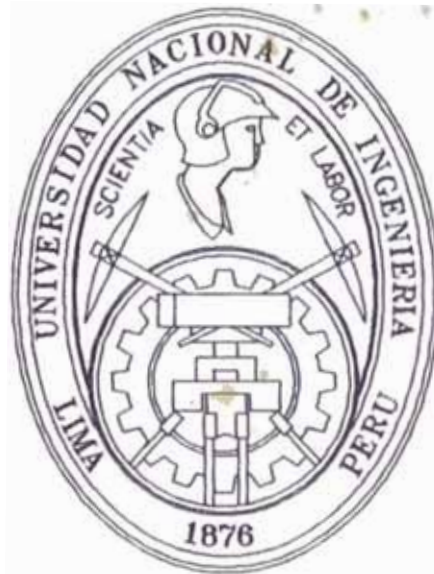


**UNIVERSIDAD NACIONAL DE INGENIERIA  
FACULTAD DE CIENCIAS**



**TITULO: "DISEÑO Y CONSTRUCCION DE UN  
ACUMULADOR DE DATOS DE TEMPERATURA  
CON OCHO CANALES"**

**T E S I S**

**Para optar el Título profesional de:  
LICENCIADO EN FISICA**

**CESAR MARTIN CRUZ SALAZAR**

**LIMA - PERU  
1994**

# INDICE

<u>CONTENIDO</u>	<u># PAGS.</u>
RESUMEN . . . . .	5
PROLOGO . . . . .	6
<b>CAPITULO I: ASPECTOS GENERALES . . . . .</b>	<b>8</b>
1.1 INTRODUCCION . . . . .	8
1.2 TEMAS A TRATAR EN LA PRESENTE TESIS . . . . .	8
1.3 DIAGRAMA DE BLOQUES DEL ACUMULADOR DE DATOS (ADATOC) . . . . .	10
<b>CAPITULO II: DESCRIPCION DEL ADATOC . . . . .</b>	<b>11</b>
2.1 CARACTERISTICAS . . . . .	11
2.2 EL ADATOC . . . . .	11
2.2.1 FORMA FISICA DEL ADATOC . . . . .	12
2.2.2 TRANSFERENCIA DE DATOS . . . . .	13
2.2.3 ACCESORIOS EXTERNOS . . . . .	14
2.2.4 COMPONENTES USADOS EN EL ADATOC . . . . .	14
2.3 FORMA DE MANEJO DEL ADATOC . . . . .	15
2.3.1 ENCENDIENDO EL EQUIPO POR PRIMERA VEZ . . . . .	15
2.3.2 USO DEL TECLADO . . . . .	16
2.3.3 LA PANTALLA O DISPLAY LCD . . . . .	19
2.3.4 EJEMPLO DE PROGRAMACION DEL ADATOC . . . . .	20
2.3.5 LA MEMORIA DEL ADATOC . . . . .	21
2.3.6 MAPA DE MEMORIA DEL ADATOC . . . . .	23
2.4 FICHA TECNICA DEL ADATOC . . . . .	24
2.5 ERROR DE MEDICION DE LA TEMPERATURA EN EL ADATOC . . . . .	25
2.5.1 CALIBRACION DEL ADATOC . . . . .	26
<b>CAPITULO III: EJEMPLO DE APLICACION DEL ADATOC . . . . .</b>	<b>27</b>
3.1 DISPOSICION DE LOS SENSORES EN LA TERMA SOLAR . . . . .	27
3.2 ACLARACION ACERCA DE LOS DATOS TOMADOS . . . . .	28
3.3 DATOS TOMADOS CADA HORA Y GRAFICADOS DURANTE 5 DIAS . . . . .	29
3.3.1 COMENTARIO ACERCA DE LOS DATOS TOMADOS . . . . .	39

3.4 DATOS ACUMULADOS DE LA TERMA SOLAR PROMEDIADOS CADA HORA . . .	40
<b>CAPITULO IV: EL HARDWARE DEL ADATOC . . . . .</b>	<b>45</b>
4.1 EL MICROPROCESADOR Z80 . . . . .	45
4.2 EL CONVERTIDOR A/D (ANALOGO/DIGITAL) . . . . .	45
4.2.1 DESCRIPCION GENERAL DEL ADC0808/09 . . . . .	45
4.2.2 CARACTERISTICAS . . . . .	46
4.2.3 DESCRIPCION FUNCIONAL . . . . .	46
4.2.4 MODO DE FUNCIONAMIENTO . . . . .	47
4.2.5 EL VALOR DIGITALIZADO SELECCIONA SU VALOR EN CENTIGRADOS . . . . .	48
4.3 FORMA DE USO DEL CTC EN EL ADATOC . . . . .	49
4.4 RELOJ DE TIEMPO REAL . . . . .	50
4.5 PANTALLA DE CRISTAL LIQUIDO DEL ADATOC . . . . .	50
4.5.1 CARACTERISTICAS DE LOS MODULOS LCD . . . . .	52
4.5.2 INTERFAZ DE SOFTWARE . . . . .	53
4.5.3 CONTROLADORES DE SOFTWARE . . . . .	54
4.5.4 USOS DE LAS PALABRAS DE CONTROL . . . . .	54
4.5.5 INICIALIZACION DEL VISUALIZADOR . . . . .	55
4.5.6 VISUALIZACION DE LOS DATOS ACTUALES . . . . .	56
4.5.7 INTERFASE DE HARDWARE . . . . .	56
4.6 EL SENSOR DE TEMPERATURA . . . . .	56
<b>CAPITULO V: EL SOFTWARE DEL ADATOC . . . . .</b>	<b>57</b>
5.1 ESPECIFICACIONES DEL MICROPROFESSOR I PLUS (MPF-IP) . . . . .	57
5.2 ESPECIFICACIONES DEL SOFTWARE DEL SISTEMA REALIZADO. . . . .	57
5.3 DESARROLLO DEL PROGRAMA . . . . .	57
5.4 PROGRAMA DE RECEPCION DE DATOS DEL SISTEMA DE DESARROLLO(MPFIP + ADATOC) . . . . .	58
5.5 ALGORITMO DEL PROGRAMA MONITOR . . . . .	65
5.6 LISTADO DEL PROGRAMA MONITOR . . . . .	67
5.7 MAPA DE DIRECCIONES DEL PROGRAMA ADATOC . . . . .	136
5.8 ACERCA DE LA CONSTRUCCION DEL ADATOC . . . . .	146
<b>CAPITULO VI: PROGRAMAS PARA COMPUTADORA . . . . .</b>	<b>148</b>
6.1 ALGORITMO DEL PROGRAMA ADCYA . . . . .	148
6.2 PROGRAMA QUE OBTIENE PROMEDIOS DE LOS DATOS . . . . .	149

6.3 ALGORITMO DEL PROGRAMA ADCYACOM . . . . .	157
6.4 PROGRAMA DE COMUNICACION SERIAL . . . . .	158
CONCLUSIONES . . . . .	162
APENDICES . . . . .	164
REFERENCIAS BIBLIOGRAFICAS .. . . .	201

## RESUMEN

Este trabajo de tesis consiste en la descripción detallada de las etapas de desarrollo necesarias para el diseño, construcción y manejo de un equipo **ACUMULADOR DE DATOS DE TEMPERATURA CON OCHO CANALES(A.DA.T.O.C)**, que sea programable, de tamaño compacto, de bajo consumo, portátil, que mida datos de temperatura por ocho canales y los visualice en forma digital, que almacene o acumule datos en memoria RAM en un tiempo de varias horas o varios días o semanas y que tenga capacidad de transmitir estos datos acumulados vía comunicación serial(interface RS232) a una computadora IBM PC o compatible.

Para cumplir con estos fines fue necesario el uso del microprocesador, y en este caso se utilizó el CPU Z80 versión CMOS, de bajo consumo, fabricado por TOSHIBA. El proceso de lectura de datos, acumulación en memoria y todas las demás funciones del ADATOC(como controlar el teclado, la visualización de los datos, etc) se realiza controlado por un programa en Lenguaje Assembly que radica en código hexadecimal en una memoria EPROM del sistema.

Se le instaló un teclado de 20 teclas para seleccionar diferentes funciones, como transmisión de los datos, selección del canal a visualizar, selección de cuantos canales se van a utilizar, selección de la frecuencia de acumulación de datos(ya que la lectura de datos no es un proceso que se toma en un tiempo cualquiera, el teclado permite que se pueda programar, en que tiempo o cada que tiempo se toman los datos), visualizar el reporte final del proceso de acumulación, lectura de direcciones de memoria, para programar al equipo fijando la hora final, la fecha final del proceso de acumulación, para actualizar la hora y la fecha, etc.

Como se requería que el equipo sea de bajo consumo se utilizó un display de cristal líquido(LCD) de dos filas, cada uno de 16 caracteres.

Para salvaguardar los datos almacenados en memoria cuando sucediera algún corte de la energía en la fuente principal del equipo, se suministró de una batería BACKUP de 4.5 voltios.

## PROLOGO

Hace algunos años cuando en el Laboratorio de energía solar hacia falta un equipo acumulador de datos de ocho canales(sólo había un equipo limitado a cuatro canales, me refiero al DATA LOGGER MODAS 84)se pensó desde ese momento en hacer realidad un proyecto de diseño y construcción de un acumulador de datos. Es así que esta tarea se le encomendó a dos integrantes(César Martín Cruz Salazar y Walter A. Arellano Espinoza) del grupo A.C.E.INS.(ASOCIACION CIENTIFICA ESPECIALIZADA EN INSTRUMENTACION), con sede en el Laboratorio R2-303 de la Fac. de Ciencias de la U.N.I., por sus conocimientos y trabajos previos realizados en esta área.

El presente trabajo de investigación tiene como objetivo principal describir en una forma clara y detallada todo lo relacionado, en el modo de funcionamiento y fases de desarrollo, con el diseño y construcción del equipo **ACUMULADOR DE DATOS DE TEMPERATURA CON OCHO CANALES(A.D.A.T.O.C)** con las características mencionadas.

Para un mejor entendimiento de lo que se pretende exponer, se ha dividido este trabajo en 6 capítulos. Aquí me parece oportuno mencionar que consideré mejor tratar en los primeros capítulos el modo de funcionamiento del equipo y su aplicación.

El capítulo I, trata de unas líneas de justificación del diseño y construcción del equipo, los temas que se van a detallar más adelante y un diagrama de bloques del equipo.

El capítulo II, es una descripción del equipo en forma global y la forma de manejo de él.

El capítulo III, hace mención de un ejemplo de aplicación tomando datos en una terma solar durante cinco días.

El capítulo IV, describe parte de la circuitería digital utilizada en el equipo, como los modos de trabajo del convertidor A/D, el CTC, el reloj de tiempo real y el display de cristal líquido.

El capítulo V, trata del desarrollo del programa del equipo así como las herramientas necesarias que se utilizaron para lograr este objetivo.

El capítulo VI, muestra los listados de programas en Turbo Pascal necesarios para la comunicación del ADATOC con una computadora.

Resulta inevitable e impostergable un agradecimiento especial a quienes dentro de la gente que me apoyó, sobresalieron por su esfuerzo y dedicación, quiero dar las gracias a mis padres y hermanos, al proyecto de secado solar de la cooperación técnica alemana(GTZ) por su apoyo económico,

a la Facultad de Ciencias de la U.N.I., al profesor Hernán Vásquez, al Dr. Manfred Horn por la asesoría de la tesis, al Dr. Guillermo Alzamora, a mi compañero en este trabajo Walter Arellano y por último con un agradecimiento muy especial a Miguel Risco y Adalberto Huamánchumo que conjuntamente con los demás muchachos de A.C.E.INS. ayudaron a la realización de este trabajo.

## CAPITULO I: ASPECTOS GENERALES

### 1.1 INTRODUCCION

La tendencia actual en todos los campos de la ciencia y la tecnología nos conducen hacia la automatización de procesos a cualquier nivel.

Esta tendencia brinda múltiples ventajas a la investigación práctica y a su vez ofrece facilidades para la enseñanza.

Sin embargo, en nuestro país no es posible muchas veces tener acceso a este tipo de equipos que son sofisticados (alta tecnología) y por consecuencia sus precios bastante elevados, a esto hay que añadir que generalmente no están hechos a nuestra realidad, además de todo esto, no se tiene el soporte técnico y la asesoría adecuada sobre el funcionamiento real de estos equipos por ser justamente importados. De tal forma que cuando se malogran quedan en desuso por tiempo indefinido.

Nuestro propósito, fue diseñar y construir un sistema (ACUMULADOR DE DATOS DE TEMPERATURA CON OCHO CANALES A.D.A.T.O.C), a nuestras necesidades (tiene múltiples aplicaciones en el campo de la investigación sobre la energía solar y en la Industria) y con el fin de desarrollar tecnología para que esto no sea de desarrollo y uso exclusivo de países desarrollados y más avanzados que el nuestro, utilizando para ello componentes y equipos disponibles en lo posible en el mercado nacional y a un precio razonable, sin embargo que tuviera lo más importante de un sistema de adquisición de datos profesional, lo cual contribuirá a que el usuario tenga un ambiente amigable de trabajo. Este equipo podemos decir que es el primero de una serie de equipos que serán destinados a otros usos, como por ejemplo la adquisición de otras variables físicas, además de temperatura.

### 1.2 TEMAS A TRATAR EN LA PRESENTE TESIS

Este trabajo, conformado por las etapas de investigación, diseño y construcción del ADATOC han sido realizados en su integridad por los señores **CESAR MARTIN CRUZ SALAZAR** y **WALTER AMERICO ARELLANO ESPINOZA**.

Este trabajo consistirá en forma global, en la descripción del **HARDWARE Y SOFTWARE** del ADATOC, así como ejemplos de aplicaciones del



equipo. El Hardware se muestra en la fig.1 en un diagrama de bloques y en los planos del Apéndice en forma detallada.

La descripción del HARDWARE se ha dividido en dos Tesis de la siguiente manera:(Vea el Diagrama de Bloques).

La Tesis del Señor César Martín Cruz Salazar tratará los temas siguientes:

CPU Z80, Memorias EPROM, Convertidor Analógico/Digital, CTC, Reloj de Tiempo Real, Pantalla LCD, Programa de recepción serial de datos(que corre en un computadora compatible) para recepcionar los datos que se envien desde el ADATOC y el programa de obtención de promedios y importación de los datos a la hoja de cálculo QPRO.

La Tesis del señor Walter A. Arellano E. tratará los temas siguientes:

Memorias RAM con Bateria Backup, El Teclado, parte análoga: Fuente de poder, descripción de los sensores y todo lo relacionado con circuito de comunicación serial.

En cuanto al Software Monitor del equipo será tratado en detalle por ambas Tesis poniendo énfasis en la parte del software que involucra el hardware seleccionado por cada uno.

Cada Tesis desarrollará una aplicación o ejemplo en el uso del equipo.

### 1.3 DIAGRAMA DE BLOQUES DEL ACUMULADOR DE DATOS (ADATOC)

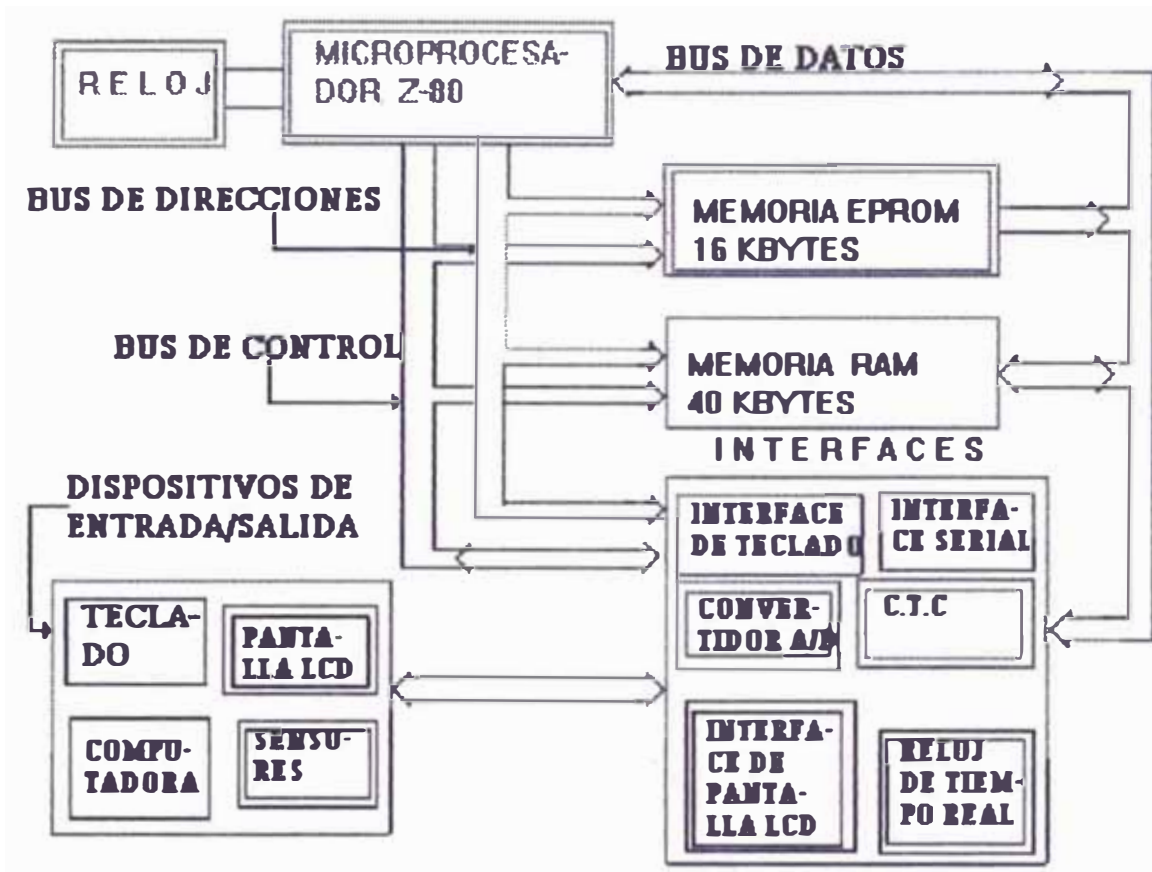


FIG.#1: En el presente trabajo de Tesis sólo se tratará en detalle los Títulos que están dentro de los recuadros a doble subrayado.

## CAPITULO II: DESCRIPCION DEL ADATOC

Este es un equipo basado en microprocesador y que se encuentra establecido en la categoría "MODAS" (Mobile Data Acquisition System) que significa Sistema Móvil de Adquisición de Datos, opera con una Batería de 12V o sino con una fuente rectificadora +12V que se conecta a la red 220V. En adición al displayado de las magnitudes de Temperatura en unidades físicas, puede también almacenar hasta 32768 datos de temperatura (en un rango que va desde -23 hasta 100 grados centígrados) en memoria.

Algunas aplicaciones típicas son:

- En Energía solar para la evaluación de termas solares y otras instalaciones solares.

- En la Industria de fabricación de refrigeradores.

- En la Industria de fabricación de transformadores. Para uso en la prueba de calentamiento del transformador almacenando datos necesarios durante varias horas que dura esta prueba.

- Otras aplicaciones diversas.

### 2.1 CARACTERISTICAS

- Mide y almacena parámetros de temperatura.

- Hasta 8 entradas análogas con 8 bits (0.4 %) de resolución; alto grado de flexibilidad.

- Consumo de corriente 230 miliamperios a 12 voltios.

- Dimensiones del equipo: 23cms de largo, 19cms de ancho y 8cms de alto.

- Referencia de instalación en la azotea del edificio de la Facultad de Ciencias de la U.N.I. para aplicaciones de energía solar.

### 2.2 EL ADATOC

- Equipo electrónico con ocho canales y resolución de 8 bits en cada canal. Displaya la magnitud de la temperatura en grados centígrados del canal seleccionado mediante el teclado.

- Lleva un teclado de 20 teclas (cada una de las teclas emite un sonido cuando se presionan) y sirve para programar la tasa (o frecuencia) de almacenamiento de datos desde tiempos que van de 1 segundo a 4 minutos),

sirve para seleccionar el número de canales a usar, para fijar la hora final y la fecha final del proceso de acumulación de datos, para actualizar la hora y la fecha, para visualizar la memoria ROM y RAM, 2 teclas destinadas a la selección de recepción y transmisión de datos respectivamente, una tecla para la ejecución de un programa que se encuentra en RAM, una tecla para la visualización del reporte parcial del proceso de acumulación (número de datos almacenados, # de canales y hora final del término de la toma de datos) con esta misma tecla se puede visualizar ACUMULACION INACTIVA en caso de no estar acumulando el equipo, otra tecla para la visualización del reporte final(hora inicial, frecuencia de la lectura de datos, # canales y # de datos).

-Contiene un display LCD de dos líneas(16 caracteres por línea) para la visualización de todo lo que se selecciona por el teclado.

-Contiene dos memorias RAM uno de 8kbytes y otro de 32kbytes. De las cuales el de 32kbytes se usa exclusivamente para almacenamiento de datos. Los 8kbytes se usa para el monitor del equipo y para correr algún programa que se cargue en ciertas posiciones de la RAM. Dispone de Bateria BACKUP de 4.5voltios(bateria de respaldo conectado a los chips de memoria RAM para no perder los datos almacenados en ella, cuando se apague el equipo.).

-Cuando termina la acumulación de datos suena una alarma durante 40 segundos como indicación de fin del proceso de acumulación de datos.

-Dispone de un reloj de tiempo real(RTC), para fijar la hora y fecha actual y contabilizar el tiempo de duración del proceso de acumulación. También es posible visualizar la hora y fecha actual en cualquier momento este o no acumulando datos, simplemente presionando la tecla correspondiente.

-Además del almacenamiento de temperatura, se almacenan también la hora inicial, fecha inicial del proceso de acumulación de datos y el número de canales usados en este.

### **2.2.1 FORMA FISICA DEL ADATOC**

El acumulador de datos de temperatura es un equipo electrónico, como

se muestra en la fig. #2. En la parte externa de la caja se encuentra un teclado(20 teclas), una pantalla LCD(display de cristal líquido), un conector para serial RS232, ocho conectores tipo DIN de los canales para conectar los sensores de temperatura(en la fig.#2 sólo se muestran dos conectores, los otros 6 están en la parte trasera de la caja), un regulador de ángulo de visión (contraste del display), un botón de RESET, un interruptor de encendido, un interruptor de Bateria backup y un indicador de encendido.

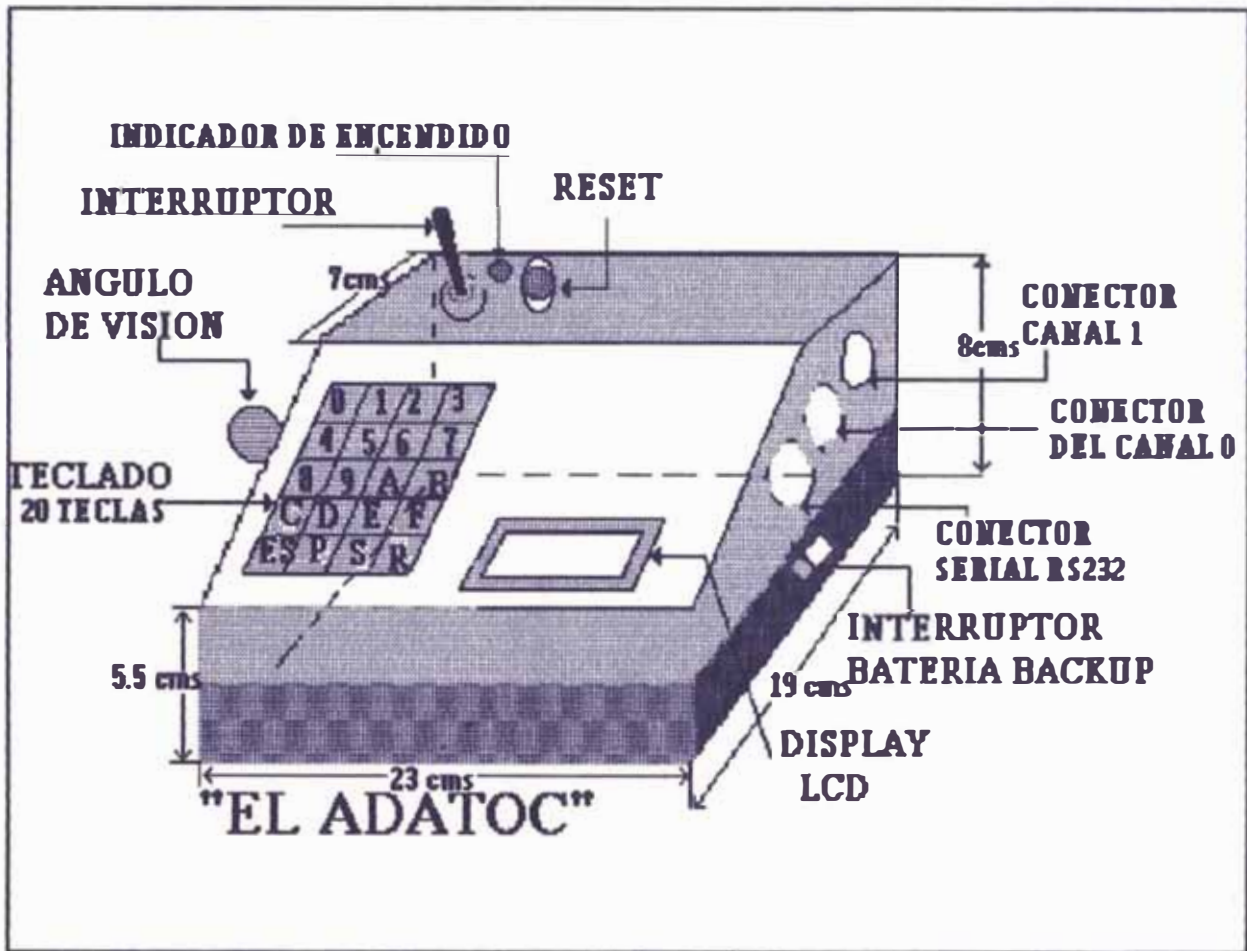


FIG.#2:Forma física y dimensiones del ADATOC.

### 2.2.2 TRANSFERENCIA DE DATOS

Los datos se transfieren del ADATOC a una computadora compatible vía comunicación serial RS232C. También se pueden transferir datos de la computadora al equipo vía comunicación serial.

### 2.2.3 ACCESORIOS EXTERNOS

-Se disponen de ocho cables coaxiales, cada uno con sensores de temperatura (sellados con pegamento epóxico) conectados en sus extremos.

-Viene con un disquette que contiene el programa de comunicación de datos, un programa que sirve para mostrar los datos en el monitor de la computadora en forma de cuadros con opción de sacar promedios, valores mínimo y máximo y con la capacidad de poder importar estos al QPRO y de esta manera poder graficar los datos y imprimirlo.

-Dispone de una fuente de poder que tiene como salida +12V, -12V y +5V que se conecta con el ADATOC mediante un cable coaxial que lleva en su extremo un conector tipo DIN.

### 2.2.4 COMPONENTES USADOS EN EL ADATOC

El acumulador de datos de temperatura consiste de un conjunto de dos tarjetas(doble cara) de circuito impreso.

#### **La primera tarjeta contiene:**

- 1.Un CPU-Z80 CMOS(TMPZ84C00AP-8 TOSHIBA)
- 2.Una memoria EPROM 27C64 de 8Kbytes más un socket vacío para colocar otra memoria EPROM 27C64.
- 3.Una memoria RAM 60L256 de 32kbytes(de bajo consumo).
- 4.Una memoria RAM 61L64 de 8kbytes(de bajo consumo).
- 5.Un circuito integrado para teclado 74C923 CMOS.
- 6.Un conector de 34 pines para la pantalla o DISPLAY LCD.
- 7.Un conector de 40 pines para conectar a la segunda tarjeta mediante un cable plano de 40 pines.
- 8.Otros componentes como compuertas, decodificadores 74HC138, diodos, flip-flops, condensadores, etc.

9. CRYSTAL DE 3.58Mhz para generar un RELOJ de 1.79MhZ.
10. Un conector de 16 pines para conectar el teclado.

**La segunda tarjeta contiene:**

1. Un CTC-Z80 (Contador y temporizador de la familia Z80).
2. Un A/D 0808 (Convertidor A/D de ocho canales multiplexado).
3. Un USART 82C51 (Interface serial).
4. Un MM58174 (Reloj de Tiempo Real, de bajo consumo).
5. Un CRYSTAL de 4.9152Mhz para la generación de la velocidad de transmisión y recepción en la comunicación serial.
6. Un regulador LM317 (Para fijar el nivel de voltaje a 3.73 voltios).
7. Una Batería de respaldo (BACKUP) de 4.5 voltios marca Duracell (no recargable) para la unidad de memoria RAM (40kbytes en total) y el Reloj en Tiempo Real (la Batería se encuentra atornillado en la caja, no en la tarjeta).
8. Otros componentes como compuertas, decodificadores 74HC138, diodos, condensadores, etc.

## **2.3 FORMA DE MANEJO DEL ADATOC**

### **2.3.1 ENCENDIENDO EL EQUIPO POR PRIMERA VEZ**

Cuando se enciende el equipo por primera vez (posición ON del interruptor de encendido), lo que se va a observar en la pantalla y el tipo de arranque que tendrá dependerá de si se encuentra el interruptor de la Batería backup en ON o OFF. Si se encuentra en OFF (ENCENDIO EN FRIO), realiza un Test de memoria RAM por unos segundos y además coloca valores FFh en toda la memoria RAM (40Kbytes) borrando todos los datos anteriores que se encontraban en memoria; en el transcurso de este tiempo no aparece

nada en la pantalla y posteriormente aparece la palabra HORA - en la primera línea de la pantalla LCD. Esto se interpreta como un indicativo de solicitud de ingreso de la hora y luego de la fecha a través del teclado. Después de hacer este ingreso aparece en la pantalla la presentación esto es, "ACUMULADOR DE DATOS" C&A. Las letras C y A son las iniciales de los apellidos de los diseñadores y constructores del ADATOC y se tratan de Cruz y Arellano. Para salir de esta presentación se presiona cualquier tecla. En caso que se desee volver a la presentación, esto se logra presionando el botón rojo(RESET) que a la vez reinicializa el funcionamiento del equipo.

Si se encuentra en ON(ENCENDIDO EN CALIENTE), no borrará la memoria y en la pantalla aparecerá la presentación "ACUMULADOR DE DATOS" C&A. Para salir de este estado se presiona cualquier tecla.

### 2.3.2 USO DEL TECLADO

Cada una de las teclas tienen diferentes usos.

La tecla marcada encima con 0, se usa para la seleccionar el canal 0 para su posterior visualización de la temperatura que entra por ese canal. Si antes de presionar esta tecla, se han presionado cualquiera de las siguientes teclas marcadas con B, B y P, entonces esta tecla significará el número 0 y no seleccionará el canal 0. Si primero se hubiera presionado la tecla S, en este caso la tecla 0 seleccionará la frecuencia de acumulación de datos de 1 segundo.

La tecla marcada encima con 1, se usa para seleccionar el canal 1 para su visualización de la temperatura que ingresa por ese canal. Si antes de presionar esta tecla, se han presionado cualquiera de las siguientes teclas marcadas con B, B y P, entonces esta tecla significará el número 1 y no seleccionará el canal 1. Si primero se hubiera presionado la tecla S, en este caso la tecla 1 seleccionará la frecuencia de acumulación de datos de 2 segundos.

En igualdad de condiciones que los dos casos anteriores descritos, la tecla marcada encima con 2, seleccionará el canal 2, significará también la tecla correspondiente al número 2 y seleccionará como frecuencia 3 segundos.



La descripción de las teclas que siguen desde la tecla 3 hasta la tecla 7 cumplirán las mismas condiciones que las teclas del 0 al 2.

La tecla marcada encima con 3, seleccionará el canal 3, significará el número 3 y seleccionará como frecuencia 4 segundos.

La tecla marcada encima con 4, seleccionará el canal 4, también será el número 4 y seleccionará como frecuencia 5 segundos.

La tecla marcada encima con 5, seleccionará el canal 5, también será el número 5 y seleccionará como frecuencia 6 segundos.

La tecla marcada encima con 6, seleccionará el canal 6, también será el número 6 y seleccionará como frecuencia 10 segundos.

La tecla marcada encima con 7, seleccionará el canal 7, también será el número 7 y seleccionará como frecuencia 15 segundos.

La tecla marcada encima con 8, en un primer modo funcionará como ingreso de la hora final de acumulación. Si antes de presionar esta tecla, se han presionado cualquiera de las siguientes teclas marcadas con 8, B y P, entonces esta tecla significará el número 8. Si primero se hubiera presionado la tecla S, en este caso la tecla 8 seleccionará la frecuencia de acumulación de datos de 20 segundos.

La tecla marcada encima con 9, en un primer modo funcionará para transmitir datos del ADATOC a la computadora. En caso de que no hubiera nada que transmitir saldrá el mensaje: "MEMORIA VACIA NO TRANSMITE".

Si antes de presionar esta tecla, se han presionado cualquiera de las siguientes teclas marcadas con 8, B y P, entonces esta tecla significará el número 9. Si primero se hubiera presionado la tecla S, en este caso la tecla 9 seleccionará la frecuencia de acumulación de datos de 25 segundos.

La tecla marcada encima con A, en un primer modo funcionará para indicar si el ADATOC está acumulando datos, si no lo está se visualiza un mensaje de "ACUMULACION INACTIVA" después de presionar esta tecla. Si se estuviera acumulando se visualizaría por ejemplo: CAN:0-0 FIN10:00 N.DATOS: 3, esto significa # de canales, hora final de acumulación y # de datos acumulados

que va variando con el tiempo como un contador.

Si antes de presionar esta tecla, se han presionado cualquiera de las siguientes teclas marcadas con B, B y P, entonces esta tecla significará el número 0Ah. Si primero se hubiera presionado la tecla S, en este caso la tecla A seleccionará la frecuencia de acumulación de datos de 30 segundos.

La tecla marcada encima con B, en un primer modo funcionará para colocar la dirección de memoria cuyo contenido se desea visualizar. Al presionar esta tecla se visualiza:M>= en este momento si vuelve a presionar la tecla B funcionará como número 0Bh. Después hay que ingresar por el teclado la dirección de memoria(4 dígitos), cuando se termina de ingresar el último dígito sale en la pantalla los contenidos de la dirección ingresada y de 11 direcciones consecutivas.

Si primero se hubiera presionado la tecla S, en este caso la tecla B seleccionará la frecuencia de acumulación de datos de 45 segundos.

La tecla marcada encima con C, en un primer modo funcionará como avance de memoria(en orden ascendente)de 12 en 12. Si antes de presionar esta tecla se presiona la tecla B, entonces la tecla C significará el número 0Ch.

Si primero se hubiera presionado la tecla S, en este caso la tecla C seleccionará la frecuencia de acumulación de datos de 60 segundos.

La tecla marcada encima con D, en un primer modo funcionará como avance de memoria(en orden descendente)de 12 en 12. Si antes de presionar esta tecla se presiona la tecla B, entonces la tecla D significará el número 0Dh.

Si primero se hubiera presionado la tecla S, en este caso la tecla D seleccionará la frecuencia de acumulación de datos de 120 segundos.

La tecla marcada encima con E, en un primer modo funcionará como ejecutor de un programa cargado en RAM, si no hubiera tal programa y si se presiona esta tecla reiniciará el sistema y se visualizará la presentación del ADATOC. Si después de presionado la tecla B se presiona esta tecla funcionará como número 0Eh.

Si primero se hubiera presionado la tecla S, en este caso la tecla E seleccionará la frecuencia de acumulación de datos de 180 segundos.

La tecla marcada encima con F, en un primer modo sirve para ver el reporte

final del proceso de acumulación de datos. Si se presiona esta tecla se visualizará por ejemplo lo siguiente: H.I.09:46 C/240S #C=1 DATOS=8, esto significa hora inicial del proceso de acumulación, frecuencia de la toma de datos, # de canales(en el ejemplo un sólo canal) y # de datos.

Si después de presionado la tecla B se presiona esta tecla funcionará como número 0Fh.

Si primero se hubiera presionado la tecla S, en este caso la tecla F seleccionará la frecuencia de acumulación de datos de 240 segundos.

La tecla marcada encima con ESC, en un primer modo sirve para visualizar la hora y la fecha. También actúa como escape de ciertas rutinas como son las que suceden si se presionan cualquiera de las teclas B, B, P, S y R.

La tecla marcada encima con P, en un primer modo sirve actualizar la hora y la fecha. Si se presiona esta tecla se visualiza inicialmente la palabra:HORA =. Esto significa que debemos ingresar la hora actual. Después sale en otra línea la palabra FECHA=, aquí debemos ingresar la fecha actual.

La tecla marcada encima con S, sirve para seleccionar la frecuencia de acumulación de datos, presionando para ello en seguida cualquiera de las teclas anteriormente descritas.

Después que se enciende en frío el ADATOC y se presiona esta tecla saldrá en pantalla por defecto 1SEG. como la tasa seleccionada para la acumulación de los datos.

La tecla marcada encima con R, sirve para recepcionar algún programa desde la computadora. Si se presiona esta tecla se visualizará el mensaje: ESPERANDO TRANSMISION... . Se sale de este estado presionando la tecla ESC y visualizando el mensaje:RECEPCION TERMINADA y si se vuelve a presionar aparecerá la hora y la fecha.

### 2.3.3 LA PANTALLA O DISPLAY LCD

En la pantalla se visualiza(en las dos filas que conforman el Display) por defecto, si es que se presiona por ejemplo la tecla 0:

**TEMPERATURA:**

**C0= 20.41°C 1S**

donde el término 1S significa que ese será la tasa de acumulación de datos, cuando se proceda a acumular.

0 si se presiona la tecla 1:

**TEMPERATURA:**

**C1= 19.93°C 1S**

Cuando se esta acumulando y si se presiona la tecla 0 aparecerá:

**ACUMULANDO: 0\***

**C0=21.38°C 1S**

En este caso, el asterisco se visualiza en forma parpadeante.

Cuando se presiona la ESC se visualiza la hora y la fecha en la forma siguiente:

**HORA =14:30:21**

**FECHA=13/09/94**

#### 2.3.4 EJEMPLO DE PROGRAMACION DEL ADATOC

Quiero programar que el ADATOC almacene datos de los ocho canales durante una hora, con una frecuencia de almacenamiento para cada dato de 4 minutos, el proceso se iniciará a las 8:30 de la mañana y se prolongará hasta las 9:30 de la mañana del mismo día.

Primeramente debo fijar la hora actual y la fecha actual en el ADATOC. Para ello presiono la tecla P y en un instante aparecerá en el display la palabra HORA =, en este momento presiono las teclas de marcadas con números para poner la hora actual. Terminando esto aparecerá la palabra FECHA =, en este momento pongo la fecha actual. Terminando aparecerá la hora y la fecha actual. Cada vez que se quisiera visualizar la hora y la fecha se presionará la tecla ESC.

Después coloco el interruptor de la Bateria BACKUP(que se encuentra en la parte baja del lado derecho) en la posición ON.

A continuación, presiono la tecla S para seleccionar el tiempo de cada cuanto se va ir almacenando cada dato en la memoria. En este caso presiono la tecla F(240 segs.) y para salir de aquí presiono la tecla ESC.

A las 8:30 de la mañana que es mi hora inicial presiono la tecla 8(hora final) y pongo la hora final(09:30)sólo los cuatro dígitos sin los dos puntos. Después me pedirá la FECHA FINAL =, en este caso presiono ESC para pasar al próximo pedido y en este momento aparecerá en el display

CANALES A ACUM.:, aquí presiono la tecla 7(8 canales), si fuera un canal presiono la tecla 0, si fuera dos canales presiono la tecla 1, etc. Terminado esto, empezó el proceso de acumulación de datos y en el display aparece la palabra ACUMULANDO: y un \* parpadeante junto al tiempo que se va incrementando, como indicativo que ya se inicio el proceso.

Una vez terminado el proceso de almacenamiento de datos, puedo apagar el equipo, pero los datos no serán borrados porque están protegidos por la Bateria BACKUP.

Para enviar los datos del ADATOC a la computador compatible conecto el cable en el ADATOC y el otro extremo en el puerto serial de la computadora. Una vez listo, y habiendo corrido ya el programa adecuado en la computadora, presiono la tecla 9(Transmisión de datos) del ADATOC y aparecerá en el display TRANSMITIENDO DATOS.... Finalizado esto, aparecerá en el display el mensaje TRANSMISION TERMINADA.

### 2.3.5 LA MEMORIA DEL ADATOC

En el ADATOC la máxima cantidad de memoria que se puede usar son sólo 56kbytes de los 64kbytes que puede direccionar como máximo el CPU Z80 .

Los primeros 16kbytes(desde 0000h hasta 3FFFh)se usan para direccionar memorias EPROMS y los 40kbytes restantes(desde 4000h hasta BFFFh y de E000h hasta FFFFh) se usan para direccionar memorias RAMS.

En la fig.# 3, se muestra el mapa de memoria del ADATOC. Aquí puede verse por ejemplo que 32kbytes son destinados para la acumulación de datos.

Las MEMORIAS EPROMS es un tipo de memoria ROM(Read Only Memory)memoria de sólo lectura desde la cual se pueden leer los datos muchas veces pero en las que no se puede escribir como en el caso de las memorias de lectura/escritura.

También la información almacenada en este tipo de memoria no se perderá cuando se apague el equipo.

En la EPROM(Erasable Programmable Read Only Memory)memoria de sólo lectura borrable y programable, los datos son programados por el usuario en la memoria. Pulsos de alto voltaje en el interior del circuito integrado, fuerzan 1s y 0s lógicos en direcciones específicas de zonas de memoria.

Los datos pueden ser borrados incidiendo luz ultravioleta en una ventana transparente que cubre el circuito integrado. Después de un tiempo

especificado de exposición de la luz, todos los datos serán borrados y el dispositivo puede entonces ser reprogramado con un nuevo ingreso de datos.

En la EPROM que tiene como direcciones(desde 0 hasta 1FFFH) se grabo el programa monitor del ADATOC de la siguiente forma:

SEGMENT NAME	START	STOP	LENGTH
RESETSEG	0000	023c	023d
TOOLSSEG	026b	03ef	0185
LOADSEG	0400	0582	0183
LCDTSEG	0600	1258	0c59
INT0SEG	1300	13e3	00e4
INT1SEG	1400	1466	0067
INT2SEG	1500	1503	0004
INT3SEG	1600	1681	0082

### 2.3.6 MAPA DE MEMORIA DEL ADATOC

DIRECCION DE MEMORIA (EN HEX)		DIRECCION DE MEMORIA (EN DEC.)
FFFF	STACK (F900H - FFFFH)	65535
E0FF	ZONA PARA CARGAR PROGRAMAS (E100H - F8FFH)	57599
E000	BUFFER DE RECEPCION	
	TABLA DE VECTORES DE INTERRUP.	57344
	ZONA NO USADA (NO HAY MEMORIA FISICA) (C000H - DFFFH)	
4000	ZONA DE MEMORIA DESTINADA A LA ACUMULACION DE DATOS (RAM) (4000H - BFFFH)	16384
3FFF		16383
2000	MEMORIA EPROM (OPCIONAL) (2000H - 3FFFH)	8192
1FFF		8191
0000	EPROM CON PROGRAMA MONITOR (0000H - 1FFFH)	0

FIG. #3: Mapa de Memoria del equipo ACUMULADOR DE DATOS.

## 2.4 FICHA TECNICA DEL ADATOC

### **GENERALIDADES:**

Equipo establecido en la categoría de MODAS(Sistema de adquisición de datos móvil), puede operar con una fuente de Bateria de 12V o 9V, o en su defecto con una fuente rectificada de +12V. En adición al displayado de las magnitudes de temperatura en unidades de grados centígrados, tiene la función de almacenar hasta 32768 datos de temperatura(de 1 byte cada uno) en memoria, en un rango que va desde -23 a 100 grados centígrados.

Tiene aplicaciones para mediciones en laboratorios y en la industria.

### **CARACTERISTICAS:**

- Mide y almacena parámetros de Temperatura.

- Resolución de 8 bits para los datos tomados.

- Consumo de corriente 230 miliamperios máximo a 12 Voltios, cuando todos los sensores están instalados.

### **DESCRIPCION:**

- Equipo electrónico con ocho canales y resolución de 8 bits en cada canal. Displaya los valores de temperatura en grados centígrados.

- Dispone de un teclado de 20 teclas para facilitar el manejo del equipo. Se usa para programar las tasas de almacenamiento de datos(desde 1 segundo a 4 minutos), para la selección del número de canales a usar, y para fijar la hora final y fecha final del proceso de acumulación de datos. Al momento de programar se puede ir visualizando en un Display LCD de dos líneas(16 caracteres por línea).

También mediante el teclado puede seleccionar la visualización de un reporte parcial del proceso de acumulación(canales que intervienen, número de datos almacenados, hora final) y cuando termina el proceso de acumulación se puede visualizar un reporte final presionando otra tecla(hora inicial, tasa de toma de datos, # de canales que intervinieron y el # de datos almacenados).

- Dispone de memorias RAMs hasta 40 kilobytes de capacidad. De las cuales 32 kilobytes se usan exclusivamente para almacenamiento de datos y los 8 kilobytes restantes es para uso del programa monitor(grabado en



EPROM) del equipo además de otros usos. Contiene una Batería BACKUP de 4.5 voltios, esta Batería sirve para no perder los datos almacenados en memoria cuando se corta la energía.

-Cuando termina la acumulación de datos suena una alarma de 40 segundos como indicación de fin de acumulación.

-Además del almacenamiento de los datos de Temperatura, se almacenan también la hora inicial, fecha inicial del proceso de acumulación y el número de canales usados.

-Se puede visualizar la hora y fecha actual en cualquier momento ya sea si esta acumulando o no, basta presionar la tecla correspondiente.

-Contiene una interfase serial RS232 para la transferencia de datos del ADATOC a la computadora IBM PC o compatible. Dispone de un interruptor que desactiva esta interfase cuando no se usa, con el fin de no consumir corriente cuando se encuentra almacenando datos.

-Dispone de 8 cables, cada uno con sensores de temperatura conectados en sus extremos.

-Viene con un disquette que contiene un programa para comunicación de los datos y un programa que muestra los datos en el monitor de la computadora en forma de una tabla, con opción de sacar promedios, valores mínimo y máximo obtenidos y con la capacidad de poder importar estos datos al QPRO, para luego imprimir las gráficas y los datos.

## **2.5 ERROR DE MEDICION DE LA TEMPERATURA EN EL ADATOC**

El sensor de temperatura LM 335 tiene un error de +/- 1°C dado por el fabricante del sensor. A este error del sensor hay que añadirle el error del convertidor Análogo a digital cuyo mayor o menor error depende de la estabilidad de la fuente de alimentación, debido a los voltajes de referencia que maneja directamente el A/D.  $V_{ref(+)}=3.73$  voltios y  $V_{ref(-)}=2.5$  voltios, estos voltajes deben ser lo más estables posibles para evitar distorsión en la medida, si esto fallara se tendría errores de +/-2 °C o más. En cambio si se controla apropiadamente la estabilidad de la

fuerza se tendría hasta un máximo de  $\pm 1^{\circ}\text{C}$  de error.

La precisión máxima con la que se puede medir la temperatura es alrededor de medio grado en el ADATOC, esto es debido a la resolución del A/D que es de 0.4%.

### **2.5.1 CALIBRACION DEL ADATOC**

La calibración fue hecha tomando como patrón un termómetro digital. Cuanto más exacto sea el patrón la calibración será mejor. La calibración de cada sensor se hace con un trimmer (potenciometro de precisión) cuyo tornillo da a la parte de afuera del ADATOC, en el caso que se desee recalibrar.

Esta calibración se realiza en una única temperatura de  $25^{\circ}\text{C}$ , un termómetro independiente debe marcar esta temperatura y la salida del sensor LM335 se debe ajustar hasta que de el valor de 2.98 voltios (aproximadamente en el display debe verse  $25.24^{\circ}\text{C}$ ).

### CAPITULO III: EJEMPLO DE APLICACION DEL ADATOC

Se tuvo que instalar los sensores del ADATOC en una TERMA SOLAR como de la **fig.#4**, para medir la temperatura en diferentes puntos de esta. Se acumularon los datos de Temperatura de ocho canales durante 5 días seguidos sin parar. Se almacenaron los datos de los ocho canales aproximadamente a una frecuencia de cada 3 minutos.

El proceso de acumulación se inicio el 4 de Agosto de 1994 a las 12:45:02 de la tarde.

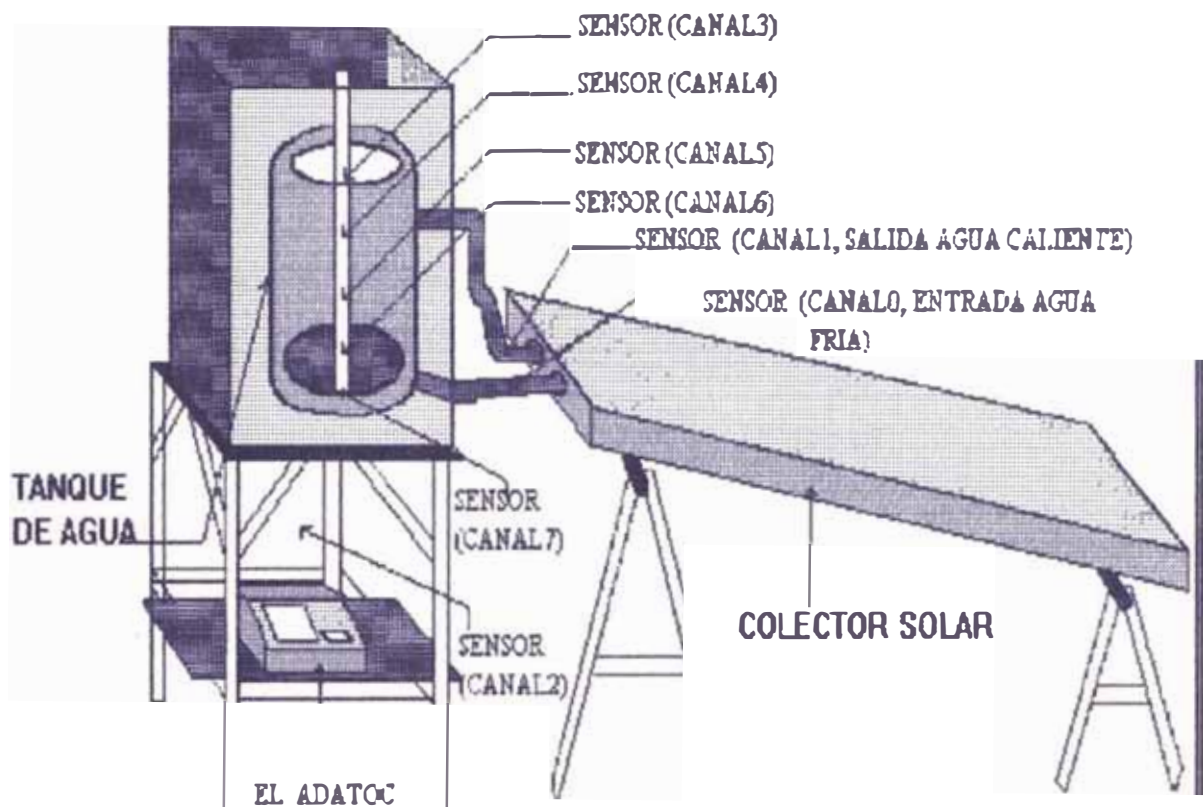


**FIG. #4:** TERMA SOLAR, en el cual se uso el ADATOC para medir varios puntos de TEMPERATURA.

#### 3.1 DISPOSICION DE LOS SENSORES EN LA TERMA SOLAR

Siete sensores se colocaron en la TERMA SOLAR, en la forma mostrada en la fig. #5 y un sensor quedo libre para medir temperatura ambiente.

De los siete sensores uno se coloco en la manguera de agua fría (canal 0), otro se coloco en la manguera de agua caliente (canal 1) y los otros cinco en una vertical dentro del tanque de agua.



## "TERMA SOLAR"

FIG. #5: Distribución de los sensores dentro de la TERMA SOLAR.

### 3.2 ACLARACION ACERCA DE LOS DATOS TOMADOS

Acerca de los datos tomados pareciera que existe un error en el hecho que si el sensor LM335 tiene una precisión de  $\pm 1^{\circ}\text{C}$  como es que el equipo mide los datos de temperatura con una precisión de  $0.48^{\circ}\text{C}$ .

La explicación a esto, está en que el hardware y software del equipo ADATOC se hizo considerando que en el mercado(extranjero) existen sensores de temperatura con una precisión de dos dígitos decimales y con la misma configuración esquemática de los pines de tal forma que solamente es cuestión de cambiar el sensor, más no la circuitería ni el programa.

Un ejm. de sensor es el LM 235 que tiene una precisión de +/-0.5°C y se puede usar este en vez del LM 335.

Con esta aclaración, los datos que a continuación paso a listar deben ser considerados como aproximados y para que se acerquen a datos exactos deben ser redondeados.

### 3.3 DATOS TOMADOS CADA HORA Y GRAFICADOS DURANTE 5 DIAS

Aunque los datos se han almacenado con una frecuencia de alrededor de tres minutos, se han seleccionado de esta tabla los datos que han sido tomados en cada hora en punto durante 5 días. Estos datos después han sido distribuidos en tablas y graficados por días usando **QPRO Versión 5.0**.

HOJA DE CALCULO DEL SISTEMA DE ADQUISICION DE DATOS C&A

UNIVERSIDAD NACIONAL DE INGENIERIA

FACULTAD DE CIENCIAS - ESCUELA PROFESIONAL DE FISICA

LABORATORIO DE ENERGIA SOLAR

Fecha: 17 de Agosto de 1994

Inicio de acumulación: 12:45:02 - 4 de Agosto

Número de canales acumulados: 8

**Primer día**

#### TEMPERATURA EN °C

HORA	CANAL0	CANAL1	CANAL2	CANAL3	CANAL4	CANAL5	CANAL6	CANAL7
:13	:18.48	:21.86	:16.07	:23.79	:19.93	:18.96	:19.93	:18.96
:14	:18.96	:22.34	:16.55	:23.31	:21.38	:18.96	:19.93	:18.96
:15	:18.48	:21.86	:16.55	:22.82	:22.34	:18.96	:19.93	:18.96
:16	:18.48	:21.38	:16.07	:22.34	:22.34	:19.93	:19.45	:18.96
:17	:17.52	:18.96	:15.59	:22.34	:22.34	:20.41	:19.45	:18.96
:18	:16.07	:15.59	:15.11	:21.86	:21.86	:19.93	:19.45	:18.48

:19	:15.11	:15.59	:14.62	:21.38	:21.38	:19.93	:19.45	:18.48
:20	:13.66	:15.11	:14.14	:21.38	:21.38	:19.45	:19.45	:18.48
:21	:13.66	:14.62	:13.66	:21.38	:21.38	:19.45	:19.45	18
:22	:12.21	:14.14	:13.66	:20.41	:20.41	:18.96	:19.45	:17.52
:23	:11.73	:13.66	:13.66	:19.93	:20.41	:18.96	:19.45	:17.52
:00	:11.73	:14.14	:13.66	:19.45	:19.93	:18.48	:18.96	:16.55
1	:11.73	:13.66	:13.66	:18.96	:19.45	:18.48	:18.96	:16.55
2	:11.73	:14.14	:13.66	:18.96	:18.96	:18.48	:18.96	:16.07
3	:12.21	:13.66	:13.66	:18.96	:18.96	18	:18.48	:15.59
4	:12.21	:13.66	:13.66	:18.96	:18.96	18	:18.48	:15.59
5	:13.66	:14.14	:13.66	:18.48	:18.96	:17.52	:18.48	:15.11
6	:13.66	:14.14	:13.66	:18.48	:18.48	:17.52	18	:15.11
7	:12.69	:13.66	:13.66	:18.48	:18.48	:17.52	18	:15.11
8	:13.66	:14.14	:14.14	18	:18.48	:17.52	:17.52	:15.11
9	:15.11	:15.11	:15.11	:18.48	:18.48	:17.52	:17.52	:15.11
:10	:15.59	:19.93	:16.07	:18.48	:18.48	:17.52	:17.52	:15.11
:11	:17.52	:22.82	:17.52	:22.34	18	:17.52	:17.52	:16.55
:12	:17.52	:23.79	:17.52	:25.24	:21.38	:17.52	18	:17.52

**Segundo día**

**TEMPERATURA EN °C**

:HORA	:CANAL0	:CANAL1	:CANAL2	:CANAL3	:CANAL4	:CANAL5	:CANAL6	:CANAL7
13	:17.52	:23.79	:17.52	:24.27	:23.79	:18.96	18	:17.52
14	:17.52	:22.82	:16.55	:24.27	:24.27	:21.38	:18.48	:17.52
15	:17.52	:22.34	:16.55	:23.79	:23.79	:22.34	:19.45	:17.52
16	:17.52	:21.38	:16.55	:23.31	:23.79	:22.34	:20.89	18
17	:17.52	:16.07	:15.11	:22.82	:23.31	:21.86	:21.38	18
18	:15.59	:16.07	:15.11	:22.34	:22.82	:21.38	:21.38	:17.52
19	:14.62	:16.07	:15.11	:22.34	:22.34	:21.38	:21.38	:17.52
20	:14.62	:15.59	:14.62	:22.34	:22.34	:20.89	:20.41	:17.52
21	:14.14	:15.59	:14.62	:21.86	:22.34	:20.41	:20.41	:17.52
22	:13.66	:15.11	:14.14	:21.38	:21.86	:20.41	:19.93	:17.52
23	:13.66	:15.11	:14.62	:21.38	:21.38	:20.41	:20.41	:17.04
00	:13.66	:15.11	:14.14	:21.38	:21.38	:19.45	:19.93	:16.55
1	:13.66	:15.11	:14.14	:21.38	:21.38	:19.45	:19.45	:16.55

2	13.66	14.62	14.14	20.41	20.89	18.96	19.45	16.55
3	13.66	15.11	14.14	20.41	20.41	18.96	18.96	16.07
4	13.66	14.62	14.14	20.41	20.41	18.96	18.96	16.07
5	13.66	14.62	13.66	19.45	20.41	18.48	18.96	15.59
6	13.66	14.62	13.66	18.96	19.93	18.48	18.48	15.59
7	13.66	14.62	14.14	18.96	19.45	18.48	18.48	15.59
8	13.66	14.62	14.62	18.96	18.96	18	18.48	15.59
9	14.62	15.11	14.62	18.96	18.96	17.52	18	15.11
10	15.59	18.48	15.59	18.48	18.96	17.52	18	15.59
11	17.52	21.38	16.07	20.41	18.96	17.52	18.48	16.55
12	18.48	27.65	18.48	26.2	19.45	17.52	18.48	17.52

**Tercer día**

**TEMPERATURA EN °C**

HORA	CANAL0	CANAL1	CANAL2	CANAL3	CANAL4	CANAL5	CANAL6	CANAL7
13	18.48	26.2	17.52	28.13	25.24	18.48	18.48	18
14	18.96	30.06	18.48	30.06	27.16	22.82	18.96	18
15	19.45	31.51	18.96	30.54	29.09	25.72	21.86	18.48
16	19.45	27.16	17.52	30.06	29.58	26.2	25.24	20.41
17	20.41	26.2	16.55	28.61	29.09	26.68	25.72	21.86
18	18	17.52	15.11	27.65	28.13	26.2	25.72	21.38
19	16.07	18	14.62	26.68	27.16	26.2	25.24	21.38
20	15.11	17.52	15.11	26.2	26.68	25.24	25.24	20.41
21	14.62	17.52	14.62	25.72	26.2	25.24	25.24	19.93
22	14.14	16.55	14.62	25.24	25.72	24.27	24.27	18.96
23	14.14	16.55	14.62	24.75	25.24	24.27	24.27	18.96
00	13.66	16.07	14.14	24.27	25.24	23.31	23.31	18.48
1	13.66	16.07	14.14	24.27	24.27	22.82	23.31	18.48
2	13.66	15.59	14.14	23.31	23.79	22.34	22.82	17.52
3	13.66	15.59	14.62	22.82	23.31	21.86	22.34	17.52
4	13.66	15.59	14.14	22.82	23.31	21.86	21.86	17.52
5	13.66	15.59	14.62	22.34	22.82	21.38	21.38	17.04
6	13.66	15.11	14.14	22.34	22.34	21.38	21.38	17.52
7	13.66	15.11	14.14	21.86	22.34	21.38	21.38	16.55
8	13.66	15.11	13.66	21.38	21.86	20.41	20.41	16.55
9	14.14	15.11	14.14	21.38	21.38	20.41	20.41	16.07

10	15.59	15.59	15.11	21.38	21.38	20.41	20.41	16.07
11	17.52	16.07	15.59	21.38	21.38	19.93	19.93	16.07
12	16.55	23.31	16.07	21.86	21.38	19.45	20.41	17.04

**Cuarto día**

**TEMPERATURA EN °C**

+-----+-----+-----+-----+-----+-----+-----+-----+-----+								
: HORA	: CANAL0	: CANAL1	: CANAL2	: CANAL3	: CANAL4	: CANAL5	: CANAL6	: CANAL7
+-----+-----+-----+-----+-----+-----+-----+-----+-----+								
13	17.52	19.45	15.11	21.38	21.38	19.45	20.41	18
14	17.52	18.96	15.11	21.38	21.38	18.96	19.93	18
15	17.52	21.38	16.07	20.41	20.41	18.96	20.41	18.48
16	17.52	20.41	15.59	20.41	20.41	18.96	20.41	18.96
17	17.52	18.96	15.11	19.93	20.41	18.96	19.93	18.48
18	16.07	15.59	15.11	19.93	20.41	18.96	19.93	18.48
19	14.62	15.59	14.62	19.45	20.41	18.48	19.45	18.48
20	14.14	15.11	14.62	19.45	19.93	18.48	19.45	17.52
21	14.14	15.11	14.62	18.96	19.45	18.48	18.96	17.52
22	13.66	15.11	14.62	18.96	19.45	18.48	18.96	17.52
23	13.66	15.11	14.14	18.96	18.96	18	18.96	17.52
00	13.66	14.62	14.62	18.96	18.96	18	18.96	16.55
1	13.66	14.62	14.62	18.48	18.96	17.52	18.48	16.55
2	13.66	14.62	14.14	18.48	18.96	17.52	18.48	16.07
3	13.66	14.62	14.14	18.48	18.48	17.52	18.48	16.07
4	13.66	14.14	14.14	18.48	18.48	17.52	18	16.07
5	13.66	14.62	14.14	18.48	18.48	17.52	18	15.59
6	13.66	14.14	13.66	18	18.48	17.52	17.52	15.59
7	13.66	13.66	13.66	18	18	17.52	17.52	15.59
8	13.66	14.14	13.66	18	18	16.55	17.52	15.11
9	15.59	15.59	15.11	17.52	18	16.55	17.52	15.11
10	15.59	21.38	16.07	18.96	17.52	16.55	17.52	15.59
11	16.07	20.41	16.07	19.93	18	16.55	17.52	16.55
12	17.52	21.86	16.07	21.38	18	16.55	17.52	17.52

**Quinto día**

**TEMPERATURA EN °C**



HORA	CANAL0	CANAL1	CANAL2	CANAL3	CANAL4	CANAL5	CANAL6	CANAL7
13	17.52	21.86	17.52	21.86	18.96	16.55	17.52	17.52
14	17.52	23.31	17.52	22.82	20.41	17.52	17.52	17.52
15	17.52	22.82	17.52	23.31	21.38	18.48	17.52	17.52
16	17.52	21.38	16.07	22.82	22.34	18.96	18	17.52
17	15.59	17.52	15.11	22.34	21.86	19.45	18.48	17.52
18	15.59	15.11	15.11	21.86	21.38	18.96	18	17.52
19	14.62	15.11	14.62	21.38	21.38	18.96	18	17.52
20	13.66	14.62	14.14	21.38	21.38	18.96	18	17.52
21	13.66	14.62	14.14	20.41	20.41	18.96	18	16.55
22	13.66	14.62	14.14	19.93	20.41	18.48	18	16.07
23	13.66	14.62	14.62	19.93	19.93	18.48	18	16.07
00	13.66	14.62	14.14	19.45	19.45	18	18	16.07
1	13.66	14.14	14.14	18.96	19.45	18	18	15.59
2	13.66	14.62	14.14	18.96	18.96	18	17.52	15.59
3	13.66	14.14	14.14	18.96	18.96	17.52	17.52	15.59
4	13.66	14.14	14.14	18.48	18.96	17.52	17.52	15.11
5	13.66	14.14	14.14	18.48	18.48	17.52	17.52	15.11
6	13.66	14.62	14.14	18.48	18.48	17.52	17.52	15.11
7	13.66	14.62	14.62	18.48	18.48	17.52	17.52	15.11
8	14.14	14.62	14.62	18	18	17.52	17.52	15.11
9	15.11	15.11	15.11	18	18.48	16.55	17.52	15.11
10	15.59	18	15.59	17.52	18	16.55	17.52	15.11
11	16.07	20.41	16.55	18.96	18	16.55	17.52	15.59
12	16.55	22.34	16.55	21.38	18.48	16.55	17.52	16.55

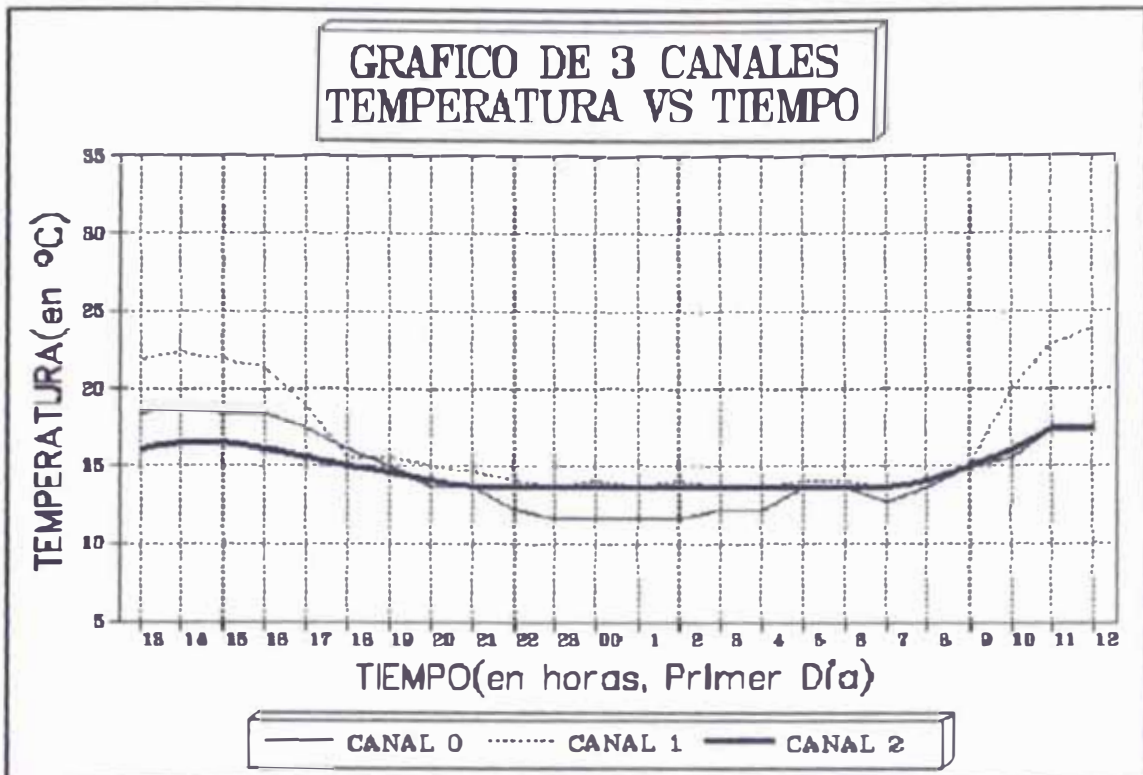


FIG. #6: Gráfico de datos Temperatura versus tiempo durante un día transcurrido.

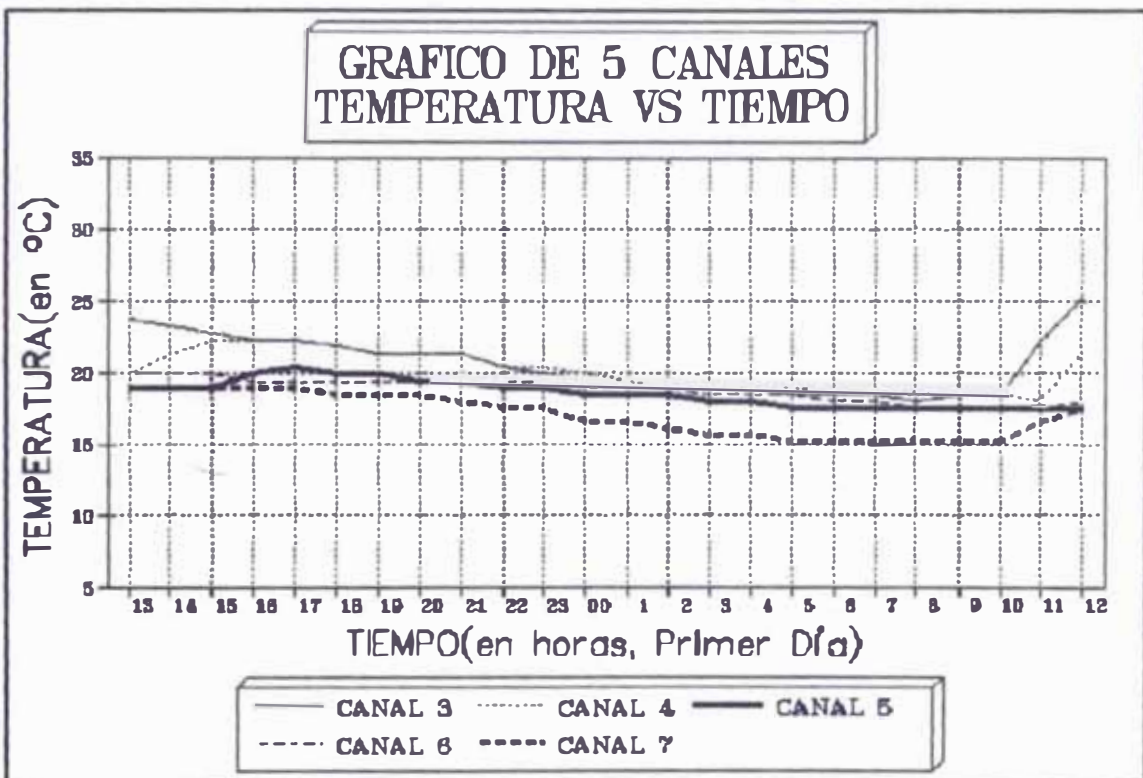


FIG. #7: Gráfico de datos de Temperatura VS. Tiempo durante un día transcurrido.

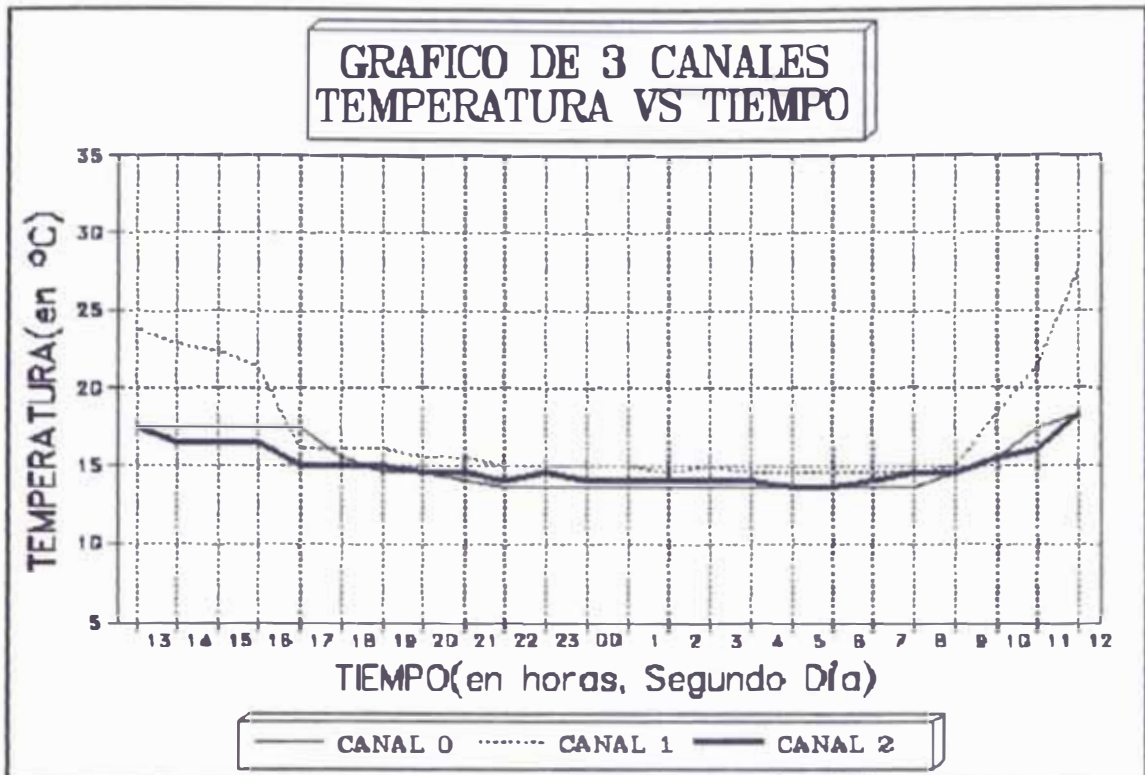


FIG. #8: Gráfico de datos de Temperatura VS. Tiempo durante un día.

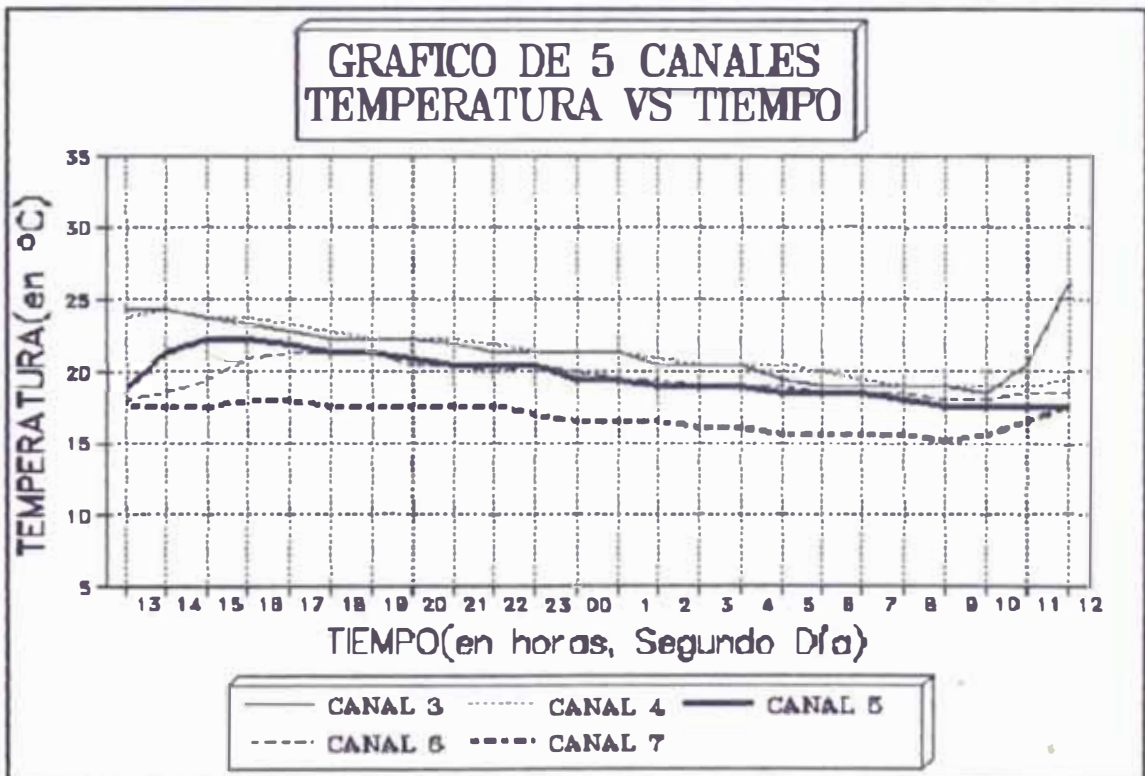


FIG. # 9: Gráfico de datos de Temperatura VS. Tiempo

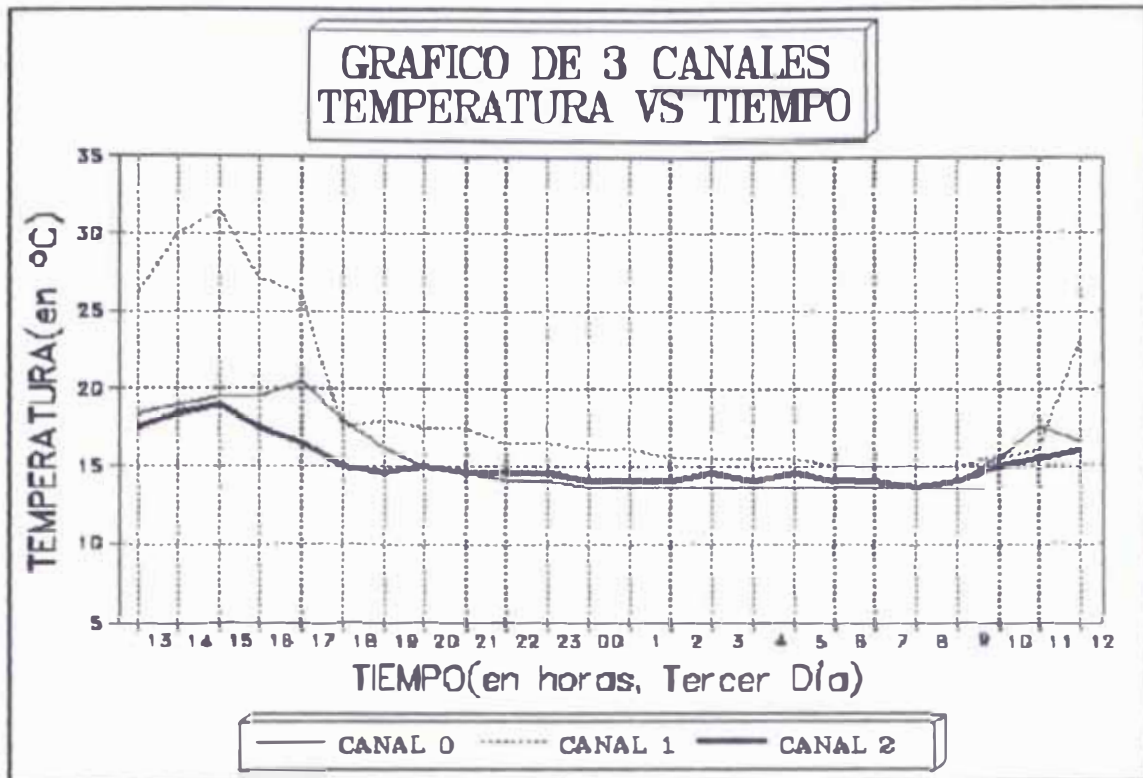


FIG. #10: Gráfico de datos de Temperatura VS. Tiempo durante un día transcurrido.

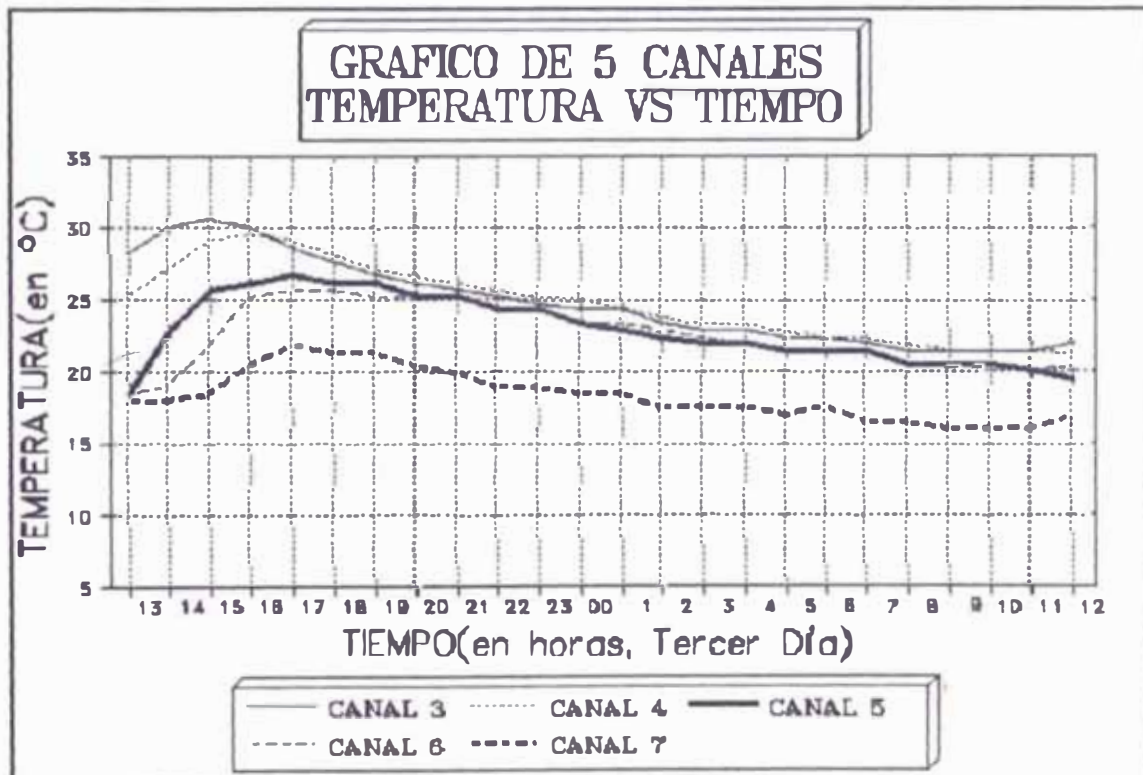


FIG. #11: Gráfico de datos de Temperatura VS. Tiempo de un día.

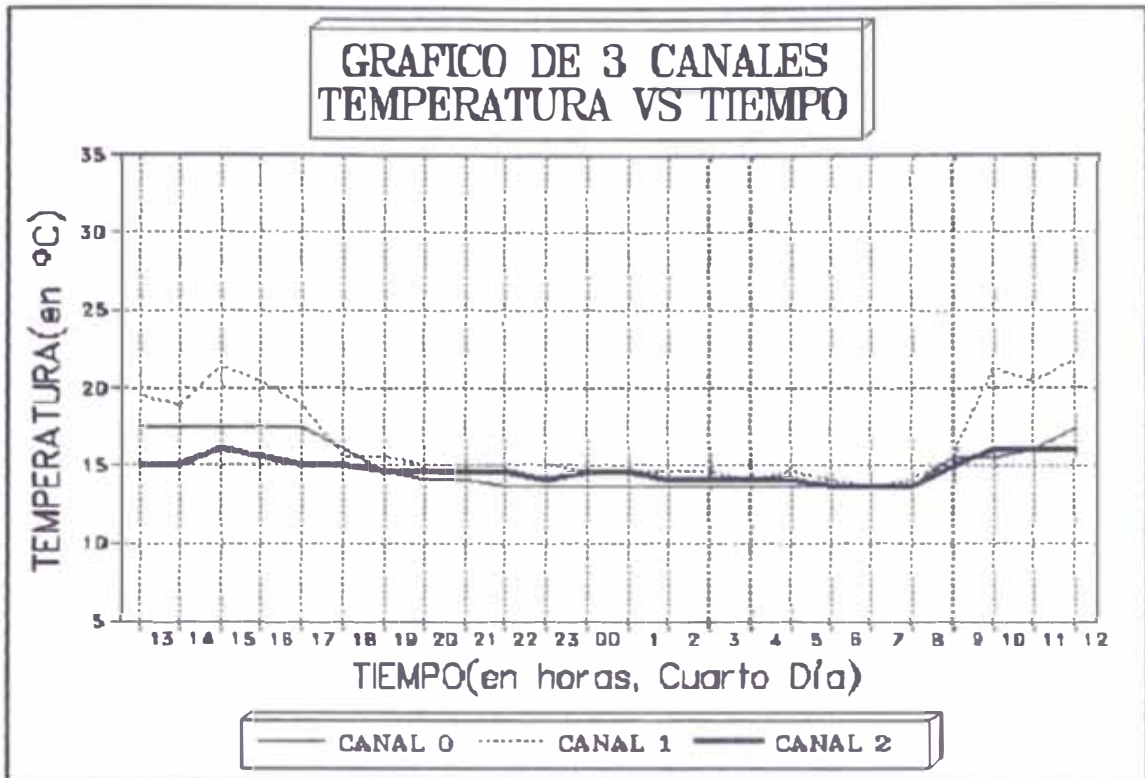


FIG. #12: Gráfico de datos de Temperatura VS. Tiempo.

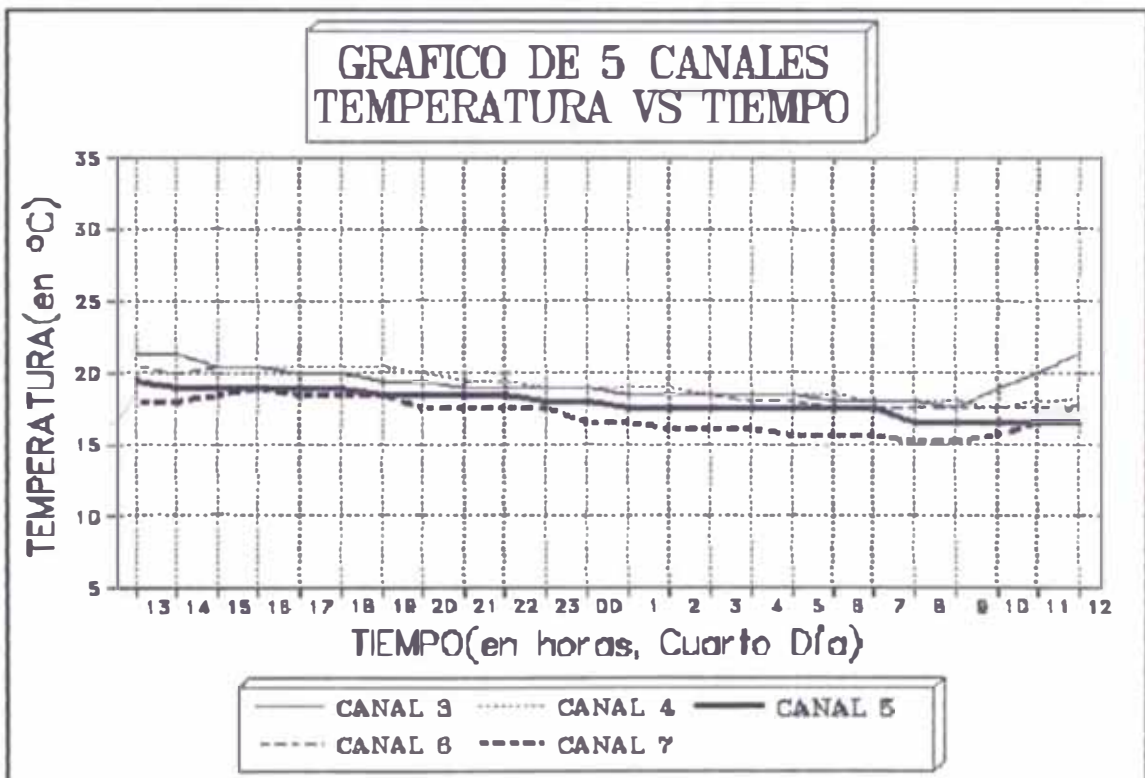


FIG. #13: Gráfico de datos de Temperatura VS. Tiempo.

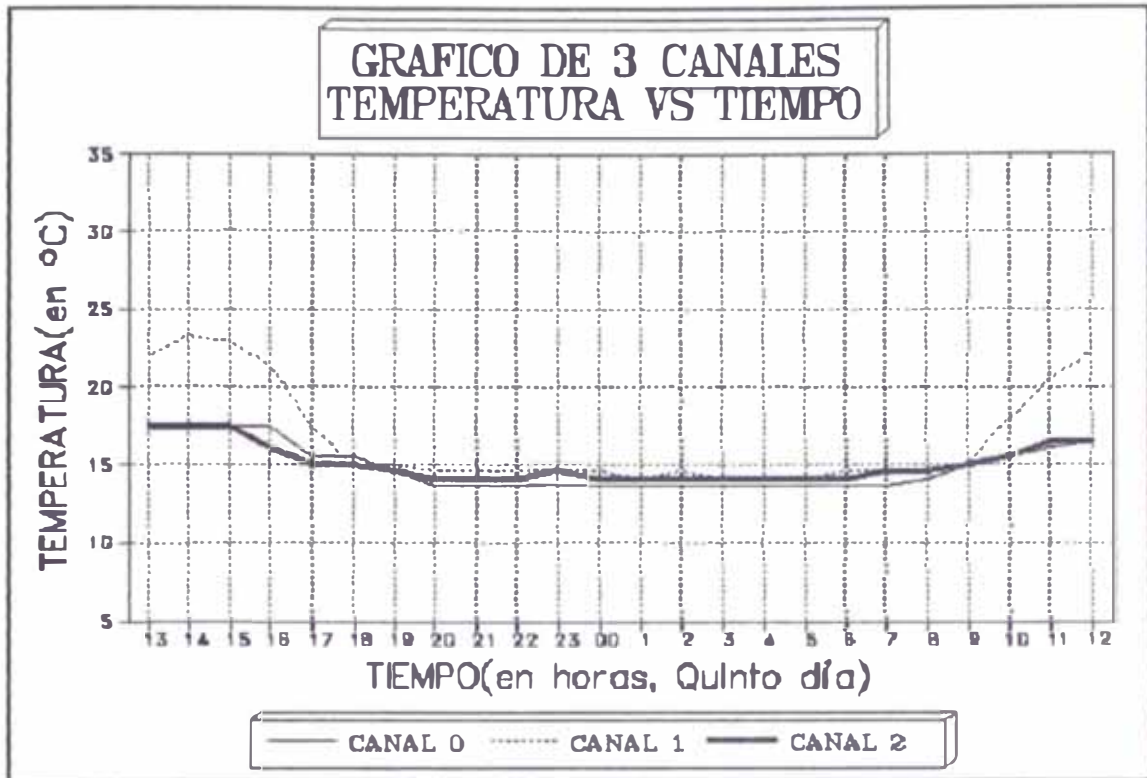


FIG. #14: Gráfico de datos Temperatura VS. Tiempo durante un día.

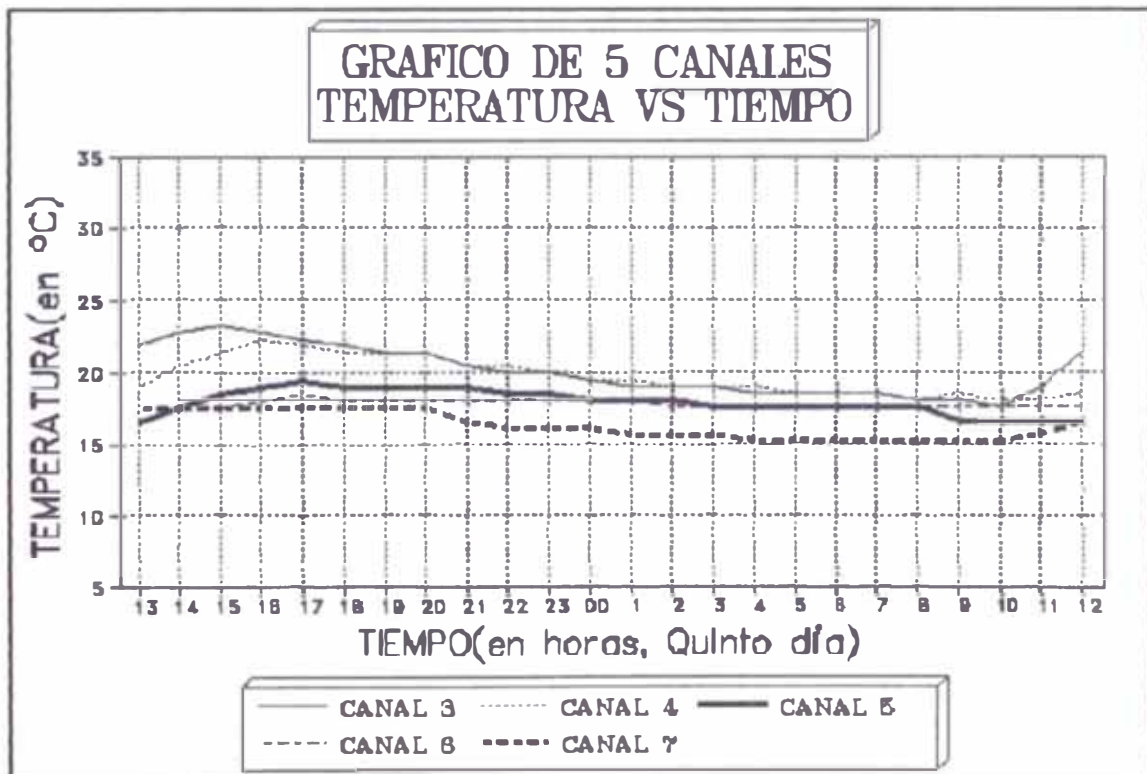


Fig. #15: Gráfico de datos de Temperatura VS. Tiempo en un día transcurrido.

### 3.3.1 COMENTARIO ACERCA DE LOS DATOS TOMADOS

En el **primer día**(desde las 13 horas del 4 de Agosto hasta las 12 horas del 5 de Agosto) se tiene una temperatura máxima de 25.24 °C(canal 3) cuando la temperatura ambiente es 17.52 °C, y una temperatura mínima de 11.73 °C(canal 0) cuando la temperatura ambiente es 13.66 °C.

Se puede notar una diferencia de temperatura mayor entre el canal 3 y el canal 2 en comparación con los demás canales. Esto es aceptable ya que el canal 3 se encuentra instalado en la parte más alta del tanque de agua y el canal 2 es temperatura ambiente. Mayores temperaturas de ambos canales, se notan alrededor de mediodía.

Es aceptable también la diferencia de temperatura que siempre se da entre el canal 0(entrada agua fría) y el canal1(salida de agua caliente), esta diferencia tiende a disminuir considerablemente en la noche.

Con respecto a los canales 7, 6, 5 y 4 se observa el progreso de la temperatura de menos a más en forma escalonada(es más notorio en la tarde), siendo el canal 7 el más frío en comparación con los canales 6, 5 y 4 y el canal 4 el más caliente también en comparación con esos canales.

Entre las 21 horas y cercano a las 5 de la mañana, la temperatura que da el canal 0(entrada de agua fría) es menor que la temperatura ambiente(canal 2). Esto se debe a que la temperatura del agua en el colector se enfría más por irradiación infrarroja. Como cuerpo sólido emite más radiación que el aire(es lo que se llama enfriamiento radiativo). Este fenómeno sólo se da si el cielo es transparente, porque en el caso que hubieran nubes estas emiten infrarrojo.

En el **segundo día**(desde las 13 horas del 5 de Agosto hasta las 12 horas del 6 de Agosto) se tiene una temperatura máxima de 27.65 °C(canal 1)cundo la temperatura ambiente es 18.48°C, y una temperatura mínima de 13.66 °C(canal 0) cuando la temperatura ambiente es 14.14°C.

Se observa que alrededor de las 12 del día el canal 1 supera en temperatura al canal 3.

Con respecto a los canales 7, 6, 5 y 4 se observa el progreso de la temperatura de menos a más en forma escalonada(es más notorio en la tarde), siendo el canal 7 el más frío en comparación con los canales 6, 5 y 4 y el canal 4 el más caliente en comparación con esos canales.

En el **tercer día**(desde las 13 horas del 6 de Agosto hasta las 12 horas del 7 de Agosto) se tiene una temperatura máxima de 31.51 °C(canal 1) cuando la

temperatura ambiente es 18.96 °C, y una temperatura mínima de 13.66 °C(canal 0) cuando la temperatura ambiente es 14.14 °C.

Con respecto a los canales 7, 6, 5 y 4 se observa el progreso de la temperatura de menos a más en forma escalonada(es más notorio en la tarde 15,16,17,18 y 19 horas), siendo el canal 7 el más frío en comparación con los canales 6, 5 y 4 y el canal 4 el más caliente también en comparación con esos canales. En la noche estas temperaturas tienden a equilibrarse.

En el cuarto día(desde las 13 horas del 7 de Agosto hasta las 12 horas del 8 de Agosto) se tiene una temperatura máxima de 21.86 °C(canal 1) cuando la temperatura ambiente es 16.07 °C, y una temperatura mínima de 13.66 °C(canal 0) cuando la temperatura ambiente es 13.66 °C.

Con respecto a los canales 7, 6, 5 y 4 no se observa claramente el progreso de la temperatura de menos a más en forma escalonada. En la noche estas temperaturas tienden a equilibrarse.

En el quinto día(desde las 13 horas del 8 de Agosto hasta las 12 horas del 9 de Agosto) se tiene una temperatura máxima de 23.31 °C(canal 3) cuando la temperatura ambiente es 17.52 °C, y una temperatura mínima de 13.66 °C(canal 0) cuando la temperatura ambiente es 14.14 °C.

Con respecto a los canales 7, 6, 5 y 4 se observa el progreso de la temperatura de menos a más en forma escalonada en la tarde. En la noche estas temperaturas tienden a equilibrarse.

### **3.4 DATOS ACUMULADOS DE LA TERMA SOLAR PROMEDIADOS CADA HORA**

Los datos de los ocho canales se tomaron aproximadamente cada 3 minutos durante 5 días, haciendo un total de 18576 datos acumulados. Después estos datos se enviaron del ADATOC vía comunicación serial a la computadora, allí mediante un software se procedió a promediar cada hora(cada 20 datos) de la siguiente forma 10 datos antes de la hora y 10 datos después de la hora.

Los datos obtenidos son los siguientes:

**UNIVERSIDAD NACIONAL DE INGENIERIA  
FACULTAD DE CIENCIAS – ESCUELA PROFESIONAL DE FISICA  
LABORATORIO DE ENERGIA SOLAR**

Fecha: 9 de Agosto de 1994



Promedio cada 20 datos.

Inicio de acumulación: 12:45:02 - 4 de Agosto

Número de canales acumulados: 8

Intervalo de tiempo en toma de datos: 185.98 segundos.

Temperatura en grados centígrados.

HORA	CANAL0	CANAL1	CANAL2	CANAL3	CANAL4	CANAL5	CANAL6	CANAL7
13:16:01	18.87	22.20	16.50	23.67	20.41	18.94	19.91	18.96
14:18:01	18.70	22.12	16.53	22.97	21.83	18.92	19.78	18.96
15:20:00	18.65	22.03	16.75	22.63	22.44	19.01	19.59	18.96
16:22:00	18.22	20.94	16.24	22.44	22.34	19.98	19.57	18.96
17:23:59	17.04	16.77	15.32	22.15	22.24	20.22	19.69	18.96
18:25:59	15.73	15.81	15.03	21.64	21.88	20.03	19.45	18.65
19:27:58	14.53	15.42	14.58	21.38	21.38	19.74	19.45	18.43
20:29:58	13.66	14.96	14.21	21.23	21.38	19.30	19.42	18.17
21:31:57	12.62	14.41	13.66	20.48	20.99	18.99	19.28	17.64
22:33:57	12.04	13.83	13.66	20.24	20.41	18.96	19.25	17.52
23:35:56	11.66	13.68	13.59	19.83	20.17	18.65	19.01	17.08
00:37:56	11.49	13.68	13.63	19.35	19.78	18.46	18.96	16.43
01:39:56	11.78	13.76	13.66	18.99	19.25	18.39	18.96	16.12
02:41:55	12.19	13.85	13.66	18.96	19.16	18.05	18.58	15.71
03:43:55	12.38	13.92	13.66	18.94	18.96	17.93	18.48	15.59
04:45:54	12.91	13.97	13.66	18.55	18.96	17.57	18.48	15.42
05:47:54	13.25	13.95	13.66	18.48	18.80	17.52	18.14	15.18
06:49:53	13.37	13.80	13.66	18.39	18.51	17.52	18.00	15.11
07:51:53	13.83	14.31	14.14	18.29	18.48	17.52	17.71	15.11
08:53:52	14.99	15.23	14.82	18.14	18.36	17.52	17.52	15.11
09:55:52	15.76	19.40	15.88	18.80	18.39	17.30	17.52	15.40
10:57:51	17.47	23.04	17.52	22.32	18.39	17.20	17.66	16.72
11:59:51	17.57	23.98	17.59	24.70	21.11	17.45	17.95	17.49
13:01:50	17.57	23.69	17.52	24.27	23.81	18.89	18.00	17.52
14:03:50	17.52	22.75	16.89	24.20	24.27	21.47	18.46	17.52
15:05:49	17.52	22.27	16.77	23.60	23.91	22.34	19.66	17.52
16:07:49	17.40	20.41	15.93	23.21	23.45	22.15	21.16	17.86
17:09:48	16.77	16.12	15.11	22.82	23.21	21.79	21.38	17.98
18:11:48	15.44	16.24	14.84	22.49	22.82	21.38	21.28	17.66

19:13:47	14.72	16.12	14.72	22.32	22.39	21.38	20.73	17.52
20:15:47	14.26	15.66	14.50	21.98	22.34	21.04	20.41	17.52
21:17:46	14.07	15.49	14.60	21.55	22.05	20.41	20.41	17.52
22:19:46	13.66	15.20	14.33	21.38	21.52	20.32	20.12	17.40
23:21:45	13.66	15.11	14.21	21.35	21.38	20.05	19.98	16.96
00:23:45	13.66	15.01	14.17	21.21	21.38	19.57	19.64	16.60
01:25:44	13.66	14.89	14.19	20.51	21.18	19.16	19.35	16.48
02:27:44	13.66	14.72	14.24	20.36	20.56	18.96	19.04	16.34
03:29:44	13.66	14.67	14.24	20.15	20.41	18.96	18.96	16.07
04:31:43	13.66	14.62	14.04	19.91	20.32	18.70	18.96	15.97
05:33:43	13.66	14.53	13.95	19.42	19.98	18.48	18.87	15.59
06:35:42	13.66	14.53	14.07	19.16	19.66	18.39	18.51	15.59
07:37:42	13.78	14.58	14.29	18.96	19.33	18.17	18.48	15.54
08:39:41	14.45	14.77	14.55	18.94	19.06	17.95	18.46	15.49
09:41:41	15.59	16.79	15.15	18.70	18.96	17.59	18.29	15.54
10:43:40	16.65	21.33	16.24	20.17	18.96	17.52	18.48	16.53
11:45:40	18.19	25.50	17.88	24.22	19.35	17.52	18.48	17.54
12:47:39	18.63	26.88	18.07	27.45	23.62	18.17	18.48	17.98
13:49:39	18.96	29.50	18.82	29.36	27.04	22.10	18.92	18.00
14:51:38	19.04	29.75	18.48	30.49	28.66	25.43	21.67	18.53
15:53:38	19.69	27.82	17.54	29.91	29.31	26.37	24.78	20.24
16:55:37	19.91	25.36	16.60	28.61	28.88	26.68	25.91	21.74
17:57:37	18.05	17.59	15.30	27.65	28.13	26.39	25.93	21.64
18:59:36	16.48	17.71	14.74	26.71	27.14	25.98	25.48	21.21
20:01:36	15.25	17.52	14.77	26.25	26.59	25.24	25.24	20.34
21:03:35	14.50	17.42	14.58	25.62	26.15	24.85	25.09	19.66
22:05:35	14.17	16.58	14.38	25.24	25.62	24.17	24.32	19.01
23:07:34	13.92	16.19	14.31	24.73	25.24	23.72	24.05	18.68
00:09:34	13.66	15.90	14.14	24.20	24.90	23.16	23.43	18.29
01:11:33	13.66	15.64	14.14	23.76	24.22	22.70	23.09	17.93
02:13:33	13.66	15.73	14.29	23.31	23.86	22.34	22.61	17.52
03:15:32	13.66	15.59	14.31	22.99	23.31	22.12	22.34	17.52
04:17:32	13.66	15.49	14.17	22.65	22.99	21.59	21.93	17.49
05:19:32	13.66	15.56	14.29	22.34	22.75	21.38	21.42	17.01
06:21:31	13.66	15.20	14.21	22.20	22.34	21.33	21.38	16.79
07:23:31	13.66	15.11	14.19	21.79	22.24	20.92	21.16	16.55
08:25:30	13.68	15.08	14.12	21.38	21.76	20.41	20.46	16.46

09:27:30	14.48	15.18	14.36	21.38	21.40	20.29	20.39	16.24
10:29:29	16.79	16.26	15.32	21.28	21.38	19.98	20.19	16.38
11:31:29	17.32	19.95	16.41	21.18	21.38	19.57	20.03	16.63
12:33:28	17.08	20.58	15.25	21.71	21.26	19.40	20.41	17.64
13:35:28	16.94	19.21	15.15	21.23	21.06	19.06	20.39	18.02
14:37:27	17.54	20.68	15.71	20.46	20.51	18.96	20.29	18.34
15:39:27	17.54	20.80	15.56	20.41	20.41	18.96	20.24	18.55
16:41:26	17.32	19.47	15.32	20.29	20.41	18.96	20.00	18.80
17:43:26	15.97	15.47	14.82	20.03	20.32	18.96	19.98	18.51
18:45:25	15.15	15.64	14.62	19.71	20.12	18.60	19.62	18.34
19:47:25	14.50	15.37	14.62	19.37	19.93	18.48	19.33	17.73
20:49:24	14.26	15.15	14.60	19.06	19.54	18.43	19.09	17.52
21:51:24	13.90	15.11	14.53	18.96	19.25	18.24	18.96	17.52
22:53:23	13.66	14.79	14.36	18.96	19.06	18.02	18.96	17.35
23:55:23	13.66	14.62	14.31	18.96	18.96	17.88	18.87	16.94
00:57:22	13.66	14.62	14.26	18.55	18.96	17.54	18.48	16.48
01:59:22	13.66	14.60	14.33	18.48	18.92	17.52	18.48	16.29
03:01:21	13.66	14.53	14.19	18.48	18.63	17.52	18.36	16.12
04:03:21	13.66	14.48	14.14	18.43	18.48	17.52	18.27	15.95
05:05:20	13.66	14.33	14.14	18.27	18.48	17.52	18.05	15.61
06:07:20	13.66	14.26	13.92	18.10	18.43	17.32	17.83	15.59
07:09:20	13.61	13.90	13.66	17.95	18.24	17.04	17.57	15.44
08:11:19	14.04	14.43	14.17	17.73	18.05	16.79	17.52	15.20
09:13:19	15.49	17.30	15.32	17.61	17.98	16.60	17.52	15.18
10:15:18	15.93	20.41	15.71	19.21	17.78	16.55	17.52	16.00
11:17:18	16.77	20.97	16.00	20.00	17.98	16.55	17.52	16.72
12:19:17	17.52	22.08	16.48	21.45	18.60	16.55	17.52	17.01
13:21:17	17.52	22.34	17.13	22.00	19.59	16.96	17.52	17.13
14:23:16	17.52	23.45	17.45	23.16	21.18	17.64	17.52	17.37
15:25:16	17.52	22.32	17.01	23.16	21.96	18.70	17.76	17.40
16:27:15	16.70	19.71	15.78	22.56	22.24	19.13	18.27	17.40
17:29:15	15.59	15.49	15.11	22.12	21.79	19.16	18.29	17.42
18:31:14	15.08	15.23	14.94	21.55	21.38	19.04	18.31	17.28
19:33:14	14.14	14.91	14.29	21.35	21.38	18.96	18.29	16.89
20:35:13	13.66	14.62	14.12	20.70	20.99	18.96	18.27	16.58
21:37:13	13.66	14.62	14.14	20.36	20.39	18.48	18.12	16.34
22:39:12	13.66	14.62	14.43	19.93	20.15	18.48	18.12	16.17

23:41:12	13.66	14.62	14.43	19.42	19.83	18.34	18.00	16.02
00:43:11	13.66	14.58	14.48	19.11	19.40	18.24	18.00	15.83
01:45:11	13.66	14.38	14.24	18.96	19.09	17.90	17.71	15.59
02:47:10	13.66	14.50	14.29	18.96	18.96	17.54	17.57	15.59
03:49:10	13.66	14.26	14.14	18.63	18.94	17.52	17.52	15.30
04:51:09	13.66	14.41	14.17	18.48	18.80	17.52	17.52	15.20
05:53:09	13.66	14.29	14.19	18.41	18.48	17.52	17.52	15.11
06:55:08	13.66	14.33	14.26	18.24	18.46	17.52	17.52	15.11
07:57:08	14.19	14.62	14.65	18.07	18.29	17.37	17.52	15.11
08:59:08	15.30	15.11	15.13	17.98	18.14	17.20	17.52	15.11
10:01:07	15.59	17.76	15.52	17.95	18.02	16.77	17.52	15.11
11:03:07	16.19	20.51	16.67	19.16	18.02	16.79	17.52	15.64
12:05:06	17.28	21.96	17.25	21.57	18.22	16.75	17.52	16.36

La siguiente Tabla contiene las temperaturas máxima, media y mínima de cada uno de los canales durante los cinco días que se almacenaron los datos.

CANAL	MAXIMA	MEDIA	MINIMA	°C
0	20.41	15.83	11.25	
1	31.51	22.58	13.66	
2	19.45	16.07	12.69	
3	31.02	24.27	17.52	
4	29.58	23.55	17.52	
5	26.68	21.62	16.55	
6	26.20	21.86	17.52	
7	22.34	18.72	15.11	

**Número total de datos almacenados en los cinco días: 18576 Datos**

## CAPITULO IV: EL HARDWARE DEL ADATOC

### 4.1 EL MICROPROCESADOR Z80

Hay muchos microprocesadores diferentes en el mercado y aunque la nomenclatura de las instrucciones es algo distinta para cada uno, los procesos básicos de cómputo lógico son semejantes en todos los dispositivos.

Se selecciono el CPU Z80 para usarlo en el ADATOC porque al ser un microprocesador muy popular en el Perú, hay disponibilidad muy amplia de libros y mucha documentación que tratan sobre el Z80, tanto en Hardware y Software. También su bajo costo y la posibilidad de comprar el CPU Z80 y su familia completa en el mercado local, aunque cuando ya se tuvo diseñado el equipo se tuvo que traer CPU Z80 en versión CMOS del Japón, para disminuir el consumo de corriente del equipo.

Otro factor que influyó en el uso del Z80 fue que se contaba con un sistema de desarrollo con todos sus accesorios en el Laboratorio de la Facultad de Ciencias de la U.N.I.

Sobre las propiedades del hardware y software del Z80 puede consultar los apéndices que tratan de él.

### 4.2 EL CONVERTIDOR A/D (ANALOGO/DIGITAL)

#### **INTRODUCCION:**

El mundo como sabemos es un mundo análogo. Por otro lado, el mundo del microprocesador es un mundo digital que hace uso de 1's y 0's. Por tanto, para que un microprocesador pueda comunicarse con el mundo análogo real, una conversión digital-a-análogo(D/A) o análogo-a-digital(A/D) debe tomar lugar.

El convertidor A/D es usado para convertir la salida de un sensor análogo a un valor digital que puede ser evaluado por el microprocesador.

#### 4.2.1 DESCRIPCION GENERAL DEL ADC0808/09

El componente de adquisición de datos ADC0808, ADC0809 es un dispositivo monolítico CMOS que contiene un convertidor análogo a digital, un multiplexor de 8 canales y control lógico compatible al microprocesador. El convertidor A/D de 8 bits usa aproximaciones sucesivas como la técnica

de conversión. El multiplexor de 8 canales puede directamente acceder cualquiera de las ocho señales análogas.

Es fácil de interfacear a microprocesadores debido al latched de las entradas de dirección del multiplexor y el latched de las salidas TTL TRI-STATE.

El diseño de ADC0808, ADC0809 ha sido optimizado porque incorporan los aspectos más deseables de varias técnicas de conversión A/D. ADC0808, ADC0809 ofrece alta velocidad, alta precisión, mínima dependencia de la temperatura, y consumo mínimo de potencia.

#### **4.2.2 CARACTERISTICAS**

RESOLUCION: 8 bits

TIEMPO DE CONVERSION: 100  $\mu$ Seg.

FUENTE UNICA DE PODER: 5Vdc.

MULTIPLEXOR DE 8 CANALES CON LATCHED EN EL CONTROL LOGICO.

FACIL DE INTERFACEAR A TODOS LOS MICROPROCESADORES.

RANGO DE VOLTAJE DE ENTRADA ANALOGA DE 0V A 5V.

EL DISPOSITIVO VIENE EN UN PAQUETE TIPO DIP(DUAL IN LINE PACKAGE)DE 28 PINES.

RANGO DE TEMPERATURA  $-40^{\circ}\text{C}$  A  $+85^{\circ}\text{C}$  O  $-55^{\circ}\text{C}$  A  $+125^{\circ}\text{C}$ .

FRECUENCIA DE RELOJ TIPICA  $f_c=640\text{kHz}$ . TIEMPO DE CONVERSION  $t_c=100\mu\text{S}$ .

FRECUENCIA DE RELOJ MAXIMA  $f_c=1280\text{kHz}$ . TIEMPO DE CONVERSION  $t_c=90\mu\text{S}$ .

BAJO CONSUMO DE POTENCIA 15mW.

SALIDA LATCHED TRI-STATE.

#### **4.2.3 DESCRIPCION FUNCIONAL**

**MULTIPLEXOR:** El dispositivo contiene un multiplexor de señales análogas de 8 canales. Una entrada de canal particular es seleccionado usando el decodificador de dirección. La tabla siguiente muestra los estados de entrada para las líneas de dirección para seleccionar cualquier canal.

CANAL ANALOGO SELECCIONADO		LINEA DE DIRECCION		
		C	B	A
ENTRADA CANAL 0	IN 0	0	0	0
ENTRADA CANAL 1	IN 1	0	0	1
ENTRADA CANAL 2	IN 2	0	1	0
ENTRADA CANAL 3	IN 3	0	1	1
ENTRADA CANAL 4	IN 4	1	0	0
ENTRADA CANAL 5	IN 5	1	0	1
ENTRADA CANAL 6	IN 6	1	1	0
ENTRADA CANAL 7	IN 7	1	1	1

#### 4.2.4 MODO DE FUNCIONAMIENTO

EL ADC0808 tiene 8 canales de entrada para ingreso de 8 señales análogas diferentes. Sólo se puede digitalizar una señal análoga a la vez que puede provenir de cualquiera de los 8 canales de entrada.

El ADC0808 tiene un pin de START (puede observar el plano# 1) que se activa en alta, una vez activado este pin el A/D empieza el proceso de digitalización de la señal análoga que proviene del canal seleccionado. Este canal es seleccionado al mismo instante en que se activa el pin

de START.

Para realizar este trabajo de inicializar el A/D y seleccionar el canal de entrada de la señal análoga el microprocesador debe ejecutar por ejemplo las siguientes instrucciones:

```
LD A,0
OUT (68H),A
```

Aquí selecciono el canal 0 de entrada de la señal análoga y además inicializó el ADC0808, cuya dirección es 68H. Para seleccionar otros canales de entrada por ejm. el canal 3, simplemente cambio el 0 por el 3 en la instrucción LD A,0 para tener LD A,3. De esta forma puedo seleccionar los demás canales de entrada 1,2,4,5,6 y 7.

El tiempo de conversión dura aproximadamente 100µSeg. y una vez finalizado este tiempo de conversión, se puede leer el valor digitalizado por el ADC. Antes de la lectura el valor digitalizado se mantiene dentro del A/D y no sale al bus de datos del CPU hasta que uno active el pin OE(OUTPUT ENABLE) que tiene como función habilitar la salida del valor digitalizado al bus de datos del CPU. Esto lo logra el CPU con la siguiente línea de instrucción:

```
IN A,(68H)
```

Con esta instrucción leemos el valor digitalizado del ADC0808. Todas las conexiones necesarias para el funcionamiento del A/D se dan en el plano # 1.

#### 4.2.5 EL VALOR DIGITALIZADO SELECCIONA SU VALOR EN CENTIGRADOS

Si aceptamos que el sensor tiene una salida de 10mv/°K, entonces partimos de una señal análoga que está en el rango de 2.33v a 3.73v(-40°C a 100°C, rango de funcionamiento del sensor LM335). Se ha fijado  $V_{ref(+)}=3.73v$  y  $V_{ref(-)}=2.5v$ . Tenemos que calcular una relación entre el valor en grados centígrados y el número digitalizado "n" por el A/D comprendido entre 0 y 255.

Sea  $V_{in}$  el voltaje de ingreso al A/D,  $(V_{in}-V_{ref(-)})$  será el voltaje digitalizado por el A/D. Luego el A/D digitalizará este voltaje en un rango de 0 a  $(V_{ref(+)}-V_{ref(-)})$ .



Para el cálculo de n se aplica la siguiente regla:

$$\frac{(V_{in}-V_{ref(-)})}{n} = \frac{(V_{ref(+)}-V_{ref(-)})}{255}$$

Entonces:

$$n = (V_{in}-V_{ref(-)}) \times 255 / (V_{ref(+)}-V_{ref(-)})$$

También:

$$V_{in} = V_{ref(-)} + n \times (V_{ref(+)}-V_{ref(-)}) / 255$$

Reemplazando los valores de  $V_{ref(+)}$  y  $V_{ref(-)}$ , tenemos:

$$V_{in} = 2.5 + n \times (1.23) / 255$$

Luego, como:

$$T^{\circ}C = (V_{in} - 2.73) \times 100$$

Entonces:

$$T^{\circ}C = ((n \times 1.23) / 255 - 0.23) \times 100$$

Esta es la fórmula que se usa para poder visualizar cualquier valor digitalizado por el A/D en grados centígrados.

El programa no calcula esta fórmula sino que usa una tabla que contiene todos los valores de temperatura calculadas con la fórmula anterior (en una calculadora aparte) que corresponde en el rango de 0 a 255.

La TABLA dentro del programa sirve para visualizar el valor medido en grados centígrados. Para ello el programa que se hizo debe usar el valor digitalizado por el A/D como puntero para apuntar a su valor equivalente en grados centígrados de la tabla.

La máxima precisión que se obtiene es  $0.48^{\circ}C$ , esto significa que todos los valores de temperatura que se midan serán con una precisión de  $0.48^{\circ}C$ .

### 4.3 FORMA DE USO DEL CTC EN EL ADATOC

El canal 0 del CTC se usa para generar una interrupción al CPU, cada vez que el RTC (RELOJ DE TIEMPO REAL) lo haga.

El canal 1 del CTC se usa para generar una interrupción al CPU, cada vez que se presione una tecla.

El canal 2 del CTC genera pulsos para obtener el pitido(sonido).

El canal 3 del CTC se usa para generar una interrupción al CPU, en el momento de recepción de datos.

#### 4.4 RELOJ DE TIEMPO REAL

Como reloj de tiempo real se ha usado el circuito integrado **MM58174A**. Este circuito funciona como reloj de tiempo real y calendario en sistemas basados en microprocesador. El tiempo puede mantenerse a un voltaje de 2.2V para permitir consumo de baja potencia en la operación de la Batería. El tiempo base es generado desde un oscilador producido por un cristal de 32768 Hz.

#### **CARACTERISTICAS**

- COMPATIBLE CON EL MICROPROCESADOR
- DECIMAS DE SEGUNDOS, SEGUNDOS, DECENAS DE SEGUNDOS, MINUTOS, DECENAS DE MINUTOS, DIA DE SEMANA, DECENAS DE DIAS, MESES, DECENAS DE MESES, REGISTROS INDEPENDIENTES.
- PROTECCION PARA LECTURA DURANTE CAMBIO DE DATOS.
- TIEMPO DE ACCESO RAPIDO(500ns).
- TTL COMPATIBLE.
- OPERACION STANDBY DE BAJA POTENCIA(2.2V, 10  $\mu$ A).
- PAQUETE TIPO DIP DE 16 PINES Y BAJO COSTO.

#### 4.5 PANTALLA DE CRISTAL LIQUIDO DEL ADATOC

Los días de los visualizadores de LED's se encuentran contados. Ahora se puede añadir un visualizador alfanumérico LCD un montaje casero de forma sencilla y económica.

En muchas ocasiones, los diseñadores han evitado ciertos proyectos porque necesitaban un visualizador que pudiese manejar números, letras e incluso símbolos. Incluso también se han desestimado otros diseños que incluían visualizadores con 10, 20 y hasta incluso 40 caracteres de largo.

Estos problemas se pueden resolver mediante la utilización de un sencillo y económico módulo alfanumérico LCD, el cual contiene el circuito integrado controlador que realiza la mayoría del trabajo por el diseñador.

Mostraremos como utilizar los LCD's con los diseños basados en microprocesador. La mayoría de los módulos pequeños utilizan el controlador Hitachi HD44780. Por lo tanto, aquí explicaremos el funcionamiento de los módulos LCD que utilizan este controlador o que son compatibles con el formato del HD44780. Entre los módulos más comunes se encuentran los fabricados por Optrex, Epson, Hitachi, Amperex y Densitron.

Un visualizador numérico de 10 dígitos necesita aproximadamente 100 hilos y unos 20 componentes. Un visualizador alfanumérico de 10 caracteres necesita incluso más cables. El visualizador equivalente (incluidos los alfanuméricos) realizado mediante un módulo LCD necesita únicamente 14 hilos. Utilizando un módulo LCD, cualquier diseñador puede incluir un visualizador que contenga hasta 80 caracteres con sólo 14 hilos, diez de los cuales conectan el visualizador al sistema microprocesador, más uno de alimentación, otro de masa, un pin de habilitación y un controlador para el ajuste de contraste en el LCD. Esto es todo.

Los módulos de visualización manejan de forma automática las funciones de refresco y de multiplexación. El sistema que incluye al visualizador únicamente escribe los datos que se deben visualizar junto con algunos códigos de control (como activación y desactivación de la pantalla de visualización, desplazamiento a la derecha o a la izquierda, etc.), en la memoria del módulo; a partir de este momento el circuito integrado controlador del módulo LCD hace el resto.

Los módulos LCD no se han utilizado de forma extensiva en el pasado, debido a sus elevados precios. Sin embargo, el costo de este tipo de dispositivos de visualización ha bajado de forma considerable.

La mayoría de los módulos pequeños y económicos LCD disponen del circuito integrado controlador de Hitachi HD44780. Esto significa que todos ellos tienen el mismo formato estándar, con el mismo interfaz de 14 patillas y, por lo tanto, son compatibles e intercambiables unos por otros. Los tamaños más normales incluyen visualizadores con formatos de 16x1, 16x2, 20x4, 24x2 y 40x2. Esto significa que se puede modificar

el tamaño del visualizador con únicamente insertar un módulo mayor. No se necesitan modificaciones adicionales del hardware; únicamente precisarán algún cambio los controladores de software de la aplicación concreta.

Los módulos LCD reconocen el formato ASCII estándar para las letras (mayúsculas y minúsculas) y números, además de una variedad de símbolos incluyendo ?, !, \$, % y ', entre otros. En todos los modelos se soportan 192 caracteres alfanuméricos y 32 símbolos especiales. Los módulos también permiten la utilización de ocho caracteres definidos por el usuario.

Los módulos LCD son del tipo de una matriz de puntos por cada carácter formados por cinco puntos de ancho y siete de alto (formatos de 5x7) o por bloques de cinco puntos de ancho por 10 de alto (formatos de 5x10). Estos formatos son seleccionados mediante un mandato de control.

También existe una línea de cursor, debajo de cada carácter. El formato de 5x10 es mucho más adecuado para algunos caracteres en minúsculas, como es el caso de la "g", la "y" y la "p" (es decir, letras y símbolos con partes que descienden por debajo del nivel de escritura del resto de los caracteres). Es de destacar que la matriz de 5x10 encuentra los límites del visualizador independientemente de que dicho módulo tenga una o dos líneas.

#### **4.5.1 CARACTERISTICAS DE LOS MODULOS LCD**

Los módulos LCD soportan una serie de posibilidades de visualización que se pueden acomodar a cada tipo de aplicación. Lo que sigue es una pequeña descripción de este tipo de posibilidades:

\*Activación/desactivación de la pantalla: permite que el usuario encienda o apague el visualizador a través del procesador.

\*Activación/desactivación del cursor: permite seleccionar el cursor de pantalla o eliminarlo.

\*Parpadeo del cursor: el usuario puede elegir entre un cursor estático o parpadeante. El carácter libre del cursor también parpadeará.

\*Desplazamiento a la derecha/izquierda: desplaza los datos de la pantalla del visualizador.

\*Vuelta al primer carácter de la pantalla: hace que el cursor vuelva a la primera posición de la pantalla de visualización (posición HOME) si se ha desplazado previamente.

#### 4.5.2 INTERFAZ DE SOFTWARE

El interfaz de software entre el módulo LCD y un microprocesador se hacen de la siguiente forma:

Operaciones de control(ej.:la activación o desactivación de la pantalla) y las operaciones con datos. Las operaciones de control actúan sobre las posibilidades de la pantalla de visualización, mientras que las operaciones de datos escriben los datos actualizados que deben ser visualizados en la pantalla del módulo LCD.

El circuito integrado que incluye los módulos LCD HD44780 contiene 80 bytes de RAM y es capaz de soportar hasta una pantalla de 40x2 caracteres(cada byte de la memoria RAM corresponde a un carácter de la pantalla). Los módulos de visualización menores simplemente no utilizan completamente los 80 bytes de RAM. La RAM de visualización se encuentra organizada de la siguiente manera:

LINEA1:

Posiciones de carácter 1,2,3,4,5,6,7,8,9....40.

Direcciones de RAM 0,1,2,3,4,5,6,7,8.....27(HEX)

Posiciones de carácter 1,2,3,4,5,6,7,8,9....40.

Direcciones de RAM 28,29,2A,2B,2C,2D,2E,2F,30,31,32,33,34,....4F(HEX)

Los módulos menores simplemente no visualizan el carácter superior asociado con las direcciones superiores. Por ejemplo, un visualizador de 16x2 utiliza las direcciones 00-0F(HEX)para la línea 1 y las direcciones 28-36 para la línea 2.

El HD44780 también dispone de 64 bytes de RAM del generador de caracteres. Esta memoria se utiliza para almacenar la configuración de los caracteres de los ocho definidos por el usuario(8 bytes por cada carácter).

Una vez que se encuentra incluido el carácter en la memoria RAM del generador, se puede acceder a él como si fuera un carácter ordinario del juego disponible.

NOTA: En el modo de la matriz de 5x10 solamente se soportan cuatro caracteres definidos por el usuario, ya que cada uno de ellos necesita 11 bytes de la memoria de generación.

### 4.5.3 CONTROLADORES DE SOFTWARE

El sistema que utiliza la pantalla LCD necesita dos controladores de software para poder trabajar con los módulos LCD, el de escritura de mandatos de control y el de escritura de datos. Las funciones mínimas que deben poder realizar cada uno de estos controladores son las siguientes:

Escritura de mandatos de control:

- \* Poner en las líneas DB0-DB7 el código de control deseado.
- \* Colocar la línea R/W a cero lógico.
- \* Colocar la línea RS a cero lógico.
- \* Activar la línea de habilitación.

Escritura de datos:

- \* Poner en las líneas DB0 a DB7 el carácter deseado.
- \* Colocar la línea R/W a cero lógico.
- \* Colocar la línea RS a uno lógico.
- \* Activar la línea de habilitación.

El usuario también puede leer los datos y los mandatos de control desde el HD44780. Los controladores para la lectura de los mandatos de control y los datos son similares a los de escritura, excepto en el hecho de que la línea R/W se debe poner a uno lógico.

DIRECCIONES DE LA PANTALLA LCD EN EL ADATOC.

- La dirección 74H se usa para escritura de mandatos de control.
- La dirección 76H se usa para escritura de datos.

### **4.5.4 USOS DE LAS PALABRAS DE CONTROL**

- La palabra de control 1H se usa para borrar la pantalla.
- La palabra de control 2H se usa para volver el cursor a la posición inicial en la pantalla.

-La palabra de control 4H se usa para decrementar el puntero de direcciones después de cada operación de lectura/escritura.

-La palabra de control 6H se usa para incrementar el puntero de direcciones después de cada operación de lectura/escritura.

-La palabra de control 0CH activa el visualizador(encendido), desactiva el cursor y selecciona el cursor estático (no parpadeo).

-La palabra de control 0FH activa el cursor y controla el parpadeo del cursor.

-La palabra de control 10H desplaza únicamente el cursor, las operaciones de desplazamiento se realizan a la izquierda.

-La palabra de control 14H desplaza únicamente el cursor, las operaciones de desplazamiento se realizan a la derecha.

-La palabra de control 24H selecciona interfase de datos de 4 bits, visualización de una línea, fuente de caracteres de 5x10 puntos.

-La palabra de control 38H selecciona interfase de datos de 8 bits, visualización de 2 líneas, fuente de caracteres de 5x7 puntos.

-La palabra de control 40H selecciona la dirección para la posterior escritura en la RAM del generador de caracteres.

-La palabra de control 80H selecciona la dirección para la posterior escritura en la RAM de visualización.

#### **4.5.5 INICIALIZACION DEL VISUALIZADOR**

La primera operación a realizar por el software es la inicialización del visualizador. La inicialización incluye el borrado de la pantalla y la utilización de los mandatos de control adecuados que preparen al visualizador con el estado que se desee para su posterior utilización.

NOTA: El módulo de visualización requiere 10 milisegundos para

inicializarse después de que se aplica la alimentación. El sistema que lo contiene, por lo tanto, deberá esperar 10 milisegundos antes de escribir nada en la pantalla después de alimentar el módulo.

#### **4.5.6 VISUALIZACION DE LOS DATOS ACTUALES**

Una vez que se ha inicializado adecuadamente el visualizador, la escritura de los datos es tan simple como escribir el código ASCII adecuado con una serie de operaciones de escritura de datos. Hay que recordar que el mandato SET DD RAM ADDRESS debe ir delante de las operaciones de datos para asegurar que estos se dirijan a la RAM de la pantalla y no a la del generador de caracteres. Del mismo modo, los datos que se deben escribir en la RAM del generador de caracteres deben incluir un mandato SET CG RAM ADDRESS al comienzo.

#### **4.5.7 INTERFASE DE HARDWARE**

Hay una variedad de formas de realizar la interfase de un módulo LCD con un procesador o microprocesador que lo soporte.

Los módulos LCD con un circuito integrado controlador HD44780 incorporado tiene dos modos de interfase por hardware: uno de 4 bits y otro de 8 bits.

En el caso del ADATOC se ha realizado una conexión de 8 bits directamente al bus de datos del microprocesador, como puede ser observado en el plano # 1.

#### **4.6 EL SENSOR DE TEMPERATURA**

Se selecciono el sensor LM 335 por su bajo costo y fácil uso. No es necesario emplear elementos(circuitos integrados) de hardware adicionales para el proceso de lectura del sensor, economizando de esta manera espacio y dinero.

Este sensor da en su salida un voltaje de 2.5v a -23 °C y un voltaje de 3.73v a 100 °C.

Su salida es de 10mv/°K y su rango de funcionamiento va desde -40°C hasta 100°C.



## CAPITULO V: EL SOFTWARE DEL ADATOC

### 5.1 ESPECIFICACIONES DEL MICROPROFESSOR I PLUS (MPF-IP)

Sistema Monitor: Monitor residente de MULTITECH.

Editor de Texto: Provee un editor de línea que permite la edición de programas en Lenguaje Ensamblador.

Ensamblador de una pasada.

Ensamblador de dos pasadas.

Desensamblador.

### 5.2 ESPECIFICACIONES DEL SOFTWARE DEL SISTEMA REALIZADO.

Cuenta con todas las facilidades antes mencionadas y además con lo siguiente:

\*Editor de texto para IBM PC O COMPATIBLES, se puede utilizar cualquier editor de texto que genere código ASCII. En este caso utilizamos el editor TEDIT(TURBO EDITOR) porque tiene muchas ventajas con relación a otros editores, una ventaja es que viene incorporado en el editor la capacidad de comunicación serial(Transmisión y Recepción). Esta capacidad de comunicación fue utilizada muy frecuentemente para la transmisión del programa en código ejecutable hacia el sistema de desarrollo (MPFIP + TARJETA DEL ADATOC).

\*Ensamblador cruzado para IBM PC O COMPATIBLES, programa que genera un archivo ejecutable que se carga al sistema de desarrollo para su posterior ejecución. En este caso se utilizó como programa ensamblador el Ensamblador cruzado de la Compañía AVOCET.

\*Como Programa Receptor se utilizo un programa diseñado por Martín Cruz y Miguel Risco. Este programa se encarga de recepcionar el programa enviado en código de máquina por la Computadora Compatible hacia la tarjeta del ADATOC.

### 5.3 DESARROLLO DEL PROGRAMA

Se escribió el programa en Lenguaje Ensamblador utilizando un Editor de texto de propósito general(TEDIT) para Microcomputadora personal IBMPC o compatible que genere código ASCII, creándose así archivos de texto con

extensión del tipo.ASM.

Este programa se ensambla mediante un Ensamblador cruzado para IBM PC, si el programa .ASM tuviera errores sería detectado por el programa ensamblador.

El programa ensamblador crea un archivo .HEX el cual tiene los códigos de máquina del programa ensamblado en un formato normalizado, este archivo es el que realmente se carga en la memoria RAM del ADATOC vía comunicación serial.

Luego de ser cargado el programa .HEX hacia la tarjeta mediante un Programa Receptor almacenado en una de las EPROMS DE LA TARJETA, este puede ejecutarse con los comandos que ofrece el Sistema Monitor del MULTITECH MPF-IP(MICROPROFESSOR I PLUS).

El programa puede ser comprobado mediante las facilidades de visualización de memoria, de registros, puntos de parada, etc del Monitor.

Este método ofrece grandes ventajas ya que permite trabajar en un Editor de Texto muy superior al original ofrecido por el kit MPFIP, como otras ventajas tenemos almacenado los archivos imprimibles y ejecutables en disco. También es posible utilizar el simulador de AVOCET llamado AVSIMZ80 para simular algunas subrutinas del programa.

Permite una profusa documentación de programas, trabajando a nivel modular lo cual minimiza el tiempo de desarrollo de programas, además permite el uso de librerías ya existentes.

#### **5.4 PROGRAMA DE RECEPCION DE DATOS DEL SISTEMA DE DESARROLLO(MPFIP + ADATOC)**

**\$AllPublic**

**\$Title(RECEPCION DE DATOS(ASCII),CALCULO DEL #BYTES Y CONVERSION A COD.HEX)**

**;Hecho por C. Martín Cruz Salazar**

**;Abril 1994**

**;Grabado en memoria EPROM desde la dirección 2000H.**

**;A partir de la dirección FAE6H almaceno el número de bytes.**

**;Para cambiar el offset se debe colocar en el programa que se transmite la línea:**

**;**:02FBF000HLL00****

**;Donde HLL es el valor del offset.**

**;0 se puede poner la siguiente cabecera en el código fuente:**

```

DEFSEG      OFFSETSEG, START=$FBF0, CLASS=DATA, ABSOLUTE
SEG         OFFSETSEG
OFFSET     DW      0H

```

-----

```

DEFSEG LOAD2GR2, START=2000H, ABSOLUTE
SEG     LOAD2GR2
URTDAT EQU 60H
URTCNT EQU 61H
CTC3 EQU 67H
LD     A, 0
LD     (0FAE5H), A
LD     (0FAE6H), A
LD     (0FAE7H), A
LD     (OFFSET), A
LD     (OFFSET+1), A
CALL  INIT
CALL  TERM
LD     IY, (0FAE3H)
JP     (IY)
RST   30H
TERM LD     A, 3
OUT   (CTC3), A
LD     A, 37H
OUT   (URTCNT), A
CALL  803H
CALL  9B9H
LD     HL, MEN1
CALL  9CAH
LD     HL, (0FAE6H)
CALL  0A89H ;Mando a visualizar el # de bytes.
LD     IX, 0FF2CH
CALL  246H
RET
MEN1 DB     'RECEP.FIN, #B:
      DB     0DH

```

```

;La siguiente subrutina inicializa interrupciones.
INIT: PUSH AF

```

```

LD    A,0    ;Inicializa las direcciones POINTER Y TRADFLG con cero
LD    (POINTER),A;
LD    (TRADFLG),A
LD    A,0FAH    ;Parte más significativa del vector de
;interrupción.
LD    I,A
LD    A,0D7H    ;Palabra de control que programa al CTC en el modo
OUT   (64H),A
OUT   (CTC3),A    ;contador(canal 3).
LD    A,1    ;Constante de tiempo que se carga al CTC.
OUT   (CTC3),A
LD    A,6    ;Parte menos significativa del vector de
;interrupcion.
OUT   (CTC3),A
LD    HL,TRANSF
LD    (0FA06H),HL
%REPT 3
LD    A,0
OUT   (URTCNT),A
NOP
%ENDREPT
LD    A,40H
OUT   (URTCNT),A
NOP
NOP
LD    A,7EH    ;Palabra modo del 8251. Paridad Par.
OUT   (URTCNT),A
LD    A,37H    ;Palabra comando del 8251. Resetea los flags de
;erro-
OUT   (URTCNT),A ;res(PE,OE y FE) del registro de estado.
IM    2
EI
LOOP2 LD    A,(TRADFLG)
CP    0FFH    ;Si es FFH llamo a la subrutina TRADUC.
CALL  Z,TRADUC
LD    A,(TRADFLG)
CP    0F0H

```

```

    JR    NZ, LOOP2
    pop   AF
    RET

TRANSF PUSH AF      ;Inicio de la subrutina de interrupción.
    PUSH BC
    PUSH HL
    IN   A,(URTCNT) ;Leo el registro de estado del 8251.
    LD   (OFAE0H),A
    AND  %00111000;
    JP   NZ, ERR0RE ;Si es diferente de cero salta a error.
    IN   A,(URTDAT) ;Leo el valor que ha sido transmitido.
    CP   0AH        ;Veo si he llegado al fin de línea.
    CALL Z, FLAG    ;Si estoy al final llamo a la subrutina FLAG.
    LD   B,A        ;Si no estoy al final, cargo un BUFFER con los
;datos.
    LD   A,(POINTER)
    CP   80
    JP   NC, ERROR  ;Si es mayor o igual a 80 voy a ERROR.
    LD   C,A
    LD   A,B
    LD   B,0
    LD   HL,BUFFER;Dirección inicial del buffer de datos.
    ADD  HL,BC
    LD   (HL),A     ;Cargo los datos recepcionados en el BUFFER.
    INC  C          ;Incremento el puntero.
    LD   A,C
    LD   (POINTER),A ;El puntero se salva en la dirección POINTER.
    POP  HL
    POP  BC
    POP  AF
    EI
    RETI

FLAG   PUSH AF
    LD   A,OFFH     ;Cargo A con FFH.
    LD   (TRADFLG),A
    POP  AF

```

RET

;La subrutina siguiente traduce de código ASCII a código hexadecimal.

```
TRADUC  PUSH    AF
        PUSH    BC
        PUSH    DE
        PUSH    HL
        PUSH    IX
        PUSH    IY
        LD      A,0 ;Inicializo las direcciones.
        LD      (POINTER),A
        LD      (TRADFLG),A
        LD      IX,BUFFER
        LD      A,(IX)
        CP      ':' ;El primer dato debe ser dos puntos.
        JP      NZ, ERROR
        INC     IX
        LD      H,(IX) ;Cargo en H el código ASCII.
        LD      L,(IX+1) ;Cargo en L el código ASCII.
        CALL   ASCBIN ;Traduce de código ASCII a código hexadecimal.
        CP      0 ;Si es cero va a FIN.
        JR      Z, FIN
        LD      E,A ;Cargo en E el # de bytes.
        INC     IX
        INC     IX
        LD      H,(IX) ;En HL cargo el byte más significativo de la di-
        LD      L,(IX+1);rección inicial del programa que se envia.
        CALL   ascbin
        LD      B,A ;Cargo en B el valor hexadecimal más significativo.
        LD      H,(IX+2) ;Parte menos significativa de la dirección inicial
        LD      L,(IX+3) ;del programa que se envia.
        %REPT 4
        INC     IX
        %ENDREPT
        CALL   ASCBIN
        LD      C,A ;Cargo en C el valor hexadecimal menos significativo.
        LD      A,(OFFSET)
        LD      H,A
```

```

LD    A,(OFFSET+1)
LD    L,A
ADD   HL,BC
PUSH  HL
POP   IY ;En IY tengo el valor completo de la dir.inicial.
LD    A,(0FAE5H)
CP    OFFH
JR    Z,NOCAR
LD    (0FAE3H),IY
LD    A,OFFH
LD    (0FAE5H),A
NOCAR LD    B,E ;Cargo en B el # de bytes de datos de la línea.
LD    D,0
LD    HL,(0FAE6H)
ADD   HL,DE ;Cálculo del # de bytes.
LD    (0FAE6H),HL
LOOP  INC    IX
INC   IX
LD    H,(IX) ;Empiezo a cargar los códigos ASCII de los datos.
LD    L,(IX+1)
CALL  ASCBIN
LD    (IY),A ;Estos bytes de datos en hexadecimal los voy posi-
INC   IY ;cionando desde la dirección inicial del programa.
DJNZ  LOOP ;Se posicionan un # contenido en B.
SALIR POP   IY
POP   IX
POP   HL
POP   DE
POP   BC
POP   AF
RET
FIN   LD    A,0F0H
LD    (TRADFLG),A
JP    SALIR
ERRORE LD  A,3
OUT   (CTC3),A
LD    A,37H

```

```

    OUT    (URTCNT),A
    CALL  803H
    CALL  9B9H
    LD    HL,MEN2
    CALL  9CAH
    LD    A,(0FAE0H)
    CALL  0A9AH
    LD    IX,0FF2CH
    CALL  246H
    DI
    RST   30H
MEN2  DB   'RECEP.ERRONEA:'
      DB   0DH
ERROR LD   A,3
      OUT  (CTC3),A
      LD   A,37H
      OUT  (URTCNT),A
      CALL 803H
      CALL 9B9H
      LD   HL,MEN
      CALL 9CAH
      LD   IX,0FF2CH
      CALL 246H
      DI
      RST  30H
MEN   DB   'RECEP. ERRONEA'
      DB   0DH

```

;La siguiente subrutina traduce de código ASCII a código hexadecimal.

```

ASCBIN: PUSH    BC
          LD     A,L
          CALL  A2HEX
          LD     B,A
          LD     A,H
          CALL  A2HEX
          RRCA
          RRCA
          RRCA

```



```

        RRCA
        OR      B
        POP    BC
        RET
A2HEX:  SUB    '0'
        CP     10
        JR C,  A2HEX1
        SUB    7
A2HEX1: RET
        ORG   OFBOOH
BUFFER  DS    80
POINTER DS    1
TRADFLG DS   1
        END

```

## 5.5 ALGORITMO DEL PROGRAMA MONITOR

Para un mejor entendimiento se ha dividido el programa en cuatro partes:

La primera parte: Inicialización de los recursos del sistema.

La segunda parte: Subrutinas de uso más frecuente dentro del programa principal. Como por ejm. Subrutina de conversión de Binario a código ASCII, Subrutina de conversión de código ASCII a Binario, Subrutina de conversión de Binario a ASCII decimal, Subrutina que vierte el contenido de la pantalla virtual en la pantalla real.

La tercera parte: Programa de Recepción de datos del ADATOC.

La cuarta parte: Programa principal del ADATOC.

Lo que sigue es la secuencia del programa(I parte y IV parte).

-Inicializo registros importantes del Z80, como el puntero de pila, el registro I.

-Chequeo de si el arranque es en FRIO o en CALIENTE.

-En caso de que el arranque es en FRIO se realiza primero un chequeo de si la memoria RAM se encuentra en buen estado. Si estuviera en buen estado se

llenen las posiciones de memoria con el valor FFH. Después dos pitidos. En caso que la memoria se encuentre en mal estado se manda un mensaje de error.

-Si el arranque es en CALIENTE no se chequea la memoria, tampoco se llenan con valores FFH. Dos pitidos.

-Cargo los valores AAH y 5AH en ciertas posiciones de memoria que van a servir como patrones del chequeo de arranque en caliente o en frio.

-Inicializo una dirección de memoria(puntero) con la dirección inicial de los datos que se van a almacenar.

-Inicialización con cero de varias posiciones de memoria importantes dentro del programa. Por ej. la dirección que contiene el # de canales.

-Inicializo la dirección que contiene la frecuencia de acumulación de datos con 1(significa 1 segundo).

-Deshabilito el hecho que al presionar una tecla no aparesca su código en la pantalla.

-Inicialización de los vectores de interrupción.

-Inicialización del Display LCD.

-Inicialización del canal 1 del CTC para usarse con el teclado. Palabra de control(11000111).

-Inicialización del canal 0 del CTC para usarse con el reloj de tiempo real. Palabra de control(11000111).

-Inicialización de la dirección que contendrá el código de la tecla presionada con el valor FFH.

-Inicialización del USART 8251.

-Inicialización del Reloj de Tiempo real.

-Se pide actualización de la hora y la fecha si fuera arranque en frío, visualizándose HORA = en el display. Se visualiza esta palabra sin usar interrupciones. La hora actual y fecha actual se fija usando el teclado.

-Presentación del nombre del equipo en el display. Este nombre se visualiza debido a la interrupción del RTC cuya rutina manda a visualizar el contenido de la pantalla virtual en el display.

-Llamo al programa principal.

-Se continua visualizándose el nombre del equipo hasta que se presiona cualquier tecla.

A partir de aquí cada tecla del teclado tiene una función definida, como por ejm.:

-Cada vez que se presiona una tecla se genera una interrupción pasando el CPU a ejecutar una rutina de interrupción donde carga en una cierta posición de memoria el código de la tecla presionada.

-Si este código es cero sale en display la temperatura del canal 0, si es 1 del canal 1, etc.

-La tecla ESC sirve para visualizar la hora y la fecha actual.

-La lectura del convertidor A/D se realiza en la rutina de interrupción que c/0.5 segundo causa el RTC en el CPU.

-El valor de la Temperatura conseguida de la tabla, para poder visualizarlo hay que convertirlo primero de binario a ASCII.

## **5.6 LISTADO DEL PROGRAMA MONITOR**

El siguiente programa fuente es el listado del programa total que se grabó en una memoria EPROM 2764 que forma parte del hardware del ADATOC. Este programa se ha dividido en 4 partes para un mejor entendimiento.

**\$title(PARTE I DEL PROGRAMA ADATOC)**

**\$subtitle(INICIALIZACION, TABLA DE VECTORES, VARIOS)**

**;SET./94**

**\$ALLPUBLIC**

EXTERN ASCBIN,BINASC,DISPLAY,RETAR,LOAD,ACTUA

EXTERN MAIN,INT0,INT1,INT2,INT3,DISDSP,RESETKE

EXTERN BEEP

**;DIRECCION INICIAL USADA EN**

**;EL ALMACENAMIENTO DE**

**;DATOS DEL PROGRAMA PRINCIPAL**

**BASEDAT EQU \$4000**

**;Definiendo direcciones del USART(8251)**

**;(Interface serial)**

DEFSEG SIOSEG,START=60H,CLASS=IOSPACE

SEG SIOSEG

URTDAT DS 1

URTCNT DS 1

**;Definiendo direcciones del CTC Z80**

**;(circuito contador y temporizador).**

DEFSEG CTCSEG,START=\$64,CLASS=IOSPACE

SEG CTCSEG

CTC0: DS 1 ; CTC 64 - 67H En I/O Space

CTC1: DS 1

CTC2: DS 1

CTC3: DS 1

**;Definiendo las direcciones para la PANTALLA LCD.**

DEFSEG LCDSEG,START=\$74,CLASS=IOSPACE

SEG LCDSEG

LCDC: DS 1 ;LCD 74 -76H En I/O Space

LCDI: DS 1

```
LCDD: DS 1
```

```
;Definiendo la dirección para el
```

```
;CONVERTIDOR ANALOGICO/DIGITAL.
```

```
DEFSEG CADSEG,START=$68,CLASS=IOSPACE
```

```
SEG CADSEG
```

```
CAD: DS 1
```

```
;Reloj de Tiempo Real
```

```
DEFSEG RTCSEG,START=$00,CLASS=IOSPACE
```

```
SEG RTCSEG
```

```
RTCTEST DS 1 ;0
```

```
RTCDECS DS 1 ;1
```

```
RTCUSEG DS 1 ;2
```

```
RTCDSEG DS 1 ;3
```

```
RTCUMIN DS 1 ;4
```

```
RTCDMIN DS 1 ;5
```

```
RTCUHOR DS 1 ;6
```

```
RTCDHOR DS 1 ;7
```

```
RTCUDIA DS 1 ;8
```

```
RTCDDIA DS 1 ;9
```

```
RTCDSEM DS 1 ;10
```

```
RTCUMES DS 1 ;11
```

```
RTCDMES DS 1 ;12
```

```
RTCYEAR DS 1 ;13
```

```
RTCINIP DS 1 ;14
```

```
RTCINTE DS 1 ;15
```

```
;Definiendo la dirección para el TECLADO
```

```
DEFSEG KEYSEG,START=$70,CLASS=IOSPACE
```

```
SEG KEYSEG
```

```
KEY DS 1
```

```
;DEFINICION DEL STACK
```

```
defseg STSEG,start=$F900,absolute,CLASS=DATA
```

```
seg stseg
```

```
DS $6FF
```

STACK EQU \$

;TABLA DE VECTORES DE INTERRUPCION

```
DEFSEG      VTABSEG,START=$E000,CLASS=DATA,ABSOLUTE
SEG  VTABSEG
VECTTAB    DS    8
FINVEC     EQU   $
```

;SEGMENTO DE DATOS VARIOS

```
DEFSEG      DATASEG,START=FINVEC,CLASS=DATA,ABSOLUTE
SEG  DATASEG
PATRON      DS    2    ;PATRON DE RESET EN CALIENTE
KEYFLAG     DS    1    ;ESTADO DEL TECLADO(contiene el
                    ;código de la última tecla)
BUFFER      DS   80    ;BUFFER DE RECEPCION(ENVIO DE DATOS
                    ;de la computadora al ADATOC)
POINTER     DS    1    ;PUNTERO AL BUFFER
TRADFLG     DS    1    ;FLAG DE TRADUCCION
ERRCOD      DS    1    ;ERROR EN LA RECEPCION
EXEADD      DS    2
ADDFLG      DS    1
DECIDT      DS    4    ;BUFFER DE SUBROUTINA BN2DEC
POINTDT     DS    2    ;DIRECCION QUE CONTIENE LA DIRECCION
                    ;SIGUIENTE AL ULTIMO DATO ACUMULADO.
MEDIOS      DS    1    ;Cambia entre 0 y 1. Controla el incre-
                    ;mento dentro de SECOND.
SECOND      DS    1    ;Esta zona va variando de acuerdo al
                    ;tiempo seleccionado en la toma de datos
HORADT      DS   11    ;A PARTIR DE AQUI SE GUARDA LA HORA,
                    ;MINUTOS Y SEGUNDOS EN CODIGO ASCII.
FECHDT      DS    9    ;A PARTIR DE AQUI SE GUARDA EL DIA,
                    ;MES Y EL AÑO EN CODIGO ASCII.
YEARDT      DS    2    ;GUARDA EL AÑO ACTUAL COMO BCD.
TEMPDT      DS    7    ;A PARTIR DE AQUI SE GUARDA EL DATO
                    ;DE TEMPERATURA EN CODIGO ASCII DEL ULTIMO
                    ;CANAL LEIDO.
DIGCONT     DS    1    ;Zona que almacena el # de digitos
```

```

;que se pueden visualizar en la pantalla
KEYBUF      DS      10      ;Zona donde se carga los códigos de las
;teclas que se van presionando.

KEYCOD      DS      1

CANAL DS      1      ;Zona donde se guarda el # de canal
ADCOD DS      1      ;Zona para guardar el código
;digitalizado por el A/D.
UNSEG DS      1      ;Contiene un número en segundos
;que ha sido seleccionado. Por defecto es 1

EVITREB     DS      1
INTCNT      DS      1
DSPENA      DS      1      ;Zona donde se coloca un 0 o
;un 1 para habilitar o
;deshabilitar el display.

ACUMFLG     DS      1      ;Zona donde se guarda un 1 o 0
;para activar o desactivar la acumulación

INIDHOR     DS      1      ;ZONA PARA GUARDAR LA HORA INICIAL
INIUHOR     DS      1      ;DEL PROCESO DE ACUMULACION
INIDMIN     DS      1      ;ZONA PARA GUARDAR MINUTOS
INIUMIN     DS      1
INIDSEG     DS      1      ;ZONA PARA GUARDAR SEGUNDOS
INIUSEG     DS      1
INIDMES     DS      1      ;ZONA PARA GUARDAR EL MES DE
INIUMES     DS      1      ;LA FECHA INICIAL
INIDDIA     DS      1      ;ZONA PARA GUARDAR EL DIA
INIUDIA     DS      1
HORAFIN     DS      4      ;A PARTIR DE AQUI SE GUARDA LA HORA FINAL.
FECHFIN     DS      4      ;A PARTIR DE AQUI SE GUARDA LA FECHA FINAL.
BUFPTR      DS      2      ;Es utilizado por la subrutina BN2DEC
CURLLEN     DS      1      ;Es también utilizado por la subrutina
;BN2DEC.

NGFLAG      DS      1
ASCBUF      DS      7
NUMCAN      DS      1      ;ZONA PARA GUARDAR EL NUMERO DE CANALES
SGUNDOS     DS      6      ;Buffer de la rutina BN2DEC. Contiene
;el # de bytes y los códigos ASCII del
;número.

```

```

CALLBUF    DS    6    ;Llamada a Ejecución de rutina auxiliar
MEMORY     DS    4
POINTME    DS    2
DSPPTR     DS    2    ;Puntero de memoria de pantalla
LIN1BUF    DS    18   ;Espacio de memoria destinado a la primera
                ;fila de pantalla.
LIN2BUF    DS    22   ;Espacio de memoria destinado a la segunda
                ;fila de pantalla.
OFFSET     DS    2

```

```

;PROGRAMA DE RESET

```

```

        DEFSEG      RESETSEG,START=$0000,CLASS=CODE
        SEG         RESETSEG
RESET LD    SP,STACK    ;Inicializo el registro SP
        DI
        IM    2        ;Selecciono modo dos
                ;de interrupción
        LD    A,$E0 ;Byte más significativo del
        LD    I,A      ;vector de interrupción
        EI
        LD    IX,PATRON ;Chequeo de si se arranca en frio
                ;o caliente
        LD    A,(IX)   ;Se carga A con el valor AAH
        XOR   $F0      ;En A queda 5AH
        XOR   (IX+1)   ;En A queda 0
        JR    Z,HOT    ;Z=1, arranque en caliente
COLD JP    INITMEM     ;Si es arranque en frio

```

```

;Inicializo algunas direcciones

```

```

;cuando se arranca en frio

```

```

MEMRET LD    IX,BASEDAT
        LD    (POINTDT),IX
        LD    A,1
        LD    (UNSEG),A
        LD    A,0
        LD    (NUMCAN),A ;Inicializa el número de canales
        CALL BEEP      ;Suena un pitido

```



```

        CALL BEEP          ;otro pitido
        CALL CONFIG
        CALL INIRTC
        CALL COLDPR
        JP   FINRES
HOT    LD   B,$FF ;Aquí viene cuando hay
                ;arranque en caliente
LOOPRET CALL RETAR ;Lazo de retardo
        DJNZ LOOPRET
        CALL BEEP          ;Activa un pitido
        CALL BEEP          ;otro pitido
        CALL CONFIG
        CALL SETRTC
        CALL COLDPR
FINRES CALL MAIN
        JP   RESET
•     FIN DE RESET

```

```

;CONFIGURACION INICIAL DEL ADATOC
;SE EJECUTA SIN IMPORTAR SI ES
;ARRANQUE EN FRIO O
;CALIENTE

```

```

CONFIG:  PUSH  AF
        LD   A,$AA
        LD   (PATRON),A
        LD   A,$5A
        LD   (PATRON+1),A
        LD   A,$0D
        LD   (TEMPDT),A
        LD   A,0
        LD   (CANAL),A
        LD   (ACUMFLG),A
        LD   (POINTME),A
        LD   (MEDIOS),A
        LD   (SECOND),A
        CALL DISDSP
        CALL SETVECT

```

```
CALL  INITIO
POP   AF
RET
```

```
INITIO    PUSH  AF
          CALL  INILCD
          CALL  INICTC1
          CALL  INICTCO
          LD   A,0           ;LSB del vector de interrupcion.
          OUT  (CTCO),A     ;PARA EL CTC
          CALL  INIKEY
          CALL  INISIO
          POP  AF
          RET
```

```
;Subrutina que carga cada dirección
;inicial de la rutina de interrupción
;a cada uno de los vectores correspondiente
```

```
SETVECT  PUSH  HL
          LD   HL,INT0
          LD   ($E000),HL
          LD   HL,INT1
          LD   ($E002),HL
          LD   HL,INT2
          LD   ($E004),HL
          LD   HL,INT3
          LD   ($E006),HL
          POP  HL
          RET
```

```
;INICIALIZACION DEL USART 8251(INTERFACE
;SERIAL)
```

```
INISIO:  PUSH  AF
          LD   A,0
          CALL  OUTSIO
          CALL  OUTSIO
```

```

CALL  OUTSIO
LD    A,40H
CALL  OUTSIO
NOP
LD    A,7EH ;1STOP,PARIDADPAR,8DATA,16(FACTOR DE VELOC.)
OUT   (URTCNT),A
LD    A,37H
OUT   (URTCNT),A
POP   AF
RET

```

```

OUTSIO  OUT   (URTCNT),A
        NOP
        RET

```

```

;Inicialización del canal 0 del
;CTC(se usa para el RELOJ DE TIEMPO
;REAL)

```

```

INICTC0  PUSH  AF
        LD    A,0C7H
        OUT   (CTC0),A
        LD    A,01H
        OUT   (CTC0),A
        LD    A,0           ;Parte menos significativa del
                           ;vector de interrupción.
        OUT   (CTC0),A
        POP   AF
        RET

```

```

;Inicialización del canal 1 del
;CTC(se usa para el teclado)

```

```

INICTC1  PUSH  AF
        LD    A,0C7H           ;Palabra de control(11000111)
        OUT   (CTC1),A       ;habilita interrupción,modo contador
                           ;flanco negativo,resetea canal.
        LD    A,1           ;Constante de tiempo.
        OUT   (CTC1),A

```

```

LD    A,0           ;Parte menos significativa del vector
OUT   (CTC0),A     ;de interrupción.
POP   AF
RET

```

```

INICTC3    PUSH  AF
           LD    A,0D7H           ;Palabra de control que programa
                                           ;al CTC en el modo
OUT   (CTC3),A     ;contador(canal 3).
           LD    A,1           ;Constante de tiempo que se carga
OUT   (CTC3),A     ;al CTC.
           POP   AF
           RET

```

```

;Inicialización del Reloj de tiempo
;real.

```

```

INIRTC:    PUSH  AF
           LD    A,0
           OUT   (RTCINTE),A
           CALL  INIRTCI
           OUT   (RTCTEST),A
           OUT   (RTCINIP),A
           CALL  ACTUA
           CALL  RESETKE
           INC   A
           OUT   (RTCINIP),A
           LD    A,$9
           OUT   (RTCINTE),A
           CALL  INIRTCI
           POP   AF
           RET

```

```

SETRTC     PUSH  AF
           LD    A,0
           OUT   (RTCTEST),A
           INC   A
           OUT   (RTCINIP),A

```

```

LD    A,$09
OUT   (RTCINTE),A
CALL  INIRTCI
POP   AF
RET

```

```

INIRTCI    PUSH  AF
           IN    A,(RTCINTE)
           IN    A,(RTCINTE)
           IN    A,(RTCINTE)
           POP   AF
           RET

```

```

;Inicializa posiciones
;de memoria con el valor 0.

```

```

INIKEY:    PUSH  AF
           PUSH  BC
           LD    A,0FFH
           LD    (KEYFLAG),A
           LD    A,20H
           LD    (EVITREB),A
           LD    A,0           ;Coloco ceros en las direcciones
           LD    (INTCNT),A   ;INTCNT Y DIGCONT.
           LD    (DIGCONT),A
           POP   BC
           POP   AF
           RET

```

```

;Subrutina de inicialización de
;la pantalla o Display LCD

```

```

INILCD:    CALL  RETAR
           CALL  RETAR
           CALL  RETAR
           PUSH  AF
           LD    A,1           ;Estas líneas se usan para

```

```

OUT   (LCDC),A      ;borrar la pantalla.
CALL  RETAR
CALL  RETAR
CALL  RETAR
LD    A,2           ;Vuelve el cursor a la posi-
OUT   (LCDC),A      ;ción inicial.
CALL  RETAR
CALL  RETAR
CALL  RETAR
LD    A,38H        ;Formato de 2 lineas, caracteres de
OUT   (LCDC),A      ;5x7 puntos.
CALL  RETAR
LD    A,0CH        ;Selecciona cursor estático.
OUT   (LCDC),A
CALL  RETAR
POP   AF
RET

```

;INICIALIZACION DE LA MEMORIA(Hace un TEST de la  
;memoria y coloca FFH en toda ella)

```

INITMEM  LD    HL,$BFFF-$4000    ;Chequeo y llenado de la
                                           ;memoria($4000 hasta $BFFF)

LD    IX,$4000
LASO LD    A,$AA
LD    (IX),A
LD    A,(IX)
CP    $AA
JP    NZ,ERRMEM    ;Si es diferente Mensaje de error
CPL                                     ;Invierte el contenido del acumulador
LD    (IX),A
LD    A,(IX)
CP    $55          ;Si es diferente Mensaje de error
JP    NZ,ERRMEM
LD    A,$FF
LD    (IX),A
LD    BC,0
INC   IX

```

```

DEC    HL
SBC    HL,BC    ;Resta HL---HL-BC
JR     NZ,LASO
LD     HL,$FFFF-$E000    ;Chequeo y llenado de la
                        ;memoria($E000 hasta $FFFF)
LD     IX,$E000
LASO2 LD     A,$AA
LD     (IX),A
LD     A,(IX)
CP     $AA
JP     NZ,ERRMEM
CPL
LD     (IX),A
LD     A,(IX)
CP     $55
JP     NZ,ERRMEM
LD     A,$FF
LD     (IX),A
LD     BC,0
INC    IX
DEC    HL
SBC    HL,BC    ;Resta HL---HL-BC
JR     NZ,LASO2
JP     MEMRET
ERRMEM: LD     IY,ERRMSG    ;Aparece Error de Memoria
LD     A,0
CALL  DISPLAY
JP     ;No continua ejecutando el Z80

;Visualización del nombre del
;equipo:ACUMULADOR DE DATOS C&A
COLDPR:    PUSH  AF
           PUSH  IY
LD     IY,COLDCAR
LD     A,0
CALL  DISPLAY
LD     IY,COLDCA1

```

```
INC    A
CALL  DISPLAY
POP   IY
POP   AF
RET
```

```
COLDCAR    DB    '    ACUMULADOR    ', $0D
COLDCA1    DB    'DE DATOS -*C&A*-', $0D
ERRMSG     DB    'ERROR DE MEMORIA', $0D
FINREST    EQU    $
        END
```

**\$title(PARTE II DEL PROGRAMA ADATOC)**

**\$subtitle(Conjunto de subrutinas que se usan con mucha frecuencia)**

**;SET./94**

**;UNIDAD DE HERRAMIENTAS PARA EL MODULO ACUMULADOR DE DATOS**

```
PUBLIC    ASCBIN,BINASC,DISPLAY,DSPCAD,DSPCAR
PUBLIC    ERASE,ENADSP,DISDSP,RETAR,BN2DEC
PUBLIC    RESETKE,FINTOOL,LINE1,LINE2,SWAPDSP
```

```
EXTERN    BUFPTR,CURLEN,NGFLAG,DSPENA,DIGCONT
EXTERN    INTCNT,KEYFLAG,DSPPTR
EXTERN    LIN1BUF,LIN2BUF,NULMSG
```

```
LCDC EQU    $74    ;UBICACION DE LA PANTALLA LCD
```

```
LCDD EQU    $76
```

```
DEFSEG    TOOLSSEG,START=$026B,CLASS=CODE
```

```
SEG        TOOLSSEG
```

**;Este programa convierte dos códigos  
;ASCII en su correspondiente valor**



```

;binario pe:
;(H)=31H,(L)=37H resultado
;(A)=17-->00010111
;dos códigos ASCII en HL
;Resultado en A

```

```

ASCBIN:PUSH    BC
            LD     A,L
            CALL   A2HEX
            LD     B,A
            LD     A,H
            CALL   A2HEX
            %REPT 4           ;REPETIR LA INST. 4 VECES
            RRCA
            %ENDREPT
            OR     B
            POP    BC
            RET
A2HEX:SUB    '0'
            CP     10
            JR     C,A2HEX1
            SUB    7
A2HEX1:RET

```

```

;Este programa convierte un valor binario
;en su valor ASCII
;pe: 00011101-->'1D', 31H=(H) y 44H=(L)
;El valor en A
;Resultado ASCII en HL

```

```

BINASC:PUSH AF
            PUSH   BC
            LD     B,A   ;Convierte nibble mas significativo.
            AND    OFOH
            RRC     A

```

```

RRC  A
RRC  A
RRC  A
CALL NASCII
LD   H,A   ;Convierte nibble menos significativo.
LD   A,B
AND  OFH
CALL NASCII
LD   L,A
POP  BC
POP  AF
RET

```

```

NASCII CP  10
        JR   C,NAS1
        ADD  A,7
NAS1  ADD  A,'0'
        RET

```

```

;CONVIERTE BINARIO DE 16 BITS A ASCII DECIMAL
;VALOR DE ENTRADA COLOCARLO EN EL REGISTRO DE
;DIRECCION BASE DEL BUFFER DE SALIDA EN HL
;POR EJ.:
;Valor a convertir:3EB7H
;Resultado en(el buffer de salida):
;05(número de bytes en el buffer)
;31(ASCII 1)
;36(ASCII 6)
;30(ASCII 0)
;35(ASCII 5)
;35(ASCII 5)

```

```

BN2DEC:PUSH AF
        PUSH BC
        PUSH DE
        PUSH HL

```

```

LD      (BUFPTR),HL
EX      DE,HL
LD      A,0
LD      (CURLLEN),A
LD      A,H
LD      (NGFLAG),A
OR      A
JP      P,CNVERT
EX      DE,HL
LD      HL,0
OR      A
SBC     HL,DE
CNVERT  LD      E,0
        LD      B,16
        OR      A
DVLOOP  RL      L
        RL      H
        RL      E
        LD      A,E
        SUB     10
        CCF
        JR      NC,DECCNT
        LD      E,A
DECCNT  DJNZ   DVLOOP
        RL      L
        RL      H
CHINS   LD      A,E
        ADD     A,'0'
        CALL    INSERT
        LD      A,H
        OR      L
        JR      NZ,CNVERT
EXIT    LD      A,(NGFLAG)
        OR      A
        JP      P,POS
        LD      A,'-'
        CALL    INSERT

```

```
POS   POP   HL
      POP   DE
      POP   BC
      POP   AF
      RET
```

```
INSERT:   PUSH  BC
          PUSH  DE
          PUSH  HL
          PUSH  AF
          LD    HL,(BUFPTR)
          LD    D,H
          LD    E,L
          INC   DE
          LD    (BUFPTR),DE
          LD    A,(CURLLEN)
          OR    A
          JR    Z,EXITMR
          LD    C,A
          LD    B,0
          LDDR
```

```
EXITMR   LD    A,(CURLLEN)
          INC   A
          LD    (CURLLEN),A
          LD    (HL),A
          EX   DE,HL
          POP   AF
          LD    (HL),A
          POP   HL
          POP   DE
          POP   BC
          RET
```

```
;RUTINA DE VISUALIZACION EN PANTALLA LCD
;NUMERO DE FILA EN A (0:PRIMERA FILA 1:SEGUNDA FILA)
;DIRECCION DE LA CADENA EN IY
;LA CADENA DEBE TERMINAR CON ODH
```

```

DISPLAY    PUSH  AF
           CP    0
           JR    NZ,FILA2
           CALL  LINE1      ;Inicializa el puntero de memoria
                           ;de pantalla(fila 1).
           JR    CONTDSP
FILA2 CALL  LINE2      ;Inicializa el puntero de memoria
                           ;de pantalla(fila 2).
CONTDSP    CALL  DSPCAD;
           POP   AF
           RET

```

```

;Carga caracter por caracter a la
;memoria de pantalla de la fila seleccionada
;hasta encontrar ODH
;dirección de la cadena en IY.

```

```

DSPCAD:    PUSH  AF
           PUSH  IY
LOOPDSP    LD    A,(IY)
           CP    $0D
           JR    Z,FINDSP
           CALL  DSPCAR
           INC   IY
           JR    LOOPDSP
FINDSP     POP   IY
           POP   AF
           RET

```

```

;Cargo un caracter puesto en A
;del contenido de la cadena
;a la memoria de pantalla(filal o
;fila2).

```

```

DSPCAR:    PUSH  IY
           LD    IY,(DSPPTR)
           LD    (IY),A
           INC   IY
           LD    (DSPPTR),IY

```

POP IY

RET

;Cambio a línea 1

;Actualiza el contenido del puntero

;con la dirección inicial de mem. de

;pantalla fila 1.

LINE1: PUSH IY

LD IY,LIN1BUF

LD (DSPPTR),IY

POP IY

RET

;Cambio a línea 2

;Actualiza el contenido del puntero

;con la dirección inicial de mem. de

;pantalla fila 2.

LINE2: PUSH IY

LD IY,LIN2BUF

LD (DSPPTR),IY

POP IY

RET

;Subrutina que borra la pantalla LCD.

ERASE: PUSH AF

PUSH IY

LD A,0

LD IY,NULMSG ;Llena de espacios

CALL DISPLAY ;en blanco ambas filas.

INC A

CALL DISPLAY

POP IY

POP AF

RET

;Retarda la operacion de la CPU B \* 4.5µs

RETAR: PUSH BC

```

        LD    B,30          ;135µs
LIS1   DJNZ  LIS1
        POP   BC
        RET

```

;Pequeño retardo para visualización

```

RETCAR    PUSH  BC
        LD    B,9          ;40.5µs
LIS2   DJNZ  LIS2
        POP   BC
        RET

```

;Rutina que habilita el display

;cada tecla que se presiona aparece

;en el display.

```

ENADSP    PUSH  AF
        LD    A,1
        LD    (DSPENA),A
        POP   AF
        RET

```

;Rutina que deshabilita el display

;cuando presiono alguna tecla

;no aparece en la pantalla.

```

DISDSP:   PUSH  AF
        LD    A,0
        LD    (DSPENA),A
        POP   AF
        RET

```

;Cambia el código de la

;última tecla por 20h y lo

;carga en KEYFLAG. Vuelve

;a cero el valor de DIGCONT y

;INTCNT.

```

RESETKE   PUSH  AF
        LD    A,$20

```

```

LD    (KEYFLAG),A
LD    A,0
LD    (DIGCONT),A
LD    (INTCNT),A
POP   AF
RET

```

```

;Vierte el contenido de la pantalla virtual en la
;pantalla real

```

```

SWAPDSP    PUSH  AF
           PUSH  BC
           PUSH  IY
           LD    B,16
           LD    IY,LIN1BUF
           LD    A,$80           ;Palabra de control que
                                   ;selecciona primera línea.
           OUT   (LCDC),A
           CALL  RETCAR
LOOPSW1    LD    A,(IY)
           OUT   (LCDD),A
           CALL  RETCAR
           INC   IY
           DJNZ  LOOPSW1
           LD    B,16
           LD    IY,LIN2BUF
           LD    A,$A8           ;Palabra de control que
                                   ;selecciona segunda línea.
           OUT   (LCDC),A
           CALL  RETCAR
LOOPSW2    LD    A,(IY)
           OUT   (LCDD),A
           CALL  RETCAR
           INC   IY
           DJNZ  LOOPSW2
           POP   IY
           POP   BC
           POP   AF

```



RET

FINTOOL EQU \$  
END

**\$title(PARTE III del programa ADATOC)**

**\$subtitle(Subrutina de recepción y conversión de código ejecutable)**

**;SET./94**

;Nota: este programa deberá ser enlazado junto con RESET, TOOLS y LOAD

PUBLIC LOAD,TRADUC,INT3  
PUBLIC FLAGFINLOAD,FININT3,NULMSG  
PUBLIC ERRSER,ERRBUF,ERRTRD,SALIR,FIN  
EXTERN POINTER,TRADFLG,ERRCOD,RESETKE  
EXTERN ASCBIN,DISPLAY,INILCD,BUFFER  
EXTERN URTDAT,URTCNT,CTC3,INICTC3,INISIO  
EXTERN KEYFLAG,EXEADD,ADDFLG  
EXTERN SWAPDSP,RTCINTE,INIRTCI,BEEP,OFFSET

;Rutina Principal

Defseg loadSEG, start=\$0400  
seg loadSEG

LOAD: DI

PUSH AF

PUSH HL

PUSH IY

LD A,0

LD (OFFSET),A ;Reinicia a 0 el offset

LD (OFFSET+1),A

LD (POINTER),A " " puntero

LD (TRADFLG),A " " el flag de datos

;disponibles

LD (ERRCOD),A " " Código de error

```

        LD    (ADDFLG),A    "    "    flag de dirección
;ejecución
        LD    IY,INIMSG1    ;Manda la palabra ESPERANDO al DISPLAY,
                            ;primera Fila.

        CALL  DISPLAY
        INC  A
        LD    IY,INIMSG2    ;Manda la palabra TRANSMISION... en
;segunda
                            ;Fila.

        CALL  DISPLAY
        CALL  INICTC3    ;Inicializa el CTC Z80 en modo contador.
        CALL  INISIO    ;Inicializa el USART 8251.
        EI
        LD    A,0
        OUT  (RTCINTE),A
        CALL  SWAPDSP
LOOP2  LD    A,(KEYFLAG)
        CP    $10
        JR    Z,FINTRA
        LD    A,(TRADFLG)
        CP    OFFH    ;Línea recibida?
        CALL  Z,TRADUC    ;Si, traducir
        LD    A,(TRADFLG)
        CP    OFH    ;Fin de transmisión?
        JR    NZ,LOOP2    ;No, seguir esperando
FINTRA  LD    A,0
        OUT  (RTCINTE),A
        CALL  INIRTCI
        LD    A,9
        OUT  (RTCINTE),A
        CALL  BEEP
        LD    A,02H
        OUT  (CTC3),A
        LD    A,0
        LD    IY,ENDMSG1
        CALL  DISPLAY
        INC  A

```

```

LD    IY,ENDMSG2
CALL  DISPLAY
CALL  RESETKE
POP   IY
POP   HL
POP   AF
RET

```

;Rutina de traducción de línea

```

TRADUC: PUSH    AF
        PUSH    BC
        PUSH    DE
        PUSH    HL
        PUSH    IX
        PUSH    IY
        LD     A,0
        LD     (POINTER),A
        LD     (TRADFLG),A
        LD     IX,BUFFER
        LD     A,(IX)
        CP     ' '
        JR     NZ,ERRTRD
        INC    IX
        LD     H,(IX)
        LD     L,(IX+1)
        CALL   ASCBIN      ;Número de datos
        CP     0           ;Es cero?
        JR     Z,FIN       ;Sí, finalizar
        LD     E,A
        INC    IX
        INC    IX
        LD     H,(IX)
        LD     L,(IX+1)
        CALL   ASCBIN      ;MSB de la dirección destino
        LD     B,A
        LD     H,(IX+2)

```

```

LD    L,(IX+3)
%REPT 4
INC   IX
%ENDREPT
CALL  ASCBIN      ;LSB de la dirección destino
LD    C,A
LD    A,(OFFSET)
LD    H,A
LD    A,(OFFSET+1)
LD    L,A
ADD   HL,BC
PUSH  HL
POP   IY
LD    A,(ADDFLG)
CP    1
JR    Z,CONTADD
LD    (EXEADD),IY
LD    A,1
LD    (ADDFLG),A
CONTADD LD    B,E
LOOP  INC   IX
      INC   IX
      LD    H,(IX)
      LD    L,(IX+1)
      CALL  ASCBIN      ;Dato
      LD    (IY),A
      INC   IY
      DJNZ  LOOP
SALIR POP   IY
      POP   IX
      POP   HL
      POP   DE
      POP   BC
      POP   AF
      RET
FIN   LD    A,0FH

```

```
LD (TRADFLG),A
JR SALIR
```

```
ERRTRD LD HL,ERRCOD
LD A,1
OR (HL)
LD (HL),A
JR FIN
```

```
INIMSG1 DB ESPERANDO ', $0D
INIMSG2 DB TRANSMISION... ', $0D
ENDMSG1 DB ' RECEPCION ', $0D
ENDMSG2 DB '-* TERMINADA *- ', $0D
EEMSG DB ' ERROR SERIE ', $0D
EBUMSG DB 'ERROR DE BUFFER ', $0D
NULMSG DB ', $0D
FINLOAD EQU $
```

;tratamiento de la interrupción

```
DEFSEG INT3SEG,START=$1600
SEG INT3SEG
```

```
INT3 DI
PUSH AF
PUSH BC
PUSH HL
IN A,(URTCNT)
AND %00111000
CALL NZ,ERRSER
IN A,(URTDAT)
CP OAH
CALL Z, FLAG
LD B,A
LD A,(POINTER)
CP 80
CALL NC,ERRBUF
```

```

LD    C,A
LD    A,B
LD    B,0
LD    HL,BUFFER
ADD   HL,BC
LD    (HL),A
INC   C
LD    A,C
LD    (POINTER),A
POP   HL
POP   BC
POP   AF
EI
RETI

```

```

FLAG   PUSH   AF
LD     A,OFFH
LD     (TRADFLG),A
POP    AF
RET

```

```

ERRSER  PUSH   AF
        PUSH  HL
        PUSH  IY
LD      HL,ERRCOD
OR      (HL)
LD      (HL),A
LD      A,0
LD      IY,ESEMSG
CALL    DISPLAY
INC     A
LD      IY,NULMSG
CALL    DISPLAY
CALL    SWAPDSP
POP     IY
POP     HL
POP     AF

```

RET

```
ERRBUF  PUSH      AF
        PUSH  HL
        PUSH  IY
        LD   HL,ERRCOD
        LD   A,2
        OR   (HL)
        LD   (HL),A
        LD   A,OFH
        LD   (TRADFLG),A
        LD   A,0
        LD   IY,EBUMSG
        CALL DISPLAY
        INC  A
        LD   IY,NULMSG
        CALL DISPLAY
        CALL SWAPDSP
        POP  IY
        POP  HL
        POP  AF
        RET
```

```
FININT3 EQU  $
        END
```

**\$title(PARTE IV DEL PROGRAMA DE ADATOC)**

**\$subtitle(PROGRAMA PRINCIPAL DEL ADATOC)**

**;SET./94**

```
EXTERN  RTCTEST,RTCDECS,RTCUSEG,RTCDSEG,RTCUMIN,RTCDMIN,RTCUHOR
EXTERN  RTCDHOR,RTCUDIA,RTCD DIA,RTCDSEM,RTCUMES,RTCDMES,RTCYEAR
EXTERN  RTCINIP,RTCINTE,KEY,KEYFLAG,CTC0,CTC1,CTC2,CTC3
EXTERN  MEDIOS,SECOND,BASEDAT,URTDAT,URTCNT,LCDC,LCDI,LCDD,CAD
EXTERN  KEYCOD,ADCOD,EVITREB,UNSEG,ACUMFLG,NUMCAN,LINE1,LINE2
EXTERN  KEYBUF,INIDHOR,INIUHOR,INIDMIN,INIUMIN,INIDSEG,INIUSEG
EXTERN  HORAFIN,INTCNT,DIGCONT,ASCBUF,EXEADD,CALLBUF,SWAPDSP
```

```

EXTERN    RETAR,DISPLAY,BINASC,BN2DEC,INICTC1,INIKEY,INISIO
EXTERN    INICTCO,DSPCAD,DSPCAR,LOAD,HORADT,FECHDT,YEARDT,DECIDT
EXTERN    POINTDT,TEMPDT,INIRTC,INIRTCI,ERASE,DSPENA,CANAL
EXTERN    ENADSP,DISDSP,SGUNDOS,RESETKE,MEMORY,POINTME
EXTERN    FECHFIN,INIDMES,INIUMES,INIDDIA,INIUDIA,NULMSG

PUBLIC    MAIN,FINMAIN,FININTO,FININT1,FININT2,IDLE,BEEP
PUBLIC    DEMORA,ACTUA,INTO,INT1,INT2

DEFSEG    LCDTSEG,START=$0600
SEG       LCDTSEG
MAIN     EI
DENU    CALL  IDLE
        JR    DENU
EXITMA  RET

;Visualiza los contenidos de direcciones
;de memoria seleccionada por teclado.
VMEMORY  PUSH  AF
        PUSH  BC
        PUSH  HL
        CALL  RESETKE
        CALL  SUBRU1
        CALL  ENADSP
VMEL00   LD    A,(KEYFLAG)
        CP    $10
        JR    Z,FINVMEM
        LD    A,(DIGCONT)
        CP    4
        JR    NZ,VMEL00
        LD    A,' '
        CALL  DSPCAR
        LD    A,(KEYBUF)
        LD    (MEMORY),A
        LD    A,(KEYBUF+1)
        LD    (MEMORY+1),A
        LD    A,(KEYBUF+2)

```



```

        LD    (MEMORY+2),A
        LD    A,(KEYBUF+3)
        LD    (MEMORY+3),A
        LD    B,4
        LD    A,(MEMORY)
ROTA   RL    A
        DJNZ ROTA
        LD    B,A
        LD    A,(MEMORY+1)
        OR    B
        LD    H,A
        LD    B,4
        LD    A,(MEMORY+2)
ROTA1  RL    A
        DJNZ ROTA1
        LD    B,A
        LD    A,(MEMORY+3)
        OR    B
        LD    L,A
        LD    (POINTME),HL
        CALL VERDAT
        CALL RESETKE
FINVMEM CALL DISDSP
        POP  HL
        POP  BC
        POP  AF
        RET

```

```

;Transmite todos los
;datos necesarios a la
;computadora

```

```

TRADA:   PUSH  AF
         PUSH  BC
         PUSH  DE
         PUSH  HL
         PUSH  IY
         LD    A,0           ;Desactiva la acumulación

```

```

LD      (ACUMFLG),A
LD      DE,BASEDAT
LD      HL,(POINTDT)
OR      A
SBC     HL,DE          ;Se calcula el # de datos
JR      NZ,SEGTRAD    ;acumulados
CALL    ERASE          ;Borra pantalla
LD      A,0            ;Si no hay datos
LD      IY,NTRAMSG    ;visualiza mensaje:
CALL    DISPLAY        ;"MEMORIA VACIA
INC     A
LD      IY,ONTRAM     ; NO TRANSMITE"
CALL    DISPLAY
JP      ENDTRAD        ;Se termina la rutina
SEGTRAD CALL ERASE     ;Viene aquí si existen
LD      A,0            ;datos
LD      IY,TRADMSG    ;Visualiza mensaje:
CALL    DISPLAY        ;TRANSMITIENDO
INC     A
LD      IY,DATMSG     DATOS....
CALL    DISPLAY
CALL    INISIO         ;Inicializo el USART
LD      B,6            ;Envia la hora,
LD      IY,INIDHOR    ;minutos y segundos
LOOPTR1 LD  A,(IY)     ;iniciales
CALL    TRANS
INC     IY
DJNZ   LOOPTR1
LD      B,4            ;Envia el mes,
LD      IY,INIDMES    ;y el día iniciales
LOOPTR2 LD  A,(IY)
CALL    TRANS
INC     IY
DJNZ   LOOPTR2
LD      A,(NUMCAN)    ;Envia el #
CALL    TRANS          ;de canales
LD      A,(UNSEG)     ;Envia el tiempo en

```

```

        CALL  TRANS          ;segundos seleccionado
OTRO1 LD   A,(KEYFLAG) ;Lee el estado del
        CP    $10          ;teclado y compara
        JR    Z,FINTRAD   ;si es ESC aborta la
        DI                      ;Transmisión
        LD    HL,(POINTDT) ;Desde aquí
        LD    A,(DE)       ;empiezo a
        CALL  TRANS          ;enviar los
        INC  DE            ;datos acumulados
        OR   A              ;en el ADATOC
        SBC  HL,DE
        EI
        JR   NZ,OTRO1
FINTRAD LD   A,0           ;Envio un cero como
        CALL  TRANS          ;fin de transmisión
        LD   IY,TRAMSG1    ;Visualiza mensaje:
        CALL  DISPLAY       ;TRANSMISION
        INC  A
        LD   IY,TERMMSG    ;TERMINADA
        CALL  DISPLAY
ENDTRAD CALL  RESETKE      ;Se cambia el estado
        POP  IY             ;del teclado a 20H
        POP  HL
        POP  DE
        POP  BC
        POP  AF
        RET

```

;Subrutina que sirve para poner la hora final y  
;la fecha final del proceso de  
;acumulación de datos.

```

ACUMULA  PUSH  AF
         PUSH  IY
         CALL  RESETKE
         CALL  ERASE      ;Borro la pantalla
         LD   A,0
         LD   (ACUMFLG),A

```

```

LD IY,HORFMSG ;Coloca el mensaje HORA FINAL:
CALL DISPLAY
CALL ENADSP ;Habilita el display
ACRUA1 LD A,(KEYFLAG) ;Escape de hora final
CP $10
JP Z,FINACUM
LD A,(DIGCONT) ;Sirve para colocar sólo 4 dígitos
CP 4
JR NZ,ACRUA1
CALL DISDSP ;Deshabilita el display
LD A,(KEYBUF)
LD (HORAFIN),A
LD A,(KEYBUF+1)
LD (HORAFIN+1),A
LD A,(KEYBUF+2)
LD (HORAFIN+2),A
LD A,(KEYBUF+3)
LD (HORAFIN+3),A
CALL RESETKE
LD A,1 ;Se visualiza FECHA FINAL:
LD IY,FECFMSG
CALL DISPLAY
CALL ENADSP
CALL FEINI
ACRUA2 LD A,(KEYFLAG)
CP $10
JR Z,CANJUMP
LD A,(DIGCONT)
CP 4
JR NZ,ACRUA2
CALL DISDSP
LD A,(KEYBUF)
LD (FECHFIN),A
LD A,(KEYBUF+1)
LD (FECHFIN+1),A
LD A,(KEYBUF+2)
LD (FECHFIN+2),A

```

```

        LD    A,(KEYBUF+3)
        LD    (FECHFIN+3),A
CANJUMP    CALL  RESETKE
        LD    A,0
        LD    IY,ACUMSG1 ;Se visualiza CANALES A ACUM.:
        CALL  DISPLAY
LOOPAC    LD    A,(KEYFLAG)
        CP    $20
        JR    Z,LOOPAC
        CP    $08
        JR    NC,FINACUM
        INC  A
        LD    (NUMCAN),A
        LD    A,0
        LD    (SECOND),A
        LD    IY,BASEDAT
        LD    (POINTDT),IY
        INC  A
        LD    (ACUMFLG),A ;ACUMULACION ACTIVA
        CALL  HOINI
FINACUM  CALL  DISDSP
        POP  IY
        POP  AF
        RET

```

;Reporte de estado de estado parcial

```

REPORT:  DI
        PUSH AF
        PUSH BC
        PUSH DE
        PUSH HL
        PUSH IY
        LD    A,(ACUMFLG)
        CP    0
        JR    Z,INACREP
        LD    A,0
        LD    IY,CANMSG

```

```

CALL DISPLAY
LD A,(NUMCAN)
DEC A
OR $30
CALL DSPCAR
LD IY,FINMSG
CALL DSPCAD
LD A,(HORAFIN)
OR A,$30
CALL DSPCAR
LD A,(HORAFIN+1)
OR A,$30
CALL DSPCAR
LD A,':'
CALL DSPCAR
LD A,(HORAFIN+2)
OR A,$30
CALL DSPCAR
LD A,(HORAFIN+3)
OR A,$30
CALL DSPCAR
LD A,1
LD IY,NDATMSG
CALL DISPLAY
LD HL,(POINTDT)
LD DE,BASEDAT
SBC HL,DE
EX DE,HL
LD HL,SGUNDOS
CALL BN2DEC
LD A,(HL)
LD B,A
LOOPREP INC HL
LD A,(HL)
CALL DSPCAR
DJNZ LOOPREP
LD A,','

```

```

%REPT 6
CALL DSPCAR
%ENDREPT
JR SALREP
INACREP LD A,0
LD IY,FACMSG1
CALL DISPLAY
INC A
LD IY,ACINMSG
CALL DISPLAY
SALREP POP IY
POP HL
POP DE
POP BC
POP AF
EI
RET

```

;Subrutina que da el reporte final del  
;proceso de acumulación de datos.

```

REPORTF PUSH AF
PUSH BC
PUSH HL
PUSH IY
LD A,0
LD IY,REFMSG
CALL DISPLAY
LD A,(INIDHOR)
OR A,$30
CALL DSPCAR
LD A,(INIUHOR)
OR A,$30
CALL DSPCAR
LD A,'.'
CALL DSPCAR
LD A,(INIDMIN)
OR A,$30

```

```

CALL  DSPCAR
LD    A,(INIUMIN)
OR    A,$30
CALL  DSPCAR
LD    A
CALL  DSPCAR
LD    IY,TIMDMMSG
CALL  DSPCAD
LD    A,(UNSEG)
LD    D,0
LD    E,A
LD    HL,SGUNDOS
CALL  BN2DEC
LD    A,(HL)
LD    B,A
SEREP INC  HL
LD    A,(HL)
CALL  DSPCAR
DJNZ  SEREP
LD    IY,S.MSG
CALL  DSPCAD
LD    A,1
LD    IY,NCANMSG
CALL  DISPLAY
LD    A,(NUMCAN)
OR    $30
CALL  DSPCAR
LD    IY,NDFMSG
CALL  DSPCAD
LD    HL,(POINTDT)
LD    DE,BASEDAT
SBC  HL,DE
EX   DE,HL
LD   HL,SGUNDOS
CALL BN2DEC
LD   A,(HL)
LD   B,A

```



```

LOOPRE      INC    HL
            LD     A,(HL)
            CALL  DSPCAR
            DJNZ  LOOPRE
            LD     A,' '
            %REPT 5
            CALL  DSPCAR
            %ENDREPT
            CALL  RESETKE
            POP   IY
            POP   HL
            POP   BC
            POP   AF
            RET

```

;Ejecuta La rutina ubicada en EXEADD

```

EXECUTE     DI
            PUSH  AF
            PUSH  IY
            LD    A,$CD
            LD    (CALLBUF),A
            LD    IY,(EXEADD)
            LD    (CALLBUF+1),IY
            LD    A,$C3
            LD    (CALLBUF+3),A
            LD    IY,RETEXE
            LD    (CALLBUF+4),IY
            JP    CALLBUF
RETEXE     LD    IY,EXEMSG
            LD    A,0
            CALL  DISPLAY
            LD    IY,TERMMSG
            INC   A
            CALL  DISPLAY
            POP   IY
            POP   AF
            EI

```

RET

;Selecciona el tiempo en segundos

;que servira como frecuencia de

;la acumulación de datos.

```
SELECTI    PUSH  AF
           PUSH  BC
           PUSH  HL
           PUSH  IY
           LD   A,0
           LD   (ACUMFLG),A
           CALL ERASE
SELE1 LD   A,(UNSEG)
           LD   HL,ASCBUF
           LD   D,0
           LD   E,A
           CALL BN2DEC           ;Convierte numero binario a ASCII DECIMAL
           LD   A,(HL)
           LD   B,A
           INC  HL
LIMO  LD   A,(HL)
           CALL DSPCAR
           INC  HL
           DJNZ LIMO
           LD   IY,SEGMSG
           CALL DSPCAD
           LD   A,' '
           %REPT 11
           CALL DSPCAR
           %ENDREPT
           LD   A,(KEYFLAG)
           CP   10H
           JR   Z,QUIT
           CP   11H
           CALL Z,NOMBRE
           CALL SELTI
           JR   SELE1
```

```

QUIT  POP  IY
      POP  HL
      POP  BC
      POP  AF
      RET

```

;Subrutina que sirve para incrementar la dirección de  
;memoria así como para visualizarlo.

```

INCRE:  PUSH  AF
        PUSH  BC
        PUSH  HL
        PUSH  IY
        CALL  SUBRU1
        LD   IY,(POINTME)
        LD   BC,12
        ADD  IY,BC
        LD   (POINTME),IY
        CALL  DIRDAT
        CALL  RESETKE
FININC  POP   IY
        POP  HL
        POP  BC
        POP  AF
        RET

```

;Subrutina que sirve para decrementar la dirección de  
;memoria así como para visualizarlo.

```

DECRE:  PUSH  AF
        PUSH  BC
        PUSH  HL
        PUSH  IY
        CALL  SUBRU1
        LD   IY,(POINTME)
        LD   B,12
DISMI  DEC  IY
        DJNZ DISMI
        LD   (POINTME),IY

```

```

        CALL  DIRDAT
        CALL  RESETKE
FINDEC      POP   IY
        POP   HL
        POP   BC
        POP   AF
        RET

```

;MHORA: Subrutina que muestra la hora y la fecha.

```

MHORA DI
        PUSH  AF
        PUSH  IY
        CALL  PONEHO           ;Mando a visualizar la palabra HORA -.
        LD    IY,HORADT
        CALL  DSPCAD
        CALL  PONEFE           ;Mando a visualizar la palabra FECHA=.
        LD    IY,FECHDT
        CALL  DSPCAD
        POP   IY
        POP   AF
        EI
        RET

```

;Subrutina que manda a actualizar la hora y la  
;fecha.

```

ACTUA:      PUSH  AF
        LD    A,0
        OUT   (RTCINTE),A
        CALL  INIRTCI
        OUT   (RTCTEST),A
        OUT   (RTCINIP),A
        CALL  ERASE
        CALL  RESETKE
        CALL  PONEHO           ;Mando a visualizar la palabra HORA -.
        LD    A,' '
        CALL  DSPCAR
        CALL  SWAPDSP

```

```

        CALL ENADSP
ACTUA1  LD   A,(DIGCONT)
        CP   4
        JR   Z,ACTUA3
        LD   A,(KEYFLAG)
        CP   10H
        JR   Z,IRAFE
        JR   ACTUA1
ACTUA3  CALL DISDSP
        CALL ACTUHO           ;Subrutina que actualiza la hora.
IRAFE   CALL PONEFE         ;Mando a visualizar la palabra FECHA=.
        CALL RESETKE
        CALL SWAPDSP
        CALL ENADSP
ACTUA4  LD   A,(DIGCONT)
        CP   6
        JR   Z,ACTUA5
        LD   A,(KEYFLAG)
        CP   10H
        JP   Z,ACTFIN
        JR   ACTUA4
ACTUA5  CALL ACTUFE
ACTFIN  CALL DISDSP
        LD   A,10H
        LD   (KEYFLAG),A
        LD   A,1
        OUT  (RTCINIP),A
        LD   A,$09
        OUT  (RTCINTE),A
        POP  AF
        RET

SUBRU1  PUSH  AF
        PUSH IY
        CALL ERASE
        LD   A,0
        LD   IY,MEMMSG

```

```
CALL DISPLAY
POP IY
POP AF
RET
```

;Subrutina que visualiza dirección y datos.

```
DIRDAT:    PUSH AF
           PUSH HL
           PUSH IY
           POP HL
           LD A,H
           CALL BINASC
           LD A,H
           CALL DSPCAR
           LD A,L
           CALL DSPCAR
           PUSH IY
           POP HL
           LD A,L
           CALL BINASC
           LD A,H
           CALL DSPCAR
           LD A,L
           CALL DSPCAR
           LD A
           CALL DSPCAR
           CALL VERDAT
           POP HL
           POP AF
           RET
```

;Muestra los datos de las diferentes direcciones de memoria  
;consecutivas.

```
VERDAT:    PUSH AF
           PUSH BC
           PUSH IY
           LD IY,(POINTME)
```

```

        LD    B,4
VIS3   CALL  VISNUH
        INC  IY
        DJNZ VIS3
        CALL LINE2
        LD    B,8
VISMAS CALL  VISNUH
        INC  IY
        DJNZ VISMAS
        POP  IY
        POP  BC
        POP  AF
        RET

```

;Subrutina que visualiza número hexadecimal de  
;una posición de memoria representado por el  
;contenido del registro IY.

```

VISNUH:  PUSH  AF
         PUSH  HL
         LD    A,(IY)
         CALL  BINASC
         LD    A,H
         CALL  DSPCAR
         LD    A,L
         CALL  DSPCAR
         POP  HL
         POP  AF
         RET

```

```

SELT I:  PUSH  AF
         PUSH  HL
         PUSH  IX
         CALL  LINE1
         LD    A,(KEYFLAG) ;De acuerdo a la
         CP    10H          ;tecla presionada
         JR    NC,QUITA    ;selecciono un valor de
         LD    L,A          ;la tabla.

```

```

LD    H,0           ;Lo que sigue selecciona un
EX    DE,HL        ;valor de la tabla.
LD    IX,TABLA1
ADD   IX,DE
LD    A,(IX)
LD    (UNSEG),A
QUITA: POP    IX
      POP    HL
      POP    AF
      RET

```

;Visualiza el nombre de la persona que  
;ha desarrollado el equipo.

```

NOMBRE:  PUSH  AF
        PUSH  IY
        LD    A,(KEYFLAG)
        CP    10H
        JR    Z,QUITA1
        LD    A,1
        LD    IY,PARAV1
        CALL  DISPLAY
        DEC   A
        LD    IY,PARAV2
        CALL  DISPLAY
        CALL  RETTAR
        LD    A,0
        LD    IY,NULMSG
        CALL  DISPLAY
        INC   A
        CALL  DISPLAY
        CALL  RETTAR1
QUITA1  POP    IY
        POP    AF
        RET

RETTAR  PUSH  DE
        LD    D,OFFH

```



```

ZAS1 LD E,OFFH
ZAS2 DEC E
      JR NZ,ZAS2
      DEC D
      JR NZ,ZAS1
      POP DE
      RET

```

```

RETTAR1 PUSH DE
        LD D,099H
ZAS3 LD E,099H
ZAS4 DEC E
      JR NZ,ZAS4
      DEC D
      JR NZ,ZAS3
      POP DE
      RET

```

```

ACTUHO PUSH AF
        PUSH BC
        PUSH IX
        LD A,1
        OUT (RTCINIP),A
        LD IX,KEYBUF
        LD B,4
        LD A,RTCDHOR
        LD C,A
PONE LD A,(IX)
      OUT (C),A
      INC IX
      DEC C
      DJNZ PONE
      POP IX
      POP BC
      POP AF
      RET

```

;Rutina que actualiza la fecha.

```
ACTUFE      PUSH  AF
            PUSH  BC
            PUSH  IX
            LD   A,1
            OUT  (RTCINIP),A
            LD   IX,KEYBUF
            LD   A,(IX)      ;Decenas de dias.
            OUT  (RTCDDIA),A
            LD   A,(IX+1)
            OUT  (RTCUDIA),A ;Unidad de dias.
            LD   A,(IX+2)    ;Decenas de meses.
            OUT  (RTCDMES),A
            LD   A,(IX+3)    ;Unidad de meses.
            OUT  (RTCUMES),A
            LD   B,4
            LD   A,(IX+4)
L000  RL   A
            DJNZ L000
            LD   B,A
            LD   A,(IX+5)
            OR   B
            LD   (YEARDT),A ;Se posiciona el ano.
            OUT  (RTCYEAR),A
            POP  IX
            POP  BC
            POP  AF
            RET
```

;Subrutina que coloca la palabra HORA - en la pantalla.

```
PONEHO      PUSH  AF
            PUSH  IY
            LD   A,0
            LD   IY,HORAMSG
            CALL DISPLAY
            POP  IY
            POP  AF
```

RET

;Subrutina que coloca la palabra FECHA -  
;en la pantalla LCD.

```
PONEFE:    PUSH  AF
           PUSH  IY
           LD    A,1
           LD    IY,FECHMSG
           CALL  DISPLAY
           POP   IY
           POP   AF
           RET
```

```
ACUTERM    PUSH  AF
           PUSH  BC
           PUSH  IY
           LD    A,0
           LD    IY,FACMSG1
           CALL  DISPLAY
           INC   A
           LD    IY,TERMMSG
           CALL  DISPLAY
           LD    B,55
```

```
ALARM CALL  BEEP1
           LD    A,(KEYFLAG)
           CP    $10
           JR    Z,GOOUT
           DJNZ  ALARM
           CALL  RESETKE
```

```
GOOUT POP   IY
           POP   BC
           POP   AF
           RET
```

```
BEEP1 PUSH  AF
           LD    A,15H ;Factor Preescalador 16
           OUT   (CTC2),A
```

```

LD    A,34
OUT   (CTC2),A
CALL  DEMORA1
LD    A,3
OUT   (CTC2),A
CALL  DEMORA2
POP   AF
RET

```

```

DEMORA1  PUSH  DE
          LD    D,255
JAS2    LD    E,255
JAS1    DEC   E
          JR    NZ,JAS1
          DEC   D
          JR    NZ,JAS2
          POP   DE
          RET

```

```

DEMORA2  PUSH  DE
          LD    D,125
XAS3    LD    E,125
XAS4    DEC   E
          JR    NZ,XAS4
          DEC   D
          JR    NZ,XAS3
          POP   DE
          RET

```

```

CANX: DI                ;Deshabilito Inter.enmascarable
          PUSH  AF        ;RUTINA PARA EL CANAL X EN A
          PUSH  BC
          PUSH  IY
          LD    (CANAL),A ;Guardo en CANAL el número de canal
          LD    B,A        ;GUARDA EN B EL NUMERO DE CANAL
          CALL  PONA       ;Pone TEMPERATURA
          LD    A,'C'     ;Mando a visualizar el caracter C

```

```

CALL LINE2          ;en la fila 2
CALL DSPCAR
LD   A,$30          ;Mando a visualizar el # que corres-
                    ;ponde al canal.

ADD  A,B
CALL DSPCAR
LD   A,'-'          ;Mando a visualizar el signo -.
CALL DSPCAR
CALL MTABLA
POP  IY
POP  BC
POP  AF
EI
RET

```

```

;Manda a visualizar la palabra TEMPERATURA: o
;la palabra ACUMULANDO: dependiendo del contenido
;de ACUMFLG. También manda a visualizar * en for-
;ma parpadeante.

```

```

PONA: PUSH  AF
      PUSH  BC
      PUSH  DE
      PUSH  HL
      PUSH  IY
      LD   A,(ACUMFLG) ;Contiene un 1 o un 0
      CP   0
      JR   Z,TALI      ;Si es cero no acumula
      LD   A,0         ;Muestro el mensaje
      LD   IY,ACUMSG2 ;ACUMULANDO :
      CALL DISPLAY
      LD   A,(SECOND) ;Empieza cuenta de segundos.
      LD   HL,SGUNDOS
      LD   D,0
      LD   E,A
      CALL BN2DEC
      LD   A,(HL)      ;Carga en A el # de bytes.
      LD   B,A

```

```

        INC    HL
PONA1  LD     A,(HL)
        CALL   DSPCAR           ;Coloca un caracter en memoria
                                   ;de pantalla.
        INC    HL
        DJNZ  PONA1
        LD     A,(MEDIOS) ;Mando a visualizar
        CP    1             ;asterisco en forma
        JR    Z,MED        ;parpadeante.
        LD     A,' '
        JR    CONPONA
MED     LD     A,'*'
CONPONA CALL   DSPCAR
        LD     A,' '       ;Mando a
        CALL   DSPCAR       ;visualizar
        CALL   DSPCAR       ;espacios en blanco
        JR    FINPONA
TALI:  LD     IY,TEMPMSG ;Muestra la palabra
        LD     A,0         ;TEMPERATURA:
        CALL   DISPLAY
FINPONA POP    IY
        POP    HL
        POP    DE
        POP    BC
        POP    AF
        RET

```

```

;Muestra el valor de la temperatura en grados
;centigrados seleccionandolo de una tabla
;también muestra °C y el tiempo seleccionado
;en segundos para la toma de datos.

```

```

MTABLA:  PUSH  AF
          PUSH  BC
          PUSH  DE
          PUSH  HL
          PUSH  IX
          PUSH  IY

```

```

LD    IY,TEMPDT
CALL  DSPCAD
LD    IY,GCMSG
CALL  DSPCAD
LD    A,(UNSEG)    ;También muestra el número de segs.
LD    HL,DECIDT
LD    D,0
LD    E,A
CALL  BN2DEC        ;Convierte numero binario a ASCII
LD    A,(HL)
LD    B,A
INC   HL
LIVO: LD    A,(HL)
CALL  DSPCAR
INC   HL
DJNZ  LIVO
LD    IY,S.MSG
CALL  DSPCAD
NO2   POP   IY
      POP   IX
      POP   HL
      POP   DE
      POP   BC
      POP   AF
      RET

```

;Convierte el valor digitalizado por el A/D en  
;unidades físicas de temperatura. Así como acondiciona  
;este valor para poder ser visualizado en pantalla LCD.

```

DATGC:    PUSH  AF
          PUSH  DE
          PUSH  HL
          PUSH  IX
          PUSH  IY
LD    A,(ADCOD)    ;Código digitalizado por el A/D
CP    48
JP    NC,POSIT

```

```

LD    A,'-'
JP    CONTSIG
POSIT LD    A,' '
CONTSIG LD    IY,TEMPDT
LD    (IY),A
INC   IY
LD    A,(ADCOD)
LD    L,A
LD    H,0           ;Lo que sigue selecciona un
ADD   HL,HL        ;valor correspondiente al valor
EX    DE,HL        ;digitalizado de una tabla.
LD    IX,TABLA
ADD   IX,DE
LD    A,(IX)
CALL  BINASC
LD    A,H
LD    (IY),A
LD    A,L
INC   IY
LD    (IY),A
LD    A,2EH        ;El '.' lo cargo en el buffer
INC   IY
LD    (IY),A
LD    A,(IX+1)
CALL  BINASC
INC   IY
LD    A,H
LD    (IY),A
LD    A,L
INC   IY
LD    (IY),A
LD    A,$0D
INC   IY
LD    (IY),A
POP   IY
POP   IX
POP   HL

```



```
POP DE
POP AF
RET
```

;Lee la hora inicial de acumulación de datos.

```
HOINI:    DI
          PUSH AF
          IN  A,(RTCUDIA)
          IN  A,(RTCDHOR)
          AND OFH
          LD  (INIDHOR),A
          IN  A,(RTCUHOR)
          AND OFH
          LD  (INIUHOR),A
          IN  A,(RTCDMIN)
          AND OFH
          LD  (INIDMIN),A
          IN  A,(RTCUMIN)
          AND OFH
          LD  (INIUMIN),A
          IN  A,(RTCDSEG)
          AND OFH
          LD  (INIDSEG),A
          IN  A,(RTCUSEG)
          AND OFH
          LD  (INIUSEG),A
          POP AF
          EI
          RET
```

;Lee la FECHA inicial de acumulación de datos.

```
FEINI:    DI
          PUSH AF
          IN  A,(RTCDSEM)
          IN  A,(RTCDDIA)      ;Leo decenas de DIAS.
          AND OFH
          LD  (INIDDIA),A
```

```

INC    IY
LD     (IY),A
IN     A,(RTCDMIN)
AND    OFH
CALL   BINASC      ;Convierte a códigos ASCII.
INC    IY
LD     A,L
LD     (IY),A
IN     A,(RTCUMIN)
AND    OFH
CALL   BINASC      ;Convierte a códigos ASCII.
INC    IY
LD     A,L
LD     (IY),A
LD     A,':'
INC    IY
LD     (IY),A
IN     A,(RTCDSEG)
AND    OFH
CALL   BINASC      ;Convierte a códigos ASCII.
INC    IY
LD     A,L
LD     (IY),A
IN     A,(RTCUSEG)
AND    OFH
CALL   BINASC      ;Convierte a códigos ASCII.
INC    IY
LD     A,L
LD     (IY),A
LD     A,','
INC    IY
LD     (IY),A
INC    IY
LD     A,$0D
LD     (IY),A
POP    IY
POP    HL

```

```
POP AF
RET
```

;Leo días, meses y el año correspondiente.

```
LEOFE:    PUSH AF
          PUSH HL
          PUSH IY
          LD IY,FECHDT
          IN A,(RTCDSEM)
          IN A,(RTCDDIA)      ;Leo decenas de DIAS.
          AND OFH
          CALL BINASC
          LD A,L
          LD (IY),A
          IN A,(RTCUDIA)     ;Leo unidad de DIAS.
          AND OFH
          CALL BINASC
          INC IY
          LD A,L
          LD (IY),A
          LD A,'/'
          INC IY
          LD (IY),A
          IN A,(RTCDMES)     ;Leo decenas de Meses.
          AND OFH
          CALL BINASC
          INC IY
          LD A,L
          LD (IY),A
          IN A,(RTCUMES)     ;Leo unidad de Meses.
          AND OFH
          CALL BINASC
          INC IY
          LD A,L
          LD (IY),A
          LD A,'/'
          INC IY
```

```

LD    (IY),A
LD    A,(YEARDT)
CALL  BINASC
INC   IY
LD    A,H
LD    (IY),A
INC   IY
LD    A,L
LD    (IY),A
INC   IY
LD    A,$0D
LD    (IY),A
POP   IY
POP   HL
POP   AF
RET

```

```

IDLE: PUSH AF
      LD    A,(KEYFLAG) ;Cargo A con el codigo de
      CP    8           ;la tecla presionada.
      JR    NC,CTRLKEY
      CALL  CANX
      POP   AF
      RET

```

```

CTRLKEY  CP    8
         CALL  Z,ACUMULA ;Va a la subrutina ACUMULA
         CP    9
         CALL  Z,TRADA
         CP    0AH
         CALL  Z,REPORT
         CP    0BH
         CALL  Z,VMEMORY
         CP    0CH
         CALL  Z,INCRE
         CP    0DH
         CALL  Z,DECRE
         CP    0EH

```

```

CALL Z,EXECUTE
CP    0FH
CALL Z,REPORTF
CP    10H
CALL Z,MHORA
CP    11H
CALL Z,ACTUA
CP    12H
CALL Z,SELECTI
CP    13H
CALL Z,LOAD
CP    $30
CALL Z,ACUTERM
POP   AF
RET

```

;Transmite un caracter

;binario en A

;a la computadora

```
TRANS:    PUSH AF
```

```
ESPR IN A,(URTCNT)
```

```
BIT 0,A
```

```
JR Z,ESPR
```

```
POP AF
```

```
OUT (URTDAT),A
```

```
RET
```

;Rutina que causa un

;sonido(pitido) en el

;ADATOC al suceder un

;arranque en frio o

;caliente y al presionar

;cualquier tecla.

```
BEEP: PUSH AF
```

```
LD    A,35H    ;Palabra de control(modos
```

```
OUT   (CTC2),A ;temporizador) del CTC.
```

```
LD    A,4      ;Constante de tiempo
```

```

OUT    (CTC2),A
CALL   DEMORA
LD     A,3           ;Palabra de control
OUT    (CTC2),A     ;que resetea el
CALL   DEMORA       ;temporizador
POP    AF
RET

DEMORA    PUSH  DE
          LD    D,110
XAS2     LD    E,110
XAS1     DEC   E
          JR    NZ,XAS1
          DEC   D
          JR    NZ,XAS2
          POP   DE
          RET

TRADMSG  DB    'TRANSMITIENDO ', $OD
DATMSG   DB    'DATOS...', $OD
SEGMSG   DB    'SEG. ', $OD
S.MSG    DB    'S. ', $OD
HORAMSG  DB    'HORA - ', $OD
FECHMSG  DB    'FECHA= ', $OD
TEMPMSG  DB    'TEMPERATURA: ', $OD
HORFMSG  DB    'HORA FINAL:', $OD
FECFMSG  DB    'FECHA FINAL:', $OD
GCMSG    DB    $DF, 'C ', $OD
ACUMSG1  DB    'CANALES A ACUM.: ', $OD
ACUMSG2  DB    'ACUMULANDO: ', $OD
FACMSG1  DB    'ACUMULACION ', $OD
CANMSG   DB    'CAN:0-', $OD
FINMSG   DB    'FIN', $OD
NDATMSG  DB    'N.DATOS: ', $OD
ACINMSG  DB    'INACTIVA ', $OD
TRAMSG1  DB    'TRANSMISION ', $OD
REFMSG   DB    'H.I.', $OD

```

```

NCANMSG      DB      'HC=' , $0D
NDFMSG       DB      ' DATOS=' , $0D
MEMMSG       DB      'M>=' , 0DH
EXEMSG       DB      ' EJECUCION      ' , $0D
TERMMSG      DB      ' TERMINADA      ' , $0D
TIMDMSG      DB      'C/' , $0D
NTRAMSG      DB      ' "MEMORIA VACIA ' , $0D
ONTRAM       DB      ' NO TRANSMITE" ' , $0D
PARAV1       DB      'C.MARTIN CRUZ S.' , 0DH
PARAV2       DB      'Desarrollado por' , 0DH

```

;Tabla de tiempos seleccionados por teclado.

```

TABLA1:      DB      1          ;TECLA
              DB      2          ;TECLA 1
              DB      3          ;TECLA 2
              DB      4          ;TECLA 3
              DB      5          ;TECLA 4
              DB      6          ;TECLA 5
              DB      10         ;TECLA 6
              DB      15         ;TECLA 7
              DB      20         ;TECLA 8
              DB      25         ;TECLA 9
              1DB     30         ;TECLA A
              DB      45         ;TECLA B
              DB      60         ;TECLA C
              DB      120        ;TECLA D
              DB      180        ;TECLA E
              DB      240        ;TECLA F

```

```

TABLA DB      23H,00H,22H,52H,22H,04H,21H,55H
      DB      21H,07H,20H,59H,20H,11H,19H,62H
      DB      19H,14H,18H,66H,18H,18H,17H,69H
      DB      17H,21H,16H,73H,16H,25H,15H,76H
      DB      15H,28H,14H,80H,14H,32H,13H,84H
      DB      13H,35H,12H,87H,12H,39H,11H,91H
      DB      11H,42H,10H,94H,10H,46H,09H,98H
      DB      09H,49H,09H,01H,08H,53H,08H,05H

```

DB 07H,56H,07H,08H,06H,60H,06H,12H  
 DB 05H,64H,05H,15H,04H,67H,04H,19H  
 DB 03H,71H,03H,22H,02H,74H,02H,26H  
 DB 01H,78H,01H,29H,00H,81H,00H,33H  
 DB 00H,15H,00H,64H,01H,12H,01H,60H  
 DB 02H,08H,02H,56H,03H,05H,03H,53H  
 DB 04H,01H,04H,49H,04H,98H,05H,46H  
 DB 05H,94H,06H,42H,06H,91H,07H,39H  
 DB 07H,87H,08H,35H,08H,84H,09H,32H  
 DB 09H,80H,10H,28H,10H,76H,11H,25H  
 DB 11H,73H,12H,21H,12H,69H,13H,18H  
 DB 13H,66H,14H,14H,14H,62H,15H,11H  
 DB 15H,59H,16H,07H,16H,55H,17H,04H  
 DB 17H,52H,18H,00H,18H,48H,18H,96H  
 DB 19H,45H,19H,93H,20H,41H,20H,89H  
 DB 21H,38H,21H,86H,22H,34H,22H,82H  
 DB 23H,31H,23H,79H,24H,27H,24H,75H  
 DB 25H,24H,25H,72H,26H,20H,26H,68H  
 DB 27H,16H,27H,65H,28H,13H,28H,61H  
 DB 29H,09H,29H,58H,30H,06H,30H,54H  
 DB 31H,02H,31H,51H,31H,99H,32H,47H  
 DB 32H,95H,33H,44H,33H,92H,34H,40H  
 DB 34H,88H,35H,36H,35H,85H,36H,33H  
 DB 36H,81H,37H,29H,37H,78H,38H,26H  
 DB 38H,74H,39H,22H,39H,71H,40H,19H  
 DB 40H,67H,41H,15H,41H,64H,42H,12H  
 DB 42H,60H,43H,08H,43H,56H,44H,05H  
 DB 44H,53H,45H,01H,45H,49H,45H,98H  
 DB 46H,46H,46H,94H,47H,42H,47H,91H  
 DB 48H,39H,48H,87H,49H,35H,49H,84H  
 DB 50H,32H,50H,80H,51H,28H,51H,76H  
 DB 52H,25H,52H,73H,53H,21H,53H,69H  
 DB 54H,18H,54H,66H,55H,14H,55H,62H  
 DB 56H,11H,56H,59H,57H,07H,57H,55H  
 DB 58H,04H,58H,52H,59H,00H,59H,48H  
 DB 59H,96H,60H,45H,60H,93H,61H,41H  
 DB 61H,89H,62H,38H,62H,86H,63H,34H



```

DB      63H,82H,64H,31H,64H,79H,65H,27H
DB      65H,75H,66H,24H,66H,72H,67H,20H
DB      67H,68H,68H,16H,68H,65H,69H,13H
DB      69H,61H,70H,09H,70H,58H,71H,06H
DB      71H,54H,72H,02H,72H,51H,72H,99H
DB      73H,47H,73H,95H,74H,44H,74H,92H
DB      75H,40H,75H,88H,76H,36H,76H,85H
DB      77H,33H,77H,81H,78H,29H,78H,78H
DB      79H,26H,79H,74H,80H,22H,80H,71H
DB      81H,19H,81H,67H,82H,15H,82H,64H
DB      83H,12H,83H,60H,84H,08H,84H,56H
DB      85H,05H,85H,53H,86H,01H,86H,49H
DB      86H,98H,87H,46H,87H,94H,88H,42H
DB      88H,91H,89H,39H,89H,87H,90H,35H
DB      90H,84H,91H,32H,91H,80H,92H,28H
DB      92H,76H,93H,25H,93H,73H,94H,21H
DB      94H,69H,95H,18H,95H,66H,96H,14H
DB      96H,62H,97H,11H,97H,59H,98H,07H
DB      98H,55H,99H,04H,99H,52H

```

```
FINMAIN      EQU
```

```

;SEGMENTOS ABSOLUTOS DE
;LAS RUTINAS DE INTERRUPCION
;INT0: Rutina que se ejecuta
;cuando hay una interrupción del RTC
      DEFSEG      INTOSEG, START=$1300
      SEG      INTOSEG
INT0:  DI
      PUSH      AF
      PUSH      BC
      PUSH      IY
      IN      A, (RTCINTE)
      IN      A, (RTCINTE)
      IN      A, (RTCINTE)
      LD      A, (ACUMFLG)
      CP      0

```

```

JR      Z,VETE                ;si no acumula salir
LD      A,(MEDIOS)
INC     A
LD      (MEDIOS),A
CP      2
JR      NZ,VETE
CALL    COMHFIN              ;Verificar hora final
LD      A,0
LD      (MEDIOS),A
LD      A,(UNSEG)
LD      C,A
LD      A,(SECOND)
INC     A
LD      (SECOND),A
CP      C                    ;Verifico si ya tengo M segundos trans-
JR      C,VETE                ;curridos. Si no lo tengo va a VETE.
LD      A,0                    ;vuelvo a cero el valor en
LD      (SECOND),A           ;SECOND.
LD      A,(NUMCAN)
LD      B,A
ACUM: LD      A,B
DEC     A
CALL    GETDAT
LD      IY,(POINTDT)
LD      A,(ADCOD)
LD      (IY),A
INC     IY
LD      (POINTDT),IY
DJNZ   ACUM
VETE   CALL    LEOHO
CALL    LEOFE
CALL    SWAPDSP              ;Causa visualización
LD      A,(CANAL)
CALL    GETDAT
CALL    DATGC
POP     IY
POP     BC

```

```
POP AF
EI
RETI
```

```
;Coge el dato del A/D del canal A
```

```
;y lo retorna en ADCOD
```

```
GETDAT:   PUSH AF
          OUT  (CAD),A           ;Selecciono el canal A del A/D.
          CALL RETAR           ;Necesario para dar tiempo
          IN   A,(CAD)         ;al A/D que termine de convertir.
          LD   (ADCOD),A
          POP  AF
          RET
```

```
;Compara Hora y fecha actual
```

```
;con la hora y fecha final
```

```
;y en caso de coincidir
```

```
;pone ACUMFLG en 0(desactiva la
;acumulación).
```

```
COMHFIN   PUSH AF
          PUSH BC
          LD   A,(FECHFIN) ;Lee FECHA final fijada
          LD   C,A
          IN   A,(RTCDSEM)
          IN   A,(RTCDDIA) ;Leo decenas de DIAS.
          AND  OFH
          CP   C           ;Compara ambas
          JR   NZ,SALCOMP
          LD   A,(FECHFIN+1)
          LD   C,A
          IN   A,(RTCUDIA) ;Leo unidad de DIAS.
          AND  OFH
          CP   C
          JR   NZ,SALCOMP
          LD   A,(FECHFIN+2)
          LD   C,A
          IN   A,(RTCDMES) ;Leo decenas de Meses.
```

```

AND    OFH
CP     C
JR     NZ,SALCOMP
LD     A,(FECHFIN+3)
LD     C,A
IN     A,(RTCUMES) ;Leo unidad de Meses.
AND    OFH
CP     C
JR     NZ,SALCOMP
LD     A,(HORAFIN) ;Lee hora final fijada
LD     C,A
IN     A,(RTCUDIA) ;Truco de lectura del RTC
IN     A,(RTCDHOR) ;Lee Decena de hora actual
AND    OFH
CP     C           ;Compara ambas
JR     NZ,SALCOMP
LD     A,(HORAFIN+1) .
LD     C,A
IN     A,(RTCUHOR) ;Lee Unidades de hora actual
AND    OFH
CP     C
JR     NZ,SALCOMP
LD     A,(HORAFIN+2) .
LD     C,A
IN     A,(RTCDMIN) ;Lee Decenas de minutos actual
AND    OFH
CP     C
JR     NZ,SALCOMP
LD     A,(HORAFIN+3)
LD     C,A
IN     A,(RTCUMIN) ;Lee Unidades de minutos actual
AND    OFH
CP     C
JR     NZ,SALCOMP
LD     A,0
LD     (ACUMFLG),A
LD     A,$30

```

```

        LD    (KEYFLAG),A
SALCOMP  POP   BC
        POP  AF
        RET

```

```

FININTO  EQU   $

```

```

;Rutina que se ejecuta
;cuando hay una interrupción
;causado por el teclado.

```

```

        DEFSEG      INT1SEG,START=$1400

```

```

        SEG    INT1SEG

```

```

INT1:  DI

```

```

        PUSH AF

```

```

        PUSH BC

```

```

        PUSH HL

```

```

        PUSH IX

```

```

        IN   A,(KEY)           ;Leo el código de

```

```

        AND  1FH           ;la tecla presionada y lo corta.

```

```

        LD   (KEYCOD),A

```

```

        LD   A,(INTCNT)

```

```

        INC  A

```

```

        LD   (INTCNT),A

```

```

        CP   1           ;Numero que evita rebotes.

```

```

        JR   NC,PON2

```

```

        LD   A,(EVITREB)

```

```

        LD   C,A

```

```

        LD   A,(KEYCOD)

```

```

        CP   C

```

```

        JR   Z,VET1

```

```

PON2:  CALL  BEEP

```

```

        LD   HL,KEYBUF

```

```

        LD   B,0

```

```

        LD   A,(DIGCONT)

```

```

        LD   C,A

```

```

        ADD  HL,BC

```

```

        LD   A,(KEYCOD)

```

```

LD    (KEYFLAG),A
LD    (HL),A
CALL  BINASC
LD    A,(DSPENA)
CP    0
JR    Z,PON3
LD    A,L
CALL  DSPCAR
CALL  SWAPDSP           ;Visualiza la tecla
                        ;presionada.
PON3  LD    A,(DIGCONT)
      INC  A
      CP   9
      JR   C,SIGDIG
      LD   A,0
SIGDIG LD    (DIGCONT),A
      LD   A,0
      LD   (INTCNT),A
      LD   A,(KEYCOD)
      LD   (EVITREB),A
VET1  POP  IX
      POP  HL
      POP  BC
      POP  AF
      EI
      RETI
FININT1 EQU  $

;NO SE LE DA USO
;COMO INTERRUPCION
      DEFSEG      INT2SEG,START=$1500
      SEG  INT2SEG
INT2  DI
      EI
      RETI
FININT2 EQU  $

```

END

## 5.7 MAPA DE DIRECCIONES DEL PROGRAMA ADATOC

AVLINK ---- LOAD MAP

For: Avocet            Z80 Assembler v1.12, #01103

### RELOCATED SEGMENTS - CLASS 'I'

SEGMENT NAME	START	STOP	LENGTH	ovl/cat	
RTCSEG	0000	000f	0010	Concat	Defined
SIOSEG	0060	0061	0002	Concat	Defined
CTCSEG	0064	0067	0004	Concat	Defined
CADSEG	0068	0068	0001	Concat	Defined
KEYSEG	0070	0070	0001	Concat	Defined
LCDSEG	0074	0076	0003	Concat	Defined

### RELOCATED SEGMENTS - CLASS 'M'

SEGMENT NAME	START	STOP	LENGTH	ovl/cat	def/undef
<b>RESETSEG</b>	0000	023c	<b>023d</b>	Concat	Defined
<b>TOOLSSEG</b>	026b	03ef	<b>0185</b>	Concat	Defined
<b>LOADSEG</b>	0400	0582	<b>0183</b>	Concat	Defined
<b>LCDTSEG</b>	0600	1258	<b>0c59</b>	Concat	Defined
<b>INT0SEG</b>	1300	13e3	<b>00e4</b>	Concat	Defined
<b>INT1SEG</b>	1400	1466	<b>0067</b>	Concat	Defined
<b>INT2SEG</b>	1500	1503	<b>0004</b>	Concat	Defined
<b>INT3SEG</b>	1600	1681	<b>0082</b>	Concat	Defined

### ABSOLUTE SEGMENTS

SEGMENT NAME	START	STOP	LENGTH	ovl/cat	def/undef	class
DATASEG	e008	e0f4	00ed	Concat	Defined	M
VTABSEG	e000	e007	0008	Concat	Defined	M
STSEG	f900	ffff	06ff	Concat	Defined	M

ZERO LENGTH SEGMENTS

SEGMENT	START
CODE	0000
DATA	0000
IOSPACE	0000

SYMBOLS -- CLASS 'I'

SYMBOL NAME	START	SYMBOL'S SEGMENT	def/undef
RTCTEST	0000	RTCSEG	Defined
RTCDECS	0001	RTCSEG	Defined
RTCUSEG	0002	RTCSEG	Defined
RTCDSEG	0003	RTCSEG	Defined
RTCUMIN	0004	RTCSEG	Defined
RTCDMIN	0005	RTCSEG	Defined
RTCUHOR	0006	RTCSEG	Defined
RTCDHOR	0007	RTCSEG	Defined
RTCUDIA	0008	RTCSEG	Defined
RTCD DIA	0009	RTCSEG	Defined
RTCDSEM	000a	RTCSEG	Defined
RTCUMES	000b	RTCSEG	Defined
RTCDMES	000c	RTCSEG	Defined
RTCYEAR	000d	RTCSEG	Defined
RTCINIP	000e	RTCSEG	Defined
RTCINTE	000f	RTCSEG	Defined
URTDAT	0060	SIOSEG	Defined
URTCNT	0061	SIOSEG	Defined
CTC0	0064	CTCSEG	Defined
CTC1	0065	CTCSEG	Defined
CTC2	0066	CTCSEG	Defined
CTC3	0067	CTCSEG	Defined
CAD	0068	CADSEG	Defined
KEY	0070	KEYSEG	Defined
LCDC	0074	LCDSEG	Defined
LCDI	0075	LCDSEG	Defined
LCDD	0076	LCDSEG	Defined



## SYMBOLS -- CLASS 'M'

SYMBOL NAME	START	SYMBOL'S SEGMENT	def/undef
RESET	0000	RESETSEG	Defined
COLD	0019	RESETSEG	Defined
MEMRET	001c	RESETSEG	Defined
HOT	0040	RESETSEG	Defined
LOOPRET	0042	RESETSEG	Defined
FINRES	0056	RESETSEG	Defined
CONFIG	005c	RESETSEG	Defined
INITIO	0088	RESETSEG	Defined
SETVECT	009e	RESETSEG	Defined
INISIO	00b9	RESETSEG	Defined
OUTSIO	00d5	RESETSEG	Defined
INICTC0	00d9	RESETSEG	Defined
INICTC1	00e8	RESETSEG	Defined
INICTC3	00f7	RESETSEG	Defined
INIRTC	0102	RESETSEG	Defined
SETRTC	0120	RESETSEG	Defined
INIRTCI	0131	RESETSEG	Defined
INIKEY	013a	RESETSEG	Defined
INILCD	0151	RESETSEG	Defined
INITMEM	0185	RESETSEG	Defined
LAS0	018c	RESETSEG	Defined
LAS02	01bb	RESETSEG	Defined
ERRMEM	01e6	RESETSEG	Defined
COLDPR	01f2	RESETSEG	Defined
COLDCAR	020a	RESETSEG	Defined
COLDCA1	021b	RESETSEG	Defined
ERRMMSG	022c	RESETSEG	Defined
FINREST	023d	RESETSEG	Defined
ASCBIN	026b	TOOLSSEG	Defined
BINASC	0285	TOOLSSEG	Defined
BN2DEC	02a9	TOOLSSEG	Defined
DISPLAY	0320	TOOLSSEG	Defined
DSPCAD	0332	TOOLSSEG	Defined
DSPCAR	0347	TOOLSSEG	Defined

LINE1	0359	TOOLSSEG	Defined
LINE2	0366	TOOLSSEG	Defined
ERASE	0373	TOOLSSEG	Defined
RETAR	0387	TOOLSSEG	Defined
ENADSP	0395	TOOLSSEG	Defined
DISDSP	039d	TOOLSSEG	Defined
RESETKE	03a5	TOOLSSEG	Defined
SWAPDSP	03b5	TOOLSSEG	Defined
FINTOOL	03f0	TOOLSSEG	Defined
LOAD	0400	LOADSEG	Defined
TRADUC	0477	LOADSEG	Defined
SALIR	04f3	LOADSEG	Defined
FIN	04fc	LOADSEG	Defined
ERRTRD	0503	LOADSEG	Defined
NULMSG	0572	LOADSEG	Defined
FINLOAD	0583	LOADSEG	Defined
MAIN	0600	LCDTSEG	Defined
ACTUA	09ff	LCDTSEG	Defined
IDLE	0e79	LCDTSEG	Defined
BEEP	0ed4	LCDTSEG	Defined
DEMORA	0ee9	LCDTSEG	Defined
FINMAIN	1259	LCDTSEG	Defined
INT0	1300	INT0SEG	Defined
FININT0	13e4	INT0SEG	Defined
INT1	1400	INT1SEG	Defined
FININT1	1467	INT1SEG	Defined
INT2	1500	INT2SEG	Defined
FININT2	1504	INT2SEG	Defined
INT3	1600	INT3SEG	Defined
FLAG	162f	INT3SEG	Defined
ERRSER	1637	INT3SEG	Defined
ERRBUF	1659	INT3SEG	Defined
FININT3	1682	INT3SEG	Defined
VECTTAB	e000	ABSOLUTE	Defined
FINVEC	e008	ABSOLUTE	Defined
PATRON	e008	ABSOLUTE	Defined
KEYFLAG	e00a	ABSOLUTE	Defined

BUFFER	e00b	ABSOLUTE	Defined
POINTER	e05b	ABSOLUTE	Defined
TRADFLG	e05c	ABSOLUTE	Defined
ERRCOD	e05d	ABSOLUTE	Defined
EXEADD	e05e	ABSOLUTE	Defined
ADDFLG	e060	ABSOLUTE	Defined
DECIDT	e061	ABSOLUTE	Defined
POINTDT	e065	ABSOLUTE	Defined
MEDIOS	e067	ABSOLUTE	Defined
SECOND	e068	ABSOLUTE	Defined
HORADT	e069	ABSOLUTE	Defined
FECHDT	e074	ABSOLUTE	Defined
YEARDT	e07d	ABSOLUTE	Defined
TEMPDT	e07f	ABSOLUTE	Defined
DIGCONT	e086	ABSOLUTE	Defined
KEYBUF	e087	ABSOLUTE	Defined
KEYCOD	e091	ABSOLUTE	Defined
CANAL	e092	ABSOLUTE	Defined
ADCOD	e093	ABSOLUTE	Defined
UNSEG	e094	ABSOLUTE	Defined
EVITREB	e095	ABSOLUTE	Defined
INTCNT	e096	ABSOLUTE	Defined
DSPENA	e097	ABSOLUTE	Defined
ACUMFLG	e098	ABSOLUTE	Defined
INIDHOR	e099	ABSOLUTE	Defined
INIUHOR	e09a	ABSOLUTE	Defined
INIDMIN	e09b	ABSOLUTE	Defined
INIUMIN	e09c	ABSOLUTE	Defined
INIDSEG	e09d	ABSOLUTE	Defined
INIUSEG	e09e	ABSOLUTE	Defined
INIDMES	e09f	ABSOLUTE	Defined
INIUMES	e0a0	ABSOLUTE	Defined
INIDDIA	e0a1	ABSOLUTE	Defined
INIUDIA	e0a2	ABSOLUTE	Defined
HORAFIN	e0a3	ABSOLUTE	Defined
FECHFIN	e0a7	ABSOLUTE	Defined
BUFPTR	e0ab	ABSOLUTE	Defined

CURLEN	e0ad	ABSOLUTE	Defined
NGFLAG	e0ae	ABSOLUTE	Defined
ASCBUF	e0af	ABSOLUTE	Defined
NUMCAN	e0b6	ABSOLUTE	Defined
SGUNDOS	e0b7	ABSOLUTE	Defined
CALLBUF	e0bd	ABSOLUTE	Defined
MEMORY	e0c3	ABSOLUTE	Defined
POINTME	e0c7	ABSOLUTE	Defined
DSPPTR	e0c9	ABSOLUTE	Defined
LIN1BUF	e0cb	ABSOLUTE	Defined
LIN2BUF	e0dd	ABSOLUTE	Defined
OFFSET	e0f3	ABSOLUTE	Defined
STACK	ffff	ABSOLUTE	Defined

SYMBOLS -- CLASS 'N'

SYMBOL NAME	START	SYMBOL'S SEGMENT	def/undef
BASEDAT	4000	ABSOLUTE	Defined

SYMBOLS BY NAME

ACTUA	09ff	LCDTSEG	Defined
ACUMFLG	e098	ABSOLUTE	Defined
ADCOD	e093	ABSOLUTE	Defined
ADDFLG	e060	ABSOLUTE	Defined
ASCBIN	026b	TOOLSSEG	Defined
ASCBUF	e0af	ABSOLUTE	Defined
BASEDAT	4000	ABSOLUTE	Defined
BEEP	0ed4	LCDTSEG	Defined
BINASC	0285	TOOLSSEG	Defined
BN2DEC	02a9	TOOLSSEG	Defined
BUFFER	e00b	ABSOLUTE	Defined
BUFPTR	e0ab	ABSOLUTE	Defined
CAD	0068	CADSEG	Defined
CALLBUF	e0bd	ABSOLUTE	Defined
CANAL	e092	ABSOLUTE	Defined
COLD	0019	RESETSEG	Defined

COLDCA1	021b	RESETSEG	Defined
COLDCAR	020a	RESETSEG	Defined
COLDPR	01f2	RESETSEG	Defined
CONFIG	005c	RESETSEG	Defined
CTC0	0064	CTCSEG	Defined
CTC1	0065	CTCSEG	Defined
CTC2	0066	CTCSEG	Defined
CTC3	0067	CTCSEG	Defined
CURLEN	e0ad	ABSOLUTE	Defined
DECIDT	e061	ABSOLUTE	Defined
DEMORA	0ee9	LCDTSEG	Defined
DIGCONT	e086	ABSOLUTE	Defined
DISDSP	039d	TOOLSSEG	Defined
DISPLAY	0320	TOOLSSEG	Defined
DSPCAD	0332	TOOLSSEG	Defined
DSPCAR	0347	TOOLSSEG	Defined
DSPENA	e097	ABSOLUTE	Defined
DSPPTR	e0c9	ABSOLUTE	Defined
ENADSP	0395	TOOLSSEG	Defined
ERASE	0373	TOOLSSEG	Defined
ERRBUF	1659	INT3SEG	Defined
ERRCOD	e05d	ABSOLUTE	Defined
ERRMEM	01e6	RESETSEG	Defined
ERRMSG	022c	RESETSEG	Defined
ERRSER	1637	INT3SEG	Defined
ERRTRD	0503	LOADSEG	Defined
EVITREB	e095	ABSOLUTE	Defined
EXEADD	e05e	ABSOLUTE	Defined
FECHDT	e074	ABSOLUTE	Defined
FECHFIN	e0a7	ABSOLUTE	Defined
FIN	04fc	LOADSEG	Defined
FININT0	13e4	INT0SEG	Defined
FININT1	1467	INT1SEG	Defined
FININT2	1504	INT2SEG	Defined
FININT3	1682	INT3SEG	Defined
FINLOAD	0583	LOADSEG	Defined
FINMAIN	1259	LCDTSEG	Defined

FINRES	0056	RESETSEG	Defined
FINREST	023d	RESETSEG	Defined
FINTOOL	03f0	TOOLSSEG	Defined
FINVEC	e008	ABSOLUTE	Defined
FLAG	162f	INT3SEG	Defined
HORADT	e069	ABSOLUTE	Defined
HORAFIN	e0a3	ABSOLUTE	Defined
HOT	0040	RESETSEG	Defined
IDLE	0e79	LCDTSEG	Defined
INICTC0	00d9	RESETSEG	Defined
INICTC1	00e8	RESETSEG	Defined
INICTC3	00f7	RESETSEG	Defined
INIDDIA	e0a1	ABSOLUTE	Defined
INIDHOR	e099	ABSOLUTE	Defined
INIDMES	e09f	ABSOLUTE	Defined
INIDMIN	e09b	ABSOLUTE	Defined
INIDSEG	e09d	ABSOLUTE	Defined
INIKEY	013a	RESETSEG	Defined
INILCD	0151	RESETSEG	Defined
INIRTC	0102	RESETSEG	Defined
INIRTCI	0131	RESETSEG	Defined
INISIO	00b9	RESETSEG	Defined
INITIO	0088	RESETSEG	Defined
INITMEM	0185	RESETSEG	Defined
INIUDIA	e0a2	ABSOLUTE	Defined
INIUHOR	e09a	ABSOLUTE	Defined
INIUMES	e0a0	ABSOLUTE	Defined
INIUMIN	e09c	ABSOLUTE	Defined
INIUSEG	e09e	ABSOLUTE	Defined
INT0	1300	INT0SEG	Defined
INT1	1400	INT1SEG	Defined
INT2	1500	INT2SEG	Defined
INT3	1600	INT3SEG	Defined
INTCNT	e096	ABSOLUTE	Defined
KEY	0070	KEYSEG	Defined
KEYBUF	e087	ABSOLUTE	Defined
KEYCOD	e091	ABSOLUTE	Defined

KEYFLAG	e00a	ABSOLUTE	Defined
LASO	018c	RESETSEG	Defined
LASO2	01bb	RESETSEG	Defined
LCDC	0074	LCDSEG	Defined
LCDD	0076	LCDSEG	Defined
LCDI	0075	LCDSEG	Defined
LIN1BUF	e0cb	ABSOLUTE	Defined
LIN2BUF	e0dd	ABSOLUTE	Defined
LINE1	0359	TOOLSSEG	Defined
LINE2	0366	TOOLSSEG	Defined
LOAD	0400	LOADSEG	Defined
LOOPRET	0042	RESETSEG	Defined
MAIN	0600	LCDTSEG	Defined
MEDIOS	e067	ABSOLUTE	Defined
MEMORY	e0c3	ABSOLUTE	Defined
MEMRET	001c	RESETSEG	Defined
NGFLAG	e0ae	ABSOLUTE	Defined
NULMSG	0572	LOADSEG	Defined
NUMCAN	e0b6	ABSOLUTE	Defined
OFFSET	e0f3	ABSOLUTE	Defined
OUTSIO	00d5	RESETSEG	Defined
PATRON	e008	ABSOLUTE	Defined
POINTDT	e065	ABSOLUTE	Defined
POINTER	e05b	ABSOLUTE	Defined
POINTME	e0c7	ABSOLUTE	Defined
RESET	0000	RESETSEG	Defined
RESETKE	03a5	TOOLSSEG	Defined
RETAR	0387	TOOLSSEG	Defined
RTCDDIA	0009	RTCSEG	Defined
RTCDECS	0001	RTCSEG	Defined
RTCDHOR	0007	RTCSEG	Defined
RTCDMES	000c	RTCSEG	Defined
RTCDMIN	0005	RTCSEG	Defined
RTCDSEG	0003	RTCSEG	Defined
RTCDSEM	000a	RTCSEG	Defined
RTCINIP	000e	RTCSEG	Defined
RTCINTE	000f	RTCSEG	Defined

RTCTEST	0000	RTCSEG	Defined
RTCUDIA	0008	RTCSEG	Defined
RTCUHOR	0006	RTCSEG	Defined
RTCUMES	000b	RTCSEG	Defined
RTCUMIN	0004	RTCSEG	Defined
RTCUSEG	0002	RTCSEG	Defined
RTCYEAR	000d	RTCSEG	Defined
SALIR	04f3	LOADSEG	Defined
SECOND	e068	ABSOLUTE	Defined
SETRTC	0120	RESETSEG	Defined
SETVECT	009e	RESETSEG	Defined
SGUNDOS	e0b7	ABSOLUTE	Defined
STACK	ffff	ABSOLUTE	Defined
SWAPDSP	03b5	TOOLSSEG	Defined
TEMPDT	e07f	ABSOLUTE	Defined
TRADFLG	e05c	ABSOLUTE	Defined
TRADUC	0477	LOADSEG	Defined
UNSEG	e094	ABSOLUTE	Defined
URTCNT	0061	SIOSEG	Defined
URTDAT	0060	SIOSEG	Defined
VECTTAB	e000	ABSOLUTE	Defined
YEARDT	e07d	ABSOLUTE	Defined

MODULE DATA

MODULE:       resetz80   NUMBER OF SEGMENTS: 13

SEGMENT NAME	OFFSET	STOP	LENGTH
CODE	0000	0000	0000
DATA	0000	0000	0000
IOSPACE	0000	0000	0000
SIOSEG	0000	0001	0002
CTCSEG	0000	0003	0004
LCDSEG	0000	0002	0003
CADSEG	0000	0000	0001
RTCSEG	0000	000f	0010
KEYSEG	0000	0000	0001



STSEG	0000	06fe	06ff
VTABSEG	0000	0007	0008
DATASEG	0000	00ec	00ed
RESETSEG	0000	023c	023d

MODULE: toolsz80 NUMBER OF SEGMENTS: 4

SEGMENT NAME	OFFSET	STOP	LENGTH
CODE	0000	0000	0000
DATA	0000	0000	0000
IOSPACE	0000	0000	0000
TOOLSSEG	0000	0184	0185

MODULE: loadz80 NUMBER OF SEGMENTS: 5

SEGMENT NAME	OFFSET	STOP	LENGTH
CODE	0000	0000	0000
DATA	0000	0000	0000
IOSPACE	0000	0000	0000
LOADSEG	0000	0182	0183
INT3SEG	0000	0081	0082

MODULE: Lcdtz80 NUMBER OF SEGMENTS: 7

SEGMENT NAME	OFFSET	STOP	LENGTH
CODE	0000	0000	0000
DATA	0000	0000	0000
IOSPACE	0000	0000	0000
LCDTSEG	0000	0c58	0c59
INT0SEG	0000	00e3	00e4
INT1SEG	0000	0066	0067
INT2SEG	0000	0003	0004

## 5.8 ACERCA DE LA CONSTRUCCION DEL ADATOC

Mencionando primero la construcción de tarjetas para circuito impreso, hubieron muchas etapas intermedias hasta llegar a la construcción

óptima.

Primero se diseñó el conexionado de pistas entre los componentes, usando primeramente el **ORCAD** para el diseño esquemático de las conexiones de los componentes y luego un paquete para computadora compatible llamado **SMARTWORK** para el diseño de las pistas. Una vez terminado esto, después de haberlo optimizado al máximo se procedió a la construcción de las tarjetas.

Inicialmente empezamos construyendo tarjetas nosotros, empleando para ello la técnica de Fotorevelado(usando fotolito positivo), pero al final el modelo más óptimo se mandó a que lo construya una empresa. El equipo consta de dos tarjetas de circuito impreso(doble cara), la serigrafía se encuentra en los Apéndices de esta Tesis.

Terminada las tarjetas empieza el montaje de los componentes, colocando sockets para cada circuito integrado.

El proceso de grabación de memorias EPROMs lo tuvimos que realizar muchas veces hasta llegar al programa óptimo.

Se compró la caja hecha en el mercado. Se hicieron los agujeros en los lados de la caja para los conectores de los ocho canales, batería BACKUP, serial, interruptor de encendido, indicador de encendido.

El teclado fue construido extrayendo las teclas de teclados más grandes que estaban en desuso, para ello también se tuvo que hacer la tarjeta de circuito impreso en una cara.

## CAPITULO VI: PROGRAMAS PARA COMPUTADORA

### 6.1 ALGORITMO DEL PROGRAMA ADCYA

Este programa entre otras cosas saca promedios de los datos.

-Tiene una serie de posibilidades que se escogen según los parámetros al ejecutar el programa.

-El programa comienza estableciendo primero valores por defecto.

-Por defecto el programa no genera reporte para hoja de calculo.

-Análisis de línea de comando todas las opciones que se pone con el nombre del programa.

-Se consideran un montón de procedimientos y funciones reunidos en un objeto TDATAKOM.

-Puedo utilizar cada uno de los métodos de un objeto, poniendo el nombre como junto al método a utilizar(procedimiento o función, variable).

-Por defecto:

PCOM=1

BPS= 9600

BP=1

BD=8

PAR="P"

-De aquí pasa a hacer un análisis de línea de comando.

-Si se inserta un valor inadecuado llama a HELP.

-HELP te muestra en pantalla las opciones que deben salir.

-El programa verifica si he insertado un nombre de archivo. Sino se inserto le asigna un nombre por defecto(adcy.txt).

-Lee los archivos DAT para convertir o activa el puerto serie si se ha ordenado para ello.

-Imprime cabecera en pantalla.

-Coge dato de la hora, el mes, segundos.

-Se van escribiendo el inicio de acumulación, el # de canales.

-El programa se encarga de hacer la corrección del tiempo.

-Luego lectura de datos de c/uno de los canales acumulados.

Calculo de la hora real, y mientras almacena valores máximos y mínimos.

## 6.2 PROGRAMA QUE OBTIENE PROMEDIOS DE LOS DATOS

Este programa además de calcular promedios desde un archivo con extensión dat, muestra en pantalla los datos en forma de columnas y crea un archivo con extensión txt que es importable a la hoja de cálculo QPRO.

En este programa uso algunas unidades como Comport, además de otros que son comerciales y han sido adquiridos en el mercado local y sólo se usan directamente, por lo tanto no serán listados.

**Program ADCYA;**

```
Uses Crt,Dos,ComPort,Utills,tools;
```

```
Var
```

```
ComParam : TComParam;
```

```
ComStat : TComStat;
```

```
Comu:TDataCom;
```

```
err:byte;
```

```
f:text;
```

```
D:file of Byte;
```

```
Dosparam:string;
```

```
name,Fdata:PathStr;
```

```
cuent,posi:byte;
```

```
error:integer;
```

```

    ver,hoja,fin,Comm:boolean;
    Dia, Mes, Ano, DiaS, Reuval,NumDat: word;
    HH,MM,SS,timeinc,dat,NumCan: byte;
    DatCan:Array[0..7] of word;
    RealInc,SSReal:real;

const
    max      :Array[0..7] of byte=($00,$00,$00,$00,$00,$00,$00,$00);
    min      :Array[0..7] of byte=($ff,$ff,$ff,$ff,$ff,$ff,$ff,$ff);
    PromCnt:Array[0..7] of word=(1,1,1,1,1,1,1,1);

Procedure Abortar;
begin
{$I-} Close(f);
    if not comm then Close(D);
{$I+} err:=IoResult;
    Writeln('Programa Abortado');
    Halt(2);
end;

function GetDat:byte;
var Dato:byte;
begin
    GetDat:=0;Dato:=0;
    if fin then exit;
    If Comm then
        Repeat
            Dato:=Ord(Comu.Getdat);
            if keypressed then fin:=readkey=#27;
        Until Comu.GetAv or fin
        else if not Eof(D) then Read(D,Dato);
        GetDat:=Dato;
end;

function Temp(Dat:real):real;
begin
    Temp:=0.482353*Dat-23;

```

```

end;

procedure ReadCan;
begin
  if fin then exit;
  For cuent:=NumCan downto 1 do
  begin
    Dat:=GetDat;
    if Dat=0 then begin fin:=true;exit end;
    If Dat>max[cuent-1] then max[cuent-1]:=Dat;
    If Dat<min[cuent-1] then min[cuent-1]:=Dat;
    if PromCnt[cuent-1]=1 then DatCan[cuent-1]:=0;
    INC(DatCan[cuent-1],dat);
    INC(PromCnt[cuent-1]);
    if PromCnt[cuent-1]>ReuVal then PromCnt[cuent-1]:=1;
    INC(NumDat);
  end;
end;

procedure Help;
Begin
  WriteLn;

  WriteLn('ADCYA[/Afilename][/Cn][/Bnnnn][/Pn[/Dn][/O;/E][/H]
[/Rnnnn]');
  WriteLn('ADCYA [/Afilename] [/FFilename] [/H] [/Rnnnn]');
  WriteLn;
  WriteLn(' /Afilename:FileName =Encaminamiento yel nombredel
Archivo. ');
  WriteLn(
Nombre de Archivo por defecto
ADCYA.TXT');
  WriteLn(' /FFilename: FileName - Encaminamientoy nombre del
Archivo que sirve como ');
  WriteLn('
fuente alternativa dedatos (por
defecto ADCYA.DAT');
  WriteLn(' /Cn : n - (1,2,3,4) Puerto Serie COMn a usar. ');
  WriteLn('
Puerto serie por defecto COM1. ');

```

```

        WriteLn('/Bnnnn:nnnn=(300,600,1200,2400,4800,9600)Velocidad
de transmisión');
        WriteLn('
                en baudios. Valor por defecto 9600');
        WriteLn(' /Pn      : n      - (1,2) Bits de Parada. ');
        WriteLn('
                Valor por defecto 1. ');
        WriteLn(' /Dn      : n      - (7,8) Bits de datos. ');
        WriteLn('
                Valor por defecto 8. ');
        WriteLn('/N      :Sin Paridad,no puedeutilizarse juntocon /Eni
/O. ');

        WriteLn(' /O      : ParidadImpar (Odd), nopuede utilizarsejunto
con /E. ');

        WriteLn('/E      : ParidadPar (Even), nopuede utilizarse junto
con /O. ');

        WriteLn('
                Por defecto paridad par. ');
        WriteLn(' /H      : Grabar formato exportable a hoja de cálculo. ');
        WriteLn('/Rnnnn :nnnn =(1..9999) Reunirypromediar nnnndatos
consecutivos. ');
        WriteLn('
                Valor por defecto 1. ');
        WriteLn; Halt(1);
end;

Begin
    ver:=true;Comm:=true;
    Hoja:=false;
    fin:=false;
    error:=0; ReuVal:=1;NumDat:=0;
    name:='';FData:='';
    With ComParam do begin
        PCOM := 1;
        BPS  := 9600;
        BP   := 1;
        BD   := 8;
        PAR  := 'P';
        If ParamCount>0 then
            For cuent:=1 to ParamCount do begin
                DosParam:=UPStr(ParamStr(Cuent));
                posi:=Pos('/',DosParam);

```

```

        if posi=0 then Help;
        Case DosParam[Posi+1] of
            'R': Val(copy(dosparam,posi+2,4),ReuVal,error);
            'C': Val(copy(dosparam,posi+2,1),PCOM ,error);
            'B': Val(copy(dosparam,posi+2,4),BPS ,error);
            'P': Val(copy(dosparam,posi+2,1),BP ,error);
            'D': Val(copy(dosparam,posi+2,1),BD ,error);
            'A': name
:=copy(dosparam,posi+2,length(Dosparam));
            'E': PAR:='P';
            'O': PAR:='I';
            'N': PAR:='N';
            'V': Ver :=true;
            'H': Hoja:=true;
            'R': Val(copy(dosparam,posi+2,4),reuval,error);
            'F': begin
                    Comm:=false;

FDData:=copy(dosparam,posi+2,length(Dosparam));
                    end;
            else help;
            end;
            if error<>0 then Help;
            end;
        end;
        if name= '' then assign(f,'ADCYA.txt') else assign(f,name);
        if FDData='' then assign(D,'ADCYA.DAT') else assign(D,FDData);
        if not Comm then
        begin
{$I-} reset(D);
{$I+} err:=IoResult;
        if err<>0 then begin WriteLn(DosErrorStr(err));Abortar;end;
        Writeln('Fuente: ',FDData);
        end;
        Rewrite(f);
        if Comm then begin Comu.Init; Comu.ActCom(ComParam) endelse
reset(D);

```



```

Writeln(f,'UNIVERSIDAD NACIONAL DE INGENIERIA');
Writeln(f,'FACULTAD DE CIENCIAS - ESCUELA PROFESIONAL DE FISICA');
Writeln(f,'LABORATORIO DE ENERGIA SOLAR');
Writeln(f);
GetDate(Ano,Mes,Dia,DiaS);
Writeln(f,'Fecha: ',Dia,' de ',MesStr[Mes],' de ',Ano);
        ifReuVal>1thenWriteLn(f,'Promediocada',ReuVal,'datos.')else
Writeln(f);
        if ver then
        begin
{
        With ComParam,ComStat do
        begin
            Comu.GetStat(ComParam,ComStat);
            Writeln('COM',PCOM, : ',BPS,',',',PAR,',',BD,',',BP);
            Writeln('INT=',INT,',DTR=',DTR,',RTS=',RTS);
        end;
}

        writeln;
        Writeln('UNIVERSIDAD NACIONAL DE INGENIERIA');
        Writeln('FACULTAD DE CIENCIAS - ESCUELA PROFESIONAL DE FISICA');
        Writeln('LABORATORIO DE ENERGIA SOLAR');
        Writeln;
        Writeln('Fecha: ',Dia,' de ',MesStr[Mes],' de ',Ano);
                ifReuVal>1thenWriteLn('Promediocada',ReuVal,'datos.')else
WriteLn;

        end;
        HH:=Getdat*10;HH:=HH+GetDat;
        MM:=Getdat*10;MM:=MM+GetDat;
        SS:=Getdat*10;SS:=SS+GetDat; if fin then abortar;
        SSReal:=SS;
                Write(f,'Inicio                de                acumulación:
',HH,':',TwoDigit(MM,true),':',TwoDigit(SS,true));
                if ver then Write('Inicio                de                acumulación:
',HH,':',TwoDigit(MM,true),':',TwoDigit(SS,true));
        MES:=Getdat*10;MES:=MES+GetDat;
        DIA:=Getdat*10;DIA:=DIA+GetDat;

```

```

Writeln(f, ' - ', DIA, ' de ', MesStr[mes]);
if ver then Writeln(' - ', DIA, ' de ', MesStr[mes]);
NumCan:=GetDat;
TimeInc:=GetDat; if fin then Abortar;
RealInc:=TimeInc*(1.0332);
Writeln(f, 'Número de canales acumulados: ', NumCan);
Writeln(f, 'Intervalo de tiempo en toma de datos: ', RealInc:6:2, '
segundos. ');
Writeln(f);
Write(f, ' HORA ');
For cuent:=1 to NumCan do Write(f, ' CANAL',cuent-1);
Writeln(f, ' °C ');
if ver then
begin
Writeln('Número de canales acumulados: ', NumCan);
Writeln('Intervalo de tiempo en toma de datos: ', RealInc:6:2, '
segundos . ');
Writeln;
Write(' HORA ');
For cuent:=1 to NumCan do Write(' CANAL',cuent-1);
Writeln(' °C');;
end;
Repeat
ReadCan;
IF (not fin) and (PromCnt[0]=1) then
begin
if ReuVal=1 then SSReal:=SSReal+RealInc else
SSReal:=SSReal+(RealInc*ReuVal/2);
if SSReal>59 then
begin
MM:=MM+Trunc(SSReal/60);
SSReal:=Frac(SSReal/60)*60;
end;
SS:=Trunc(SSReal);
if ReuVal<>1 then SSReal:=SSReal+(RealInc*ReuVal/2);
while MM>59 do begin inc(HH); MM:=MM-60; end;
while HH>23 do HH:=HH-24;

```

```

Write(f,TwoDigit(HH,true),':',TwoDigit(MM,true),':',TwoDigit(SS,true));
      For   cuent:=1   to   NumCan   do   Write(f,'
',Temp(DatCan[cuent-1]/ReuVal):7:2);
      Writeln(f);
      if ver then
      begin

Write(TwoDigit(HH,true),':',TwoDigit(MM,true),':',TwoDigit(SS,true));
      For   cuent:=1   to   NumCan   do   Write('
',Temp(DatCan[cuent-1]/ReuVal):7:2);
      Writeln;
      end;
      end;
      Until keyPressed or fin;
      if not hoja then
      begin
      Writeln(f);
      Writeln(f,'CANAL MAXIMA MEDIA MINIMA °C');
      Writeln(f,'-----');
      For   cuent:=1 to   NumCan do Writeln(f,'   ',cuent-1,'
',Temp(Max[cuent-1]):5:2,'
(Temp(Max[cuent-1])+Temp(Min[cuent-1]))/2:5:2,'   ',Temp(Min[cuent-1]):5:2);
      writeln(f);
      writeln(f,NumDat,' Datos recibidos. ');
      end;
      if ver then
      begin
      Writeln;
      Writeln('CANAL MAXIMA MEDIA MINIMA °C');
      Writeln('-----');
      For   cuent:=1 to   NumCan do Writeln('   ',cuent-1,'
',Temp(Max[cuent-1]):5:2,'
(Temp(Max[cuent-1])+Temp(Min[cuent-1]))/2:5:2,'   ',Temp(Min[cuent-1]):5:2);
      end;

```

```

writeln;
writeln(NumDat,' Datos recibidos.');
```

writeln;

```

if comm then
begin
    err:=Comu.GetErr;
    writeln(err,':',Comu.GetErrStr(err));
    Comu.Done;
end else Close(D);
close(f);
End.
```

### **6.3 ALGORITMO DEL PROGRAMA ADCYACOM**

-Establece algunos valores por defecto.

-Verifica líneas de comando para alterar los valores por defecto según información ingresada como parámetros.

-En caso de algún error en el ingreso de los parámetros llama a un procedimiento HELP que muestra el uso correcto de los valores.

-Analiza si le he cambiado de nombre. Si no es así le asigna un nombre por defecto "ADCYA.DAT".

-Abre un archivo para escritura.

-Luego se activa el puerto serie.

-Muestro en pantalla los parámetros que estoy utilizando, con el fin de verificar.

-Luego empieza a recibir información del puerto serie si esta disponible.

-Si el dato es válido se incrementa una variable el cual contiene la cantidad de datos que va llegando.

- Escribo en el archivo el dato que ha llegado.
- Se visualiza el dato en hexadecimal.
- Se para el programa si recibe un "0" después del dato #14.
- Verifica si ha ocurrido un error serial.
- Desactiva el sistema serial y cierra el archivo.

#### 6.4 PROGRAMA DE COMUNICACION SERIAL

Este programa recepciona los datos en la computadora enviados por el ADATOC y además crea un archivo con extensión dat que es usado por el programa anterior para cálculo de promedios.

##### **Program ADCYACOM;**

```
Uses Crt,Dos,ComPort,Utills;
```

```
Var c : Char;
    ComParam : TComParam;
    ComStat : TComStat;
    Comu:TDataCom;
    err:byte;
    f:text;
    Dosparam:string;
    name:PathStr;
    cuent, posi:byte;
    error:integer;
    ver:boolean;
    NumDat:word;
```

```
procedure Help;
```

```
Begin
```

```
    WriteLN;
```

```

        WriteLn('ADCYACOM[/Afilename][/Cn][/Bnnnn][/Pn][/Dn][/O;/E]
[/V]');
        WriteLn;
        WriteLn(' /Afilename:FileName =Encaminamiento yel nombredel
Archivo. ');
        WriteLn('
Nombre de Archivo pordefecto
ADCYA.DAT');
        WriteLn(' /Cn      : n      - (1,2,3,4) Puerto Serie COMn a usar. ');
        WriteLn('
Puerto serie por defecto COM1. ');
        WriteLn(' /Bnnnn:nnnn=(300,600,1200,2400,4800,9600)Velocidad
de transmisión ');
        WriteLn('
en baudios. Valor por defecto 9600 ');
        WriteLn(' /Pn      : n      - (1,2) Bits de Parada. ');
        WriteLn('
Valor por defecto 1. ');
        WriteLn(' /Dn      : n      - (7,8) Bits de datos. ');
        WriteLn('
Valor por defecto 8. ');
        WriteLn(' /N      :Sin Paridad,no puedeutilizarse juntocon /Eni
/O. ');
        WriteLn(' /O      :Paridad Impar,no puedeutilizarse juntocon /E
n1 /N. ');
        WriteLn(' /E      :Paridad Par,no puedeutilizarse juntocon /Oni
/N. ');
        WriteLn('
Por defecto hay paridad PAR (/E. ');
        WriteLn(' /V      : Habilitala visualizacióndelos datosrecibidos
en ASCII. ');
        WriteLn; Halt(1);
end;

Begin
    ClrScr;
    Comu.Init;
    ver:=false;
    error:=0;
    WriteLn;

    name:= '';
    With ComParam do begin

```

```

PCOM := 1;
BPS := 9600;
BP := 1;
BD := 8;
PAR := 'P';
If ParamCount>0 then
  For cuent:=1 to ParamCount do begin
    DosParam:=UPStr(ParamStr(Cuent));
    posi:=Pos('/',DosParam);
    if posi=0 then Help;
    Case DosParam[Posi+1] of
      'C': Val(copy(dosparam,posi+2,1),PCOM,error);
      'B': Val(copy(dosparam,posi+2,4),BPS,error);
      'P': Val(copy(dosparam,posi+2,1),BP,error);
      'D': Val(copy(dosparam,posi+2,1),BD,error);
      'A': name:=copy(dosparam,posi+2,length(Dosparam));
      'E': PAR:='P';
      'O': PAR:='I';
      'N': PAR:='N';
      'V': Ver:=true;
    else help;
    end;
    if error<>0 then Help;
  end;
end;
if name='' then assign(f,'ADCYA.DAT') else assign(f,name);
Rewrite(f);
Comu.ActCom(ComParam);
With ComParam,ComStat do begin
  Comu.GetStat(ComParam,ComStat);
  Writeln('COM',PCOM,': ',BPS,',',PAR,',',BD,',',BP);
  Writeln('INT=',INT,',DTR=',DTR,',RTS=',RTS);
end;
writeln;
Repeat
  Repeat c:=Comu.Getdat Until Comu.GetAv or keyPressed;
  Inc(NumDat);

```

```

write(f,c);
if ver then write(c) else
begin
            gotoxy(2,7);Write('Dato           ',NumDat,'
',Num2Hex(Ord(c)));clreol;
end;
Until keyPressed or ((NumDat>14) and (c=#0));
writeln;
writeln;
err:=Comu.GetErr;
writeln(err,':',Comu.GetErrStr(err));
Comu.Done;
close(f);
End.

```



## CONCLUSIONES

Las principales conclusiones que se han extraído del presente trabajo son las siguientes:

\*La ventaja que brinda el microprocesador dentro del equipo es considerable frente a otros equipos que no lo tienen.

\*El diseño y construcción de este equipo posibilitó aprender y desarrollar, en base a tecnologías extranjeras, equipos propios. Para evitar acostumbrarnos a importar todo tipo de equipos y empezar a preocuparnos por desarrollar algo de técnicas propias.

\*Permitir la capacitación de personal competente en áreas de la investigación en Instrumentación Científica.

\*La capacidad de almacenamiento de datos y su interfase serial del ADATOC abre la posibilidad de un almacenamiento permanente de datos en diskettes así como impresión de los datos por impresora.

\*El uso del microprocesador y componentes digitales hizo factible conseguir un sistema compacto, confiable, liviano y de bajo consumo.

\*Hizo posible ganar experiencia, por parte de los diseñadores en todas las fases que comprenden el desarrollo de un equipo basado en microprocesador.

\*Este trabajo deja precedentes teóricos y técnicos que puedan servir de base para el desarrollo de nuevos equipos, es así que se abre un gran campo de posibilidades de seguir construyendo otros equipos basados en microprocesador con diferentes aplicaciones científicas e industriales.

\*Con este Equipo se pueden cubrir las necesidades propias de los Laboratorios de Investigación de las Universidades del País y otras Instituciones a un precio accesible a nuestro medio.

\*Es una herramienta para la enseñanza de Microprocesadores en el área

de la Instrumentación Científica.

\*Este trabajo es un ejemplo de saber aprovechar los escasos recursos con que disponemos.

\*Para poder obtener menores índices de consumo de corriente en el equipo, es necesario reducir el número de componentes digitales. Para ello se debe usar en vez del microprocesador un microcontrolador, que tiene un convertidor A/D dentro del mismo integrado además de una interfase serial, temporizadores, etc.

## REFERENCIAS BIBLIOGRAFICAS

- \* ZILOG Inc. MANUAL DE ESPECIFICACIONES TECNICAS DE LA FAMILIA Z80. EEUU. Febrero 1984.
- \* Willian Bardem, Jr. SAMS. EL Z80 MICROCOMPUTER HANDBOOK. EEUU.
- \* Motorola Inc. MANUAL DE MICROPROCESADORES Y PERIFERICOS DE 8 BITS DE MOTOROLA Inc. EEUU 1984/85.
- \* Intel Corp. MANUAL DE MICROPROCESADORES Y MICROCONTROLADORES DE LA FAMILIA INTEL. EEUU 1993.
- \* Motorola Inc. MANUAL MEMORY DATA DE MOTOROLA INC. EEUU 1989.
- \* NATIONAL SEMICONDUCTOR. MANUALES DE ESPECIFICACIONES TECNICAS DE CIRCUITOS TTL, CMOS, CIRCUITOS LINEALES DE ADQUISICION DE DATOS. EEUU 1989.
- \* Multitech Industrial Corp. MANUAL DEL USUARIO DEL KIT MICROPROFESSOR I PLUS(MPF-IP). Taiwán 1983.
- \* Multitech Industrial Corp. MANUAL DE EXPERIMENTOS DEL MICROPROFESSOR I PLUS. Taiwán 1983.
- \* Motorola Inc. MANUAL LINEAR AND INTERFACE INTEGRATED CIRCUITS DE MOTOROLA INC. EEUU 1988.
- \* Steve Ciarcia. CONSTRUYA UNA MICROCOMPUTADORA BASADO EN EL Z80(GUIA DE DISEÑO Y FUNCIONAMIENTO). McGraw-Hill de México 1985.
- \* INICTEL. LIBRO SOBRE MICROPROCESADORES. PERU 1988.
- \* Elizabeth A. Nichols, Joseph C. Nichols, Peter R. Rony. PROGRAMACION DEL MICROPROCESADOR Z80. MARCOMBO. EEUU. 1981.
- \* Joseph C. Nichols, Elizabeth A. Nichols y Peter R. Rony. MICROPROCESADOR Z-80, PROGRAMACION E INTERFASES. MARCOMBO. EEUU. 1982.

- \* Gordon Mccomb. THE ROBOT BUILDER'S BONANZA 99 INEXPENSIVE ROBOTICS PROJECTS. TAB BOOKS Inc. EEUU 1987.
- \* Lance A. Leventhal, Winthrop Saville. Z80 ASSEMBLY LANGUAGE SUBROUTINES. McGraw-Hill. EEUU 1983.
- \* Multitech Ind. Corp. IOM-MPF-IP OPERATION MANUAL. Taiwán 1983.
- \* Philips ECG Company. MANUAL ECG SEMICONDUCTORS. EEUU.1989.
- \* Electrónica Popular. APUNTES DE ELECTRONICA DE INGENIERIA I. Perú 1986.
- \* James W. Coffron. Z80 APPLICATIONS. SYBEX Inc. EEUU 1983.
- \* ORCAD. MANUAL ORCAD Schematic, Design, Tools. EEUU 1989.
- \* Robert C. Brenner. HOWARD W. SAMS & COMPANY. IBM PC ADVANCED TROUBLESHOOTING & REPAIR. EEUU 1988.
- \* ELEKTOR. REVISTA ELEKTOR N°125.FRANCIA. 1990
- \* Edward Pasahow. ELECTRONICA MANUAL PRACTICO DE REFERENCIAS. McGraw Hill. México 1987.
- \* Peter Norton. GUIA DEL PROGRAMADOR PARA EL IBM PC. re1 andes. Colombia 1987
- \* Stephen K. O'Brien, Steve Nameroff. TURBO PASCAL 7 MANUAL DE REFERENCIA. Osborne/McGraw-Hill. España 1993.
- \* José M. Angulo. MICROPROCESADORES Fundamentos, diseño y aplicaciones en la Industria y en los microcomputadores. PARANINFO. España 1988.