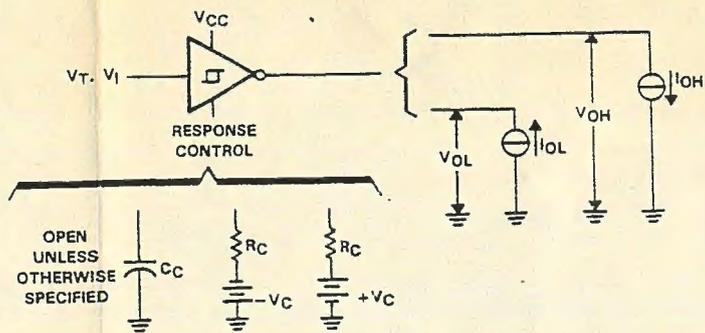
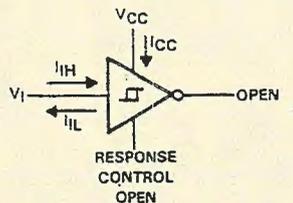


FIGURE 1. V_{T+} , V_{T-} , V_{OH} , V_{OL}



I_{CC} is tested for all four receivers simultaneously

FIGURE 2. I_{iH} , I_{iL} , I_{CC}

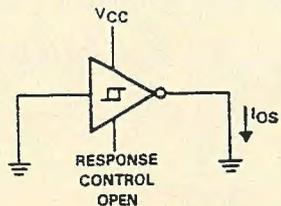
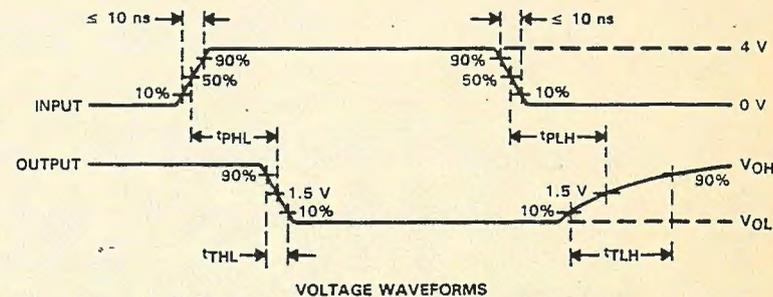
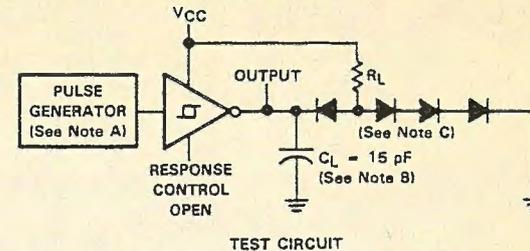


FIGURE 3. I_{OS}

†Arrows indicate actual direction of current flow. Current into a terminal is a positive value.

PARAMETER MEASUREMENT INFORMATION†



VOLTAGE WAVEFORMS

- NOTES: A. The pulse generator has the following characteristics: $Z_{out} \approx 50 \Omega$, $t_w = 500 \text{ ns}$.
- B. C_L includes probe and jig capacitances.
- C. All diodes are 1N3064 or equivalent.

FIGURE 4. SWITCHING TIMES

†Arrows indicate actual direction of current flow. Current into a terminal is a positive value.

TYPICAL CHARACTERISTICS†

SN55189, SN75189
 OUTPUT VOLTAGE
 vs
 INPUT VOLTAGE

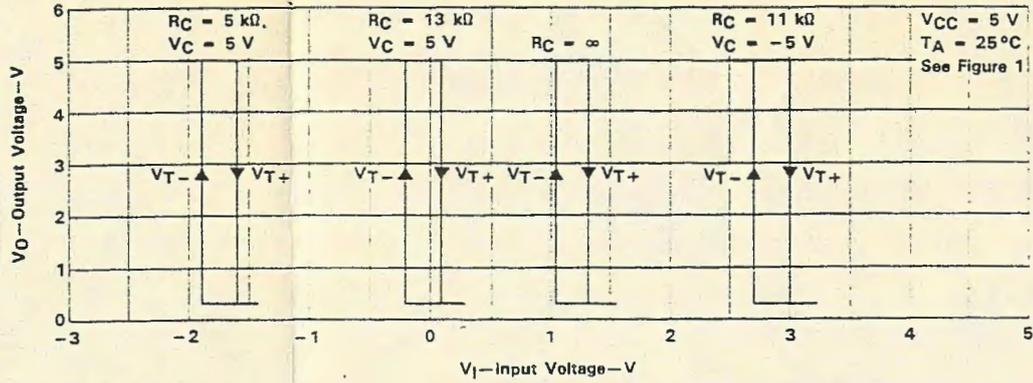


FIGURE 5

SN55189A, SN75189A
 OUTPUT VOLTAGE
 vs
 INPUT VOLTAGE

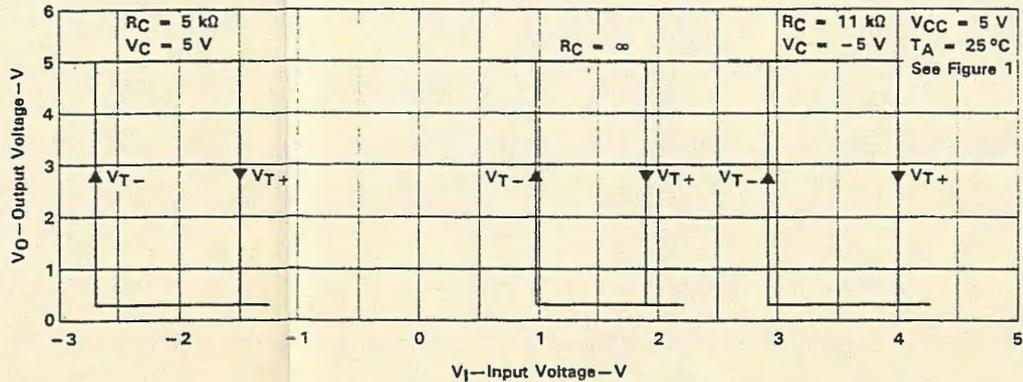


FIGURE 6

†Data for free-air temperatures below 0°C and above 70°C are applicable to SN55189 and SN55189A circuits only.

TYPICAL CHARACTERISTICS†

INPUT THRESHOLD VOLTAGE
 vs
 SUPPLY VOLTAGE

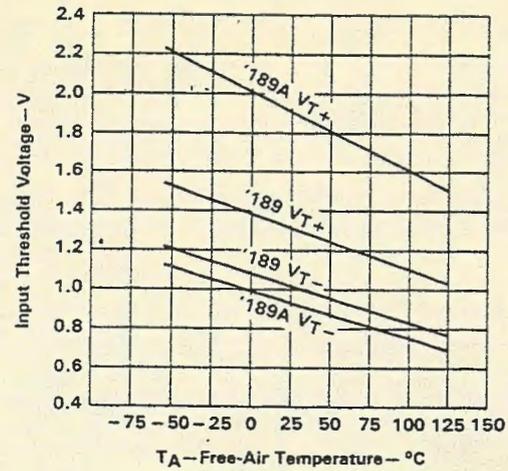


FIGURE 7

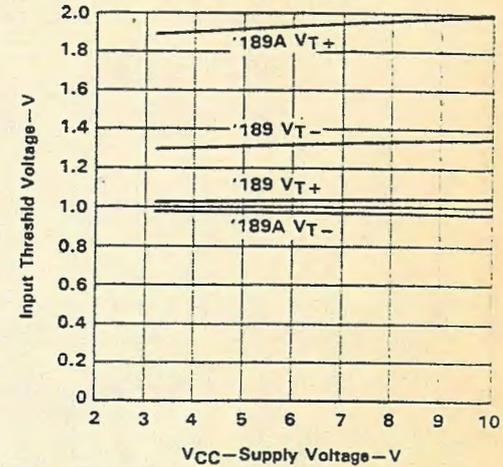


FIGURE 8

†Data for free-air temperatures below 0°C and above 70°C are applicable to SN55189 and SN55189A circuits only.