

33.4
24.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DESARROLLO DE UN SISTEMA ELECTRONICO
AUTOMATIZADO, PARA LA ADQUISICION,
RECUPERACION Y PROCESAMIENTO DE DATOS
DE DEMANDA, ENERGIA ELECTRICA Y TIEMPO
DE INTERRUPCION

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A N :

JAIME CONTRERAS NIETO

ADOLFO GARCIA PEÑA



DIRECTOR DE TESIS,
ING. MARCO ANTONIO BORJA DIAZ

MEXICO, D. F.

TESIS CON
FALLA DE ORIGEN

1989



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

INDICE

CAPITULO 1.- INTRODUCCION.

1.1.-	Antecedentes.....	1
1.2.-	Diseño Conceptual.....	2

CAPITULO 2.- ESPECIFICACIONES DE DISEÑO.

2.1.-	Especificaciones Técnicas de la Unidad de Adquisición de Datos.....	7
2.1.1.-	Modo de Adquisición de Datos.....	8
2.1.2.-	Tipo y Capacidad de Memoria a Utilizar.....	9
2.1.3.-	Base de Tiempo.....	9
2.1.4.-	Interfaz de Comunicación al Exterior.....	10
2.1.5.-	Fuente de Alimentación.....	10
2.2.-	Especificaciones Funcionales de la UAD.....	11
2.2.1.-	Comando de Modo de Interrogación.....	11
2.2.2.-	Comando de Acceso.....	12
2.2.3.-	Comando de Transferencia de Datos.....	12
2.2.4.-	Comando de Programación de Reloj.....	12
2.3.-	Especificaciones Mecánicas de la UAD.....	13
2.4.-	Especificaciones Técnicas del Equipo de Interrogación/ Programación Externa (IPE).....	13
2.4.1.-	Configuración.....	13
2.4.2.-	Interfaz Hombre-Máquina.....	14
2.4.3.-	Capacidad de Memoria de Programación.....	14
2.4.4.-	Capacidad de Memoria de Recuperación de Datos.....	15
2.4.5.-	Interfaz de Comunicación con la UAD.....	15
2.4.6.-	Interfaz de Comunicación para Transferencia de Datos al Computador Personal.....	16
2.4.7.-	Fuente Ininterrumpible de Voltaje.....	16
2.5.-	Especificaciones Funcionales del Equipo IPE.....	16
2.5.1.-	Autodiagnóstico.....	17
2.5.2.-	Modo de Espera de Comandos.....	17
2.5.3.-	Modo de Interrogación.....	18
2.5.4.-	Modo de Acceso.....	18
2.5.5.-	Modo de Transferencia.....	18
2.5.6.-	Modo de Programación de Reloj.....	19
2.5.7.-	Modo de Transferencia a Computador Personal.....	19
2.6.-	Especificaciones Mecánicas del Equipo IPE.....	19

CAPITULO 3.- DISEÑO ELECTRONICO

3.1.-	Introducción.....	21
3.2.-	Diseño Electrónico de las Unidades de adquisición de Datos (UAD).....	23
3.2.1.-	Unidad de Control y Comunicación.....	24
3.2.2.-	Unidad de Captura de Datos.....	30
3.2.3.-	Unidad de Almacenamiento de Información.....	30
3.2.4.-	Fuente Ininterrumpible de Alimentación.....	31
3.3.-	Diseño Electrónico del equipo de Interrogación / Programación Externa (IPE).....	33
3.3.1.-	Unidad Mínima.....	36
3.3.2.-	Interfaz Hombre-Máquina.....	39
3.3.3.-	Unidad de Comunicación.....	40
3.3.4.-	Unidad de Almacenamiento de Información.....	44
3.3.5.-	Fuente de Alimentación Ininterrumpible.....	44
3.4.-	Listas de Partes.....	46
3.4.1.-	Lista de Proveedores.....	47
3.4.2.-	Lista de Catálogos.....	48

CAPITULO 4.- DISEÑO DEL SOFTWARE.

4.1.-	Introducción.....	63
4.2.-	Software de la Unidad de Adquisición de Datos.....	65
4.3.-	Software para el Equipo de Interrogación/ Programación Externa.....	122

CAPITULO 5.- DISEÑO MECANICO.

5.1.-	Introducción.....	245
5.2.-	Diseño Mecánico de las Unidades de Adquisición de Datos (UAD).....	247
5.3.-	Diseño Mecánico del Equipo de Interrogación/ Programación Externa (IPE).....	252

CAPITULO 6.- SISTEMA OPERATIVO.

6.1.-	Introducción.....	268
6.2.-	Modo de Espera de Instrucciones (MEI).....	269
6.2.1.-	Modo de Acceso (A).....	269
6.2.2.-	Modo de Interrogación (B).....	270
6.2.3.-	Modo de Transferencia (C).....	272
6.2.4.-	Modo de Reprogramación (D).....	274
6.2.5.-	Modo PC (E).....	276
6.2.6.-	Mensajes de Error.....	277

REFERENCIAS.....	279
------------------	-----

BIBLIOGRAFIA.....	281
-------------------	-----

ANEXO I.- Fotografías.....	285
----------------------------	-----

Capítulo 1

Introducción.

1.1 Antecedentes.

El Instituto de Investigaciones Eléctricas, ha venido realizando proyectos de infraestructura, con el fin de desarrollar tecnología Nacional de alta calidad, que permita la eliminación de importaciones de equipo de medición de parámetros eléctricos. Este tipo de equipos es utilizado comunmente para la realización de estudios de comportamiento y caracterización de circuitos de distribución y de subestaciones eléctricas, de uso común dentro del Sector Eléctrico Nacional.

El conocimiento del comportamiento de las demandas y consumo de energía que se presentan en los circuitos, subestaciones de control y centros de consumo, es fundamental en la planeación del equipo y estructuración de los mismos, tendientes a mejorar la confiabilidad y calidad del servicio suministrado a los usuarios, así como a establecer programas para el uso racional de la energía.

Así mismo, la distribución del tiempo durante el cual un circuito o subestación de energía eléctrica deja de operar, es información que al correlacionarse con la de demanda y energía, permite profundizar los estudios mencionados.

Para llevar a cabo un estudio de esta naturaleza, se presentan varios obstáculos como son:

1. La gran cantidad de datos que se requieren para obtener los resultados de interés.
2. El necesario procesamiento computacional de los mismos.

3. La carencia de equipo de fabricación Nacional, para la adquisición de datos.

4. El alto costo de equipos comerciales disponibles en el mercado internacional.

En la época actual, el poder realizar los estudios mencionados en forma repetitiva, demanda el uso de sistemas modernos de adquisición de datos compatibles con sistemas de cómputo, dentro de una economía accesible que justifique la inversión.

Existen en el mercado internacional diferentes tipos de sistemas de medición, que van desde graficadores de valores instantáneos, hasta modernos equipos automatizados. Sin embargo, en ambos casos, estos equipos no se fabrican en México y su importación resulta bastante costosa, además de los inconvenientes para la economía nacional que ello implica.

El presente proyecto fué realizado con objeto de proporcionar una alternativa para satisfacer dicha carencia en el Sector Eléctrico Nacional, tanto para nuevas instalaciones como para la sustitución de equipo que por sus características resulta obsoleto en la época actual.

Así mismo, el producto de este proyecto se orientó a el desarrollo de tecnología transferible a la Industria Eléctrica Nacional, de tal manera que sus especificaciones de diseño, funcionalidad, calidad y costo, lo hicieran reproducible a nivel Nacional y competitivo con equipos similares de fabricación extranjera.

1.2 Diseño Conceptual.

De acuerdo a lo mencionado en la sección anterior, el objetivo del presente proyecto es el desarrollo de un sistema electrónico automatizado, para la adquisición, recuperación y procesamiento de datos de demanda, energía eléctrica y tiempo de interrupción en circuitos de distribución y subestaciones.

En el Instituto de Investigaciones Eléctricas, se ha recopilado y revisado información relativa a sistemas comerciales de esta índole, realizando estudios comparativos entre los diferentes sistemas que se ofrecen en el mercado internacional, así como recomendaciones realizadas por posibles usuarios nacionales.

En este tipo de estudios, se revisaron los siguientes aspectos:

- Confiabilidad.

- Exactitud.
- Facilidad de operación.
- Facilidad de mantenimiento.
- Tiempo de obsolescencia.
- Calidad.
- Tamaño.
- Costo de reproducción.
- Facilidad de reproducción.
- Nivel de tecnología utilizado.

Del resultado de estos estudios, se obtuvo un diseño conceptual de un sistema que intenta optimizar la combinación de los tópicos mencionados. Dicho diseño conceptual, se ilustra en forma de diagrama de bloques en la figura No. 1.1, y sus características de tipo general, se describen a continuación.

En términos generales, el sistema está formado por Unidades de Adquisición de Datos (UAD), instalables dentro de Watthorímetros convencionales. Estas unidades tienen como función principal la adquisición de datos de demanda promedio y/o energía eléctrica consumida y tiempo de interrupción, en intervalos consecutivos de 15 minutos, hasta por periodos ininterrumpidos de 31 días.

La información adquirida, es almacenada en las unidades de adquisición, en una memoria no volátil de estado sólido (EEPROM), hasta que es recuperada por medio de un equipo externo de concepto portátil, denominado equipo de Interrogación Programación Externa (IPE).

Por medio de órdenes generadas a través de una interface hombre máquina, integrada al equipo IPE, la información almacenada en las unidades de adquisición de datos, es transferida al equipo externo, -vía una interface de comunicación óptica-, el cual a su vez, almacena dicha información en un cartucho de memoria no volátil de estado sólido (EEPROM).

El cartucho donde se almacena la información recuperada, es de concepto removible, lo que significa que no existe un límite físico de la cantidad de información que puede ser recuperada por el IPE y es independiente de la cantidad de unidades de adquisición de datos operadas.

**SISTEMA INTEGRAL PARA LA ADQUISICION Y RECUPERACION
DE DATOS DE DEMANDA Y/O ENERGIA ELECTRICA
SIADE-1-11E**

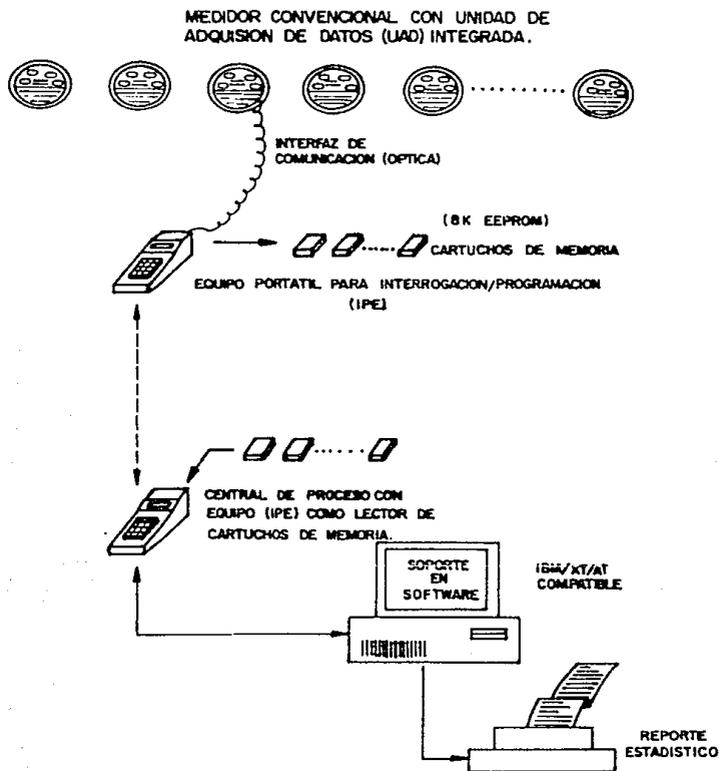


Figura 1.1: Diseño Conceptual del Sistema

El equipo externo (IPE), tiene la capacidad de comunicación con un computador personal -via un puerto serie RS 232-. esto con el objeto de que los cartuchos de memoria donde se guarda temporalmente la información recuperada, puedan ser leídos y la información transferida automáticamente hacia el computador personal donde será procesada.

De acuerdo a la configuración conceptual dada al sistema, se observa que el objetivo fundamental, es obtener un diseño simple, compacto y de bajo costo, para las unidades de adquisición de datos (UAD), esto sin sacrificar su confiabilidad, calidad y capacidad funcional. Lo anterior, debido a que de estas unidades, deberán existir tantas como circuitos de distribución y/o subestaciones se deseen medir, minimizando así el monto de la inversión requerida.

Por su parte, el diseño del equipo externo (IPE), será mas complejo en cuanto a las funciones a desarrollar y se esperará un costo mayor que el de las unidades de adquisición de datos, sin embargo, bastará una sola unidad para operar varias UAD, lo cual también abate el monto de la inversión requerida.

Para el establecimiento de la comunicación entre ambos equipos (UAD e IPE), se utilizará un puerto serie aislado ópticamente, con el objeto de proporcionar una independencia eléctrica entre ambas unidades y de que la unidad de adquisición de datos esté contenida en su totalidad, dentro de la cúpula de vidrio de los Watthorímetros donde será instalada, sin existir ningún conector hacia el exterior.

La computadora utilizada para procesamiento de datos, será del tipo Computador Personal IBM/PC/XT AT ó compatible, ya que en la actualidad este tipo de sistemas es accesible económicamente y técnicamente a los posibles usuarios del sistema.

Tal como se muestra en la configuración conceptual dada al sistema, se requerirá software de soporte para la microcomputadora, con el objeto de manejar el protocolo de comunicación con el (IPE), al funcionar este último como lector de cartuchos de memoria de estado sólido, en el proceso de recuperación de la información capturada.

Así mismo, se requerirá un soporte en software para el procesamiento de datos y la obtención de resultados finales.

En ambos aspectos, el Instituto de Investigaciones Eléctricas, cuenta con dos paquetes computacionales denominados ALE (ref. 1) y DEMANDAS (ref. 2), que en conjunto permiten lo siguiente:

1. Manejar un protocolo de comunicación con dispositivos periféricos via

puerto serie RS-232, para la interrogación de los mismos y transferencia de datos almacenados (orientado a lectores de memoria de estado sólido, desarrollados en el IIE).

2. Creación y decodificación de archivos de datos en disco, a partir de la información transferida desde el exterior.
3. Procesamiento de archivos de datos de demanda y energía eléctrica, para el análisis de comportamiento de circuitos y subestaciones.

Capítulo 2

Especificaciones de Diseño.

A partir del diseño conceptual, se procedió a determinar las especificaciones de diseño que deberán ser satisfechas por los subsistemas a desarrollar como parte de este proyecto.

Dado el estado del arte actual, en cuanto a la tecnología de circuitos electrónicos, la especificación fundamental que se aplica tanto a la Unidad de Adquisición de Datos (UAD) y a la unidad de Interrogación y Programación Externa (IPE) es que su diseño deberá basarse en el uso de micro-controladores y periféricos compatibles de tecnología CMOS.

Las especificaciones de diseño determinadas para ambos subsistemas, pueden agruparse de la siguiente manera:



2.1 Especificaciones Técnicas de la UAD.

Por especificaciones técnicas de la UAD. definiremos a aquellas características que deberán ser satisfechas desde el punto de vista del diseño electrónico.

2.1.1 Modo de Adquisición de Datos.

Dado que en el mercado existe una gran variedad de marcas comerciales de Watthorímetros, con el objeto de que la UAD sea compatible eléctricamente con todos ellos, se observa que la forma más adecuada de realizar la medición de demanda promedio y/o energía eléctrica, será el uso de "generadores de pulsos".

En "generador de pulsos", consiste en un arreglo de foto-emisores, foto-receptores y circuitos analógicos, que generan un tren de pulsos de frecuencia proporcional a la velocidad de giro del disco de un Watthorímetro convencional. De esta manera el conteo de los pulsos generados, dentro de un intervalo de tiempo, proporcionará la suficiente información para el cálculo de la energía eléctrica medida por el Watthorímetro. En estas condiciones la demanda promedio en dicho intervalo de tiempo, será calculada de la energía medida.

Lo anterior tiene las siguientes ventajas:

- La UAD, no requerirá calibración ya que no se realizará ninguna etapa de acondicionamiento analógico de señales.
- La precisión de la UAD, dependerá únicamente de la precisión con que se haya calibrado el Watthorímetro donde se instalará esta, procedimiento que actualmente está perfectamente controlado por las compañías suministradoras de energía eléctrica.
- La señal adquirida no dependerá del voltaje de trabajo del Watthorímetro, por lo que se hace compatible con mediciones a diferentes voltajes o capacidades de circuitos.
- La señal adquirida es independiente del tipo de configuración de medición dada a los Watthorímetros convencionales, por lo que podrá ser utilizado indistintamente en medidores configurados para medición de KWh o KVAh.
- No existirá ningún acoplamiento eléctrico entre la señal adquirida por la UAD y las señales utilizadas por el Watthorímetro para su propia medición, por lo que su introducción, no podrá afectar de ninguna forma a la medición original.

En el Instituto de Investigaciones Eléctricas, se desarrolló un iniciador de pulsos muy compacto, que será integrado a la UAD (ref. 3).

2.1.2 Tipo y Capacidad de Memoria a Utilizar.

De acuerdo a un estudio realizado por la compañía General Electric, con respecto a los generadores de pulsos mencionados (ref. 4), un generador que proporcione 10 pulsos por revolución del disco del Wathorímetro, tendrá una resolución suficiente para las diferentes capacidades de circuitos a medir.

Así mismo, en dicho estudio se deduce que para mediciones normales, los medidores utilizados en conjunto con "generadores de pulsos" que tengan una respuesta de 10 pulsos/revolución, proporcionarán una frecuencia máxima de 4 pulsos/segundo, lo que equivale a 3600 pulsos en un intervalo de 15 minutos.

Si estos pulsos son contados en forma binaria, se requerirá de 1.5 bytes para almacenar la cuenta máxima.

Por otra parte, de acuerdo con procedimientos de manejo de información de compañías nacionales de suministro de energía eléctrica, la medición del tiempo de interrupción dentro del intervalo de 15 minutos, puede realizarse con resolución de subintervalos de 1 minuto, lo que utilizará 4 bits en un conteo binario.

De esta manera, la información a capturar puede almacenarse en dos bytes de memoria, asignando el primer byte y los cuatro bits menos significativos del segundo byte al dato de energía eléctrica y los cuatro bits más significativos del segundo byte al dato de tiempo de interrupción.

Ya que se desea obtener estos datos en periodos de hasta 1 mes, será necesario almacenar 2976 datos que corresponden a 5952 bytes.

Para cubrir esta necesidad, se selecciona una memoria con capacidad de 8 K bytes, lo cual deja un espacio suficiente para almacenar información adicional, tal como: identificación de la unidad, fecha y hora de inicio de adquisición de datos, etc.

La memoria a utilizar, debe ser del tipo no volátil, con objeto de asegurar la conservación de la información adquirida, aún en el caso de ausencia prolongada de alimentación de energía eléctrica. Lo anterior sumado con la condición de que la UAD permanezca dentro del Wathorímetro implica la selección de una memoria EEPROM.

2.1.3 Base de Tiempo.

Ya que la precisión de la UAD, estará determinada por la precisión y estabilidad de su base de tiempo, se integrará un reloj en tiempo real, cuya base de tiempo sea generada a partir de un cristal de cuarzo.

Este reloj deberá tener la capacidad de proporcionar al microcontrolador, una señal cada que se finalice un intervalo de 15 minutos, con el objeto que el microcontrolador interrumpa cualquier función en proceso y almacene el dato adquirido.

Así mismo, deberá tener capacidad para proporcionar al microcontrolador, la información correspondiente a la fecha y hora corrientes, lo cual será utilizado para "etiquetar" los archivos de información con la hora y fecha de inicio de adquisición de datos. Este reloj, deberá operar en forma continua independientemente de fallas en la alimentación de energía principal.

2.1.4 Interface de Comunicación al exterior.

Para efectuar la transferencia de información, entre la UAD y el IPE, dado que la UAD estará permanentemente contenida dentro de la cúpula de vidrio del Watthorímetro, la mejor alternativa es utilizar un puerto de comunicación óptico tipo serie.

Esto tendrá la ventaja de que no se requerirá la modificación de la cúpula de vidrio del Watthorímetro, para la instalación de algún conector hacia el exterior. De igual forma, no se tendrá acceso a la UAD si no es a través del equipo IPE.

Para ello, se selecciona un arreglo de foto-emisores y foto-detectores de espectro infrarrojo, con lo cual pueden eliminarse los posibles efectos adversos causados por la iluminación artificial y luz de día.

2.1.5 Fuente de Alimentación.

Obviamente, la forma más práctica de alimentar eléctricamente a los circuitos de la UAD, será utilizando la línea medida por el Watthorímetro (en baja tensión), sin embargo dado que se prevén interrupciones en dicha línea, se hace necesario contar con una fuente ininterrumpible de alimentación, que mantenga en operación continua al equipo bajo las condiciones de interrupción.

Para ello, se podrá utilizar un arreglo de baterías recargables integradas al circuito electrónico de la UAD, tal que sumado a la utilización de micro-circuitos de tecnología CMOS, proporcione el mayor soporte posible, dentro de la disponibilidad actual de capacidad de baterías recargables, en función a las restricciones de tamaño para su introducción en el Watthorímetro convencional.

2.2 Especificaciones Funcionales de la UAD.

Por especificaciones de tipo FUNCIONAL para la UAD, definiremos a aquellas que se deberán realizar, en respuesta a las órdenes o comandos enviados por el equipo de Interrogación/ Programación Externo (IPE).

Obviamente, estas especificaciones fueron determinadas en conjunto con las correspondientes a realizar por el IPE, las cuales se detallan en la sección 2.5 de este capítulo.

Como especificación fundamental, se determinó, que la UAD deberá realizar como tarea prioritaria, la adquisición de datos. Esto significa que cualquier interrogatorio o interrupción generada desde el IPE, deberá ser atendida, sin interrupción del proceso de adquisición de datos. Solamente se considera una excepción, que corresponderá a una orden de finalización del período en progreso y la iniciación de uno nuevo.

La UAD, deberá responder a cuatro comandos básicos generados desde el IPE, que son:

2.2.1 Comando de Modo de Interrogación.

En respuesta a este comando, la UAD, deberá transmitir via el puerto de comunicación óptico, la siguiente información:

- A - Fecha de inicio de adquisición de datos.
- B - Hora de inicio de adquisición de datos.
- C - Número de días que faltan para que la memoria interna sea saturada de información.
- D - Código de identificación del estado de operación de la unidad.
operación normal.
falla con diagnóstico.
- E - Fecha corriente del reloj interno.
- F - Hora corriente del reloj interno.

Este comando, será utilizado para que el operador pueda verificar la correcta operación de la UAD, y podrá ser utilizado en cualquier momento.

2.2.2 Comando de Acceso.

Este comando está orientado a la protección contra el acceso de personal no autorizado, a la información adquirida por la UAD, así como a la modificación de la fecha y hora corrientes del reloj interno.

En el momento de la recepción de dicho comando, la UAD, contestará con una petición de un código de acceso, el cual será propio y único de cada UAD.

El código correspondiente será enviado desde el IPE. En el caso de corresponder a un código falso, la UAD responderá con un mensaje de "acceso negado", de lo contrario el acceso será permitido enviando un mensaje de "acceso permitido" y el usuario podrá entrar en los modos de operación que se describen en seguida.

2.2.3 Comando de Transferencia de Datos.

Cuando este comando sea enviado, la UAD procederá a la lectura de la información almacenada en su memoria y a la transmisión de esta vía el puerto óptico de comunicación hacia el IPE.

Una vez que la totalidad de la información haya sido transferida, se generará un mensaje de "fin de memoria", quedando en espera de otra orden que le indicará si se desea iniciar un nuevo período de adquisición de datos o si se desea continuar con el actual.

Este hecho tiene por objeto, permitir al usuario la recuperación parcial de la información adquirida en el momento que lo desee, sin la interrupción de períodos programados, que comúnmente se estima a nivel mensual.

Cuando se inicie un nuevo período de adquisición de datos, la memoria deberá ser borrada en su totalidad y en sus primeras localidades, se almacenará la información correspondiente a la fecha y hora en que se inicia el nuevo período, así como un código que identifique al "archivo de datos" en cuanto a la UAD en que fué programada.

Esto será de gran utilidad al usuario, ya que no se requerirá la etiquetación manual de los cartuchos de memoria utilizados por el IPE para recuperación de la información.

2.2.4 Comando de reprogramación de Reloj.

Este comando será de utilidad para la instalación de la UAD.

Su función será la de permitir al usuario programar el reloj interno con la fecha y hora correcta.

Cuando se reciba este comando, la UAD quedará en espera de todos los datos necesarios para la programación del reloj. Un protocolo de petición y envío de dichos datos, deberá ser manejado entre la UAD y el IPE.

Una vez que la UAD reciba la información necesaria, el reloj será programado e iniciada su operación.

2.3 Especificaciones Mecánicas de la UAD.

Dado que se requiere que la información adquirida por la UAD, sea de uso exclusivo de las compañías suministradoras de energía eléctrica ó en su caso del usuario que haya adquirido el equipo para estudios propios, se hace necesario protegerlo en contra del uso no autorizado ó de actos que pudieran afectar su funcionamiento. Una de las formas de lograr esto, es integrando la unidad de adquisición de datos dentro del mismo Watthorímetro, ya que estos generalmente se protegen con sellos para detectar su violación.

Este hecho tiene además la ventaja de reducir el costo del equipo, ya que no se hará necesario el uso de ningún gabinete para montaje de la unidad.

No obstante, existe el inconveniente de que las diferentes marcas comerciales son de distinta construcción mecánica, lo cual implica que el diseño mecánico de la UAD será dependiente del Watthorímetro a utilizar.

Para este proyecto, la UAD será diseñada para el medidor marca General Electric, modelo M-50, utilizado comunmente para la medición en circuitos primarios de distribución y servicios industriales.

2.4 Especificaciones Técnicas del Equipo de Interrogación / Programación Externo (IPE).

Al igual que para la UAD, las especificaciones técnicas del IPE, se refieren a las características que deberán ser satisfechas desde el punto de vista del diseño electrónico.

2.4.1 Configuración.

En términos generales, las funciones que realizará el equipo IPE son:

- Manejo de una interfaz hombre-máquina para establecimiento de comunicación entre el usuario y el sistema.

- Manejo de una interfaz de comunicaciones para interrogación y programación de las unidades de adquisición de datos.
- Almacenamiento temporal de la información recuperada de las UAD.
- Manejo de una interfaz de comunicaciones para establecimiento de comunicación con un computador personal y transferencia de la información almacenada

Dado su concepto de equipo portátil, su diseño electrónico deberá basarse en circuitos integrados multifuncionales, que le permitan integrar todas las funciones mencionadas en el menor espacio posible.

2.4.2 Interface Hombre-Maquina.

Una comunicación fácil y explícita entre el operador y el sistema, es fundamental para su aceptación por los usuarios y para su confiable operación.

Para ello, el IPE deberá contener un teclado y un monitor alfanumérico, teniendo ambos la capacidad necesaria para la introducción y despliegado de las órdenes generadas por el operador.

En un sistema de este tipo, es muy importante que el operador observe en el monitor la instrucción o datos que está tecleando, así como que se le permita corregir lo introducido en caso de error, antes de que el sistema ejecute la instrucción.

2.4.3 Capacidad de Memoria de Programación.

Con objeto de minimizar la capacidad de memoria de programación de las unidades de adquisición de datos, la mayor parte del software utilizado para el protocolo de comunicación y despliegado de mensajes, será programado en el IPE.

Lo anterior sumado a la cantidad de funciones que debe realizar el IPE, hace suponer que se requerirá una capacidad amplia de memoria de programación, que a priori puede estimarse en 8 K bytes.

Observando el mercado actual, una memoria EPROM de 128 K bytes, de tecnología CMOS, puede alcanzar una cotización de hasta 6 U.S.D. (precio unitario), cuando una memoria de 8 K bytes de tecnología MOS, se cotiza en 4.3 U.S.D.. Esta pequeña diferencia no justifica la limitación de memoria a 8 K bytes, ya que se observa además que la tendencia de precio de la memoria de 128 K bytes tipo CMOS es a la baja, mientras que el precio de la de 8 K bytes de tecnología MOS tiende a subir.

Ante estas circunstancias, es obvio que la demanda de memorias de mayor capacidad se irá incrementando y tenderán a obsoletar a aquellas de menor capacidad. Igualmente las ventajas en cuanto al consumo de corriente que presenta la tecnología CMOS contra la MOS, propician que en la actualidad se prefiera el uso de las primeras, sobre todo en sistemas de concepto portátil.

La capacidad de memoria que no sea utilizada en el IPE, proporcionará la libertad de ampliar su programa operativo para su utilización en otras posibles aplicaciones.

2.4.4 Capacidad de Memoria para Recuperación de Datos.

Para el almacenamiento temporal de la información recuperada de las UAD, el IPE requerirá de un banco de memoria, cuyo tamaño debe determinarse.

Es obvio que si se maneja un concepto de memoria integrada, siempre existirá un límite físico en cuanto a la cantidad de UAD que puedan interrogarse en una misma sesión de recuperación de datos, sin tener que efectuar la transferencia de la información a la computadora.

Por ejemplo, considérese que un operador tiene como tarea la lectura en el mismo día de 20 unidades de adquisición de datos, las cuales estén instaladas en diferentes lugares. Si la capacidad del IPE se limitara a 5 veces la capacidad de una UAD, el operador tendría que realizar 5 lecturas y regresar a la computadora para realizar la transferencia de datos adquiridos.

Así mismo, un banco de memoria integrada, solamente aumentaría el tamaño y complejidad de la unidad portátil.

Para solucionar estos problemas, se determinó que la forma más conveniente de dotar al IPE de memoria temporal, es mediante el uso de "CAR-TUCHOS" removibles que contengan memorias de la misma capacidad que las de las UAD.

Así, la operación se hará insertando un cartucho de memoria en el IPE, antes de cada lectura de una UAD y removiéndolo al final de cada evento, siendo su manejo similar al de las cintas magnéticas o "cassettes".

Implicitamente, lo anterior significa que la memoria utilizada en los cartuchos, deberá ser del tipo NO volátil, por lo que se selecciona una memoria EEPROM de 8 K bytes (igual que la de la UAD).

2.4.5 Interfaz de Comunicación con la UAD.

Obviamente, esta interfaz deberá ser igual a la interfaz utilizada en la UAD, que consiste en un par foto-emisor y foto-detector infrarrojos, para trans-

misión y recepción tipo serie.

2.4.6 Interfaz de Comunicación para Transferencia de Datos al Computador Personal.

La forma más adecuada de establecer una comunicación entre el IPE y el computador personal, es a través de un puerto serie RS-232 con el que cuentan las computadoras para el manejo de periféricos.

Va que un puerto de este tipo (en su concepción mas sencilla), requiere únicamente de una línea de transmisión, una de recepción y otra común, se podrá utilizar la misma interfaz de comunicación con la UAD, tomando la señal eléctrica (no óptica) y convirtiéndola a los niveles de voltaje requeridos por la norma estándar de un puerto RS-232.

2.4.7 Fuente Ininterrumpible de Voltaje.

Dado el concepto portátil del equipo IPE, y por el tipo de instalaciones en donde comunmente se instalarán las UAD, es preferible que se integre al IPE una fuente ininterrumpible de alimentación, basada en baterías recargables.

Esta fuente, deberá proporcionar la capacidad de operación del equipo IPE conectado a una línea de alimentación convencional de 110 V.c.a., y en su ausencia la operación conmutada automáticamente por medio de baterías.

La autonomía de operación que dicha fuente deberá proporcionar al IPE, - al operarse con baterías -, deberá ser de un mínimo de 2.5 horas en modo de transferencia de datos.

Obviamente, para optimizar está condición, todos los circuitos integrados utilizados en el IPE deberán ser de tecnología CMOS.

2.5 Especificaciones Funcionales del Equipo IPE.

Por especificaciones FUNCIONALES del equipo IPE, definiremos a aquellas que deberán realizarse con objeto de:

- Capturar, interpretar y transmitir a las UAD, órdenes generadas por el operador a través de la interfaz hombre-máquina, indicando en su monitor el progreso de las mismas.
- Recibir, interpretar y desplegar en el monitor alfanumérico, los mensajes generados por la UAD en respuesta a los comandos enviados.

- Establecer el protocolo de transferencia de información almacenada en las UAD, recuperando dicha información y almacenándola a su vez en los cartuchos de memoria.
- Capturar, interpretar y ejecutar órdenes generadas por el operador a través de un computador personal, con el objeto de realizar rutinas de transferencia de información almacenada en los cartuchos de memoria.

A continuación se describen las especificaciones determinadas.

2.5.1 Autodiagnóstico.

Para obtener una operación confiable, es importante que al activar el equipo IPE, se realicen automáticamente pruebas de autodiagnóstico.

En el caso de existir una falla, se deberá mostrar al operador un código de identificación de la falla y de lo contrario el equipo entrará en modo de operación normal.

Este hecho, permitirá que si el equipo IPE tiene una falla identificable por las pruebas de autodiagnóstico, el operador se percate de ello y no intente operar el sistema, ya que de lo contrario la falla de operación, se podría atribuir equivocadamente a la UAD en interrogación.

2.5.2 Modo de Espera de Comandos.

En operación normal, el IPE estará en espera de 5 posibles comandos que son:

- MODO DE INTERROGACION.
- MODO DE ACCESO.
- MODO DE TRANSFERENCIA.
- MODO DE PROGRAMACION.
- MODO DE COMUNICACION A COMPUTADOR PERSONAL.

El IPE, deberá verificar que el comando introducido, sea válido o de lo contrario deberá desplegar un mensaje de "opcion inválida"

2.5.3 Modo de Interrogación.

Quando se seleccione este modo, el IPE transmitirá el comando correspondiente a la UAD y esperará por la información solicitada. La información transmitida por la UAD, deberá ser mostrada al operador en el monitor alfanumérico.

Al igual que para todos los comandos enviados, el IPE esperará por una respuesta de la UAD, solamente durante un intervalo de tiempo y en el caso de no recibirla, desplegará un mensaje de "NO RESPUESTA".

2.5.4 Modo de Acceso.

Con este comando, el IPE, deberá solicitar al operador el código que será enviado a la UAD, el cual deberá contener cuatro dígitos. Una vez capturado dicho código, se enviará a la UAD, el comando de modo de acceso, y esperará por una respuesta de petición de envío del primer dígito, al recibir esta respuesta, lo transmitirá y así sucesivamente hasta terminar con el código.

Finalmente, esperará por un código de respuesta de acceso negado ó permitido, y desplegará un mensaje en el monitor indicando al operador el resultado.

Si el código de acceso fué aceptado, el IPE, permitirá al usuario la introducción de los dos comandos siguientes, de lo contrario se mostrará un mensaje de "no acceso previo".

2.5.5 Modo de Transferencia.

Este modo se refiere a la recuperación de la información adquirida por las UAD.

Para iniciar una rutina de transferencia de datos, el IPE deberá verificar que el operador haya colocado un cartucho de memoria, que esté en buen estado, así como el de los circuitos utilizados para su programación.

La forma más sencilla y confiable de lograrlo, es programando la primera localidad de memoria y verificando su correcta programación. Si dicha prueba NO es satisfactoria, se deberá indicar al operador por medio de un mensaje en el monitor.

Una vez pasada esta prueba, se transmitirá el comando de transferencia y se esperará por la información contenida en el primer byte de la memoria de la UAD. A su recepción, el byte será programado en la primera localidad de memoria del cartucho colocado en el IPE. Posteriormente, se solicitará el

envío del siguiente byte y el procedimiento continuará hasta llegar al final de la memoria.

Al término de la transferencia, el IPE preguntará al operador si se desea iniciar un nuevo período de adquisición de datos ó no. En cualquier caso, se transmitirá un código que identifique la opción seleccionada y se regresará al modo de espera de comandos.

2.5.6 Modo de Programación de Reloj.

Al recibir este comando, el IPE, solicitará al operador la información necesaria para programar en el reloj interno de la UAD, la fecha y hora deseadas.

Una vez que la información esté completa, se transmitirá a la UAD el comando correspondiente y se establecerá un protocolo de petición y envío de información.

A su término, el IPE solicitará al operador una orden de INICIO, lo cual permitirá iniciar la operación del reloj en el momento que se alcance la hora programada.

2.5.7 Modo de Transferencia a Computador Personal.

Cuando el equipo IPE sea utilizado para lectura de cartuchos de memoria y transferencia de la información a un computador personal, deberá operar como un dispositivo periférico a el computador, obedeciendo a órdenes generadas desde este último.

De esta manera al entrar en este modo de operación, el IPE quedará en espera de una solicitud de comunicación por parte del computador, la que deberá ser contestada.

Posteriormente, quedará en espera de una orden para realizar la lectura y transmisión de un número específico de bytes, lo cual se repetirá hasta que la memoria colocada sea leída en su totalidad.

Esta rutina de lectura de memorias, podrá ser accedada repetitivamente con el objeto de facilitar al operador la lectura de varios cartuchos de memoria sin tener que abandonar el modo seleccionado.

2.6 Especificaciones Mecánicas del Equipo IPE.

Dado que se pretende obtener la mayor economía y funcionalidad en los equipos a diseñar, las características mecánicas del equipo IPE, se eligieron como las siguientes:

- El equipo deberá ser compacto y portátil, ya que se tratará de un instrumento de campo.
- Su presentación debe ser excelente, ya que será la imagen del sistema debido a que por su parte la UAD estará contenida dentro de los Wattthorímetros.
- Su diseño mecánico deberá ser muy simple desde el punto de vista de la reproducción del mismo a nivel industrial.

Capítulo 3

DISEÑO ELECTRONICO

3.1 Introducción.

A partir de las especificaciones determinadas en el capítulo No. 2, se procedió al análisis de las diferentes alternativas de diseño electrónico para los equipos UAD e IPE.

El paso inicial, fué la selección del microcontrolador a utilizar. Al respecto, se determinó la importancia de que tanto el equipo IPE como las UAD, basaran su diseño en microcontroladores de la misma familia, con lo que su compatibilidad es asegurada, y en su desarrollo y reproducción, se manejará una misma tecnología (lenguaje, periféricos, equipo de desarrollo, proveedores, etc.).

Después de un estudio detallado, se seleccionaron microcontroladores de la familia MCS-51 de la marca INTEL (ref. 5), de acuerdo a lo siguiente:

Para la UAD	87C51	ó	80C51
Para el IPE	80C31		

Las características de esta familia se presentan a continuación:

1. Unidad de procesamiento central (CPU) de 8 bits.
2. Procesador booleano.
3. Ciento once instrucciones de programación.

4. Memoria RAM integrada con capacidad de 128 bytes.
5. 32 líneas de entrada/salida.
6. Dos "timer/counter" de 16 bits.
7. Puerto serie integrado "Full-duplex".
8. Cinco fuentes de interrupción con prioridad programable.
9. Emplea una sola fuente de alimentación (5 Volts).

La versión 87C51 tiene una memoria EPROM integrada con 4 K bytes de capacidad.

La versión 80C51 tiene una memoria ROM integrada (4 K bytes), "mask programmable" que puede ser programada de fábrica para reproducción del equipo en gran escala, a un bajo costo.

La versión 80C31 utiliza memoria externa de programación.

Las características mencionadas, hablan por sí mismas de las ventajas que presenta la familia MCS-51. Sin embargo, a continuación se mencionan algunos de los aspectos más importantes que motivaron su selección.

1. El contar con una interfaz de comunicaciones integrada, es una ventaja muy grande, ya que de lo contrario se haría necesario el uso de un periférico adicional tal como el 82C51, aumentando el tamaño del circuito impreso (factor tan importante en el diseño de las UAD) y disminuyendo la disponibilidad de líneas de entrada/salida.
2. Así mismo, la integración de una memoria EPROM con capacidad de 4 K bytes en la versión 87C51, reducirá el tamaño de las UAD, y la versión 80C51 permitirá la reproducción de las UAD (en gran escala), a precios muy bajos.
3. La versión 80C31, permite el uso de una memoria externa de programación con 128 K bytes de capacidad para el equipo IPE, tal como se determinó en las especificaciones de diseño, manteniendo la compatibilidad al 100 por ciento con las UAD.
4. En las versiones de tecnología CMOS, se dispone de una instrucción que programa al microcontrolador en modo "Idle" (perezoso), en el cual se consume el mínimo de energía eléctrica (3.7 mA), este hecho resulta muy importante para las UAD cuando exista una falla en la alimentación principal.

El segundo punto común al diseño del equipo IPE y de las UAD, es la selección de la memoria de almacenamiento de información adquirida.

Tal como se determinó en las especificaciones de diseño, la memoria debe ser del tipo EEPROM, de tecnología CMOS y con una capacidad de 8 K bytes.

Para ello se seleccionó la memoria 28C64 de la marca THOMPSON (ó su equivalente en otras marcas), a continuación se enumeran algunas de sus características más relevantes (ref. 6).

1. Tiempo de acceso de 150 nano-segundos.
2. Bajo consumo de potencia.
 - Modo activo 25 mili-Amperes.
 - Modo "Standby" 100 micro-Amperes.
3. Operación de escritura simple, 5 mS para escritura de un byte.
4. Utiliza una sola fuente de alimentación (5 Volts).
5. Diseñada para soportar 10,000 ciclos de escritura /lectura.
6. Almacenamiento de información hasta por 10 años.

Este tipo de memorias, es de reciente introducción al mercado internacional, y no obstante que su precio actual es relativamente elevado (23 U.S.D./ unidad), se observa una rápida tendencia a la baja y una creciente competitividad entre diversos fabricantes, lo cual indica que su selección es apropiada desde el punto de vista de actualidad tecnológica y tiempo de obsolescencia esperado.

Una vez seleccionada la familia de microcontroladores a utilizar y la memoria de almacenamiento de información, se seleccionaron los periféricos compatibles y se procedió a configurar el arreglo electrónico correspondiente. En las siguientes secciones, se presentan las configuraciones electrónicas diseñadas y se describe en forma concreta su funcionamiento.

3.2 Diseño Electrónico de la UAD.

En la página siguiente (fig. 3.1), se presenta la configuración electrónica escogida para las UAD, como se puede observar en el mismo, el sistema fué

diseñado con una muy baja densidad de componentes, solamente 4 circuitos integrados, que son:

1. Microcontrolador (87C51).
2. Memoria EEPROM (28C64).
3. Reloj en tiempo real (58274).
4. Expansor de puertos (14174).

Para la descripción modular de este circuito, la UAD puede considerarse constituida por las siguientes secciones:

1. Unidad de Control y Comunicación.
2. Unidad de Captura de Datos.
3. Unidad de Almacenamiento de Datos.
4. Fuente Ininterrumpible de alimentación.

3.2.1 Unidad de Control y Comunicación.

Esta unidad formada básicamente por el microcontrolador 87C51 (U1), el reloj en tiempo real 58274 (U4), un foto-emisor y un foto-detector infrarrojos, constituye el "cerebro" de la UAD. Aquí se controlan todas las señales necesarias para el funcionamiento de la unidad, así como la comunicación con el equipo IPE.

El programa de operación está grabado en la memoria EPROM interna del microcontrolador y su descripción detallada se presenta en el capítulo No. 4.

El programa de operación, mantiene al microcontrolador en el estado "IDLE", que corresponde al mínimo consumo de energía eléctrica. En este estado el microcontrolador no ejecuta ninguna instrucción del programa, pero las siguientes condiciones son respetadas:

1. "Timer/ Counter (s)" en actividad.
2. Puerto de comunicaciones habilitado.
3. Interrupciones externas habilitadas. activadas.

4. Se mantiene la información anterior en:

"Stack Pointer."
"Program Counter."
"Program Status Word."
Acumulador.
Registros.
Memoria RAM.

El microcontrolador solamente puede abandonar el estado "IDLE" cuando se presenta una interrupción externa, que previamente haya sido habilitada y de acuerdo a las prioridades establecidas.

El programa de operación, solamente habilita dos posibles interrupciones:

1. Interrupción generada por el reloj en tiempo real, programada con prioridad 1 (máxima).
2. Interrupción generada en el puerto de comunicaciones, por la presencia de un byte transmitido desde el exterior.

El reloj en tiempo real (U4), está conectado al microcontrolador tal como se muestra en el diagrama esquemático de la fig. 3.1. Este dispositivo, responde a una base de tiempo generada por un cristal de cuarzo y una vez que se ha programado en su memoria interna los datos iniciales y que su operación ha sido iniciada (en la etapa final de construcción de la UAD o bien a su instalación), el conteo de tiempo es actualizado continuamente y la información correspondiente es almacenada en su memoria interna. (ref. 7).

Para asegurar la operación continua de este reloj, se le alimenta por medio de una pequeña batería de Litio (larga vida), que sumado a su bajo consumo de energía, asegura una operación ininterrumpida por 4 años.

La información que se maneja internamente en el reloj, corresponde a:

AÑO	(Unidades y decenas).
MES	(Unidades y decenas).
DIA	(Unidades y decenas).

HORA (Unidades y decenas).
MINUTOS (Unidades y decenas).
SEGUNDOS (Unidades y decenas).

Esta información se encuentra almacenada en localidades de memoria, que pueden ser leídas por el microcontrolador. Las direcciones de la localidad de memoria que se desea leer, son colocadas en las líneas AD0-AD3 las cuales son manejadas por las líneas P1.4-P1.7 de entrada/ salida del microcontrolador y la información contenida en la localidad de memoria seleccionada, es presentada al exterior en los pines DB0-DB3, de donde puede ser leída por las líneas P0.0-P0.3 de entrada salida del microcontrolador a las que están conectadas.

El flujo de información es bidireccional, lo que significa que la información contenida en las localidades de memoria del reloj en tiempo real, pueden ser leída o modificada de acuerdo al programa de operación del sistema.

La selección del reloj como periférico del microcontrolador, así como la dirección del flujo de la información, es controlada por tres líneas más de entrada/ salida P2.1,P2.2 y P2.5 tal y como se muestra en el diagrama esquemático.

La lectura o programación de un dato del reloj, se realiza de acuerdo al diagrama de tiempos mostrado en la figura 3.2.

El reloj en tiempo real, tiene la capacidad de generar una señal con un período programable. En este caso, se programa la generación de dicha señal cada minuto.

Esta señal es utilizada como fuente de interrupción externa al microprocesador P3.2, lo que hará que el microcontrolador sea "despertado" del modo "Idle" y atienda la interrupción, realizando una subrutina que corresponde al monitoreo de la línea de entrada salida P3.3, en la que se tendrá un nivel alto en el caso de que la alimentación principal esté presente, y un nivel bajo en caso contrario.

La ausencia de alimentación principal, es contada - con una resolución de un minuto -, y dicha cuenta es almacenada en un registro interno. Así mismo, el microcontrolador mantiene un conteo interno de interrupciones generadas por el reloj, la cual es comparada en cada evento con el número 15 (15 minutos).

Cuando ha transcurrido un periodo de 15 minutos, el microcontrolador procede a realizar una subrutina de captura, manejo y almacenamiento de datos.

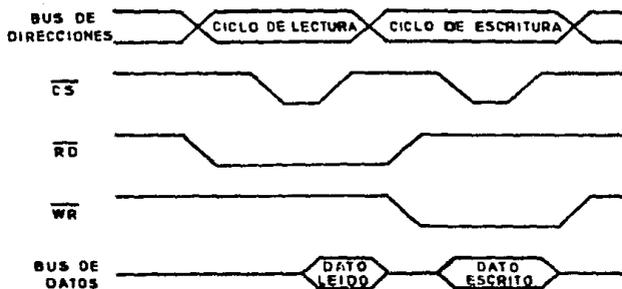


Figura 3.2: Diagrama de tiempos del reloj.

Por lo que respecta al puerto de comunicaciones, las líneas P3.0/RXD y P3.1/TXD, corresponden a las líneas de recepción y transmisión respectivamente, del puerto de comunicaciones tipo serie integrado al microcontrolador.

La línea correspondiente a recepción (P3.0), está conectada al colector de un foto-detector infrarrojo (FT), en donde se presentarán niveles altos o bajos dependiendo del estado de corte o saturación en que trabaje el foto-detector, atendiendo a la presencia o ausencia de un haz de luz de espectro infrarrojo, emitido por un foto-emisor conectado a la línea de transmisión en el equipo IPE.

Las señales generadas por el foto-detector, son detectadas en la línea de recepción del microcontrolador y almacenadas en un registro de corrimiento interno de 8 bits, que constituye un registro de recepción.

Una vez que el registro de recepción almacena 8 bits o un Byte, se genera automáticamente una interrupción que ocasiona que el microcontrolador "despierte" de su modo "Idle" y realice una subrutina de lectura del byte recibido. De acuerdo al programa de operación, dicho byte es comparado con los códigos válidos y se procede a realizar las subrutinas correspondientes a los modos de INTERROGACION, ACCESO, TRANSFERENCIA Y REPROGRAMACION, de acuerdo a las especificaciones funcionales determinadas en el capítulo No. 2.

La transmisión de datos es realizada por medio un foto-emisor (FE) de espectro infrarrojo, que es manejado por un transistor (Q1) tal como se ilustra en el diagrama esquemático.

La velocidad de comunicación de la UAD, fue programada a 1200 BAUDS y de acuerdo a las características propias del microcontrolador, se requirió el uso de un cristal de cuarzo para una frecuencia de oscilación de 11.059 MHz.

La comunicación programada para la UAD, tiene las siguientes características:

- Comunicación Asíncrona.
- Un bit de inicio.
- 8 bits de longitud de palabra.
- Un bit de parada.

3.2.2 Unidad de Captura de Datos.

Como se mencionó en las especificaciones de diseño, el dato de energía eléctrica, será adquirido por medio del conteo de pulsos proporcionados por el "generador de pulsos" que se integrará a la UAD.

El "generador de pulsos", es exitado por medio del voltaje V_{cc} , el cual como se puede apreciar en el diagrama esquemático (fig. 3.1), estará presente solamente en el caso de que la alimentación principal este a su vez presente.

Este hecho tiene por objeto ahorrar energía de la batería correspondiente a la fuente ininterrumible de la UAD, ya que el arreglo de foto-emisores y foto-detectores del "generador de pulsos" presenta un relativo alto consumo de energía cuando está activado. Obviamente si la alimentación principal no está presente, no girará el disco del Watt-hourmetro y no se producirá generación de pulsos, por lo que el efecto es el mismo al desactivar la excitación del "generador de pulsos".

La salida del "generador de pulsos" es conectada directamente a la línea P3.4 del microcontrolador que corresponde a uno de sus contadores internos (T0).

Tal como se mencionó en la descripción de la Unidad de Control y Comunicación, el microcontrolador permanece en modo "Idle", pero el contador no es desactivado, por lo que todo pulso presente en la línea P3.4 será contado y la cuenta acumulativa será almacenada en un registro de 16 bits correspondiente al registro del contador T0.

Cuando un período de 15 minutos finaliza, la cuenta acumulativa es leída y el contador es inmediatamente limpiado para iniciar un nuevo conteo.

El dato capturado de energía eléctrica, junto con el correspondiente al tiempo de interrupción, es preparado para su almacenamiento en la memoria de estado sólido tal como se detalla en la siguiente sección.

3.2.3 Unidad de Almacenamiento de Información.

Esta unidad está formada por el microcontrolador (U1) y la memoria EEPROM 28C64 (U3). La memoria se conecta al microcontrolador tal y como se muestra en el diagrama esquemático.

Las direcciones de la localidad de memoria en donde se desea almacenar el dato (1 byte), son colocadas en las líneas A0-A12, para lo que se utiliza uno de los puertos del microcontrolador P1 (8 líneas), expandido por un circuito 14174 (U2) que almacenará temporalmente los cinco bits más significativos de la dirección a programar.

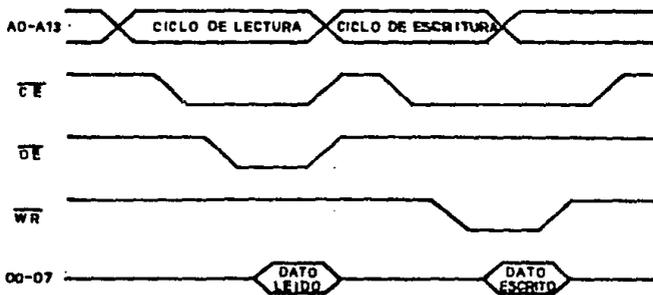


Figura 3.3: Diagrama de tiempos de la memoria.

Los datos se transfieren del microcontrolador a la memoria o viceversa, por medio del "bus" (P0) de datos que en este caso será utilizado como un puerto adicional.

Por medio de 3 líneas más de entrada salida P2.0, P2.1 y P2.2, se generan las señales de control para selección de la memoria, modo de programación o modo de lectura.

La programación de un byte en la memoria EEPROM, se realiza atendiendo al diagrama de tiempos requerido, el cual se ilustra en la figura No. 3.3.

3.2.4 Fuente Ininterrumpida de Alimentación.

La fuente ininterrumpida de alimentación empleada en la UAD, mantendrá al sistema en operación continua, hasta por 10 horas en caso de falta de energía en la línea medida (baja tensión).

Esta fuente, se muestra en el diagrama esquemático y consta de un transformador (T1), un puente de diodos (PD), un cargador de baterías y un regulador de voltaje (REG). El voltaje de alimentación empleado por la U.A.D. es de 5 V.c.d., debido a que este valor es necesario para la operación de la memoria EEPROM seleccionada.

El primario del transformador es conectado a la línea de alimentación medida (en baja tensión), lo que para el Watthorimetro utilizado en este diseño, corresponde a 125 V.c.a. El voltaje del devanado secundario del transformador y la corriente que debe soportar se calcularon con las siguientes relaciones:

$$V_{ac} = \frac{(V_{out} + V_{reg} + V_{rect} + V_{rizo}) V_{nom}}{0.92 V_{ll} \sqrt{2}}$$

Donde:

- 0.92 = Eficiencia típica del rectificador.
- V_{out} = Voltaje de salida en c.d. (5 volts).
- V_{reg} = Diferencia de voltajes de salida y entrada en el regulador. (3 Volta).
- V_{rect} = Voltaje en cada diodo del puente rectificador (1.25 Volts).
- V_{rizo} = Voltaje de rizo (10 % V_{out}).
- V_{nom} = Voltaje de línea (125 V.c.a.)
- V_{ll} = Voltaje mínimo de línea (100 V.c.a.)
- V_{ac} = Voltaje en el secundario del transformador.

$$I_{rms} = 1.8 \times I_{pl}$$

Donde:

- I_{pl} = Corriente a plena carga.

I_{rms} = Corriente nominal en el secundario.

Para obtener un voltaje regulado de 5 V.c.d. y proporcionar una corriente nominal de 150 mA, se obtiene como resultado $V_{ac} = 10$ V.c.a. y $I_{rms} = 270$ mA, aplicando un 10% de tolerancia a la corriente, se obtiene $I_{rms} = 300$ mA.

El cargador de baterías, consiste en un arreglo de diodos. Cuando el voltaje de la batería es menor que el del diodo zener D5, hay un flujo de corriente a través del diodo D2 hacia la batería. Cuando el voltaje de la batería se aproxima al voltaje del diodo zener, lo suficiente para no polarizar en forma directa al diodo D2, cesa el flujo de corriente hacia la batería.

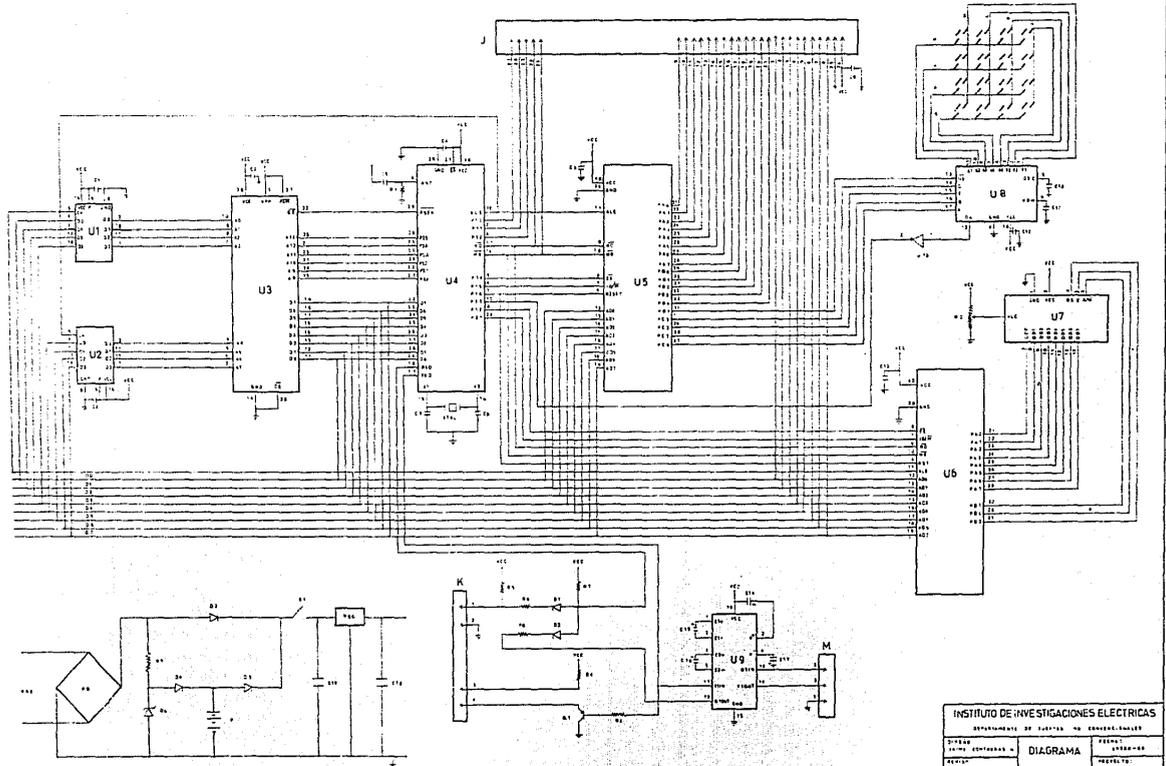
Cuando existe suministro de energía eléctrica principal, el transformador, puente rectificador y regulador alimentarán a la UAD. En estas condiciones el diodo D1 conduce mientras que el diodo D3 no lo hace. Cuando falta la alimentación principal, la batería alimentará a la UAD, conduciendo el diodo D3, mientras que D1 y D2 no lo hacen.

La batería seleccionada corresponde a una del tipo 9V de construcción Ni-Ca, cuyo voltaje de operación es de 7.5 volts y proporciona una capacidad de 70 mA-h.

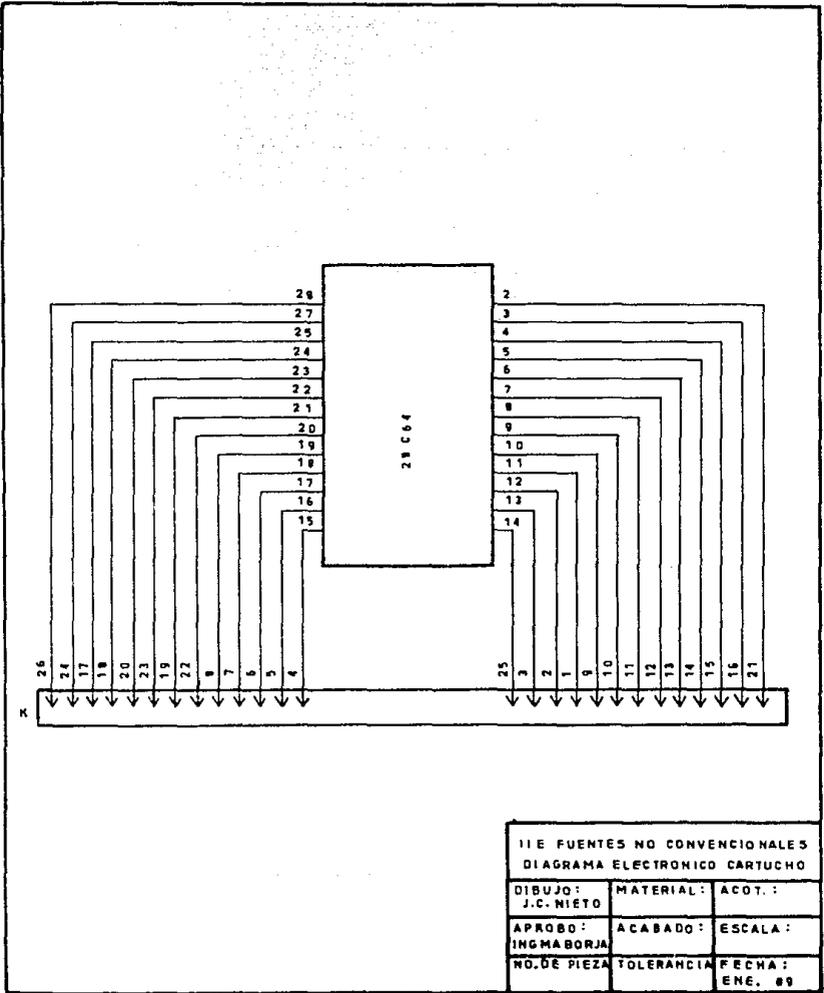
3.3 Diseño Electrónico del Equipo IPE.

En la página siguiente (fig. 3.4), se presenta la configuración electrónica dada del equipo IPE. Los circuitos y dispositivos principales que fueron seleccionados para el diseño electrónico son:

1. Microcontrolador 80C31 (U4).
2. Memoria EPROM, de 16 K bytes, 27C128 (U3).
3. Expansor de puertos 14042 (U1 y U2).
4. Expansor de puertos 81C55 (U5 y U6).
5. Decodificador de teclado 54C922 (U8).
6. Display de cristal líquido M1641LCDM (U7).
7. Teclado matricial de 4X4.
8. Alimentador/Receptor de línea MAX232 (U9).



INSTITUTO DE INVESTIGACIONES ELECTRICAS		
DEPARTAMENTO DE ELECTRONICA Y COMUNICACIONES		
TITULO	DIAGRAMA	FECHA
AUTORES	ELECTRONICO	1988-88
REVISOR		
APROBADO		
FECHA DE ENTREGA	1988-1-11-E	NO. PAGOS
		1.0



11 E FUENTES NO CONVENCIONALES		
DIAGRAMA ELECTRONICO CARTUCHO		
DIBUJO: J.C. NIETO	MATERIAL:	ACOT.:
APROBO: INGMABORJA	ACABADO:	ESCALA:
NO. DE PIEZA	TOLERANCIA	FECHA: ENE. 89

9. Inversor 4069 (U10).

10. Cartuchos removibles de 8 K bytes EEPROM, 28C64.

Para su descripción, el IPE puede considerarse formado por las siguientes unidades:

1. UNIDAD MINIMA.
2. INTERFAZ HOMBRE - MAQUINA.
3. UNIDAD DE COMUNICACION.
4. UNIDAD DE ALMACENAMIENTO DE DATOS.
5. FUENTE DE ALIMENTACION ININTERRUMPIDA.

En la figura 3.5, se muestra un diagrama a bloques de las unidades que forman el equipo y la descripción del funcionamiento de las mismas, se presenta a continuación.

3.3.1 Unidad Mínima.

Esta unidad, constituye el "cerebro" del equipo, realizando todas las funciones de control necesarias para su operación.

La unidad de control, básicamente esta formada por el microcontrolador 80C31 (U4), la memoria EPROM 27C128 (U3), los circuitos 14042 (U1) y (U2) y los expansores de puertos 81C55 (U5 y U6).

El programa operativo, está contenido en la memoria EPROM y el microcontrolador se configura para operación con memoria externa, para lo cual la línea EA (pin 31) es conectada permanentemente a Vcc.

El microcontrolador, direcciona a la memoria externa por medio del puerto P2 y de los circuitos U1 y U2. Cuando el 80C31 se configura para memoria externa, el direccionamiento se realiza por medio de un registro de 16 bits que corresponde al contador de programa.

Durante un ciclo de búsqueda de memoria externa, los ocho bits más significativos de la dirección (A8-A15), aparecen en el puerto P2, y permanecen ahí hasta el final del ciclo.

Los ocho bits menos significativos de la dirección, aparecen solamente durante un corto tiempo (respecto a la duración del ciclo de búsqueda de memoria externa) en el "bus" de datos, que corresponde al puerto P0 (marcado en el diagrama esquemático como DB0-DB7), por lo que deben almacenarse en registros que mantengan la información. Para esto son utilizados

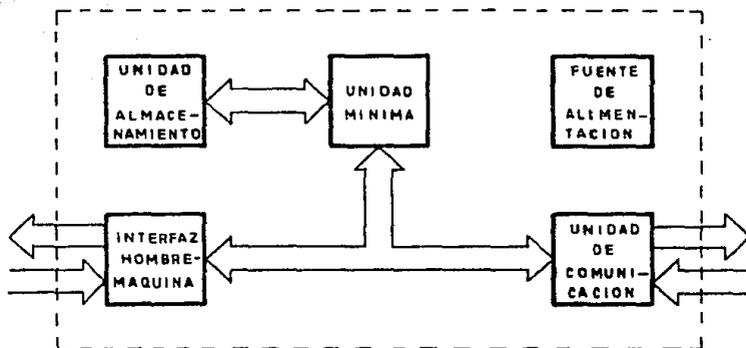


Figura 3.5: Diagrama a bloques de las unidades del equipo IPE.

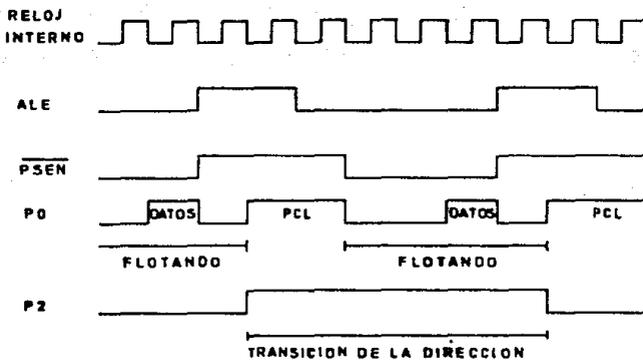


Figura 3.6: Diagrama de tiempos de la memoria externa.

los circuitos U1 y U2 que son "FLIP FLOPS" tipo D (cuatro por circuito), con reloj común.

La entrada de datos de los "flip flops", se conecta directamente al "bus" de datos, mientras que la entrada común del reloj se conecta a la señal ALE generada automáticamente por el microcontrolador, cuya función principal es precisamente permitir el almacenamiento de los 8 bits menos significativos de la dirección en registros externos.

Una vez direccionada la memoria, su "bus" de salida es habilitado por medio de la señal PSEN - generada automáticamente por el microcontrolador - y la instrucción es leída y ejecutada.

El ciclo de búsqueda de instrucciones en la memoria externa se ilustra en el diagrama de tiempos mostrado en la figura 3.6.

Los expansores de puertos 81C55 (U5 y U6), son utilizados para ampliar la capacidad de las líneas de entrada/salida del microcontrolador. Cada uno de estos circuitos, tiene 2 puertos de 8 líneas y uno de 4 líneas, lo que

proporciona una capacidad de expansión de 40 líneas adicionales.

Los puertos de dichos expansores, son utilizados para el manejo de los periféricos que se conectan al microcontrolador. La información o estado de dichos puertos, puede ser leída o modificada tal como se hace con una memoria de datos externa. Cada que se desea leer o escribir algún dato en dichos puertos, se selecciona el circuito correspondiente por medio de las señales CE ("chip enable") y mediante una instrucción de lectura de memoria externa (MOVX# RX), el puerto seleccionado es direccionado a través del "bus" de datos del microcontrolador, iniciándose un ciclo de lectura o escritura en dicho puerto.

La información correspondiente a el dato leído o escrito en los puertos seleccionados, es transferida al microcontrolador a través del "bus" de datos.

El programa operativo, mantiene al microcontrolador en modo "Idle" al igual que en la UAD y en este caso, solo puede ser "despertado" por interrupciones externas generadas por la INTERFAZ HOMBRE-MAQUINA.

3.3.2 Interfaz Hombre-Máquina.

Para que el equipo IPE realice alguna función, es necesaria una orden generada por el operador. El dar a conocer estas órdenes al microcontrolador e indicar al operador las acciones tomadas al respecto, así como la petición de información adicional para su ejecución, es la función de la interfaz hombre-máquina.

Esta interfaz está formada por un teclado matricial de 16 teclas (4X4), un decodificador de teclado 54C922 (U8), y display de cristal líquido M1641 LCDM (U7).

Cuando una tecla es oprimida por el operador, el decodificador de teclado (U8) almacena en un registro interno de 4 bits, un código binario que corresponde al número de tecla oprimida, y genera automáticamente una señal en su línea DA que está conectada a través de un inversor - a la línea P3.2 del microcontrolador que corresponde a uno de sus puertos de interrupción externa.

De esta manera, el microcontrolador es "despertado" y procede a realizar una subrutina de servicio que consiste en la lectura del código de la tecla oprimida.

Para ello activa la salida del decodificador de teclado, haciendo uso del expansor de puertos U5. De esta manera, la línea PB7 del expansor es puesta en nivel bajo y la información contenida en el registro interno del decodificador de teclado es presentada en las líneas A,B,C,D.

Esta información es leída por el microcontrolador, haciendo uso nuevamente del expansor de puertos U5. El código de la tecla oprimida es comparado con los códigos válidos de acuerdo al programa operativo y en su aceptación se procede a realizar la subrutina correspondiente a la orden proporcionada.

Los mensajes generados por el microcontrolador al operador, son mostrados en el "display" alfanumérico (U7), el cual es a su vez manejado a través del expansor de puertos U6.

El display seleccionado tiene las siguientes características (ref. 8):

- Bajo consumo de energía (Cristál líquido).
- Alto contraste.
- Amplio ángulo de visión.
- 16 caracteres por 1 línea.
- Alfanumérico.
- Posee una matriz de 5X8 + 1 cursor.
- Alimentación con una sola fuente de poder (5 Volts).

Este "display" está construido en una pequeña tarjeta de circuito impreso, en donde tiene integrado un circuito que realiza las funciones de control. El "display" es manejado como una memoria RAM, tal como se observa en el diagrama esquemático. el "display" cuenta con un "bus" de datos (DB0-DB7) y tres señales de control (RS, E.R, W).

Cuando se desea escribir un caracter en una de las 16 posibles posiciones, se envía por el "bus" la dirección de la posición a escribir y en seguida el código del caracter que se desea mostrar. Para ello, las señales de control y la información, son manejadas de acuerdo a los diagramas de tiempos de las figuras No. 3.7A y No. 3.7B.

3.3.3 Unidad de Comunicación.

En el equipo IPE, la Unidad de Comunicación, tiene dos funciones:

1. Manejar la comunicación con la UAD.
2. Manejar la comunicación con un computador personal.

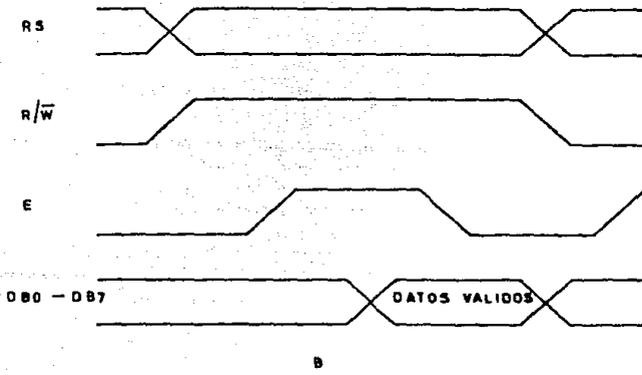
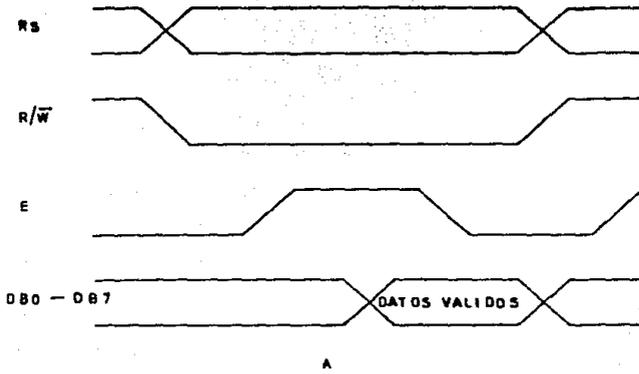


Figura 3.7: Diagramas de tiempos del display.

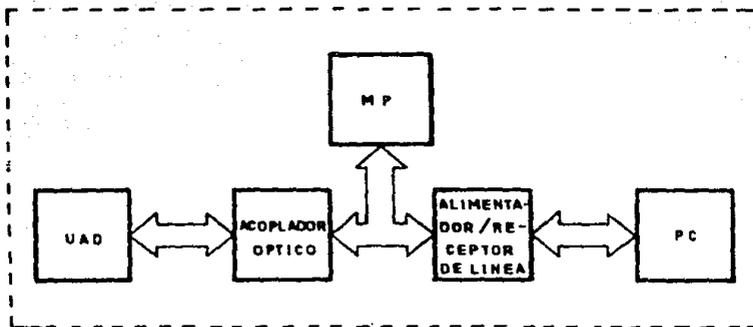


Figura 3.8: Diagrama a bloques de la unidad de comunicación.

El diagrama a bloques de esta unidad se muestra en la figura No. 3.8 como se puede observar en dicha figura. la unidad está formada por el microcontrolador 80C31 (U7), un circuito alimentador receptor de línea MAX 232 (U9) y un arreglo de foto-emisor y foto-detector infrarrojos.

Para la comunicación con la UAD. la línea de transmisión del puerto serie del microcontrolador (TXD), es conectada a la base de un transistor 2N2907, que manejará al foto-emisor infrarrojo dependiendo del estado de corte o saturación en que opere, en función al nivel lógico transmitido por el microcontrolador.

La línea de recepción del puerto serie del microcontrolador, se conecta directamente al colector del foto-detector infrarrojo. el cual proporcionará un nivel lógico alto o bajo, dependiendo de la incidencia o no de un haz de luz de espectro infrarrojo, sobre su base. Al igual que para la UAD, un ciclo de recepción es iniciado por una transición de "1" a "0" en la línea de recepción, lo cual es interpretado como el bit de inicio. Las transiciones

siguientes son detectadas y almacenadas en un registro de corrimiento hasta completar 1 byte.

Cuando el registro de corrimiento contiene el byte transmitido y se detecta el bit de parada, automáticamente se genera una señal de interrupción, que indica al microcontrolador que una subrutina de servicio deberá ser realizada para la lectura e interpretación del byte recibido.

Al igual que para la UAD, las características de la comunicación son las siguientes:

1. Velocidad de comunicación 1200 Bauds.
2. Longitud de palabra, 8 bits.
3. Un bit de inicio.
4. Un bit de parada.
5. Comunicación asíncrona.

Estas características se utilizan también para la comunicación con el microcomputador personal.

El microcontrolador, realiza funciones internas que permiten la detección de pulsos falsos de inicio y eliminación de ruido, lo cual es ampliamente explicado en la ref. 5, página 7-20.

Para la comunicación con el computador personal, se utiliza el mismo puerto serie del microcontrolador. La diferencia es que en lugar de señales de comunicación del tipo óptico, se emplean señales eléctricas con niveles de voltaje de +9 V.c.d. para el estado lógico "1" y -9 V.c.d. para el nivel lógico "0".

La función del alimentador/receptor de línea MAX 232 (U9), es la de transformar los niveles de voltaje generados por el microcontrolador para transmisión (+5/0 V.c.d.) a (-9 V.c.d) para su alimentación a la línea de recepción del computador personal y viceversa.

Las características principales del circuito MAX 232, se presentan a continuación (ref. 9):

- Opera con una sola fuente de alimentación (5 V.c.d.).
- Cumple con las especificaciones RS-232 y V.28
- Tiene múltiple transmisor y receptor.

- Posee un convertidor CD a CD.
- +/- 9 V.c.d. de salida con +/- 5 V.c.d. de entrada.
- Soporta +/- 30 V.c.d. de nivel de entrada.
- Salidas "tri-state" compatibles con TTL y CMOS.
- Bajo consumo de energía.

3.3.4 Unidad de Almacenamiento de Información.

Esta unidad tiene como función principal almacenar en un cartucho de memoria de estado sólido, la información que es transmitida por una UAD cuando el operador genera la orden de modo de TRANSFERENCIA.

Esta unidad esta formada por el microcontrolador (U4), un expensor de puertos (U5) y una memoria EEPROM (U10).

La memoria EEPROM está contenida dentro de un "cartucho" removible que se conecta al IPE por medio del conector J1. Una vez hecha la conexión, las líneas de direccionamiento de la memoria, quedan conectadas a los puertos PA y PB del expensor U5 y las correspondientes al "bus" de datos quedan conectadas al "bus" de datos del microcontrolador.

Las líneas de control WE,OE,CE y R/B, quedan conectadas directamente a las líneas de entrada/salida P1.1, P1.2, P1.3 y P1.0 respectivamente.

Tal y como se mencionó en las especificaciones de diseño del capítulo 2, la información almacenada en la UAD, se transfiere al IPE a través del puerto de comunicación óptica, estableciendo un protocolo de petición y envío de 1 byte.

De esta manera, el microcontrolador del IPE, genera una petición de envío de 1 byte a la UAD y queda en espera de la respuesta. Cuando recibe el byte solicitado por medio del puerto de comunicaciones, el microcontrolador realiza una rutina de almacenamiento del byte recibido en el "cartucho" de memoria conectado.

Ya que la memoria EEPROM utilizada es igual a la memoria de la UAD, el control y manejo de las señales se realiza de acuerdo al diagrama de tiempos de la figura 3.3.

3.3.5 Fuentes de Alimentación Ininterrumpible.

De acuerdo a las especificaciones de diseño mencionadas en el capítulo 2, la fuente de alimentación del equipo IPE, debe proporcionar la facilidad de

operación con una línea convencional de 110 V.c.a. y la opción de operación con baterías con una autonomía de hasta 2.5 horas, operando en el modo de transferencia de información.

Este tiempo fué determinado para la lectura de la información contenida en 10 UAD (6 minutos) y un tiempo adicional para interrogación y reinicialización, con un total estimado de 15 minutos por unidad.

En el modo de transferencia de datos, se espera un consumo máximo de 300 mA. de acuerdo al consumo individual de cada componente.

En el mercado, existe un tipo de baterías recargables tipo GEL/CELL, que proporcionan una considerable capacidad de corriente respecto a su tamaño y que tienen un costo relativamente bajo respecto a las de Ni-Ca.

Estas baterías son de la marca Globe-Union y el modelo seleccionado se describe a continuación:

Modelo.	GC-610-1
Voltaje.	6 V.c.d.
Capacidad.	0.9 A-h.
Dimensiones.	5.08 X 4.91 X 5.68 (Cm.)
Precio.	16.50 U.S.D.

No es posible obtener mejores características al considerar arreglos de baterías Ni-Ca.

Una batería de este tipo, será suficiente para obtener la autonomía deseada.

La fuente de alimentación para el equipo IPE, tiene un arreglo similar a la de las UAD con las siguientes diferencias:

1. Batería de mayor capacidad.
2. Transformador externo (Transformador de pared).
3. Voltaje de trabajo 6.0 V.c.d.

El funcionamiento del arreglo correspondiente, fué descrito en la sección 3.2.4 de este capítulo.

3.4 Lista de Partes.

En la presente sección, se presenta la información relativa a las listas de partes que forman las Unidades de Adquisición de Datos y el equipo de Interrogación/ Programación Externa.

En dichas listas, se incluyen los datos necesarios para su localización en los diagramas esquemáticos, así como para su adquisición.

La información contenida en dichas tablas, se presenta según el siguiente código:

- A Número de parte.
- B Descripción (español/inglés).
- C Clave de localización en el plano.
- D Cantidad requerida.
- E Número de parte comercial.
- F Fabricante.
- G Precio unitario.
- H Precio total.
- I Tipo de moneda (D = U.S.D.).
- J Mes y año de cotización.
- K Proveedor de acuerdo a lista anexa.
- L Catálogo de acuerdo a lista anexa.
- M Número de página de catálogo.

3.4.1 Lista de Proveedores.

- 1.- KLEIGO SUPPLY, INC.
2418 Treadslow Lane
Houston, Texas 77067
U.S.A.
Tel. (713) 893-4681

- 2.- DISTRIBUIDORA SAGO, S.A.
Rep. del Salvador 20-F
06000 México D.F.
Tel. 510-8845

- 3.- HEHCA ELECTRONICA, S.A. DE C.V.
Eje Central 50, local 37
06000 México, D.F.
Tel. 518-6437

Los dispositivos marcados como proveedor "A"
son de uso general y pueden adquirirse en
establecimientos de refacciones electronicas.

3.4.2 Lista de Catálogos.

- 1.- INTEL COMPONENT DATA CATALOG.
Intel Corporation.
Literature Department.
3065 Bowers Avenue,
Santa Clara, Ca. 95051
U.S.A.

- 2.- VOLTAJE REGULATOR HANDBOOK.
National Semiconductor Corporation.
2900 Semiconductor Drive.
Santa Clara, California 95051
U.S.A.

- 3.- MOTOROLA CMOS DATA BOOK.
CMOS Marketing Dep.
Motorola Semiconductor Products Inc.
3501 Ed. Bluestein Blvd.
Austin, Texas 78721
U.S.A.

- 4.- NEWARK ELECTRONICS CATALOG.
4801 N. Ravenswood
Chicago, IL. 60640
U.S.A.

- 5.- MEMORIES DATA BOOK.
THOMSON COMPONENTS MOSKET CORPORATION.
1310 Electronics Drive.
Carrollton, Texas 75006
U.S.A.

- 6.- COMPONENT HANDBOOK.
National Semiconductor Corp.
2900 Semiconductor Drive.
Santa Clara, Ca. 95051
U.S.A.

- 7.- THE TRANSISTOR AND DIODE DATA BOOK.
Texas Instrument Inc.
P.O. BOX 3640, M.S. 84
Dallas, Texas 75221
U.S.A.

- 8.- THE OPTOELECTRONICS DATA BOOK
Texas Instruments Inc.
P.O. BOX 3640, M.S. 84
Dallas, Texas 75221
U.S.A.

- 9.- MCS-85 USER'S MANUAL
Intel Corporation
P.O. Box 58130
Santa Clara, CA. 95052-8130
U.S.A.

- 10.- MOS MEMORY PRODUCTS DATA BOOK
Toshiba America, Inc.
13111 North Central Expy
Dallas, Texas.
U.S.A.

- 11.- CMOS DATA BOOK
National Semiconductor Corp.
P.O. Box 80876
Sunnyvale, CA. 94088
U.S.A.

- 12.- AMP INC.
AMPLITE CATALOG
Harrisburg Pennsylvania 17105
U.S.A

- 13.- BERG ELECTRONICS CATALOG (A,B)
New Cumberlan, PA. 17070
U.S.A.

- 14.- COMERCIAL/AEROSPACE SWITCHES & RELAYS
Cutler Hammer Speciatly Products Groups.
Milwakee, Wis. 53201
U.S.A.

- 15.- ENGINEERING CATALOG
Gray Hill Inc.
561 Hillgrove Ave. P.O. Box 10373
Grange, Illinois
U.S.A.

TABLA 3.1

LISTA DE PARTES ELECTRONICAS UAD-1-III

A	B	C	D	E	F	G	H	I	J	K	L	M
	CIRCUITOS INTEGRADOS INTEGRATED CIRCUITS											
CEL01	MICROCOMPUTADOR CON EPROM INTEGRADA. SINGLE CHIP EPROM MICROCOMPUTER.	U1	1	MCS87C51H	INTEL	63.25	63.25	D	ENE/89	1	1	7-1
CEL02	MEMORIA EEPROM DE 8 KBYTES. 8 KBYTES ELECTRICALLY ERASABLE PROM.	U2	1	TS28C64	THOMPSON	23.01	23.01	D	ENE/89	1	5	2-53
CEL03	FLIP FLOP SEXTUPLE TIPO D. HEX D FLIP FLOP.	U3	1	MC14174	MOTOROLA	0.82	0.82	D	ENE/89	2	3	2-198
CEL04	RELOJ DE TIEMPO REAL. REAL TIME CLOCK.	U4	1	MM58274	NATIONAL SEMICONDUC	9.08	9.08	D	ENE/89	1	6	7-113
CEL05	REGULADOR DE VOLTAJE POSITIVO 5 VOLTS. 3-TERMINAL POSITIVE REGULADOR.	REG	1	LN7805	NATIONAL SEMICONDUC	0.56	0.56	D	ENE/89	3	2	10-07
CEL06	PUNTE RECTIFICADOR DE DIODOS. BRIDGE DIODE RECTIFIER.	PD	1	MDA100A	MOTOROLA	2.10	2.10	D	ENE/89	1	4	27

TABLA 3.2

LISTA DE PARTES ELECTRONICAS	WAD-1-III
------------------------------	-----------

A	B	C	D	E	F	G	H	I	J	K	L	M
	CAPACITORES CAPACITORS											
CEL07	MONOLITICOS DE CERAMICA 0.1 uF/50V. MONOLITHIC CERAMIC 0.1 uF/50V.	C1, C2 C3, C4	4	C22BC-104H	SPRAGUE	0.08	0.16	D	ENE/89	A	4	171
CEL08	CERAMICO 22pF. CERAMIC 22pF.	C8, C9	2	SEGUN MARCA	INDIFER.	0.07	0.14	D	ENE/89	A	4	170
CEL09	CERAMICA 33pF. CERAMIC 33pF.	C5, C6	2	SEGUN MARCA	INDIFER.	0.07	0.14	D	ENE/89	A	4	170
CEL10	ELECTROLITICO 10 uF/160. ELECTROLYTIC 10 uF/160.	C7	1	UTL-10516	MALLORY	0.15	0.15	D	ENE/89	A	4	174
CEL11	ELECTROLITICO 220 uF/250. ELECTROLYTIC 220 uF/250.	C10, C11	2	TE1213	SPRAGUE	0.25	0.50	D	ENE/89	A	4	175
	RESISTENCIAS FIJAS. FIXED RESISTOR.											
CEL12	0.2 KOHMS 1/4W 5%.	R1	1	SEGUN MARCA	INDIFER.	0.06	0.06	D	ENE/89	A	-	-

TABLA 3.3

LISTA DE PARTES ELECTRONICAS UAD-1-III

A	B	C	D	E	F	G	H	I	J	K	L	M
CEL13	220 OHMS 1/4W 5x	R2	1	SEGUN MARCA	INDIFER.	0.06	0.06	D	ENE/89	A	-	-
CEL14	1 KOHMS 1/4W 5x	R3,R8	2	SEGUN MARCA	INDIFER.	0.06	0.12	D	ENE/89	A	-	-
CEL15	1.2 KOHMS 1/4W 5x	R4	1	SEGUN MARCA	INDIFER.	0.06	0.06	D	ENE/89	A	-	-
CEL16	5.6 KOHMS 1/4W 5x	R5	1	SEGUN MARCA	INDIFER.	0.06	0.06	D	ENE/89	A	-	-
CEL17	10 KOHMS 1/4W 5x	R6,R7	2	SEGUN MARCA	INDIFER.	0.06	0.12	D	ENE/89	A	-	-
CEL18	10 KOHMS 1/4W 5x	R9 R10	2	SEGUN MARCA	INDIFER.	0.06	0.12	D	ENE/89	A	-	-
CEL19	<p>DIODOS. DIODE.</p> <hr/> <p>RECTIFICADOR DE USO GENERAL. GENERAL PURPOSE RECTIFIER.</p>	D1,D2 D3	3	1N4001	TEXAS INS- TRUMENTS.	0.09	0.27	D	ENE/89	A	7	10-32
CEL20	<p>DIODO REGULADOR DE VOLTAJE. VOLTAGE REGULATOR DIODES.</p>	D4	1	1N4732	TEXAS INS- TRUMENTS.	0.25	0.25	D	ENE/89	2	7	10-52
CEL21	<p>DIODO REGULADOR DE VOLTAJE. VOLTAGE REGULATOR DIODE.</p>	D5	1	1N4739	TEXAS INS- TRUMENTS.	0.25	0.25	D	ENE/89	2	7	10-52

TABLA 3.4

LISTA DE PARTES ELECTRONICAS MAD-1-11X

A	B	C	D	E	F	G	H	I	J	K	L	M
	INFRAROJOS. INFRARED DEVICES.											
CEL22	EMISOR DE INFRAROJO. INFRARED EMITER.	FE	1	TIL 38	TEXAS INSTRUMENTS.	1.88	1.88	D	ENE/89	2	8	5-26
CEL23	FOTOTRANSISTOR. PHHOTOTRANSISTOR.	FT	1	TIL 99	TEXAS INSTRUMENTS.	1.88	1.88	D	ENE/89	3	8	5-110
	BASES PARA CIRCUITOS INTEGRADOS. INTEGRATED CIRCUITS DIP SOCKETS.											
CEL24	BASE DE 48 PINES. DIP SOCKET 48 PINES.	U1	1	248-AG39D	AUGAT	8.44	8.44	D	ENE/89	1	4	425
CEL25	BASE DE 28 PINES. DIP SOCKET 28 PINES.	U2	1	228-AG39D	AUGAT	8.39	8.39	D	ENE/89	1	4	425
CEL26	BASE DE 16 PINES. DIP SOCKET 16 PINES.	U3,U4	2	216-AG39D	AUGAT	8.24	8.48	D	ENE/89	1	4	425
	CRISTALES. CRYSTAL.											
CEL27	CRISTAL DE CUARZO DE 11.859 MHZ. 11.859 MHZ CRYSTAL.	X1	1	-	CRICUSA	13.85	13.85	D	ENE/89	-	-	-

TABLA 3.5

LISTA DE PARTES ELECTRONICAS UAD-1-11E

A	B	C	D	E	F	G	H	I	J	K	L	M
CEL28	CRISTAL DE CUARZO DE 32.768 KHZ. 32.768 KHZ CRISTAL.	X2	1	-	-	0.52	0.52	D	ENE/89	2	-	-
CEL29	VARIOS. <hr/> TRANSFORMADOR 117/120. 117/120V. TRANSFORMER.	T1	1	-	-	2.17	2.17	D	ENE/89	2	-	-
CEL30	BATERIA RECARGABLE. RECHARGABLE BATTERY.	B	1	-	-	15.21	15.21	D	ENE/89	3	-	-
CEL31	TRANSISTOR PNP. PNP TRANSISTOR.	Q1	1	2N2907	TEXAS INSTRUMENT	0.44	0.44	D	ENE/89	2	7	4-151
CEL32	INICIADOR DE PULSOS.	-	1	-	11E	-	-	-	-	-	-	-
CEL33	TARJETA DE CIRCUITO IMPRESO. CIRCUIT PRINT BOARD.											
CEL34	CONECTOR MACHO 2P. STROING SINGLE ROW.	-	1	65514	BERG	1.96	1.96	D	ENE/89	1	13A	15
CEL35	CONECTOR HEMBRA DOBLE HORIZONTAL. DOUBLE ROW HORIZONTAL	-	1	65781-814	BERG	2.53	2.53	D	ENE/89	1	13B	9

TABLA 3.6

LISTA DE PARTES ELECTRONICAS IPE-1-III

A	B	C	D	E	F	G	H	I	J	K	L	M
	CIRCUITOS INTEGRADOS. INTEGRATED CIRCUITS.											
01	MICROCOMPUTADOR 8 BIT-CONTROLADOR.	U4	1	M88C31BH	INTEL	9.95	9.95	D	EHE/89	1	1	7-1
02	MEMORIA EPROM DE 16384 X 8 BITS. 16384 X 8 BITS UV. ERASABLE PROM.	U3	1	TMM27128	THOSHIBA	6.95	6.95	D	EHE/89	1	10	D-27
03	LATCH CUADRUPLE. QUAD LATCH.	U1,U2	2	MC14842B	NATIONAL	0.56	1.12	D	EHE/89	1	11	2-111
04	EXPANSOR DE PUERTOS CON RAM Y TIMER. 2048 BIT STATIC MOS RAM WITH I/O PORTS AND TIMER.	U5,U6	2	M81C55	INTEL	4.12	8.24	D	EHE/89	1	9	5-17
05	CODIFICADOR DE TE- CLADO. 16 KEY ENCODER.	U8	1	MM54C922	NATIONAL	4.48	4.48	D	EHE/89	1	11	1-153
06	CIRCUITO INVERSOR. INVERTER CIRCUITS.	U1E	1	CD4869C	NATIONAL	0.25	0.25	D	EHE/89	3	11	2-156
07	CIRCUITO ALIMENTADOR /RECEPTOR DE LINEA. +5 V POWER DE-232 DRIVERS/RECEIVERS.	U9	1	MAX 232	MAXIM	4.14	4.14	D	EHE/89	1	-	---
08	REGULADOR DE VOLTAJE POSITIVO +5 VOLTS. 3-TERMINAL POSITIVE REGULATOR.	REG	1	LM7805	NATIONAL	0.56	0.56	D	EHE/89	3	2	10- 123

TABLA 3.7

LISTA DE PARTES ELECTRONICAS IPE-1-11R

A	B	C	D	E	F	G	H	I	J	K	L	M
09	PUENTE RECTIFICADOR DE DIODOS. BRIDGE DIODE RECTIFIER.	PD	1	MRA100DA	MOTOROLA	2.10	2.10	D	ENE/89	1	4	27
10	DISPLAY DE CRISTAL LIQUIDO. LIQUID CRYSTAL DISPLAY MODULE-CHARACTER TYPE.	U7	1	M1641LCDM	SEIKO	22.89	22.89	D	ENE/89	1	-	-
11	MEMORIA EPROM DE 85536 BITS. 65536-BIT (8Kx8) ELECTRICALLY ERASABLE PROM.	-	1	TS28C64	THOMSON COMPONENTS	23.01	23.01	D	ENE/89	1	5	2-53
12	CAPACITORES. CAPACITORS. ELECTROLITICO 220 UF/160 ELECTROLYTIC 220 UF/160	C18, C19	2	220S16	NAVORY	0.25	0.50	D	ENE/89	A	4	193
13	ELECTROLITICO 10 UF/160 ELECTROLYTIC 10 UF/160	C5 C18, C17	3	U1L-10S16	NAVORY	0.15	0.45	D	ENE/89	A	4	193
14	ELECTROLITICO 4.7 UF/160 ELECTROLYTIC 4.7 UF/160	C11, C15, C16	3	501D475F-863LL	SPRAGUE	0.15	0.45	D	ENE/89	A	4	177
15	MONOLITICO 0.1UF/50V MONOLYTHIC 0.1UF/50V	C1, C2, C3, C6, C12, C13	8	7075U104X-805001	SPRAGUE	0.08	0.64	D	ENE/89	A	4	171

TABLA 3.8

LISTA DE PARTES ELECTRONICAS IPE-1-III

A	B	C	D	E	F	G	H	I	J	K	L	M
16	CERAMICO 20PF/1000V CERAMIC 20 PF/1000V	C7, C8	2	DD-20B	CENTRALAB	0.07	0.14	D	EHE/89	A	4	199
17	CERAMICO .022UF/600V CERAMIC .022UF/600V	C18	1	DD-103	CENTRALAB	0.07	0.07	D	EHE/89	A	4	190
18	RESISTENCIAS FIJAS FIXED RESISTORS 8.2 KOHMS 1/4W 5 %	R1	1	SEGUN MARCA	INDIFERENTE.	0.06	0.06	D	EHE/89	A	-	-
19	1.0 KOHMS 1/4 W 5 %	R3, R6	3	SEGUN MARCA	INDIFERENTE.	0.06	0.10	D	EHE/89	A	-	-
20	10 KOHMS 1/4 W 5 %	R5	1	SEGUN MARCA	INDIFERENTE.	0.06	0.06	D	EHE/89	A	-	-
21	560 KOHMS 1/4 W 5 %	R7	1	SEGUN MARCA	INDIFERENTE.	0.06	0.06	D	EHE/89	A	-	-
22	8.2 KOHMS 1/4 W 5 %	R4	1	SEGUN MARCA	INDIFERENTE.	0.06	0.06	D	EHE/89	A	-	-
23	120 OHMS 1/4 W 5 %	R9	1	SEGUN MARCA	INDIFERENTE.	0.06	0.06	D	EHE/89	A	-	-
24	POTENCIONETROS. POTENTIOMETERS. POTENCIONETRO 10K , TRIMPOT. POTENTIOMETER 10 KOHMS.	R2	1	20P-1-103	BOURNS	0.77	0.77	D	EHE/89	A	4	145
	DIODOS , DIODE .											

TABLA 3.9

LISTA DE PARTES ELECTRONICAS IPE-I-III

A	B	C	D	E	F	G	H	I	J	K	L	M
25	DIODO DE USO GRAL. SILICON RECTIFIER DIODE.	D1, D2 D3, D4	5	1N4001	TEXAS INS- TRUMENTS.	0.88	0.40	D	ENE/89	A	7	10-32
26	DIODO ZENER REGULATOR DIODE	D6	1	1N4739A	TEXAS INS- TRUMENTS.	0.25	0.25	D	ENE/89	2	7	10-53
27	CONECTORES. CONNECTORS. HOCHO DE 25 POSICIO- NES (AB-25) CS. 25 POSITIONS HOLE CONNECTOR (AB-25)C.	M	1	206604	AMP	5.0	5.0	D	ENE/89	2	12	16
28	HOCHO DE 14 POSICIO- NES ANGULO RECTO SERCILLO. 14 POSITIONS RIGHT ANGLE SINGLE ROW.	1/2J	1	65597	BERG	1.96	1.96	D	ENE/89	1	13A	16
29	HOCHO DE 14 POSICIO- NES ANGULO RECTO SERCILLO. 14 POSITIONS RIGHT ANGLE SINGLE ROW.	1/2J	1	65603	BERG	1.96	1.96	D	ENE/89	1	13A	16
30	MEMBRA DE 14 POSI- CIONES DOBLE HORI- ZONTAL. 14 POSITIONS DOUBLE ROW HORIZONTAL.	-	1	65671-014	BERG	2.53	2.53	D	ENE/89	1	13B	9
31	MODULAR SIN SEGURO NON KEVED MODULAR JACK.	X	1	205-32201- 24	AMPHENOL CONNECTORS	1.40	1.40	D	ENE/89	3	4	301

TABLA 3.18

LISTA DE PARTES ELECTRONICAS IPE-1-III

A	B	C	D	E	F	G	H	I	J	K	L	M
32	MODULAR O MODULAR CON CABLE ENROLLADO MODULAR TO MODULAR COILED CORD.	-	1	TE-128	ELECTRONIC RESOURCES	1.58	1.58	D	ENE/89	2	-	-
33	PARA TELEFONO MINIA- TURA. "TINI PLUG" MINIATU- RE PHONE PLUG.	-	1	39F746	SWITCHCRA- FT.	8.78	8.78	D	ENE/89	2	4	458
34	PARA TELEFONO TIPO HI-D HI-D JAW ENCLOSED PHONE JACKS.	-	1	M112B	SWITCHCRA- FT.	8.54	8.54	D	ENE/89	2	4	461
	INTERRUPTORES SWITCHES.											
35	INTERRUPTOR SPST. PUSHBUTTON SWITCHES ROUND BASE.	81	1	8CT1	CUTLER HARNER	1.89	1.89	D	ENE/89	1	14	43
	VARIOS											
36	CRISTAL DE CUARZO DE 11.859 MHZ. 11.859 MHZ CRYSTAL	XTAL	1	-	CRICUBA	13.05	13.05	D	ENE/89	-	-	-
37	TECLADO MATRICIAL DE 4 X 4 . KEYBOARD 4 X 4 .	-	1	88-853	GRAYHILL	6.25	6.25	D	ENE/89	1	15	C14
38	TARJETA DE CIRCUITO IMPRESO. PRINT CIRCUIT BOARD	-	1	-	-	-	-	-	-	-	-	-

TABLA 3.11

LISTA DE PARTES ELECTRONICAS IPE-1-III

A	B	C	D	E	F	G	H	I	J	K	L	M
39	TRANSFORMADOR 120/12 V 300 MA. 120/12 V 300 MA TRANSFORMER.	-	1	-	-	2.17	2.17	D	ENE/89	-	-	-
40	DISIPADOR PARA RE- GULADOR DE VOLTAJE. HEAT SINK FOR VOLTA- JE REGULATOR.	REG	1	U2-80B	EGG MAKE FIELD EN- GINEERING	8.78	8.78	D	ENE/89	A	4	141
41	INFRAROJOS. INFRARED. <hr/> DIODO EMISOR DE LUZ INFRARROJA. INFRARED EMITTING DIODE.	-	1	T1L-38	TEXAS INS- TRUMENTS.	1.88	1.88	D	ENE/89	2	-	-
42	FOTOTRANSISTOR INFRARROJO. PHOTOTRANSISTOR INFRARED.	-	1	T1L99	TEXAS INS- TRUMENTS.	1.88	1.88	D	ENE/89	3	-	-
43	BASES PARA CIRCUITO INTEGRADO. INTEGRATED CIRCUITS DIP SOCKETS. <hr/> BASE DE 48 PINES DIP SOCKET 48 PINES	U4, U5 U6	3	248-A639D	CAMBION	8.44	1.32	D	ENE/89	1	4	576
44	BASE DE 28 PINES DIP SOCKET 28 PINES	U3	1	228-A639D	CAMBION	8.39	8.39	D	ENE/89	1	4	576
45	BASE DE 18 PINES DIP SOCKET 18 PINES	U8	1	218-A639D	CAMBION	8.22	8.22	D	ENE/89	1	4	576

TABLA 3.12

LISTA DE PARTES ELECTRONICAS IPE-1-III

A	B	C	D	E	F	G	H	I	J	K	L	M
46	BASE DE 16 PINES DIP SOCKET 16 PINES	U1, U2 U9	3	216-A639D	CAMBION	8.24	8.72	D	ENE/89	1	4	576
47	BASE DE 14 PINES DIP SOCKET 14 PINES	U10	1	214-A639D	CAMBION	8.21	8.21	D	ENE/89	1	4	576
48	TRANSISTORES. TRANSISTORS. <hr/> TRANSISTOR PNP. PNP SILICON TRANSIS- TOR.	Q1	1	2N2987	TEXAS INS- TRUMENTS.	8.44	8.44	D	ENE/89	2	7	4-151
49	BATERIAS. BATTERIES. <hr/> BATERIA RECARGABLE RECHARGEABLE BATTIE- RIES BANK.	P	1	6C-610-1	GLOBE- UNION	16.50	16.50	D	ENE/89	1	4	672

Capítulo 4

DISEÑO DEL SOFTWARE

4.1 Introducción.

En el presente capítulo, se documenta el software o programas de operación de los microcontroladores, que fué desarrollado en este proyecto, con el objeto de obtener las características funcionales determinadas en las especificaciones de diseño del capítulo 2.

Los programas realizados para ambos equipos (UAD e IPE), fueron desarrollados en lenguaje ensamblador, utilizando metodologías recomendadas para programación estructurada.

El programa desarrollado para la UAD, está formado por un bloque principal y 23 subrutinas, requiriendo un espacio de memoria de programación de 1.2 K bytes.

Para el equipo IPE, el programa desarrollado consta de un bloque principal y 52 subrutinas, ocupando un espacio de memoria de programación de 4 K bytes.

Para la realización de ambos programas, se utilizó un computador personal, paquetes de software comerciales y un emulador, programador de memorias con capacidad de interconexión y manejo desde el computador personal, lo que en conjunto forma un "Sistema de Desarrollo".

En la primera etapa del desarrollo de los programas, se utilizó un paquete de software comercial, para la simulación de los microcontroladores de la familia MCS-51.

Por medio de otro paquete de software comercial, los programas fueron compilados a lenguaje de máquina y en un proceso repetitivo, las subrutinas desarrolladas fueron probadas en los prototipos correspondientes, haciendo para ello uso de la capacidad de emulación del "Sistema de Desarrollo".

Los programas finales, fueron programados en una memoria 27C128 para el equipo IPE, y en la memoria interna del microcontrolador 87C51 para la UAD, haciendo uso de la capacidad de programación del "Sistema de Desarrollo".

En la sección 4.2, se documenta el programa correspondiente a las Unidades de Adquisición de Datos (UAD) y en la sección 4.3 el correspondiente a el Equipo de Interrogación / Programación Externa (IPE).

En dichas secciones, se presenta una descripción funcional de los programas, desde un punto de vista de su operación global, mencionando las funciones de los bloques principales y de aquellas subrutinas de primer orden, que manejan los posibles modos de operación de los sistemas.

Cada una de las subrutinas desarrolladas, se presenta en lenguaje ensamblador, incluyendo comentarios y un diagrama de flujo para su interpretación.

4.2 Software para la UAD.

En la figura No. 4.1, se presenta un diagrama de flujo que representa la estructura dada al software desarrollado para la UAD, el cual se explica a continuación.

El primer bloque del diagrama, corresponde a aquellas instrucciones que se ejecutan inmediatamente después a la aplicación de energía eléctrica al circuito, o sea en este caso, a la aplicación de un pulso de "reset" al microcontrolador.

Estas instrucciones corresponden a las siguientes actividades:

- Deshabilita todos los dispositivos periféricos.
- Programa e inicializa el puerto serie.
- Programa el contador T0, como contador de pulsos externos.
- Inicializa contadores utilizados por el programa.
- Inicializa "banderas" utilizadas en el programa.
- Programa prioridad de interrupciones externas.
- Programa condiciones de operación del reloj externo.
- Realiza una subrutina de prueba inicial (autodiagnóstico).
- Coloca el resultado de autodiagnóstico en un registro.
- Habilita interrupción por puerto de comunicaciones.
- Se coloca en modo "idle".

Una vez que las condiciones iniciales de operación han sido establecidas, el microcontrolador permanecerá en modo "idle" sin realizar ninguna función, quedando en espera de órdenes enviadas desde el equipo IPE, para la inicialización de su operación normal.

Ante una interrupción provocada por la presencia de un comando en el puerto de comunicaciones, el microcontrolador realizará la subrutina "ON-COM", en la cual se compara el código enviado con los códigos válidos y a su reconocimiento se permiten los modos de INTERROGACION, ACCESO, REPROGRAMACION Y TRANSFERENCIA, para lo cual el programa ejecuta las subrutinas correspondientes.

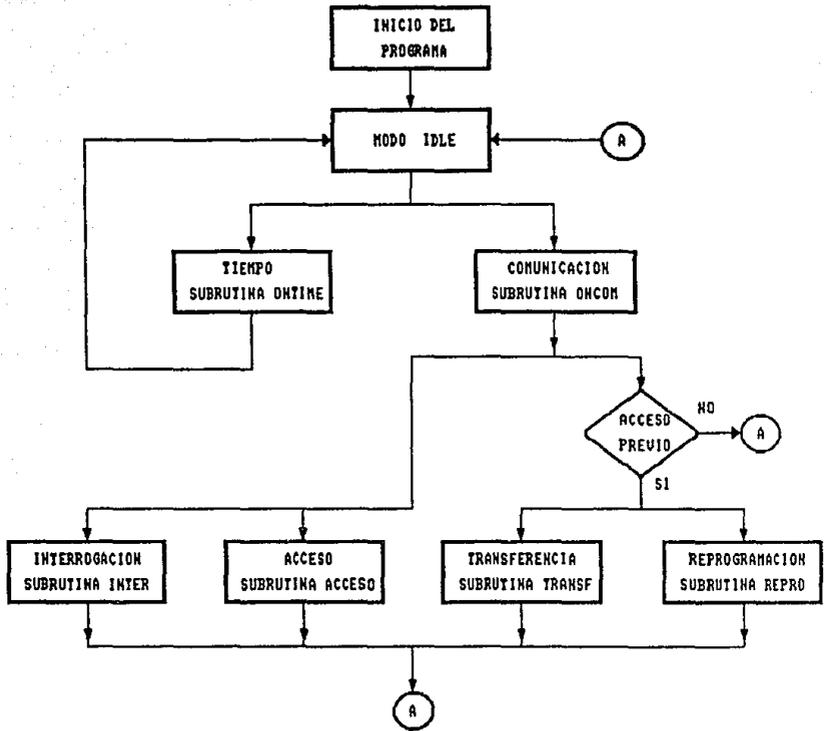


FIG. 4.1 ESTRUCTURA DEL SOFTWARE DE LA UAD-1-11E.

Nótese que las subrutinas de REPROGRAMACION Y TRANSFERENCIA, solamente podrán ser ejecutadas si previamente se llevó a cabo un proceso de acceso a la UAD.

Para iniciar la operación normal de la UAD, se requerirá el uso del equipo IPE. Como primer paso se deberá realizar una rutina de interrogación, con el objeto de que el operador pueda conocer el resultado de las pruebas de autodiagnóstico realizadas por la UAD.

En el MODO DE INTERROGACION, la UAD, transmitirá la información correspondiente a:

- FECHA Y HORA ACTUAL EN EL RELOJ EN TIEMPO REAL.
- ESTADO DE OPERACION.
 - a.- Operación normal (adquiriendo datos).
 - b.- No se ha inicializado la operación normal.
 - c.- Falla de operación en la memoria EEPROM.
 - d.- Falla de operación en el reloj.
- FECHA Y HORA EN QUE SE INICIO LA ADQUISICION DE DATOS.
- NUMERO DE DIAS QUE FALTAN PARA SATURAR LA MEMORIA.

Si no existe falla, por medio del IPE el operador colocándose en modo de ACCESO, podrá enviar la contraseña solicitada la cual será comparada con el código propio de cada UAD generando y transmitiendo un mensaje de reconocimiento o no del código enviado. Si el código es reconocido, se permitirá la operación en modo de REPROGRAMACION.

Una vez en modo de REPROGRAMACION, el operador podrá introducir al reloj de la UAD la fecha y hora reales e iniciar el conteo de tiempo.

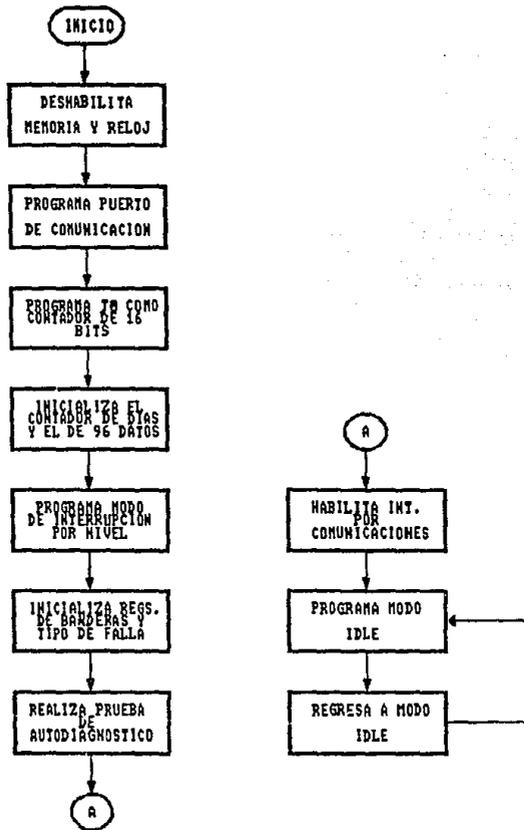
Posteriormente, en el mismo modo de REPROGRAMACION, el operador podrá proporcionar la orden de inicio de adquisición de datos, para lo cual la UAD realizará el monitoreo continuo del reloj en tiempo real y en el momento en que dicho tiempo corresponda a un valor "múltiplo" de 15 minutos (00:15 ,00:3012:15 ... 19:00 ... etc.). la UAD entrará en modo de adquisición de datos u operación normal.

En este momento, la interrupción externa correspondiente a la generada por el reloj en tiempo real es habilitada y el microcontrolador regresa a el estado "idle".

El proceso de captura y almacenamiento de información, será ejecutado en forma continua por medio de la subrutina "ONTIME", la cual será ejecutada cada que se presente una interrupción externa generada por el reloj en tiempo real.

Así mismo, en operación normal la UAD estará habilitada para responder a cualquier comando generado por el IPE a través del puerto de comunicaciones.

PROGRAMA PRINCIPAL



```

;PROGRAMA PRINCIPAL
;ESTE PROGRAMA REALIZA LAS OPERACIONES NECESARIAS PARA INICIALIZAR
;LAS FUNCIONES QUE SE LLEVAN ACABO POR EL UAD Y ADEMAS MANTIENE AL
;MICROPROCESADOR EN MODO IDLE.

```

```

DEFSEG ZERO,CLASS=CODE,START=0000H
DEFSEG MAIN,CLASS=CODE,START=100H
DEFSEG INTRE,CLASS=CODE,START=03H
DEFSEG PUERTO,CLASS=CODE,START=23H
SEG ZERO
JMP PRINCI
SEG INTRE
JMP ONTIME
SEG PUERTO
JMP ONCDM
SEG MAIN
JMP PRINCI
ORG 100H

```

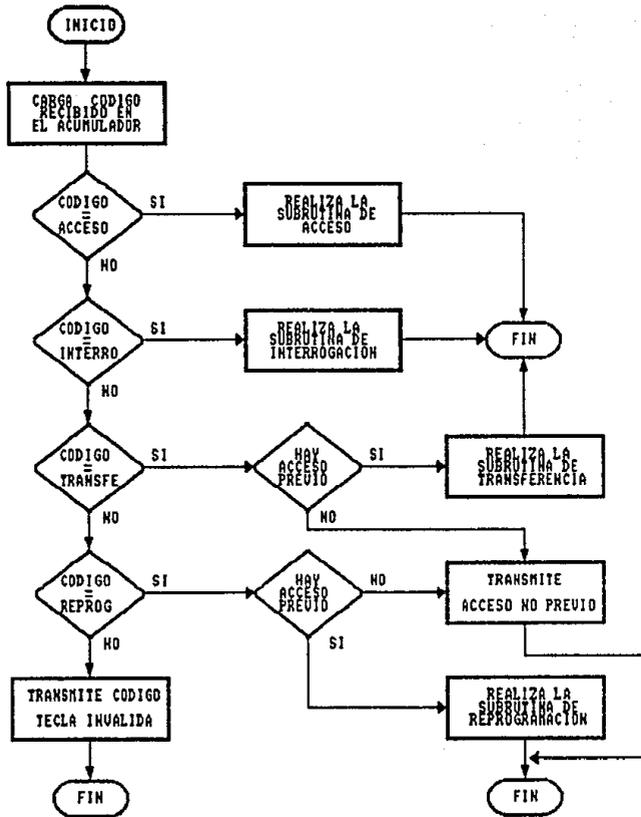
```

PRINCI: MOV SP,#057H ;POR SEGURIDAD
MOV P2,#0EFH ;DESHABILITA MEMORIA, RELOJ Y OE
MOV SCON,#050H ;INICIALIZA PUERTO SERIE
MOV TMOD,#025H ;PROGRAMA CONT. DE 16 BITS A TO
MOV TH1,#0E6H ;INICIALIZA T1 PARA AUTO-RECARGA
SETB TR1 ;ARRANCA CONT. 1
SETB RSO ;SELECCIONA BANCO 1
MOV R6,#000H ;INICIALIZA CONTADOR DE 96 DATOS
MOV R7,#01FH ;INICIALIZA CONTADOR DE DIAS
CLR RSO ;SELECCIONA BANCO 0
MOV A,#011H ;CARGA LA DIR. DEL RELOJ
MOV R1,#024H ;CARGA LA DIRECCION DE LA RAM
CARGA: ADD A,#010H ;INCREMENTA LA DIR. DEL RELOJ
MOV @R1,A ;CARGA DATOS INICIALES AL RELOJ
INC R1 ;INCREMENTA LA DIR. DE LA RAM
CJNE R1,#032H,CARGA ;COMPARA R1
ACALL RELOJ ;PROGRAMA LOS DATOS DEL RELOJ

```

SETB	PX0	;NIVEL ALTO PRIORIDAD INT. EXT.0
MOV	R7,#000H	;INICIALIZA REG. DE BANDERAS
MOV	R6,#000H	;INICIALIZA TIPO DE FALLA
ACALL	TEST1	;REALIZA AUTO-DIAGNOSTICO
MOV	IE,#090H	;HABILITA INT. DE COMUNICACIONES
ESPERA: MOV	PCON,#001H	;MODO IDLE
SJMP	ESPERA	

SUBROUTINA ONCOM



```

;SUBROUTINA "ONCOM"
;ESTA ES LA SUBROUTINA DE SERVICIO QUE SE PROPORCIONA ANTE UNA
;INTERRUPCION GENERADA POR EL PUERTO DE COMUNICACIONES.
;CAPTURA EL COMANDO ENVIADO POR EL IPE, LO INTERPRETA Y EJECUTA
;LA ACCION CORRESPONDIENTE.

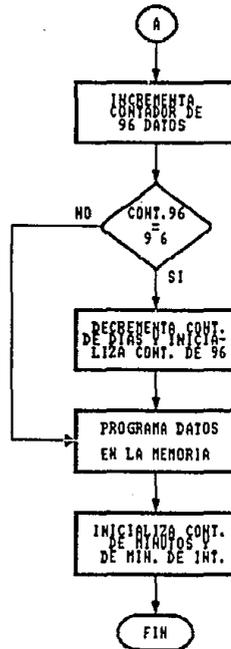
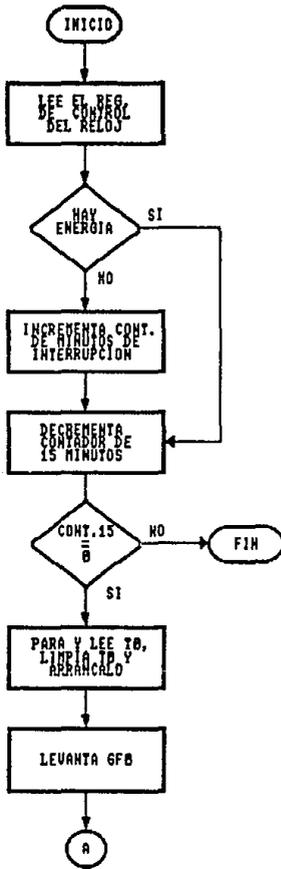
```

```

ONCOM: CLR      RI          ;BORRA LA BANDERA RI
      JNB     P3.3,FIN1    ;VERIFICA SI HAY ENERGIA ELEC.
      MOV     A,SBUF       ;CARGA COMANDO EN ACUMULADOR
      CJNE   A,#041H,C_INTE ;COMPARA CON CODIGO DE ACCESO
      ACALL  ACCESO        ;SI ES IGUAL SE VA ACCESO
      SJMP   FIN1         ;REGRESA A IDLE
C_INTE: CJNE   A,#042H,C_TRAN ;COMPARA CODIGO DE INTERROGACION
      ACALL  INTER        ;SI ES IGUAL VE A INTERROGACION
      SJMP   FIN1         ;REGRESA A IDLE
C_TRAN: CJNE   A,#043H,C_REP ;COMPARA CODIGO DE TRANSFERENCIA
      MOV     A,R7        ;CARGA EL REG. DE BANDERAS
      JNB     ACC.1,ERR2   ;SI NO HAY ACCESO PREVIO ERROR 2
      ACALL  TRANSF       ;SE VA A MODO DE TRANSFERENCIA
      SJMP   FIN1         ;REGRESA A IDLE
C_REP:  CJNE   A,#044H,ERR1 ;COMPARA CODIGO DE REPROGRAMACION
      MOV     A,R7        ;CARGA EL REG. DE BANDERAS
      JNB     ACC.1,ERR2   ;SI NO HAY ACCESO PREVIO ERROR 2
      ACALL  REPRO        ;SE VA A MODO DE REPROGRAMACION
      SJMP   FIN1         ;REGRESA A IDLE
ERR1:   MOV     A,#0F1H    ;CARGA CODIGO DE "TECLA INVALIDA"
      ACALL  TRANS        ;LO TRANSMITE AL IPE
      SJMP   FIN1         ;REGRESA A IDLE
ERR2:   MOV     A,#0F2H    ;CARGA CODIGO "ACCESO NO PREVIO"
      ACALL  TRANS        ;LO TRANSMITE AL IPE
FIN1:   ACALL  RETAR       ;REALIZA UN RETARDO
      ACALL  RETAR       ;REALIZA UN RETARDO
      MOV     A,R7        ;CARGA EL REG. DE BAND.
      CLR     ACC.5       ;BORRA BANDERA
      MOV     R7,A        ;SALVA REG. DE BAND.
      CLR     RI         ;BORRA RI
      RETI

```

SUBROUTINA ONTIME



```

;SUBROUTINA "ONTIME"
;ESTA ES LA SUBROUTINA DE SERVICIO QUE SE PROPORCIONA ANTE
;UNA INTERRUPCION GENERADA POR EL RELOJ CADA MINUTO.
;ESTA SUBROUTINA TIENE COMO FUNCION MUESTREAR CADA
;MINUTO SI HAY AUSENCIA DE ENERGIA ELECTRICA Y MANDAR
;ALMACENAR LOS DATOS DE DEMANDA DE ENERGIA ELECTRICA.

```

```

ONTIME: PUSH ACC           ;SALVA EL ACUMULADOR
        CLR A             ;CARGA DIR. DEL REG. DE CONTROL
        ACALL LEERE      ;LEE EL REG. DE CONTROL
        JB P3.3,DECREM   ;HAY ENERGIA ELEC. VE A DECREM
        INC R5           ;INCREMENTA CONT. DE MIN. DE INT
DECREM: DJNZ R4,FIN2     ;CONT. DE 15 MIN. ES O CONTINUA
        CLR TRO         ;PARA EL CONT. TO
        MOV R3,TRO      ;SALVA LA LECTURA DE TO
        MOV R2,TLO
        MOV THO,#000H   ;LIMPIA EL CONT. TO
        MOV TLO,#000H
        SETB TRO        ;ARRANCA TO
        SETB FO         ;LEVANTA LA BANDERA GFO
        SETB RSO        ;SELECCIONA EL BANCO 1
        INC R6          ;INCREMENTA CONT. DE 96 DATOS
        CJNE R6,#060H,NODEC;COMPARA SI SON 96 DATOS
        DEC R7          ;DECREMENTA CONT. DE DIAS
        MOV R6,#000H   ;LIMPIA CONT. DE 96 DATOS
NODEC:  ACALL PROGRA    ;PROGRAMA DATOS EN MEMORIAN
        CLR RSO        ;SELECCIONA BANCO 0
        MOV R4,#00FH   ;INICIALIZA CONT. DE 15 MIN.
        MOV R5,#000H   ;LIMPIA CONT. DE MIN. DE INT.
        FIN2: POP ACC  ;RESTABLECE EL ACUMULADOR
        RETI

```

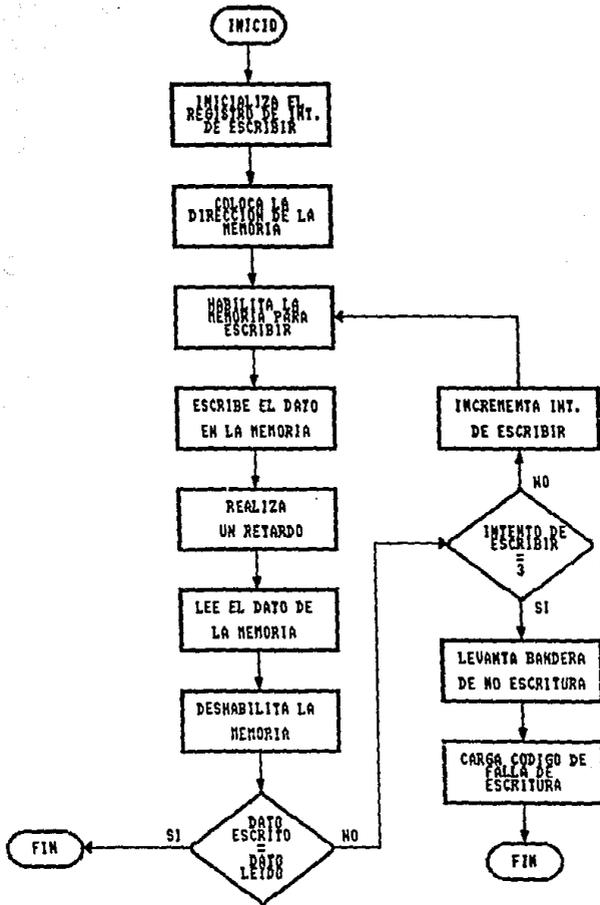
SUBROUTINA DIREC



;SUBROUTINA "DIREC"
;ESTA SUBROUTINA TIENE COMO FUNCION DIRECCIONAR A LA MEMORIA PARA
;UNA ESCRITURA O UNA LECTURA SEGUN SEA EL CASO.

```
DIREC:  MOV      P1,DPH      ;COLOCA LA DIR. DEL BYTE ALTO
        MOV      P2,#OFFH   ;DA UN PULSO AL EXPANSOR
        NOP
        NOP
        MOV      P2,#OEFH
        MOV      P1,DPL     ;COLOCA LA DIR. DEL BYTE BAJO
        RET
```

SUBROUTINA ESCRI



ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

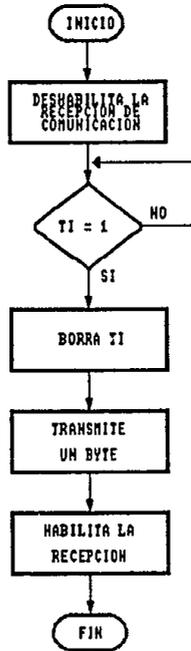
;SUBROUTINA "ESCRI"

;LA FUNCION DE ESTA SUBROUTINA ES LA DE ESCRIBIR UN DATO EN LA

;MEMORIA Y VERIFICAR QUE SEA ESCRITO CORRECTAMENTE.

```
ESCRI:  MOV     B,A           ;SALVA EL DATO A ESCRIBIR
        ACALL  DIREC        ;DIRECCIONA LA MEMORIA
        MOV     R1,#003H    ;CARGA NUM. INTENTOS DE ESCRITURA
REPITE: MOV     A,B         ;CARGA EL DATO A ESCRIBIR
        MOV     PO,A        ;SACA EL DATO POR PO
        MOV     P2,#OEAH    ;HABILITA MEMORIA PARA ESCRITURA
        ACALL  RETAR        ;REALIZA UN RETARDO
        NOP
        NOP
        MOV     P2,#OEFH    ;DESHABILITA LA MEMORIA
        ACALL  RETAR        ;REALIZA UN RETARDO
        ACALL  LECTU        ;LEE EL DATO
        CLR     C           ;BORRA EL CARRY
        SUBB   A,B         ;RESTA DATO LEIDO CON EL ESCRITO
        JZ     FIN3        ;REGRESA SI EL RESULTADO ES CERO
        DJNZ  R1,REPITE    ;R2 NO ES O TRATA DE ESCRIBIRLO
        MOV     A,#00BH    ;CARGA LA BAND. DE NO ESCRITURA
        ORL   A,R7        ;SUMALE EL REG. DE BANDERAS
        MOV     R7,A       ;SALVA EL REG. DE BANDERAS
        MOV     R6,#001H   ;CARGA CODIGO "FALLA DE ESCRITURA"
FIN3:   RET
```

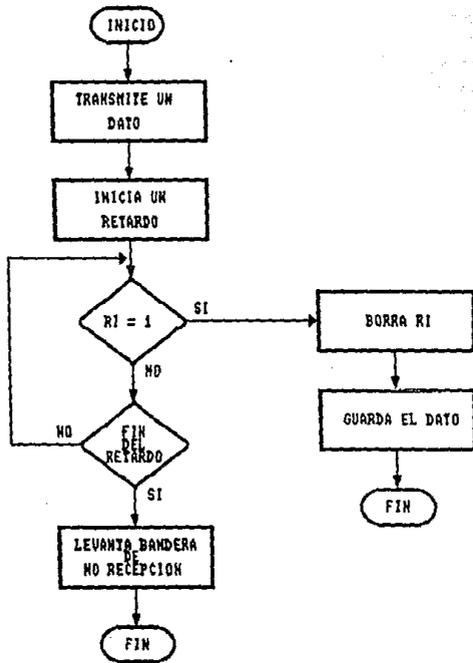
SUBROUTINA TRANS



;SUBROUTINA "TRANS"
;ESTA SUBROUTINA TIENE COMO FUNCION TRANSMITIR UN BYTE AL IPE.

TRANS:	CLR	REN	;DESHABILITA LA RECEPCION
	MOV	SBUF,A	;TRANSMITE EL BYTE AL IPE
	JNB	TI,\$;ESPERA QUE TI=1
	CLR	TI	;BORRA TI
	NOP		
	NOP		
	NOP		
	SETB	REN	;HABILITA RECEPCION
	RET		

SUBROUTINA RECEP



```

;SUBROUTINA "RECEP"
;LA FUNCION DE ESTA SUBROUTINA ES LA DE TRANSMITIR UN BYTE AL IPE
;Y CONTINUAR HASTA RECIBIR UN BYTE DEL IPE.

```

```

RECEP:  ACALL  TRANS      ;TRANSMITE UN BYTE AL IPE
        SETB  RS1        ;SELECCIONA EL BANCO 2
        MOV   R2,#OFFH
RECI2:  MOV   R3,#OFFH
RECI1:  JB    R1,NOCOM    ;VE SI HAY COMUNICACION
        DJNZ  R3,RECI1
        DJNZ  R2,RECI2
        CLR  RS1         ;SELECCIONA EL BANCO 0
        MOV  A,R7        ;CARGA REG. DE BAND
        SETB ACC.5       ;LEVANTA BAND. DE NO RECEPCION
        MOV  R7,A        ;SALVA EL REG. DE BAND
NOCOM:  CLR  RS1         ;SELECCIONA EL BANCO 0
        CLR  RI          ;BORRA RI
        MOV  A,SBUF      ;ALMACENA EL BYTE
        RET

```

SUBROUTINA LECTU

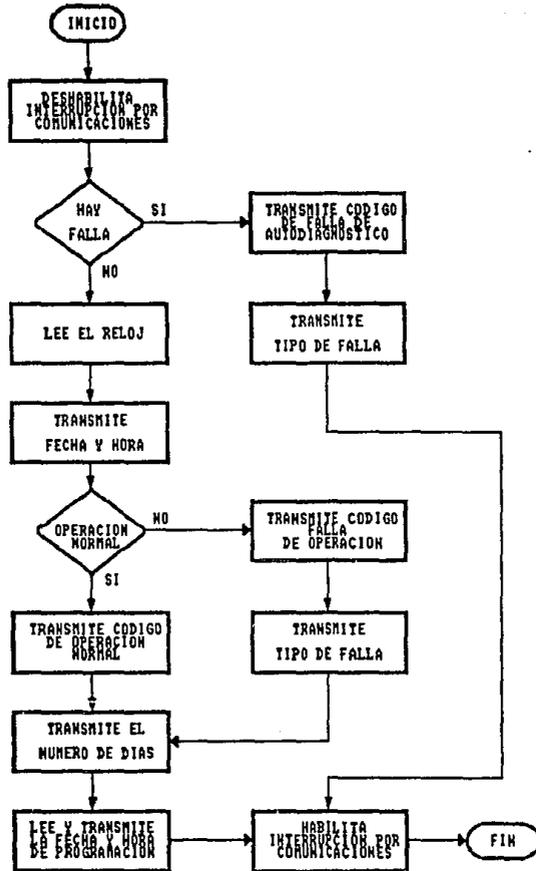


:SUBROUTINA "LECTU"

:ESTA SUBROUTINA TIENE COMO FUNCION LEER UN DATO DE LA MEMORIA Y
:SALVA EL DATO EN EL ACUMULADOR.

```
LECTU:  ACALL  DIREC      ;DIRECCIONA LA MEMORIA
        MOV   P2,#OECH   ;HABILITA LA MEMORIA PARA LEER
        ACALL RETAR      ;REALIZA UN RETARDO
        NOP
        NOP
        MOV   A,PO       ;LEE EL DATO DEL PO
        MOV   P2,#OEFH   ;DESHABILITA LA MEMORIA
        RET
```

SUBROUTINA INTER



```

;SUBROUTINA "INTER"
;LA FUNCION QUE REALIZA ESTA SUBROUTINA ES LA DE TRANSMITIR AL IPE
;LA FECHA Y HORA ACTUAL, MODO DE OPERACION, NUM. DE DIAS QUE LE
;RESTAN A LA MEMORIA DE ALMACENAMIENTO, FECHA Y HORA DE
;PROGRAMACION.

```

```

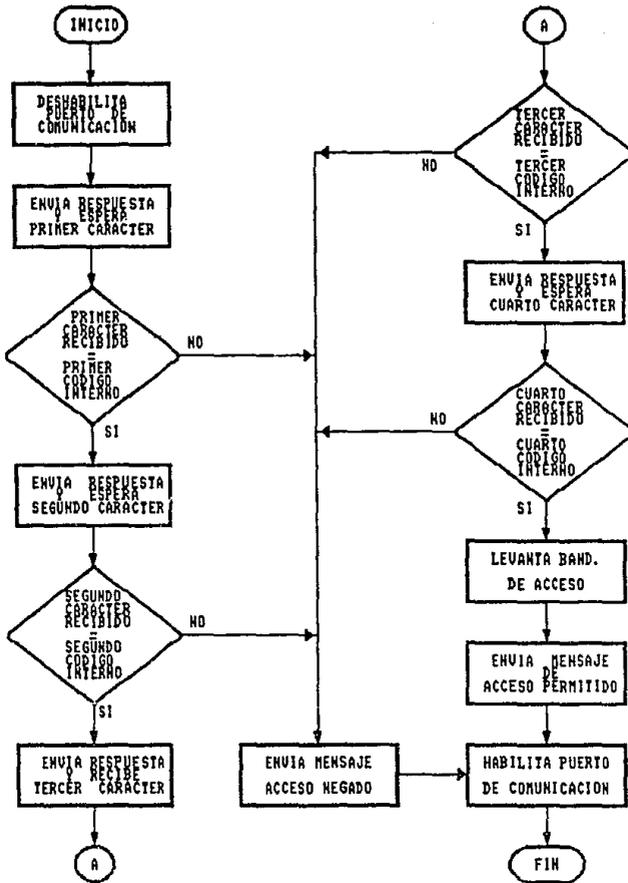
INTER: CLR     ES           ;DESHABILITA PUERTO DE COMUNI.
      MOV     A,R7        ;CARGA EL REG. DE BANDERAS
      JNB    ACC.2,ENVIA  ;NO HAY FALLA DE AUTC-DIAG.
      MOV     A,#0F6H     ;CARGA CODIGO "FALLA AUTO-DIAG."
      ACALL  RECEPT    ;TRANSMITELO Y ESPERA RESPUESTA
      MOV     A,R6        ;CARGA EL TIPO DE FALLA
      ACALL  TRANS       ;TRANSMITELO AL IPE
      SJMP   FIN4

ENVIA: ACALL  LRELOJ     ;LEE EL RELOJ
      MOV     RO,#00AH    ;
      MOV     R1,#026H   ;INICIALIZA DIR. DE RAM
      TRANSM: MOV    A,R1 ;CARGA EL DATO
      ACALL  RECEPT    ;TRANSMITE Y ESPERA CONTESTACION
      INC    R1          ;INCREMENTA DIR. DE RAM
      DJNZ  RO,TRANSM   ;ES FIN DE LA TRANSFERENCIA
      MOV     A,R7        ;CARGA EL REG. DE BANDERAS
      JNB    ACC.0,FALLA1 ;SI BAND.0=0 VE A FALLA1
      MOV     A,#0F3H    ;CARGA CODIGO "OPERACION NORMAL"
      CONTI: ACALL  RECEPT ;TRANSMITE Y ESPERA CONTESTACION
      SETB   RSO         ;SELECCIONA BANCO 1
      MOV     A,R7        ;CARGA NUMERO DE DIAS
      CLR    RSO         ;SELECCIONA BANCO 0
      MOV     B,#00AH    ;CARGA EL NUM. 10 EN B
      DIV   AB          ;DIVIDE EL NUM. DE DIAS ENTRE 10
      XCH   A,B         ;INTERCAMBIA CONTENIDO DE REGS.
      ACALL  RECEPT    ;TRANSMITE LAS UNIDADES
      MOV     A,B        ;CARGA DECENAS DEL NUM. DE DIAS
      ACALL  RECEPT    ;TRANSMITE Y ESPERA CONTESTACION
      MOV     DPTR,#0001H ;CARGA LA DIR. DE LA MEMORIA
      MOV     RO,#009H   ;CARGA NUM. DE DATOS A TRANSFERIR
      LEE1: ACALL  LECTU  ;LEE EL DATO DE LA MEMORIA

```

	ACALL	RECEP	;TRANSMITE Y ESPERA CONTESTACION
	INC	DPIR	;INCREMENTA DIR. DE MEMORIA
	DJNZ	RO,LEE1	;SI RO=0 YA TRANSFIRIO LOS DATOS
	ACALL	LECTU	;LEE EL DATO DE LA MEMORIA
	ACALL	TRANS	;TRANSMITE EL DATO
	ACALL	RETAR	;REALIZA UN RETARDO
	SJMP	FIN4	
FALLA1:	MOV	A,#OF5H	;CARGA CODIGO "FALLA DE OPERACION"
	ACALL	RECEP	;TRANSMITE Y ESPERA CONTESTACION
	MOV	A,R6	;CARGA TIPO DE FALLA
	SJMP	CONTI	
FIN4:	SETB	ES	;HABILITA INT. PUERTO DE COMUNI.
	RET		

SUBROUTINA ACCESO



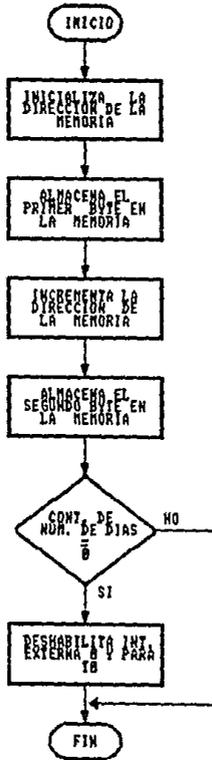
;SUBROUTINA "ACCESO"

;LA FUNCION DE ESTA SUBROUTINA ES LA DE PERMITIR EL ACCESO AL IPE

;Y ENVIAR EL CODIGO DE "ACCESO PERMITIDO" O "ACCESO NEGADO".

```
ACCESO: CLR      ES                ;DESHABILITA INT. PUERTO COMUNI.
        ACALL   RECEP             ;CONTESTA Y ESPERA CHARACTER
        CJNE   A,#031H,DIFE       ;COMPARA EL CODIGO DEL 1ER. BYTE
        ACALL   RECEP             ;CONTESTA Y ESPERA CHARACTER
        CJNE   A,#032H,DIFE       ;COMPARA EL CODIGO DEL 2DO. BYTE
        ACALL   RECEP             ;CONTESTA Y ESPERA CHARACTER
        CJNE   A,#033H,DIFE       ;COMPARA EL CODIGO DEL 3ER. BYTE
        ACALL   RECEP             ;CONTESTA Y ESPERA CHARACTER
        CJNE   A,#034H,DIFE       ;COMPARA EL CODIGO DEL 4TO. BYTE
        MOV    A,R7               ;CARGA REG. DE BANDERAS
        SETB   ACC.1              ;LEVANTA BANDERA DE ACCESO
        MOV    R7,A               ;SALVA EL REG. DE BANDERAS
        MOV    A,#0F8H            ;CARGA CODIGO "ACCESO PERMITIDO"
        ACALL   TRANS             ;TRANSMITELO
        SJMP   FIN7
DIFE:   MOV    A,#0F7H            ;CARGA CODIGO "ACCESO NEGADO"
        ACALL   TRANS             ;TRANSMITELO
FIN7:   SETB   ES                ;HABILITA INT. PUERTO COMUNI.
        RET
```

SUBROUTINA PROCRA

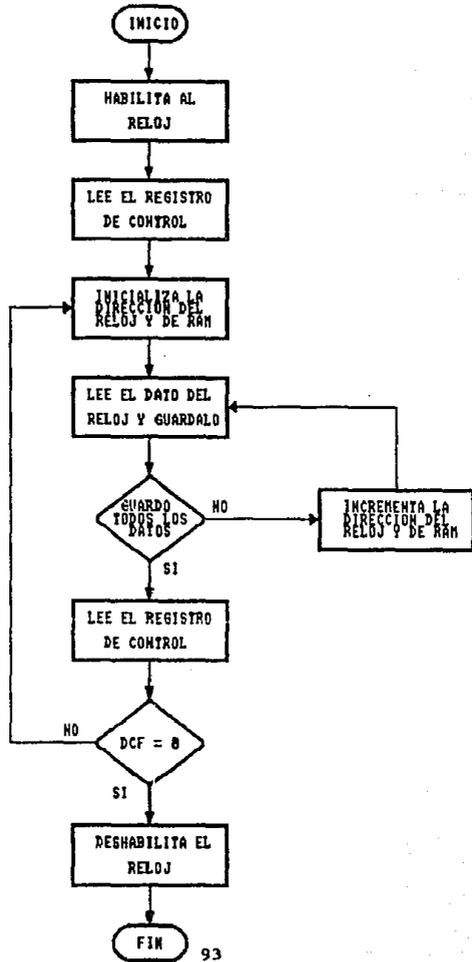


;SUBROUTINA "PROGRA"

;ESTA SUBROUTINA TIENE COMO FUNCION ALMACENAR LOS DATOS DE DEMANDA
;DE ENERGIA ELECTRICA EN LA MEMORIA EEPROM. DONDE EN EL PRIMER
;BYTE SE TIENE "TLO" Y EN EL SEGUNDO BYTE SE TIENE "THO" Y LOS
;MINUTOS DE INTERRUPCION DE ENERGIA ELEC.

```
PROGRA: PUSH   DPH           ;SALVA EL DATO DE DPH
        PUSH   DPL           ;SALVA EL DATO DE DPL
        MOV    DPH,R5        ;CARGA PARTE ALTA DIR. DE MEMORIA
        MOV    DPL,R4        ;CARGA PARTE BAJA DIR. DE MEMORIA
        CLR    RSO           ;SELECCIONA BANCO 0
        MOV    A,R2          ;CARGA TLO
        ACALL  ESCRI         ;ALMACENA EL PRIMER BYTE
        INC    DPTR          ;INCREMENTA DIR. DE MEMORIA
        MOV    A,R5          ;CARGA LOS MIN. DE INT.
        SWAP   A             ;INTERCAMBIA PARTE ALTA CON BAJA
        ORL    A,R3          ;UNE THO CON MIN. DE INT.
        ACALL  ESCRI         ;ALMACENA EL SEGUNDO BYTE
        SETB   RSO           ;SELECCIONA BANCO 1
        INC    DPTR          ;INCREMENTA DIR. DE MEMORIA
        MOV    R5,DPH        ;GUARDA LA PARTE ALTA DE LA DIR
        MOV    R4,DPL        ;GUARDA LA PARTE BAJA DE LA DIR
        MOV    A,R7          ;CARGA EL CONT. DE DIAS
        JNZ   FIN5           ;CONT. DE DIAS NO ES 0 REGRESA
        CLR    EXO           ;DESHABILITA INT. EXT. 0
        CLR    TRO           ;PARA CONT. TO
FIN5:   POP    DPL           ;RESTABLECE EL VALOR DE DPL
        POP    DPH           ;RESTABLECE EL VALOR DE DPH
        RET
```

SECUENCIA RELOJ



```

;SUBROUTINA "LRELOJ"
;ESTA SUBROUTINA TIENE COMO FUNCION LEER LOS DATOS DEL RELOJ Y
;VERIFICAR QUE SEAN LEIDOS SIN QUE OCURRA UN CAMBIO EN UNO DE
;LOS REGISTROS.

```

```

LRELOJ: MOV     P2,#OCFH      ;HABILITA EL RELOJ
        ACALL  RETAR        ;REALIZA UN RETARDO
        CLR   A              ;CARGA DIR. DEL REG. DE CONTROL
VUELVE: MOV     R1,#026H    ;CARGA LA DIR. DE LA RAM
        ACALL  LEERE        ;LEE EL DATO (PARA BORRAR DCF)
        MOV     A,#030H    ;CARGA DIR. DE LOS REG. DEL RELOJ
CONT3:  ADD     A,#010H    ;INC. DIR. DE LOS REG. RELOJ
        MOV     RO,A        ;SALVA DIR. DE LOS REG. DEL RELOJ
        ACALL  LEERE        ;LEE EL DATO DEL RELOJ
        ANL   A,#00FH      ;SALVA EL DATO
        MOV     @R1,A       ;ALMACENA EL DATO LEIDO
        INC   R1            ;INCREMENTA DIR. DE RAM
        MOV     A,RO        ;CARGA DIR. DE LOS REG. DEL RELOJ
        CJNE  R1,#030H,CONT3;COMPARA R1 YA LLEGO A 30H
        CLR   A              ;CARGA DIR. DEL REG. DE CONTROL
        ACALL  LEERE        ;LEE EL REG. DE CONTROL
        JNB   ACC.3,FIN6    ;SI DCF=1 VUELVE A LEER EL RELOJ
        CLR   A              ;CARGA EL REG. DE CONTROL
        SJMP  VUELVE
FIN6:  MOV     P2,#0EFH    ;DESHABILITA EL RELOJ
        RET

```

SUBROUTINA LEERE



;SUBROUTINA "LEERE"
;LA FUNCION DE ESTA SUBROUTINA ES LA DE HABILITAR AL RELOJ PARA
;PODER LEER UN DATO DE EL RELOJ.

```
LEERE:  MOV      P1,A           ;SACA LA DIR. POR P1
        MOV      P2,#0CDH      ;HABILITA LA LECTURA
        ACALL    RETAR         ;REALIZA UN RETARDO
        NOP
        NOP
        MOV      A,PO          ;LEE EL DATO
        MOV      P2,#0CFH      ;DESHABILITA LA LECTURA
        ACALL    RETAR         ;REALIZA UN RETARDO
        RET
```

SUBROUTINA RELOJ



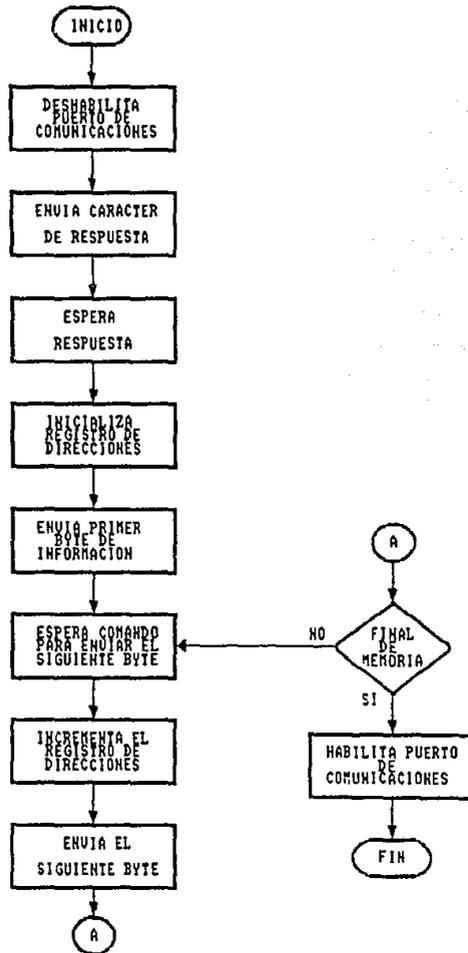
;SUBROUTINA "RELOJ"

;LA FUNCION DE ESTA SUBROUTINA ES LA DE INICIALIZAR EL RELOJ Y

;CARGAR LOS DATOS DE TIEMPO.

```
RELOJ:  MOV    R1,#020H      ;CARGA LA DIR. DE RAM
        MOV    R1,#00FH    ;RELOJ EN MODO PRUEBA Y MODO INT.
        INC    R1          ;INCREMENTA DIR
        MOV    R1,#0FOH    ;SE PONE EN NO INTERRUPCION
        INC    R1          ;INCREMENTA DIR
        MOV    R1,#005H    ;SALE MODO PRUEBA
        INC    R1          ;INCREMENTA DIR
        MOV    R1,#0F1H    ;SELECCIONA EL MODO DE 24 HRS.
        MOV    R1,#032H    ;CARGA LA DIR. 32H DE LA RAM
        MOV    R1,#007H    ;SELECCIONA REG. DE INT. DEL RELOJ
        INC    R1          ;INCREMENTA LA DIR
        MOV    R1,#0FFH    ;INT. REPETITIVA CADA 60 SEG.
        INC    R1          ;INCREMENTA DIR
        MOV    R1,#000H    ;SE PONE EN MARCHA EL RELOJ
        MOV    R1,#020H    ;INICIALIZA LA DIR. DE LA RAM
        MOV    P2,#0CFH    ;HABILITA EL RELOJ
        ACALL  RETAR       ;REALIZA UN RETARDO
CONT4:  MOV    A,R1         ;SACA EL DATO
        MOV    P1,A        ;SACA LA DIR. POR P1
        MOV    P0,A        ;SACA EL DATO POR P0
        MOV    P2,#0CBH    ;HABILITA LA ESCRITURA
        ACALL  RETAR       ;REALIZA UN RETARDO
        NOP
        NOP
        MOV    P2,#0CFH    ;DESHABILITA LA ESCRITURA
        ACALL  RETAR       ;REALIZA UN RETARDO
        INC    R1          ;INCREMENTA LA DIR
        CJNE  R1,#035H,CONT4 ;YA ESCRIBIO LOS DATOS
        MOV    P2,#0EFH    ;DESHABILITA EL RELOJ
        RET
```

SUBROUTINA TRANSF

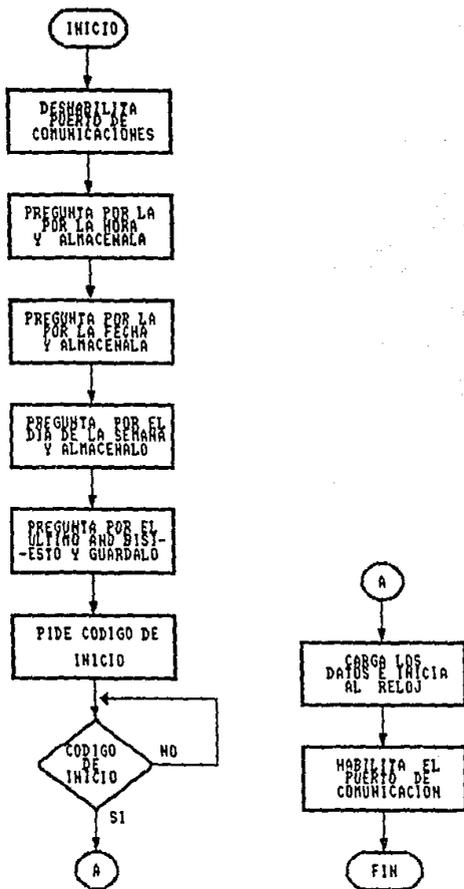


:SUBROUTINA "TRANSF"

:ESTA SUBROUTINA TIENE COMO FUNCION TRANSFERIR LOS DATOS DE LA
:MEMORIA EEPROM HACIA EL IPE (TRANSFIERE LOS 8K).

```
TRANSF: CLR     ES           ;DESHABILITA INT. PUERTO COMUNI.
        SETB    RSO         ;SELECCIONA BANCO 1
        MOV     A,R5        ;SALVA DIR. DEL ULTIMO DATO TRANS.
        MOV     R3,A        ;PARTE ALTA
        MOV     A,R4
        MOV     R2,A        ;PARTE BAJA
        CLR     RSO         ;SELECCIONA BANCO 0
        ACALL   RECEP
        MOV     DPTR,#00000H ;INICIALIZA DIR. DE LA MEMORIA
SIGUE:  ACALL   LECTU        ;LEE EL DATO DE LA MEMORIA
        ACALL   RECEP        ;TRANSMITE Y ESPERA CONTESTACION
        MOV     A,R7        ;CARGA REG. DE BAND
        JB     ACC.5,FINN2   ;NO HAY CONTESTACION VE FINN2
        INC     DPTR        ;INCREMENTA DIR. DE MEMORIA
        MOV     A,DPL       ;CARGA EL BYTE BAJO DE LA DIR.
        CJNE   A,#0FFH,SIGUE ;EL BYTE NO ES IGUAL VE A SIGUE
        MOV     A,DPH       ;CARGA EL BYTE ALTO DE LA DIR
        CJNE   A,#01FH,SIGUE ;EL BYTE NO ES IGUAL VE A SIGUE
        ACALL   LECTU        ;LEE EL DATO
        ACALL   TRANS        ;TRANSMITE EL DATO
FINN2:  SETB    ES           ;HABILITA INT. PUERTO COMUNI.
        RET
```

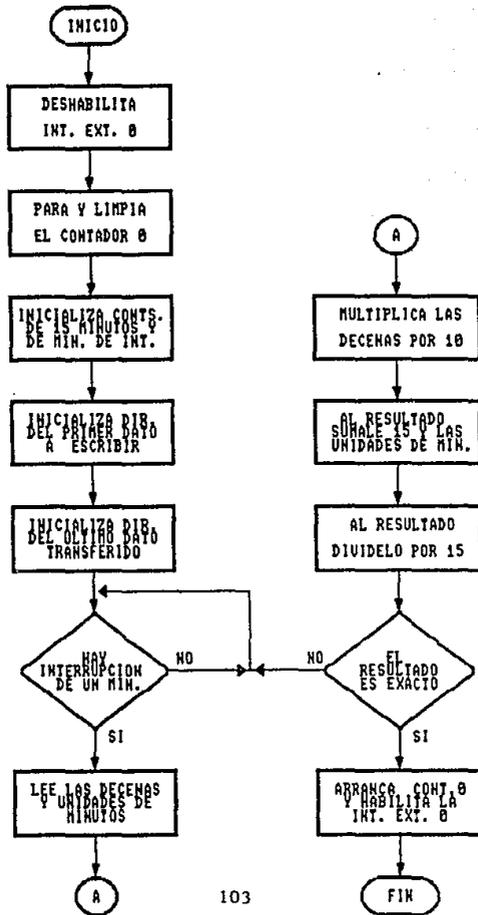
SUBROUTINA AJUSTE



;SUBROUTINA "AJUSTE"
;ESTA SUBROUTINA TIENE COMO FUNCION LA DE AJUSTAR EL RELOJ Y
;ARRANCARLO.

AJUSTE:	CLR	ACC,0	;BANDERA 0 = 0
	MOV	A,#020H	;INICIALIZA LA DIR. DEL RELOJ
	MOV	R1,#024H	;INICIALIZA LA DIR. DE LA RAM
	MOV	@R1,A	;CARGA DATO DE UNIDADES DE SEG.
	INC	R1	;INCREMENTA DIR. DE RAM
	ADD	A,#010H	;INCREMENTA DIR. DE RELOJ
	MOV	@R1,A	;CARGA DATO DE DECENAS DE SEG.
DATO:	INC	R1	;INCREMENTA DIR. DE RAM
	ADD	A,#010H	;INCREMENTA DIR. DE RELOJ
	MOV	RO,A	;SALVA DIR. DE RELOJ
	ACALL	RECEP	;PIDE EL DATO
	ANL	A,#00FH	;SALVA EL DATO
	ORL	A,RO	;UNE DATO (BAJA) CON DIR. (ALTA)
	MOV	@R1,A	;ALMACENALO EN RAM
	MOV	A,RO	;CARGA DIR. DE RELOJ
	CJNE	R1,#030H,DATO	;YA GUARDO TODOS LOS DATOS
	ACALL	RECEP	;PIDE OCURRENCIA ULTIMO AÑO BISI
	INC	R1	;INCREMENTA DIR. DE RAM
	RL	A	;ROTA A LA IZQ. EL DATO
	RL	A	;ROTA A LA IZQ. EL DATO
	ORL	A,#0F1H	;UNE DATO Y DIR. (MODO 24 HRS.)
	MOV	@R1,A	;ALMACENALO EN RAM
	JNB	RI,\$;ESPERA ORDEN DE INICIO
	CLR	RI	;BORRA RI
	ACALL	RELOJ	;CARGA DATOS DEL RELOJ
	RET		

SUBROUTINA EXACTO



```

:SUBROUTINA "EXACTO"
:LA FUNCION DE ESTA SUBROUTINA ES LA DE DETERMINAR CUANDO SE TIENE
:UN TIEMPO DE QUINCE MINUTOS CERRADOS.

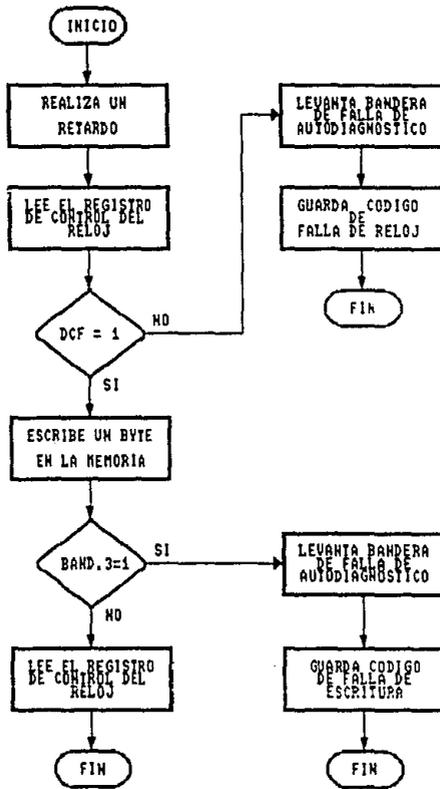
```

```

EXACTO: CLR      EXO          ;DESHABILITA INT. EXT. O
        CLR      TRO          ;PARA EL CONT. O
        MOV      TMO,#000H    ;LIMPIA EL CONT. O
        MOV      TLO,#000H
        MOV      R4,#00FH     ;INICIALIZA CONT. DE 15 MIN
        MOV      R5,#000H     ;INICIALIZA CONT. DE MIN. DE INT.
        SETB     RSO          ;SELECCIONA BANCO 1
        MOV      R5,#000H     ;CARGA DIR. DEL 1ER. DATO A
        MOV      R4,#00BH     ;      ESCRIBIR EN MEMORIA.
        MOV      R3,#000H     ;CARGA DIR. ULTIMO DATO TRANS.
        MOV      R2,#00BH
        CLR      RSO          ;SELECCIONA BANCO 0
AQU1:   JB       P3.2,$       ;ESPERA QUE EXISTA INT. DE MIN.
        ACALL    LRELOJ       ;LEE REG. DE CONTROL Y EL RELOJ
        MOV      R1,#027H     ;CARGA DIR. DE DECENAS DE MIN.
        MOV      A,@R1        ;LEE DATO DE DECENAS DE MIN.
        MOV      B,#00AH     ;CARGA 10 EN B
        MUL      AB           ;MULT. POR 10 LAS DECENAS DE MIN.
        DEC      R1           ;CARGA DIR. DE UNIDADES DE MIN.
        ADD      A,@R1        ;SUMALO AL RESULTADO DE LA MULT.
        ADD      A,#00FH     ;SUMALE 15 AL RESULTADO ANTERIOR
        MOV      B,#00FH     ;CARGA 15 EN EL REG. B
        DIV      AB           ;DIVIDE POR 15 EL TOTAL
        MOV      A,B          ;CARGA EL RESIDUO DE LA DIV.
        JNZ     AQU1         ;SI EL RESIDUO NO ES 0 VE A AQUI
        SETB     TRO          ;ARRANCA CONT. O
        SETB     EXO          ;HABILITA INT. EXT.O
        RET

```

SUBROUTINA TEST1

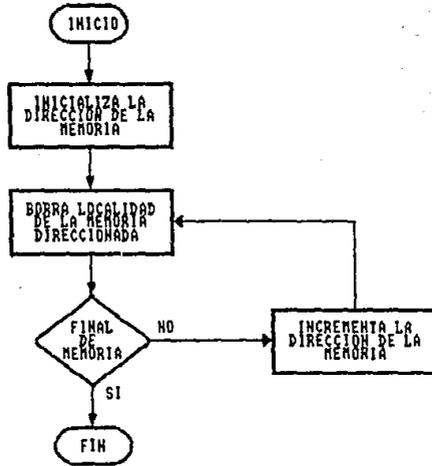


;SUBROUTINA "TEST1"

;ESTA SUBROUTINA REALIZA LA PRUEBA DE ESCRITURA DE UN BYTE EN LA
;MEMORIA Y VERIFICA QUE EL RELOJ ESTE FUNCIONANDO.

TEST1:	CLR	A	;CARGA DIR. DEL REG. DE RELOJ
	ACALL	LEERE	;LEE EL REG.DE CONTROL
	MOV	R1,#OFFH	;INICIA RETARDO
RETAR1:	ACALL	RETAR	
	DJNZ	R1,RETAR1	
	CLR	A	;CARGA DIR. DEL REG. DE CONTROL
	ACALL	LEERE	;LEE EL REG. DE CONTROL (DCF)
	JB	ACC.3,PMEMO	;SI DCF=1 VE A FIN10
	MOV	R6,#002H	;CARGA CODIGO "FALLA DE RELOJ"
	MOV	A,R7	;CARGA EL REG. DE BAND.
	SJMP	LEVAN	
PMEMO:	MOV	DPTR,#1FFFH	;CARGA DIR. DE MEMORIA
	MOV	A,#0AAH	;CARGA UN BYTE
	ACALL	ESCR1	;ESCRIBELO EN MEMORIA
	MOV	A,R7	;CARGA EL REG. DE BAND
	JNB	ACC.3,FIN10	;HAY FALLA DE ESCRITURA
	MOV	R6,#001H	;CARGA CODIGO "FALLA DE ESCRITURA"
LEVAN:	SETB	ACC.2	;LEVANTA BAND. FALLA DE AUTO-DIAG.
	MOV	R7,A	;SALVA EL REG. DE BAND.
FIN10:	RET		

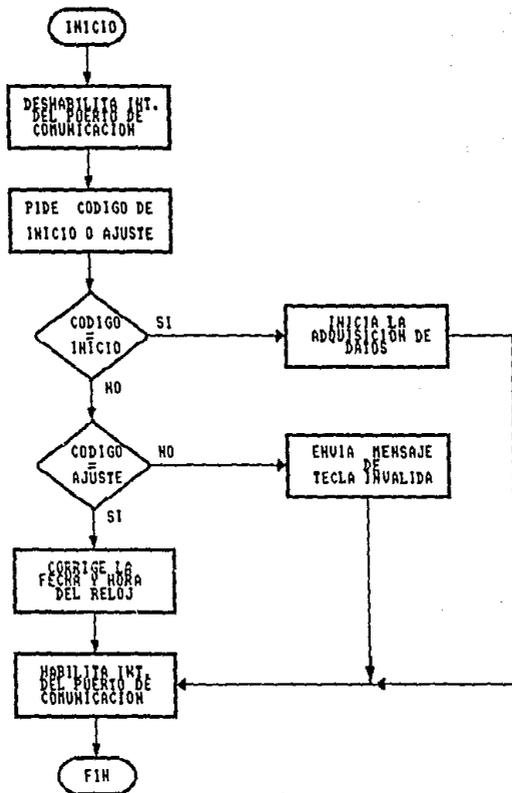
SUBROUTINA BORRA



;SUBROUTINA "BORRA"
;LA FUNCION DE ESTA SUBROUTINA ES LA DE BORRAR LA MEMORIA EEPROM.

```
BORRA: MOV    DPL,R4          ;INICIALIZA DIR. DE LA MEMORIA
        MOV    DPH,R5
LIMPIA: MOV    A,#0FFH       ;CARGA FFH AL ACUMULADOR
        CLR    RSO           ;SELECCIONA EL BANCO 0
        ACALL  ESCR1         ;ESCRIBELO EN LA MEMORIA
        SETB   RSO           ;SELECCIONA EL BANCO 1
        INC   DPTR           ;INCREMENTA DIR. DE MEMORIA
        MOV    A,DPH         ;CARGA LA PARTE ALTA DE LA DIR
        ADD   A,#0E0H        ;SUMALE E0H
        JNZ   LIMPIA        ;SI NO ES 0 VE A LIMPIA
        MOV    A,DPL         ;CARGA LA PARTE BAJA DE LA DIR
        JNZ   LIMPIA        ;SI NO ES 0 VE A LIMPIA
        RET
```

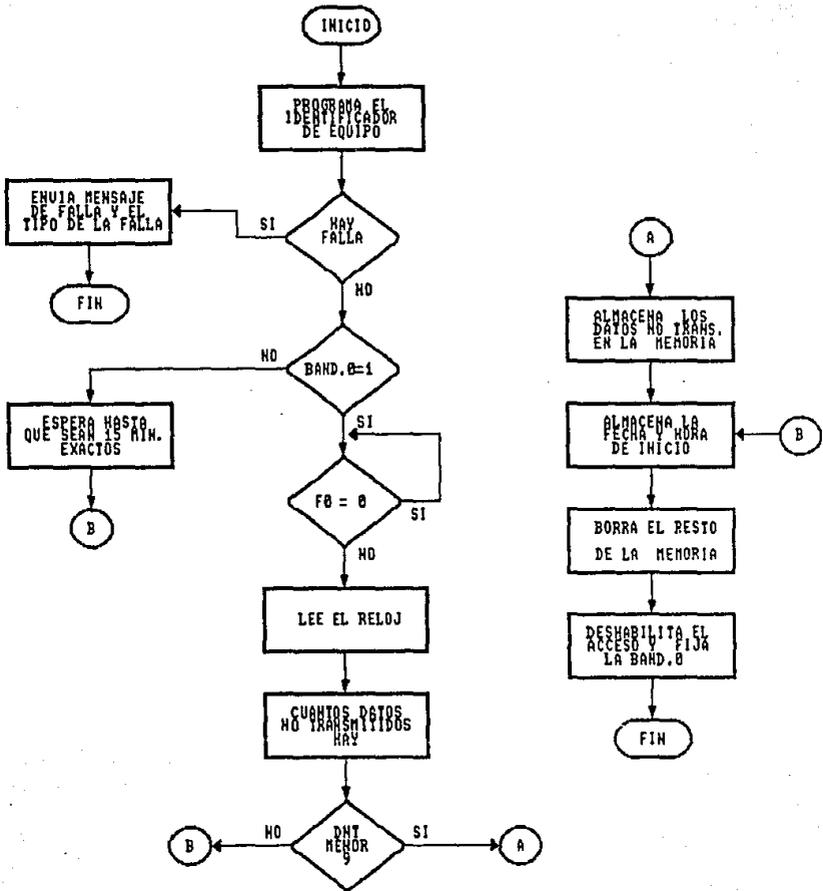
SUBMITIMA REPRO



;SUBROUTINA "REPRO"
;ESTA SUBROUTINA TIENE COMO FUNCION INICIAR LA ADQUISICION DE DATOS
;O AJUSTAR EL RELOJ, DEPENDIENDO DEL CODIGO QUE SEA RECIBIDO.

```
REPRO: CLR      ES           ;DESHABILITA INT. PUERTO COMUNI.  
       ACALL    RECEP       ;PIDE CODIGO  
       CJNE    A,#031H,ARELOJ ;COMPARALO CON EL CODIGO DE ADQUI.  
       ACALL    INICIO      ;SI ES IGUAL INICIA LA ADQUISICION  
       SJMP    FIN8  
ARELOJ: CJNE    A,#044H,ERROR ;COMPARALO CON CODIGO DE AJUSTE  
       ACALL    AJUSTE      ;SI ES IGUAL AJUSTA EL RELOJ  
       SJMP    FIN8  
ERROR: MOV     A,#0F1H      ;CARGA CODIGO "TECLA INVALIDA"  
       ACALL    TRANS       ;TRANSMITELO AL IPE  
FIN8:  SETB    ES           ;HABILITA INT. PUERTO COMUNICACION  
       RET
```

SUBROUTINA INICIO



;SUBROUTINA "INICIO"

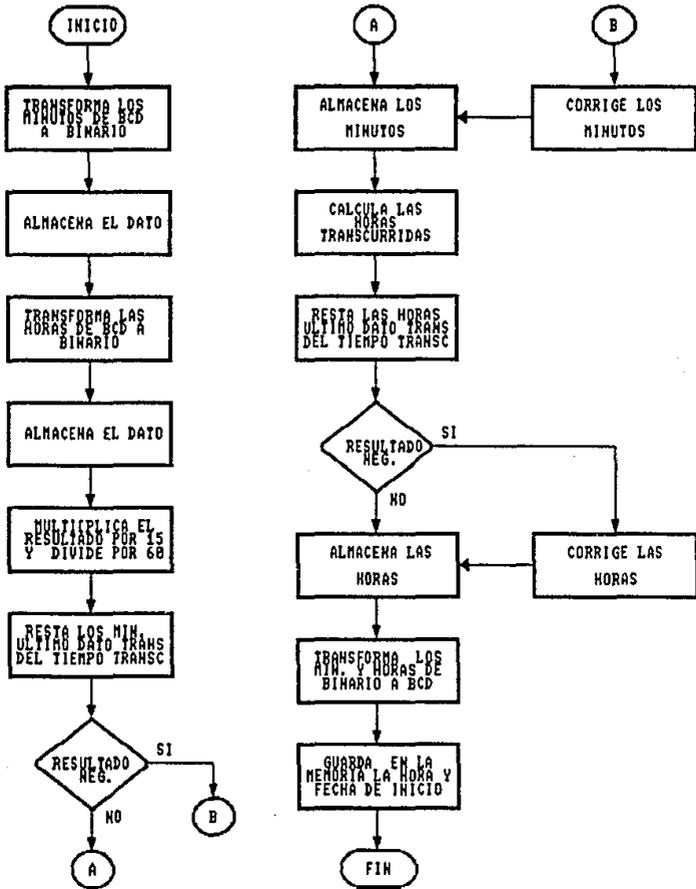
;LA FUNCION DE ESTA SUBROUTINA ES LA DE INICIAR LA ADQUISICION DE
;DATOS, ADEMÁS DE ALMACENAR LOS DATOS QUE NO SE MANDARON EN LA
;TRANSFERENCIA Y ALMACENAR LA HORA Y FECHA DE INICIO DE LA
;ADQUISICION.

```
INICIO: MOV    DPTR,#00000H    ;CARGA DIR. DE MEMORIA
        MOV    A,#001H      ;CARGA IDENTIFICADOR DEL EQUIPO
        ACALL  ESCR1        ;ESCRIBELO EN LA MEMORIA
        MOV    A,R7        ;CARGA REG. DE BANDERAS
        JB     ACC.3,MAL    ;VE SI HAY FALLA
        MOV    A,#0F9H     ;SI NO HAY CARGA CODIGO "INICIO"
        ACALL  TRANS       ;TRANSMITELO AL IPE
        MOV    A,R7        ;CARGA EL REG. DE BANDERAS
        JB     ACC.0,INICIA ;VE A INICIA SI ACC.0 ES 1
        ACALL  EXACTO      ;ESPERA QUE SEAN 15 MIN. EXACTOS
        SETB  RSO         ;SELECCIONA EL BANCO 1
        MOV    RO,#025H    ;INICIALIZA LA DIR. DE RAM
        ACALL  HYF        ;ALMACENA FECHA Y HORA DE INICIO
        MOV    R5,#000H    ;INICIALIZA LA DIR. DE LOS DATOS
        MOV    R4,#00BH
        SJMP  LIMP1       ;VE A LIMP1
INICIA: CLR    FO         ;BORRA FO
        JNB   FO,$        ;ESPERA HASTA QUE FO SEA 1
        ACALL  LRELOJ     ;LEE EL RELOJ
        SETB  RSO        ;SELECCIONA BANCO 1
        MOV    R1,#000H    ;INICIALIZA "DATOS NO TRANS."
        MOV    DPL,R2     ;CARGA LA DIRECCION DEL ULTIMO
        MOV    DPH,R3     ;          DATO TRANSMITIDO
CON:    INC   DPTR       ;INC. DIR. ULTIMO DATO TRANS.
        INC   R1        ;INCREMENTA DATOS NO TRANS.
        CLR   C         ;BORRA EL CARRY
        MOV   A,R5      ;CARGA DIR. ULTIMO DATO ALMACENADO
        SUBB  A,DPH     ;RESTALE DIR. ULTIMO DATO TRANS
        JNZ  CON       ;SI NO ES 0 VE A CON
        CLR  C         ;BORRA EL CARRY
        MOV  A,R4      ;CARGA DIR. ULTIMO DATO ALMACENADO
        SUBB A,DPL     ;RESTALE DIR. ULTIMO DATO TRANS.
```

	JNZ	COH	:SI NO ES O VE A CONT
	MOV	A,R1	:CARGA LOS DATOS NO TRANSMITIDOS
	MOV	R5,A	:SALVA LOS DATOS NO TRANS
	MOV	DPTR,#000BH	:INICIALIZA LA DIR. DE MEMORIA
	MOV	B,#013H	:CARGA EN B EL NUM. 19
	DIV	AB	:DIVIDE DATOS NO TRANS. ENTRE 19
	JNZ	HOFE	:BRINCA SI EL RESULTADO NO ES O
	MOV	RO,#035H	:INICIALIZA DIR. DE RAM
	MOV	DPL,R2	:CARGA DIR. DEL ULTIMO DATO TRANS.
	MOV	DPH,R3	
LEEDS:	CLR	RSO	:SELECCIONA EL BANCO O
	ACALL	LECTU	:LEE EL DATO DE LA MEMORIA
	SETB	RSO	:SELECCIONA EL BANCO 1
	MOV	RO,A	:ALMACENALO EN LA RAM
	INC	RO	:INCREMENTA LA DIR. DE RAM
	INC	DPTR	:INC. DIR. ULTIMO DATO TRANS.
	DJNZ	R1,LEEDS	:SI R1 NO ES O VE A LEEDS
	MOV	A,R5	:SALVA LOS DATOS NO TRANS
	MOV	R2,A	
	MOV	RO,#035H	:INICIALIZA LA RAM
	MOV	DPTR,#000BH	:INICIALIZA LA DIR. DE LA MEMORIA
SACAD:	MOV	A,RO	:SACA EL DATO DE RAM
	CLR	RSO	:SELECCIONA EL BANCO O
	ACALL	ESCR1	:ESCRIBELO EN LA MEMORIA
	SETB	RSO	:SELECCIONA EL BANCO 1
	INC	RO	:INCREMENTA DIR. DE RAM
	INC	DPTR	:INCREMENTA DIR. DE MEMORIA
	DJNZ	R5,SACAD	:SI R5 NO ES O VE A SACAD
HOFE:	MOV	R4,DPL	:GUARDA LA DIR. DE LA MEMORIA
	MOV	R5,DPH	
	ACALL	HOYFE	:ALMACENA HORA Y FECHA DE INICIO
LIMP1:	ACALL	BORRA	:BORRA EL RESTO DE LA MEMORIA
	MOV	R6,#000H	:INICIALIZA CONT. DE 96 DATOS
	MOV	R7,#01FH	:INICIALIZA CONT. DE DIAS
	CLR	RSO	:SELECCIONA EL BANCO O
	MOV	A,R7	:CARGA EL REG. DE BANDERAS
	SETB	ACC.0	:BANDERA O ES 1
	CLR	ACC.1	:DESHABILITA ACCESO
	MOV	R7,A	:SALVA REG. DE BANDERAS

```
          SJMP    FIN11
MAL:      MOV     A,#0F5H      ;CARGA CODIGO DE FALLA
          ACALL  RECP         ;TRANSMITE Y ESPERA CONTESTACION
          MOV     A,#001H     ;CARGA CODIGO "FALLA DE ESCRITURA"
          ACALL  TRANS        ;TRANSMIENDO AL IPE
FIN11:    RET
```

SUBROUTINA HOYFE



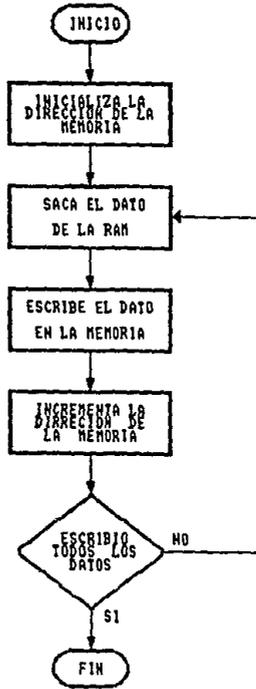
;SUBROUTINA "HOYFE"

;LA FUNCION DE ESTA SUBROUTINA ES LA DE ALMACENAR EN LA MEMORIA LA
;FECHA Y HORA DE INICIO DE LA ADQUISICION DE DATOS DE DEMANDA Y
;ENERGIA ELECTRICA.

```
HOYFE : MOV    RO,#029H      ;INICIALIZA DIR. DE RAM
BINARI: MOV    B,#00AH      ;CARGA EN B EL NUM. 10
MOV    A,@RO              ;SACA EL DATO
MUL    AB                 ;MULTIPLICA EL DATO POR 10
DEC    RO                 ;DECREMENTA LA DIR. DE RAM
ADD    A,@RO              ;SUMA EL DATO DE LA RAM
MOV    @RO,A              ;GUARDA EL RESULTADO
DEC    RO                 ;DECREMENTA LA DIR DE RAM
CJNE   RO,#025H,BINARI    ;VE A BINARI SI FALTAN DATOS
MOV    B,#002H            ;CARGA B EL NUM. 2
MOV    A,R2               ;CARGA LOS DATOS NO TRANSFERIDOS
DIV    AB                 ;DIVIDE DATOS NO TRANS. POR 2
MOV    B,#00FH            ;CARGA EN B EL NUM. 15
MUL    AB                 ;MULTIPLICA POR 15 EL RESULTADO
MOV    B,#03CH            ;CARGA EN B EL NUM. 60
DIV    AB                 ;DA EL TIEMPO TRANSCURRIDO
MOV    R3,#002H           ;CARGA EN R3 EL NUM. 2
MOV    R1,#03CH           ;CARGA EL DATO DE CORRECCION DE MIN.
CLR    C                  ;BORRA EL CARRY
BUENO1: INC    RO          ;INCREMENTA LA DIR. DE RAM
XCH    A,B                ;INTERCAMBIA LOS CONTENIDOS
XCH    A,@RO              ;INTERCAMBIA LOS CONTENIDOS
SUBB   A,@RO              ;RESTA EL DATO DEL ACC. CON EL RAM
INC    RO                  ;INCREMENTA LA DIR. DE RAM
MOV    @RO,A              ;SALVA EL RESULTADO
JNC    BUENO              ;BRINCA SI EL RESULTADO ES NEG.
MOV    A,R1                ;CARGA EL DATO DE CORRECCION
ADD    A,@RO              ;CORRIGE EL RESULTADO NEG.
SETB   C                  ;FLJA EL CARRY
MOV    @RO,A              ;SALVA EL DATO
BUENO : MOV    R1,#017H    ;CARGA EL DATO DE CORRECCION DE HORAS
DJNZ   R3,BUENO1         ;SI R3=1 VE A BUENO1
```

	MOV	R3,#002H	;CARGA R3 CON EL NUM. 2
CAMBIA:	MOV	B,#00AH	;CARGA B CON EL NUM. 10
	MOV	A,@RO	;CARGA EL DATO DE LA RAM
	DIV	AB	;DIVIDE EL DATO ENTRE 10
	MOV	@RO,A	;SALVA DATO DE LAS DECENAS
	MOV	A,B	;CARGA DATO DE LAS UNIDADES
	DEC	RO	;DECREMENTA LA DIR. DE RAM
	MOV	@RO,A	;SALVA DATO DE LAS UNIDADES
	DEC	RO	;DECREMENTA LA DIR. DE RAM
	DJNZ	R3,CAMBIA	;SI R3 NO ES O VE A CAMBIA
	ACALL	HYF	;GUARDA FECHA Y HORA DE INICIO
	RET		

SUBRUTINA HVP

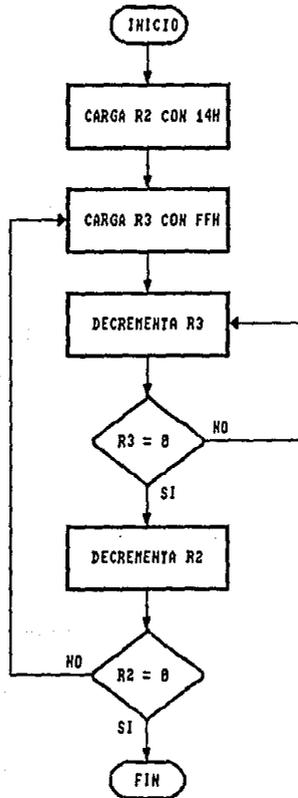


;SUBROUTINA "HYF"

;LA FUNCION DE ESTA SUBROUTINA ES LA DE ALMACENAR LOS DATOS EN LA
;EEPROM.

```
HYF:  MOV    DPTR,#0001H    ;CARGA LA DIR. DE MEMORIA
GUARDA: INC   RO           ;INCREMENTA LA DIR. DE RAM
      MOV    A,RO          ;SACA EL DATO DE LA RAM
      CLR    RSO           ;SELECCIONA EL BANCO 0
      ACALL  ESCRI        ;ESCRIBELO EN LA MEMORIA
      SETB   RSO           ;SELECCIONA EL BANCO 1
      INC   DPTR          ;INCREMENTA LA DIR. DE MEMORIA
      CJNE  RO,#02FH,GUARDA ;SI RO NO ES 2FH VE A GUARDA
      RET
```

SUBROUTINA RETAR



:SUBROUTINA "RETAR"
: LA FUNCION DE ESTA SUBROUTINA ES LA DE REALIZAR UN RETARDO DE
:10 mseg.

RETAR:	SETB	RS1	:SELECCIONA EL BANCO 2
	MOV	R2,#014H	:RETARDO DE 10 mseg
V1:	MOV	R3,#OFFH	
	DJNZ	R3,\$	
	DJNZ	R2,V1	:FIN DEL RETARDO
	CLR	RS1	:SELECCIONA BANCO 0
	RET		
	EHD		

4.3 Software para el equipo IPE.

En la figura 4.2, se presenta un diagrama de flujo de la estructura principal del programa de operación del IPE, el cual se describe a continuación.

Al igual que para la UAD, el primer bloque corresponde a las instrucciones de inicialización de operación del circuito al aplicar la energía eléctrica.

En estas circunstancias, se realizan las instrucciones necesarias para:

- Deshabilita circuitos periféricos.
- Programa e inicializa el puerto de comunicaciones.
- Programa prioridades de interrupción.
- Inicializa contadores usados en el programa.
- Inicializa "banderas" usadas en el programa.
- Realiza una subrutina de prueba de autodiagnóstico.
- Despliega mensaje de resultado de autodiagnóstico.
- Habilita interrupción por teclado en caso de diagnóstico positivo.
- El microcontrolador se coloca en modo "idle".

El equipo IPE, no requiere de un procedimiento de inicialización por parte del operador para su operación normal, y en el caso de que la prueba de autodiagnóstico arroje un resultado positivo, el equipo entrará en modo de operación normal, aceptando los comandos generados por el operador por medio del teclado.

Cuando una tecla es oprimida por el operador, se generará una interrupción en el puerto de comunicaciones, la cual es atendida realizando la subrutina "ONTEC".

En esta subrutina, se compara la tecla oprimida con los códigos de aquellos comandos aceptables que son:

A modo de ACCESO.

B modo de INTERROGACION.

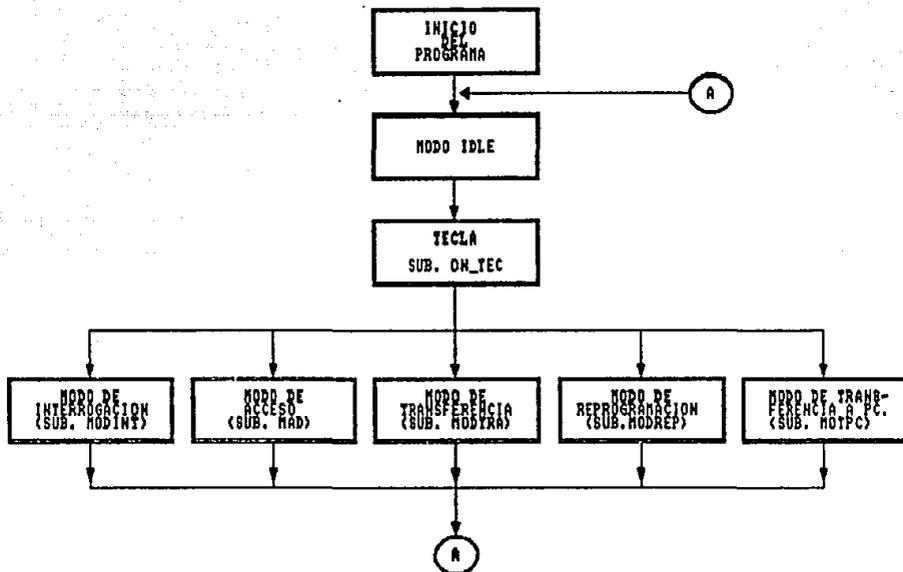


FIG. 4.2 ESTRUCTURA DEL SOFTWARE DEL EQUIPO IPE-1-11E.

C modo de TRANSFERENCIA PARA UNA UAD.

D modo de REPROGRAMACION.

E modo de TRANSFERENCIA A UN COMPUTADOR PERSONAL.

En el caso de que la tecla oprimida no corresponda a uno de los códigos aceptables, se mostrará al operador en el monitor alfanumérico un mensaje de "TECLA INVALIDA", durante diez segundos y se regresará al modo "idle", para la espera de la introducción de una nueva tecla.

Si la tecla oprimida corresponde a un código válido, se atenderá el comando realizando la subrutina correspondiente.

En términos generales, cada una de las subrutinas realizadas por el reconocimiento de un comando, realiza las siguientes actividades:

MODO DE ACCESO.

- Por medio del monitor, se solicita al operador la introducción por teclado de un código de 4 dígitos.
- El código introducido es enviado a la UAD y se adquiere una respuesta.
- Se muestra al operador el mensaje de "ACCESO PERMITIDO", en el caso de que la UAD reconozca el código enviado. De lo contrario, un mensaje de "ACCESO NEGADO" es presentado en el monitor.

MODO DE INTERROGACION.

- Se envía a la UAD el comando correspondiente.
- Se espera por una respuesta de la UAD.
- Se establece un protocolo de petición y envío de datos, hasta que toda la información determinada en las especificaciones de diseño del capítulo 2, sea transferida al IPE.
- El IPE, muestra al operador la información recibida, haciendo uso del monitor alfanumérico.

MODO DE TRANSFERENCIA PARA UNA UAD.

- El IPE verifica que exista un cartucho de memoria instalado en su conector y verifica su buen estado, generando un mensaje en el caso de falla.
- Se envía a la UAD el comando correspondiente.
- Se espera por una respuesta de la UAD.
- Se establece un protocolo de petición-envío de la información almacenada en la memoria EEPROM de la UAD.
- Cada byte recibido es almacenado en el cartucho de memoria instalado en el IPE.
- Se genera un mensaje de FIN, indicando con esto que toda la información ha sido transferida.

MODO DE REPROGRAMACION.

- Se pregunta al operador si desea iniciar un nuevo período de adquisición de datos o si desea reprogramar el reloj en tiempo real.
- Para iniciar un nuevo período de adquisición de datos, el comando correspondiente es enviado a la UAD y se genera un mensaje en el monitor de "INICIO DE LA UAD".
- Para reprogramación del reloj, se solicita al operador toda la información necesaria (hora, fecha, etc.). Dicha información es transferida a la UAD, precedida por un comando de modo de operación y mediante un protocolo de petición-envío de información.
- Se pide al operador que en el momento que desee iniciar el conteo de tiempo, introduzca un comando el cual es transmitido a la UAD.

MODO DE TRANSFERENCIA A COMPUTADOR PERSONAL.

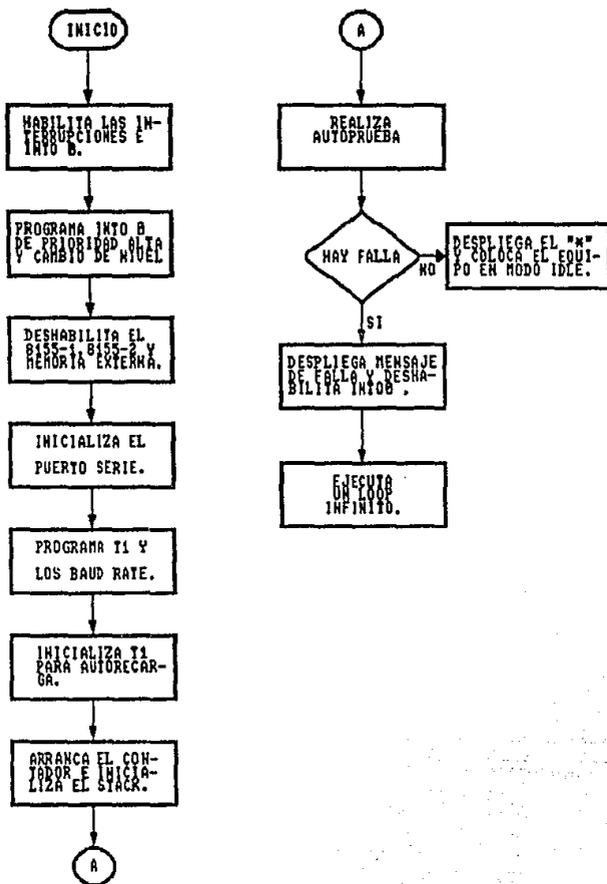
- En este modo de operación, el IPE funciona como un equipo periférico a un computador personal, aceptando órdenes enviadas desde este último.
- Al recibir un código reconocible, contesta con un mensaje para indicar al computador que la comunicación es aceptada.

- Al recibir un comando de transferencia (A), procede a leer y transmitir hacia el computador personal la información almacenada en el cartucho de memoria instalado.

En todos los modos de operación, existen subrutinas que permiten al operador un pleno conocimiento de lo que está sucediendo, por ejemplo:

- Cuando el operador oprime una tecla no reconocible por el equipo, se genera automáticamente un mensaje de "TECLA INVALIDA".
- Cuando la UAD no responde a un comando enviado por el IPE, se despliega un mensaje de "NO HAY RESPUESTA".
- Cuando el operador intenta las opciones de REPROGRAMACION o TRANSFERENCIA sin haber realizado un procedimiento de acceso, se genera un mensaje de "NO ACCESO PREVIO".

PROGRAMA PRINCIPAL.



page

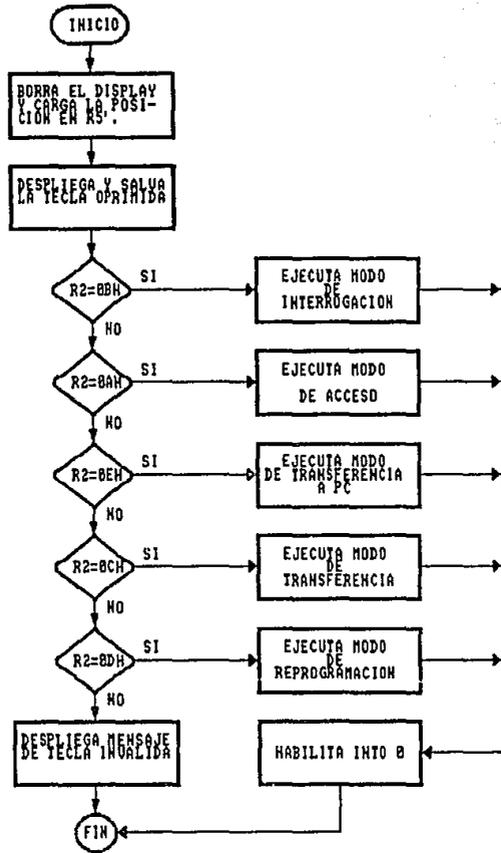
;PROGRAMA PRINCIPAL.
;EN ESTE PROGRAMA SE SELECCIONA LA FUNCION QUE DEBE SER
;REALIZADA POR EL EQUIPO IPE-1-IIE.
;LAS FUNCIONES QUE SE PUEDEN SELECCIONAR SON:MODO DE
;INTERROGACION A UN EQUIPO UAD-1-IIE,MODO DE ACCESO A UN
;UAD-1-IIE,MODO DE TRANSFERENCIA A UNA PC,MODO DE
;TRANSFERENCIA DE DATOS Y REPROGRAMACION DEL UAD-1-IIE.

```
DEFSEG ZERO,CLASS=CODE,START=000H
DEFSEG MAIN,CLASS=CODE,START=100H
DEFSEG RESTART,CLASS=CODE,START=03H
SEG ZERO
ORG 00H
JMP INICIO
SEG RESTART
ORG 03H
JMP ON_TEC
SEG MAIN
ORG 100H
```

```
INICIO: SETB IE.7 ;HABILITA LAS INTERRUPTACIONES
SETB IE.0 ;HABILITA INTO
SETB IP.0 ;PROGRAMA INTO DE PRI. ALTA
CLR TCON.0 ;PROGRAMA INTO COMO NIVEL
MOV P1,#9EH ;DESHABILITA EXP. Y MEM. EXT.
MOV SCON,#01010010B ;INICIALIZA EL PUERTO SERIE
MOV TMOD,#00100000B ;PROGRAMA T1
MOV PCON,#000H ;PROGRAMA LOS BAUD RATE
MOV TH1,#0E9H ;TIMER1 PARA AUTORECARGA
SETB TR1 ;ARRANCA EL CONTADOR
MOV SP,#57H ;CARGA EN EL STACK EL #57
CALL AUTO ;REALIZA AUTOPRUEBA
MOV R7,#002H ;CARGA COD. DE DIAG. POSI.
```

```
CALL    DISP      ;COLOCA MENSAJE EN EL DISPLAY
LOOP_0: JMP     $
        JMP     LOOP_0
```

SUBROUTINA ON_TEC



```

;SUBROUTINA ON_TEC
;EN ESTA SUBROUTINA SE ATIENDE LA INTERRUPCION GENERADA POR
;UNA TECLA OPRIMIDA EN EL TECLADO.

```

```

ON_TEC: CLR      IE.0           ;DESHABILITA INTO 0
        CALL    BORRA         ;BORRA EL DISPLAY
        SETB   RSO           ;SELECCIONA EL BANCO 1
        MOV    R5,#080H      ;CARGA EN R5' LA POSICION
        CLR    RSO           ;SELECCIONA EL BANCO 0
        CALL    MRT0         ;LLAMA A MRT0
        CJNE   R2,#00BH,COMP_1 ;BRINCA SI R2 NO ES 00BH
        CALL    MODINT       ;ENTRA MODO DE INTERROGACION
        SETB   IE.0         ;HABILITA INTO 0
        CALL    MAD          ;EJECUTA MODO DE ACCESO
        SETB   IE.0         ;HABILITA INTO 0
        RETI

COMP_2: CJNE   R2,#00EH,COMP_3 ;BRINCA SI R2 NO ES 00E
        CALL    MOTPC        ;MODO DE TRANSF. A PC
        SETB   IE.0         ;HABILITA INTO 0
        RETI

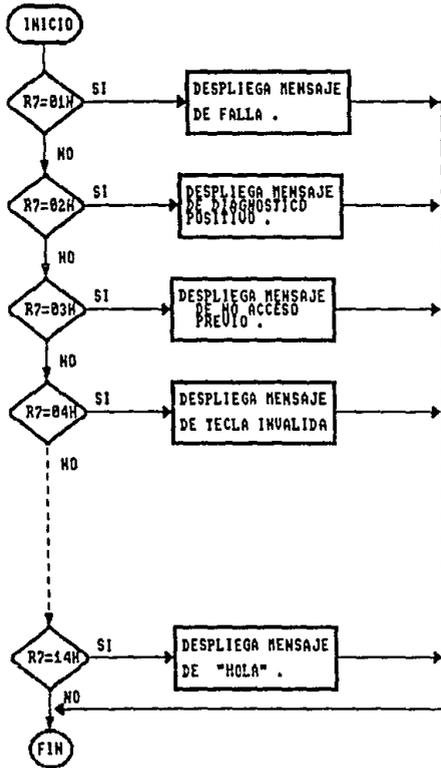
COMP_3: CJNE   R2,#00CH,COMP_4 ;BRINCA SI R2 NO ES 00C
        CALL    MODTRA       ;ENTRA MODO DE TRANSF.
        SETB   IE.0         ;HABILITA INTO 0
        RETI

COMP_4: CJNE   R2,#00DH,MENTIN ;BRINCA SI R2 NO ES 00D
        CALL    MODREP       ;ENTRA MODO DE REPROGRA.
        SETB   IE.0         ;HABILITA INTO 0
        RETI

MENTIN: MOV    R7,#004H      ;CARGA EN R7 EL #04H.
        CALL    DISP         ;DESPLIEGA EN EL DISPLAY
        SETB   IE.0         ;HABILITA INTO 0
        RETI

```

SUBROUTINA DISP



```

;SUBROUTINA DISP
;ESTA SUBROUTINA SE ENCARGA DE COMPARAR Y DESPLEGAR EL
;MENSAJE CORRESPONDIENTE A CADA CODIGO ENVIADO EN EL
;REGISTRO R7.

```

```

DISP : CJNE R7,#001H,ETIQ_1 ;BRINCA SI R7 NO ES 001
      CALL MENS_1 ;LLAMA A LA SUBROUTINA MENS_1
      JMP OUT_S ;BRINCA A OUT_S
ETIQ_1: CJNE R7,#002H,ETIQ_2 ;BRINCA SI R7 NO ES 002
      CALL MENS_2 ;LLAMA A LA SUBROUTINA MENS_2
      JMP OUT_S ;BRINCA A OUT_S
ETIQ_2: CJNE R7,#003H,ETIQ_3 ;BRINCA SI R7 NO ES 003
      CALL MENS_3 ;LLAMA A LA SUBROUTINA MENS_3
      JMP OUT_S ;BRINCA A OUT_S
ETIQ_3: CJNE R7,#004H,ETIQ_4 ;BRINCA SI R7 NO ES 004
      CALL MENS_4 ;LLAMA A LA SUBROUTINA MENS_4
      JMP OUT_S ;BRINCA A OUT_S
ETIQ_4: CJNE R7,#005H,ETIQ_5 ;BRINCA SI R7 NO ES 005
      CALL MENS_5 ;LLAMA A LA SUBROUTINA MENS_5
      JMP OUT_S ;BRINCA A OUT_S
ETIQ_5: CJNE R7,#006H,ETIQ_6 ;BRINCA SI R7 NO ES 006
      CALL MENS_6 ;LLAMA A LA SUBROUTINA MENS_6
      JMP OUT_S ;BRINCA A OUT_S
ETIQ_6: CJNE R7,#007H,ETIQ_7 ;BRINCA SI R7 NO ES 007
      CALL MENS_7 ;LLAMA A LA SUBROUTINA MENS_7
      JMP OUT_S ;BRINCA A OUT_S
ETIQ_7: CJNE R7,#008H,ETIQ_8 ;BRINCA SI R7 NO ES 008
      CALL MENS_8 ;LLAMA A LA SUBROUTINA MENS_8
      JMP OUT_S ;BRINCA A OUT_S
ETIQ_8: CJNE R7,#009H,ETIQ_9 ;BRINCA SI R7 NO ES 009
      CALL MENS_9 ;LLAMA A LA SUBROUTINA MENS_9
      JMP OUT_S ;BRINCA A OUT_S
ETIQ_9: CJNE R7,#00AH,ETIQ_A ;BRINCA SI R7 NO ES 00A
      CALL MENS_A ;LLAMA A LA SUBROUTINA MENS_A
      JMP OUT_S ;BRINCA A OUT_S
ETIQ_A: CJNE R7,#00BH,ETIQ_B ;BRINCA SI R7 NO ES 00B

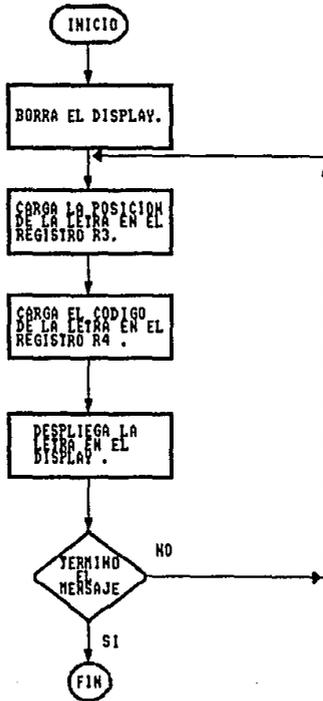
```

```

CALL MENS_B ;LLAMA A LA SUBRUTINA MENS_B
JMP OUT_S ;BRINCA A OUT_S
ETIQ_B: CJNE R7,#00CH,ETIQ_C ;BRINCA SI R7 NO ES 00C
CALL MENS_C ;LLAMA A LA SUBRUTINA MENS_C
JMP OUT_S ;BRINCA A OUT_S
ETIQ_C: CJNE R7,#00DH,ETIQ_D ;BRINCA SI R7 NO ES 00D
CALL MENS_D ;LLAMA A LA SUBRUTINA MENS_D
JMP OUT_S ;BRINCA A OUT_S
ETIQ_D: CJNE R7,#00EH,ETIQ_E ;BRINCA SI R7 NO ES 00EH
CALL MENS_E ;LLAMA A LA SUBRUTINA MENS_E
JMP OUT_S ;BRINCA A OUT_S
ETIQ_E: CJNE R7,#00FH,ETIQ_F ;BRINCA SI R7 NO ES 00F
CALL MENS_F ;LLAMA A LA SUBRUTINA MENS_F
JMP OUT_S ;BRINCA A OUT_S
ETIQ_F: CJNE R7,#010H,ETIQ_G ;BRINCA SI R7 NO ES 10H
CALL MENS_G ;LLAMA A LA SUBRUTINA MENS_G
JMP OUT_S ;BRINCA A OUT_S
ETIQ_G: CJNE R7,#011H,ETIQ_H ;BRINCA SI R7 NO ES 11H
CALL MENS_H ;LLAMA A LA SUBRUTINA MENS_H
JMP OUT_S ;BRINCA A OUT_S
ETIQ_H: CJNE R7,#012H,ETIQ_I ;BRINCA SI R7 NO ES 12H
CALL MENS_I ;LLAMA A LA SUBRUTINA MENS_I
JMP OUT_S ;BRINCA A OUT_S
ETIQ_I: CJNE R7,#013H,OUT_S ;BRINCA SI R7 NO ES 13H
CALL MENS_J ;LLAMA A LA SUBRUTINA MENS_J
OUT_S : RET

```

SUBROUTINA KENS_1 .



```

;SUBROUTINA MENS_1
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR EN EL DISPLAY UN
;MENSAJE DE FALLA EN EL EQUIPO IPE-1-1IE.

```

```

MENS_1: CALL BORRA ;LLAMA A LA SUBROUTINA BORRA
MOV R3,#081H ;CARGA EN R3 LA POSICION
MOV R4,#046H ;CARGA EN R4 EL CODIGO
CALL DISPL_ ;LLAMA A LA SUBROUTINA DISPL_
MOV R3,#083H ;CARGA EN R3 LA POSICION SIG.
MOV R4,#041H ;CARGA EN R4 EL COD. DE "A"
CALL DISPL_ ;LLAMA A LA SUBROUTINA DISPL_
MOV R3,#085H ;CARGA EN R3 LA POSICION SIG.
MOV R4,#04CH ;CARGA EN R4 EL COD. DE "L"
CALL DISPL_ ;LLAMA A LA SUBROUTINA DISPL_
MOV R3,#087H ;CARGA EN R3 LA POSICION SIG.
MOV R4,#04CH ;CARGA EN R4 EL COD. DE "L"
CALL DISPL_ ;LLAMA A LA SUBROUTINA DISPL_
MOV R3,#0C1H ;CARGA EN R3 LA POSICION SIG.
MOV R4,#041H ;CARGA EN R4 EL COD. DE "A"
CALL DISPL_ ;LLAMA A LA SUBROUTINA DISPL_
MOV R3,#0C3H ;CARGA EN R3 LA POSICION SIG.
MOV R4,#0BOH ;CARGA EN R4 EL COD. DE "-"
CALL DISPL_ ;LLAMA A LA SUBROUTINA DISPL_
MOV R3,#0C5H ;CARGA EN R3 LA POSICION SIG.
SETB RSO ;SELECCIONA EL BANCO 1
MOV A,R4 ;CARGA EN A EL REGISTRO R4'
CLR RSO ;SELECCIONA EL BANCO 0
MOV R4,A ;CARGA EN R4 EL ACUMULADOR
CALL DISPL_ ;LLAMA A LA SUBROUTINA DISPL_
RET

```

;SUBROUTINA MENS_2
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR EN EL DISPLAY UN
;MENSAJE DE DIAGNOSTICO POSITIVO EN EL EQUIPO IPE-1-III.

MENS_2:	CALL	BORRA	:BORRA EL DISPLAY
	MOV	R3,#080H	:CARGA EN R3 LA POSICION
	MOV	R4,#02AH	:CARGA EN R4 EL COD. DE "+"
	CALL	DISPL_	:LLAMA A LA SUBROUTINA DISPL_
	RET		

```

;SUBROUTINA MENS_3
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR EN EL DISPLAY UN
;MENSAJE DE AVISO,RECORDANDO AL OPERADOR DEL EQUIPO QUE SE
;DEBE REALIZAR OTRA ACCION PREVIA A LA QUE SE DESEA REALIZAR.

```

```

MENS_3: CALL   BORRA           ;BORRA EL DISPLAY
        MOV    R3,#080H       ;INICIALIZA R3
        MOV    R4,#04EH       ;DESPLIEGA "N"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        MOV    R4,#04FH       ;DESPLIEGA "O"
        CALL   DISPL_
        MOV    R3,#083H       ;CARGA LA DIR. SIGUIENTE
        MOV    R4,#041H       ;DESPLIEGA "A"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        MOV    R4,#043H       ;DESPLIEGA "C"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        MOV    R4,#043H       ;DESPLIEGA "C"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        MOV    R4,#045H       ;DESPLIEGA "E"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        MOV    R4,#053H       ;DESPLIEGA "S"
        CALL   DISPL_
        MOV    R3,#0C0H       ;CARGA LA SIGUIENTE DIR.
        MOV    R4,#04FH       ;DESPLIEGA "O"
        CALL   DISPL_
        MOV    R3,#0C2H       ;CARGA LA DIR. SIGUIENTE
        MOV    R4,#050H       ;DESPLIEGA "P"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        MOV    R4,#052H       ;DESPLIEGA "R"

```

```
CALL  DISPL_
INC   R3           ;INCREMENTA R3
MOV   R4,#045H    ;DESPLIEGA "E"
CALL  DISPL_
INC   R3           ;INCREMENTA R3
MOV   R4,#056H    ;DESPLIEGA "V"
CALL  DISPL_
INC   R3           ;INCREMENTA R3
MOV   R4,#049H    ;DESPLIEGA "I"
CALL  DISPL_
INC   R3           ;INCREMENTA R3
MOV   R4,#04FH    ;DESPLIEGA "O"
CALL  DISPL_
RET
```

```

;SUBROUTINA MENS_4
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR UN MENSAJE DE ERROR
;AL OPRIMIR UNA TECLA NO VALIDA.

```

```

MENS_4: CALL   BORRA           ;BORRA EL DISPLAY
        MOV    R3,#081H       ;CARGA LA POSICION EN R3
        MOV    R4,#054H       ;DESPLIEGA "I"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#045H       ;DESPLIEGA "E"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#043H       ;DESPLIEGA "C"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#04CH       ;DESPLIEGA "L"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#041H       ;DESPLIEGA "A"
        CALL   DISPL_
        MOV    R3,#087H       ;CARGA EN R3 LA POSICION
        MOV    R4,#049H       ;DESPLIEGA "I"
        CALL   DISPL_
        MOV    R3,#0CCH       ;CARGA EN R3 LA POSICION
        MOV    R4,#04EH       ;DESPLIEGA "N"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#056H       ;DESPLIEGA "V"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#041H       ;DESPLIEGA "A"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#04CH       ;DESPLIEGA "L"

```

```
CALL  DISPL_
INC   R3           ;INCREMENTA E3
MOV   R4,#049H    ;DESPLIEGA "-"
CALL  DISPL_
INC   R3           ;INCREMENTA E3
MOV   R4,#044H    ;DESPLIEGA "D"
CALL  DISPL_
INC   R3           ;INCREMENTA E3
MOV   R4,#041H    ;DESPLIEGA "A"
CALL  DISPL_
RET
```

```

;SUBROUTINA MENS_5
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR EL MENSAJE DEL MODO
;DE INTERROGACION.

```

```

MENS_5: CALL   BORRA           ;BORRA EL DISPLAY
        MOV    R3,#080H       ;CARGA EN R3 LA POSICION
        MOV    R4,#049H       ;DESPLIEGA "I"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#04EH       ;DESPLIEGA "N"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#054H       ;DESPLIEGA "T"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#045H       ;DESPLIEGA "E"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#052H       ;DESPLIEGA "R"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#052H       ;DESPLIEGA "R"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#04FH       ;DESPLIEGA "O"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#047H       ;DESPLIEGA "G"
        CALL   DISPL_
        MOV    R3,#C30H       ;CARGA EN R3 LA POSICION
        MOV    R4,#041H       ;DESPLIEGA "A"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3
        MOV    R4,#043H       ;DESPLIEGA "C"
        CALL   DISPL_
        INC    R3              ;INCREMENTA R3

```

```
MOV    R4,#049H      ;DESPLIEGA "I"
CALL   DISPL_
INC    R3             ;INCREMENTA R3
MOV    R4,#04FH      ;DESPLIEGA "O"
CALL   DISPL_
INC    R3             ;INCREMENTA R3
MOV    R4,#04EH      ;DESPLIEGA "H"
CALL   DISPL_
MOV    R3,#0C7H      ;CARGA EN R3 LA POSICION
MOV    R4,#03FH      ;DESPLIEGA "?"
CALL   DISPL_
RET
```

```

;SUBROUTINA MENS_6
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR LOS DATOS ACTUALES
;DEL RELOJ,ASI COMO LOS DATOS DE INICIO DEL EQUIPO UAD-1-IIE.

```

```

MENS_6: CALL BORRA           ;BORRA EL DISPLAY
        MOV R3,#0C7H        ;CARGA EN R3 LA POSICION
        MOV A,#0R0
        CALL ACOND_
        MOV R4,A           ;COLOCA LAS UNIDADES DE MIN.
        CALL DISPL_
        CALL AYUDA1        ;COLOCA LAS DECENAS DE MIN.
        DEC R3             ;DECREMENTA R3
        MOV R4,#03AH       ;DESPLIEGA "-"
        CALL DISPL_
        CALL AYUDA1        ;COLOCA LAS UNIDADES DE HORA
        CALL AYUDA1        ;COLOCA LAS DECENAS DE HORA
        MOV R3,#0B7H       ;CARGA LA POSICION EN R3
        INC R0             ;INCREMENTA R0
        MOV A,#0R0
        CALL ACOND_
        MOV R4,A           ;COLOCA LAS UNIDADES DE DIA
        CALL DISPL_
        CALL AYUDA1        ;COLOCA LAS DECENAS DE DIA
        DEC R3             ;DECREMENTA R3
        MOV R4,#0BOH       ;DESPLIEGA "-"
        CALL DISPL_
        CALL AYUDA1        ;COLOCA LAS UNIDADES DE MES
        CALL AYUDA1        ;COLOCA LAS DECENAS DE MES
        DEC R3             ;DECREMENTA R3
        MOV R4,#0BOH       ;DESPLIEGA "-"
        CALL DISPL_
        CALL AYUDA1        ;COLOCA LAS UNIDADES DE A&O
        CALL AYUDA1        ;COLOCA LAS DECENAS DE A&O
        MOV R3,#0C8H       ;CARGA EN R3 LA POSICION

```

CALL DISPL_

RET

```

;SUBROUTINA MENS_7
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR EL ESTADO DEL EQUIPO
;UAD-1-1IE Y EL NUMERO DE DIAS QUE LE RESTAN A SU MEMORIA DE
;ALMACENAMIENTO.

```

```

MENS_7: CALL   BORRA           ;BORRA EL DISPLAY
        MOV    R3,#0C2H       ;CARGA LA POSICION EN R3
        MOV    R4,#053H       ;DESPLIEGA "S"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        MOV    R4,#054H       ;DESPLIEGA "T"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        MOV    R4,#03DH       ;DESPLIEGA "="
        CALL   DISPL_
        MOV    A,#0RO
        CJNE   A,#0F3H,FALL_1 ;BRINCA SI "A" NO ES 0F3H
        INC    R3             ;INCREMENTA R3
        MOV    R4,#04EH       ;DESPLIEGA "N"
        CALL   DISPL_
        JMP    FALL_2
FALL_1: INC    R3             ;INCREMENTA R3
        MOV    R4,#046H       ;DESPLIEGA "F"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        INC    RO             ;INCREMENTA RO
        MOV    A,#0RO
        CALL   ACOND_
        MOV    R4,A           ;COLOCA EL NO. DE LA FALLA
        CALL   DISPL_
FALL_2: MOV    R3,#080H       ;CARGA EN R3 LA POSICION
        MOV    R4,#04EH       ;DESPLIEGA "N"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        MOV    R4,#044H       ;DESPLIEGA "D"
        CALL   DISPL_

```

```

INC      R3                ;INCREMENTA R3
MOV      R4,#03DH         ;DESPLIEGA "="
CALL     DISPL_
INC      RO                ;INCREMENTA RO
MOV      A,@RO
CALL     ACOND_
MOV      R4,A              ;COLOCA LAS UNI. DE NO.DE DIAS
MOV      R3,#084H         ;CARGA EN R3 LA POSICION
CALL     DISPL_
INC      RO                ;INCREMENTA RO
MOV      A,@RO
CALL     ACOND_
MOV      R4,A              ;COLOCA LAS DEC. DE NO.DE DIAS
MOV      R3,#083H         ;CARGA EN R3 LA POSICION
CALL     DISPL_
MOV      R3,#0C8H         ;CARGA EN R3 LA POSICION
CALL     DISPL_
RET

```

;SUBROUTINA MENS_8
 ;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR UN MENSAJE, CUANDO
 ;NO SE HA ESTABLECIDO COMUNICACION CON EL EQUIPO UAD-1-IE.

```

MENS_8: CALL  BORRA           ;BORRA EL DISPLAY
          MOV   R3,#080H      ;CARGA EN R3 LA POSICION
          MOV   R4,#04EH      ;DESPLIEGA "H"
          CALL  DISPL_
          INC   R3            ;INCREMENTA R3
          MOV   R4,#04FH      ;DESPLIEGA "O"
          CALL  DISPL_
          MOV   R3,#083H      ;CARGA EN R3 LA POSICION
          MOV   R4,#048H      ;DESPLIEGA "H"
          CALL  DISPL_
          INC   R3            ;INCREMENTA R3
          MOV   R4,#041H      ;DESPLIEGA "A"
          CALL  DISPL_
          INC   R3            ;INCREMENTA R3
          MOV   R4,#059H      ;DESPLIEGA "Y"
          CALL  DISPL_
          MOV   R3,#087H      ;CARGA EN R3 LA POSICION
          MOV   R4,#052H      ;DESPLIEGA "R"
          CALL  DISPL_
          MOV   R3,#0C0H      ;CARGA EN R3 LA POSICION
          MOV   R4,#045H      ;DESPLIEGA "E"
          CALL  DISPL_
          INC   R3            ;INCREMENTA R3
          MOV   R4,#053H      ;DESPLIEGA "S"
          CALL  DISPL_
          INC   R3            ;INCREMENTA R3
          MOV   R4,#050H      ;DESPLIEGA "P"
          CALL  DISPL_
          INC   R3            ;INCREMENTA R3
          MOV   R4,#055H      ;DESPLIEGA "U"
          CALL  DISPL_
  
```

```
INC      R3           ;INCREMENTA R3
MOV      R4,#045H    ;DESPLIEGA "E"
CALL     DISPL_
INC      R3           ;INCREMENTA R3
MOV      R4,#053H    ;DESPLIEGA "S"
CALL     DISPL_
INC      R3           ;INCREMENTA R3
MOV      R4,#054H    ;DESPLIEGA "T"
CALL     DISPL_
INC      R3           ;INCREMENTA R3
MOV      R4,#041H    ;DESPLIEGA "A"
CALL     DISPL_
RET
```

;SUBROUTINA MENS_9
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR UN MENSAJE INDICANDO
;QUE SE ENCUENTRA EL OPERADOR EN MODO DE ACCESO.

```
MENS_9: CALL   BORRA           ;BORRA EL DISPLAY
        MOV    R3,#080H       ;CARGA EN R3 LA POSICION
        MOV    R4,#041H       ;DESPLIEGA "A"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        MOV    R4,#043H       ;DESPLIEGA "C"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        MOV    R4,#043H       ;DESPLIEGA "C"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        MOV    R4,#045H       ;DESPLIEGA "E"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        MOV    R4,#053H       ;DESPLIEGA "S"
        CALL   DISPL_
        INC    R3             ;INCREMENTA R3
        MOV    R4,#04FH       ;DESPLIEGA "O"
        CALL   DISPL_
        RET
```

```

;SUBROUTINA MENS_A
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR UN MENSAJE EN EL
;DISPLAY DE ACCESO NEGADO.

```

```

MENS_A: CALL    BORRA           ;BORRA EL DISPLAY
        MOV     R3,#081H       ;CARGA EN R3 LA POSICION
        MOV     R4,#041H       ;DESPLIEGA "A"
        CALL    DISPL_
        INC     R3
        MOV     R4,#043H       ;DESPLIEGA "C"
        CALL    DISPL_
        INC     R3
        MOV     R4,#043H       ;DESPLIEGA "C"
        CALL    DISPL_
        INC     R3
        MOV     R4,#045H       ;DESPLIEGA "E"
        CALL    DISPL_
        INC     R3
        MOV     R4,#053H       ;DESPLIEGA "S"
        CALL    DISPL_
        INC     R3
        MOV     R4,#04FH       ;DESPLIEGA "O"
        CALL    DISPL_
        MOV     R3,#0C1H       ;CARGA EN R3 LA POSICION
        MOV     R4,#04EH       ;DESPLIEGA "N"
        CALL    DISPL_
        INC     R3
        MOV     R4,#045H       ;DESPLIEGA "E"
        CALL    DISPL_
        INC     R3
        MOV     R4,#047H       ;DESPLIEGA "G"
        CALL    DISPL_
        INC     R3
        MOV     R4,#041H       ;DESPLIEGA "A"
        CALL    DISPL_
        INC     R3
        MOV     R4,#044H       ;DESPLIEGA "D"

```

```
CALL  DISPL_  
INC   R3  
MOV   R4,#04FH ;DESPLIEGA "0"  
CALL  DISPL_  
RET
```

```

;SUBROUTINA MENS_B
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR UN MENSAJE DE ACCESO
;PERMITIDO.

```

```

MENS_B: CALL BORRA ;BORRA EL DISPLAY
MOV R3,#080H ;CARGA EN R3 LA POSICION
MOV R4,#041H ;DESPLIEGA "A"
CALL DISPL_
INC R3
MOV R4,#043H ;DESPLIEGA "C"
CALL DISPL_
INC R3
MOV R4,#043H ;DESPLIEGA "C"
CALL DISPL_
INC R3
MOV R4,#045H ;DESPLIEGA "E"
CALL DISPL_
INC R3
MOV R4,#053H ;DESPLIEGA "S"
CALL DISPL_
INC R3
MOV R4,#04FH ;DESPLIEGA "O"
CALL DISPL_
MOV R3,#087H ;CARGA EN R3 LA POSICION
MOV R4,#050H ;DESPLIEGA "P"
CALL DISPL_
MOV R3,#0C0H ;CARGA EN R3 LA POSICION
MOV R4,#045H ;DESPLIEGA "E"
CALL DISPL_
INC R3
MOV R4,#052H ;DESPLIEGA "R"
CALL DISPL_
INC R3
MOV R4,#04DH ;DESPLIEGA "H"
CALL DISPL_
INC R3
MOV R4,#049H ;DESPLIEGA "I"

```

```
CALL  DISPL_
INC   R3
MOV   R4,#054H   ;DESPLIEGA "T"
CALL  DISPL_
INC   R3
MOV   R4,#049H   ;DESPLIEGA "I"
CALL  DISPL_
INC   R3
MOV   R4,#044H   ;DESPLIEGA "D"
CALL  DISPL_
INC   R3
MOV   R4,#04FH   ;DESPLIEGA "O"
CALL  DISPL_
RET
```

```

;SUBROUTINA MENS_C
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR UN MENSAJE INDICANDO
;QUE EL SISTEMA SE ENCUENTRA EN MODO DE TRANSFERENCIA.

```

```

MENS_C: CALL    BORRA           ;BORRA EL DISPLAY
        MOV     R3,#081H       ;CARGA EN R3 LA POSICION
        MOV     R4,#054H       ;DESPLIEGA "T"
        CALL    DISPL_
        INC     R3             ;INCREMENTA R3
        MOV     R4,#052H       ;DESPLIEGA "R"
        CALL    DISPL_
        INC     R3             ;INCREMENTA R3
        MOV     R4,#041H       ;DESPLIEGA "A"
        CALL    DISPL_
        INC     R3             ;INCREMENTA R3
        MOV     R4,#04EH       ;DESPLIEGA "H"
        CALL    DISPL_
        INC     R3             ;INCREMENTA R3
        MOV     R4,#053H       ;DESPLIEGA "S"
        CALL    DISPL_
        INC     R3             ;INCREMENTA R3
        MOV     R4,#046H       ;DESPLIEGA "F"
        CALL    DISPL_
        INC     R3             ;INCREMENTA R3
        MOV     R4,#045H       ;DESPLIEGA "E"
        CALL    DISPL_
        MOV     R3,#0COH       ;CARGA EN R3 LA POSICION
        MOV     R4,#052H       ;DESPLIEGA "R"
        CALL    DISPL_
        INC     R3             ;INCREMENTA R3
        MOV     R4,#045H       ;DESPLIEGA "E"
        CALL    DISPL_
        INC     R3             ;INCREMENTA R3
        MOV     R4,#04EH       ;DESPLIEGA "H"
        CALL    DISPL_
        INC     R3             ;INCREMENTA R3

```

```
MOV    R4,#043H    ;DESPLIEGA "C"
CALL   DISPL_
INC    R3          ;INCREMENTA R3
MOV    R4,#040H    ;DESPLIEGA "I"
CALL   DISPL_
INC    R3          ;INCREMENTA R3
MOV    R4,#041H    ;DESPLIEGA "A"
CALL   DISPL_
MOV    R3,#0C0H
CALL   DISPL_
RET
```

;SUBROUTINA MENS_D
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR UN MENSAJE INDICANDO
;QUE LA TRANSFERENCIA DE DATOS HA TERMINADO.

```
MENS_D: CALL BORRA           ;BORRA EL DISPLAY
MOV R3,#080H ;CARGA EN R3 LA POSICION
MOV R4,#046H ;DESPLIEGA "F"
CALL DISPL_
INC R3       ;INCREMENTA R3
MOV R4,#049H ;DESPLIEGA "I"
CALL DISPL_
INC R3       ;INCREMENTA R3
MOV R4,#04EH ;DESPLIEGA "N"
CALL DISPL_
MOV R3,#094H ;CARGA EN R3 LA POSICION
MOV R4,#07FH ;DESPLIEGA "<-"
CALL DISPL_
RET
```

```

;SUBROUTINA MENS_E
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR UN MENSAJE INDICANDO
;QUE EL SISTEMA SE ENCUENTRA EN MODO DE REPROGRAMACION.

```

```

MENS_E: CALL   BORRA           ;BORRA EL DISPLAY
        MOV    R3,#081H       ;CARGA EN R3 LA POSICION
        MOV    R4,#052H       ;DESPLIEGA "R"
        CALL   DISPL_
        INC    R3
        MOV    R4,#045H       ;DESPLIEGA "E"
        CALL   DISPL_
        INC    R3
        MOV    R4,#050H       ;DESPLIEGA "P"
        CALL   DISPL_
        INC    R3
        MOV    R4,#052H       ;DESPLIEGA "R"
        CALL   DISPL_
        INC    R3
        MOV    R4,#04FH       ;DESPLIEGA "O"
        CALL   DISPL_
        INC    R3
        MOV    R4,#047H       ;DESPLIEGA "G"
        CALL   DISPL_
        INC    R3
        MOV    R4,#052H       ;DESPLIEGA "R"
        CALL   DISPL_
        MOV    R3,#0COH       ;CARGA EN R3 LA POSICION
        MOV    R4,#041H       ;DESPLIEGA "A"
        CALL   DISPL_
        INC    R3
        MOV    R4,#04DH       ;DESPLIEGA "M"
        CALL   DISPL_
        INC    R3
        MOV    R4,#041H       ;DESPLIEGA "A"
        CALL   DISPL_

```

```
INC R3
MOV R4,#043H ;DESPLIEGA "C"
CALL DISPL_
INC R3
MOV R4,#049H ;DESPLIEGA "I"
CALL DISPL_
INC R3
MOV R4,#04FH ;DESPLIEGA "O"
CALL DISPL_
INC R3
MOV R4,#04EH ;DESPLIEGA "N"
CALL DISPL_
MOV R3,#0CBH
CALL DISPL_
RET
```

```
;SUBROUTINA MENS_F
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR EL MENU DEL MODO DE
;REPROGRAMACION .
```

```
MENS_F: CALL    BORRA           ;BORRA EL DISPLAY
        MOV     R3,#080H       ;CARGA EN R3 LA POSICION
        MOV     R4,#034H       ;DESPLIEGA "4"
        CALL    DISPL_
        INC     R3
        MOV     R4,#03DH       ;DESPLIEGA "="
        CALL    DISPL_
        INC     R3
        MOV     R4,#049H       ;DESPLIEGA "I"
        CALL    DISPL_
        MOV     R3,#0C0H       ;CARGA EN R3 LA POSICION
        MOV     R4,#035H       ;DESPLIEGA "5"
        CALL    DISPL_
        INC     R3
        MOV     R4,#03DH       ;DESPLIEGA "="
        CALL    DISPL_
        INC     R3
        MOV     R4,#052H       ;DESPLIEGA "R"
        CALL    DISPL_
        MOV     R3,#0C8H       ;CARGA EN R3 LA POSICION
        CALL    DISPL_
        RET
```

;SUBROUTINA MENS_G
 ;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR UN MENSAJE INDICANDO
 ;QUE EL EQUIPO UAD-1-1IE HA INICIADO LA ADQUISICION DE DATOS.

```

MENS_G: CALL  BORRA           .BORRA EL DISPLAY
          MOV   R3,#080H      :CARGA EN R3 LA POSICION
          MOV   R4,#040H      :DESPLIEGA "I"
          CALL  DISPL_
          INC   R3
          MOV   R4,#04EH      :DESPLIEGA "N"
          CALL  DISPL_
          INC   R3
          MOV   R4,#049H      :DESPLIEGA "I"
          CALL  DISPL_
          INC   R3
          MOV   R4,#043H      :DESPLIEGA "C"
          CALL  DISPL_
          INC   R3
          MOV   R4,#040H      :DESPLIEGA "I"
          CALL  DISPL_
          INC   R3
          MOV   R4,#04FH      :DESPLIEGA "O"
          CALL  DISPL_
          MOV   R3,#0C1H      :CARGA EN R3 LA POSICION
          MOV   R4,#045H      :DESPLIEGA "E"
          CALL  DISPL_
          INC   R3
          MOV   R4,#04CH      :DESPLIEGA "L"
          CALL  DISPL_
          MOV   R3,#0C5H      :CARGA EN R3 LA POSICION
          MOV   R4,#041H      :DESPLIEGA "A"
          CALL  DISPL_
          INC   R3
          MOV   R4,#044H      :DESPLIEGA "D"
          CALL  DISPL_
  
```

```
INC R3
MOV R4,#044H ;DESPLIEGA "D"
CALL DISPL_
RET
```

;SUBROUTINA MENS_H
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR LAS DIFERENTES LETRAS
;PIDIENDO DATOS PARA REPROGRAMAR EL RELOJ DEL UAD-1-1IE.

MENS_H:	CALL	BORRA	;BORRA EL DISPLAY
	MOV	R3,#080H	;CARGA EN R3 LA POSICION
	SETB	RS0	;SELECCIONA EL BANCO 1
	MOV	A,R6	;CARGA EL CODIGO DE LA LETRA
	CLR	RS0	;SELECCIONA EL BANCO 0
	MOV	R4,A	;CARGA EL CODIGO EN R4
	CALL	DISPL_	
	INC	R3	
	MOV	R4,#03DH	;DESPLIEGA "="
	CALL	DISPL_	
	RET		

;SUBROUTINA MENS_I
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR UN MENSAJE INDICANDO
;QUE EL EQUIPO SE ENCUENTRA EN MODO DE PC.

```
MENS_I: CALL    BORRA           ;BORRA EL DISPLAY
        MOV     R3,#080H       ;CARGA EN R3 LA POSICION
        MOV     R4,#04DH       ;DESPLIEGA "M"
        CALL    DISPL_
        INC     R3
        MOV     R4,#04FH       ;DESPLIEGA "O"
        CALL    DISPL_
        INC     R3
        MOV     R4,#044H       ;DESPLIEGA "D"
        CALL    DISPL_
        INC     R3
        MOV     R4,#04FH       ;DESPLIEGA "O"
        CALL    DISPL_
        MOV     R3,#0C0H       ;CARGA EN R3 LA POSICION
        MOV     R4,#050H       ;DESPLIEGA "P"
        CALL    DISPL_
        INC     R3
        MOV     R4,#043H       ;DESPLIEGA "C"
        CALL    DISPL_
        MOV     R3,#0C8H
        CALL    DISPL_
        RET
```

```

;SUBROUTINA MENS_J
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR UN MENSAJE DE "HOLA"
;RECORRIENDO TODO EL DISPLAY.

```

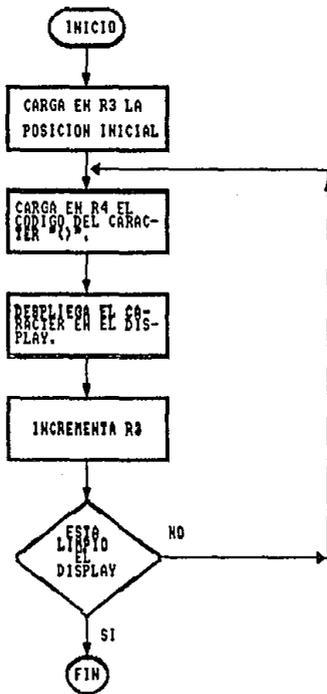
```

MENS_J: CALL BORRA           ;BORRA EL DISPLAY
        SETB RS1           ;SELECCIONA EL BANCO 2
        MOV R3,#000H      ;INICIALIZA R3''
        CLR RS1           ;SELECCIONA EL BANCO 0
        MOV R3,#080H      ;CARGA EN R3 LA POSICION
PONLO_: MOV R4,#048H      ;DESPLIEGA "H"
        CALL DISPL_
        INC R3             ;INCREMENTA R3
        MOV R4,#04FH      ;DESPLIEGA "O"
        CALL DISPL_
        INC R3             ;INCREMENTA R3
        MOV R4,#04CH      ;DESPLIEGA "L"
        CALL DISPL_
        INC R3             ;INCREMENTA R3
        MOV R4,#041H      ;DESPLIEGA "A"
        CALL DISPL_
        MOV R7,#00BH      ;EJECUTA UN RETARDO DE 2 SEG.
        CALL DELAY
        SETB RS1           ;SELECCIONA EL BANCO 2
        INC R3             ;INCREMENTA R3''
        CJNE R3,#001H,PONLO1
        CLR RS1           ;SELECCIONA EL BANCO 0
        CALL BORRA        ;BORRA EL DISPLAY
        MOV R3,#084H      ;CARGA EN R3 LA POSICION
        JMP PONLO_
PONLO1: CJNE R3,#002H,PONLO2
        CLR RS1           ;SELECCIONA EL BANCO 0
        CALL BORRA        ;BORRA EL DISPLAY
        MOV R3,#0C0H      ;CARGA EN R3 LA POSICION
        JMP PONLO_
PONLO2: CJNE R3,#003H,PONLO3
        CLR RS1           ;SELECCIONA EL BANCO 0

```

```
CALL      BORRA           ;BORRA EL DISPLAY
MOV       R3,#0C4H       ;CARGA EN R3 LA POSICION
JMP       PONLO_
PONLO3: CLR      RS1      ;SELECCIONA EL BANCO 0
RET
END
```

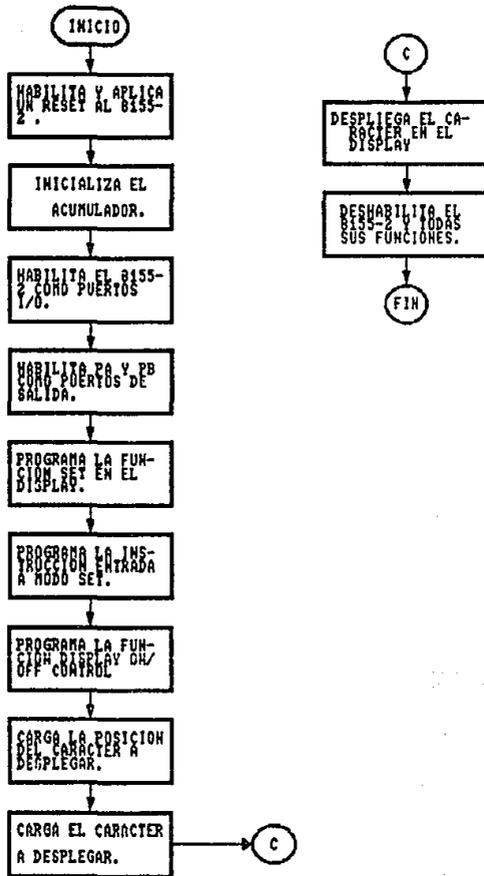
SUBRUTINA BORRA .



**:SUBROUTINA BORRA.
:ESTA SUBROUTINA SE ENCARGA DE LIMPIAR EL DISPLAY,BORRANDO
:LOS MENSAJES QUE SE ENCONTRABAN EN EL.**

```
BORRA : MOV      R3,#080H      ;CARGA EN R3 LA POS.INICIAL
SIGUE1: MOV      R4,#020H      ;CARGA EN R4 EL COD. DE"()"
        CALL     DISPL_
        INC      R3            ;INCREMENTA R3
        CJNE    R3,#08BH,SIGUE1;BRINCA SI R3 NO ES 8BH
        MOV      R3,#0C0H      ;CARGA EN R3 LA POSI. INICIAL
SIGUE2: MOV      R4,#020H      ;CARGA EN R4 EL COD. DE"()"
        CALL     DISPL_
        INC      R3            ;INCREMENTA R3
        CJNE    R3,#0CBH,SIGUE2;BRINCA SI R3 NO ES 0CBH
        RET
```

SUBROUTINA DISPL_



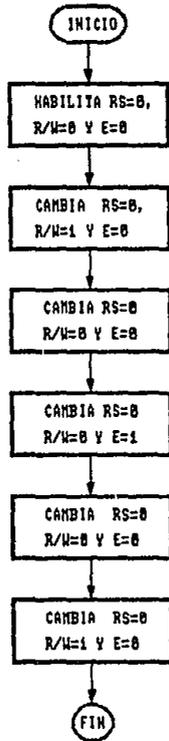
;SUBROUTINA DISPL_
 ;ESTA SUBROUTINA SE ENCARGA DE PROGRAMAR TODAS LAS FUNCIONES
 ;QUE TIENE QUE REALIZAR EL DISPLAY PARA PODER DESPLEGAR EN
 ;SU PANTALLA UN CARACTER.

```

DISPL_: CLR      P1.7      ;HABILITA EL CE DEL 8155-2
        SETB     P1.6      ;APLICA UN RESET AL 8155-2
        NOP
        NOP
        CLR      A         ;INICIALIZA ACUMULADOR
        CLR      P1.6      ;BORRA EL RESET
        SETB     P2.7      ;SELEC. LOS PTOS. DEL 8155-2
        MOV      A,#003H
        MOV      R1,#000H
        MOVX     @R1,A     ;HABILITA PA Y PB COMO SALIDA
        MOV      A,#002H   ;CARGA EN A EL CODIGO A SACAR
        MOV      R1,#002H  ;SELECCIONA EL PUERTO PB
        MOVX     @R1,A     ;INICIALIZA RS=0,R/W=1,E=0
        MOV      A,#038H   ;COD. DE FUNCION SET
        MOV      R1,#001H  ;SELECCIONA EL PUERTO PA
        MOVX     @R1,A     ;FUNCION SET
        CALL     WRITE     ;LLAMA A LA SUBROUTINA WRITE
        MOV      A,#006H   ;COD. DE ENTRADA A MODO SET
        MOV      R1,#001H  ;SELECCIONA EL PUERTO PA
        MOVX     @R1,A     ;ENTRADA A MODO SET
        CALL     WRITE     ;LLAMA A LA SUBROUTINA WRITE
        MOV      A,#00EH   ;COD. DISPLAY ON/OFF CONTROL
        MOV      R1,#001H  ;SELECCIONA EL PUERTO PA
        MOVX     @R1,A     ;DISPLAY ON/OFF CONTROL
        CALL     WRITE     ;LLAMA A WRITE
        MOV      A,R3      ;CARGA EN A LA POSICION
        MOV      R1,#001H  ;SELECCIONA EL PUERTO PA
        MOVX     @R1,A     ;PROGRAMA LA POS. DEL DISP.
        CALL     WRITE     ;LLAMA A WRITE
        MOV      A,#006H   ;CARGA EN A EL #006
        MOV      R1,#002H  ;SELECCIONA EL PUERTO PB
  
```

MOVX	@R1,A	;INICIALIZA RS=1,R/W=1,E=0
MOV	A,R4	;CARGA EL CODIGO DEL CARACTER
MOV	R1,#001H	;SELECCIONA EL PUERTO PA
MOVX	@R1,A	
CALL	WRITE1	;LLAMA A LA SUBROUTINA WRITE1
CLR	P2.7	;DESHABILITA LA SELEC. PTOS.
SETB	P1.7	;DESHABILITA EL 8155-2
RET		

SUBROUTINA WRITE .



```

:SUBROUTINA WRITE
:ESTA SUBROUTINA REALIZA LAS SENALES NECESARIAS PARA PODER
:PROGRAMAR COMANDOS EN EL DISPLAY.

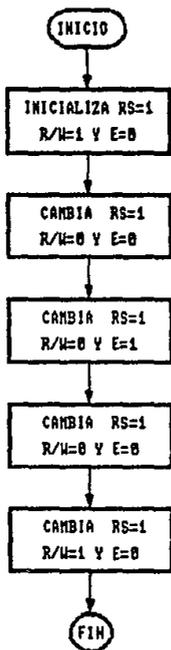
```

```

WRITE : MOV      A,#000H
        MOV      R1,#002H      ;SELECCIONA EL PUERTO FB
        MOVX    @R1,A          ;RS=0,R/W=0,E=0
        MOV      A,#002H
        MOV      R1,#002H      ;SELECCIONA EL PUERTO FB
        MOVX    @R1,A          ;RS=0,R/W=1,E=0
        MOV      A,#000H
        MOVX    @R1,A          ;RS=0,R/V=0,E=0
        MOV      A,#001H
        NOP
        NOP
        NOP
        MOVX    @R1,A          ;RS=0,R/V=0,E=1
        MOV      A,#000H
        NOP
        NOP
        NOP
        MOVX    @R1,A          ;RS=0,R/W=0,E=0
        MOV      A,#002H
        NOP
        NOP
        NOP
        MOVX    @R1,A          ;RS=0,R/W=1,E=0
        RET

```

SUBROUTINA WRITE 1



```

;SUBROUTINA WRITE1
;ESTA SUBROUTINA GENERA LAS SENALES NECESARIAS PARA PODER
;ESCRIBIR UN CARACTER EN EL DISPLAY.

```

```

WRITE1: MOV     A,#006H
        MOV     R1,#002H      ;SELECCIONA EL PUERTO PB
        MOVX    @R1,A        ;RS=1,R/W=1,E=0
        MOV     A,#004H
        MOV     R1,#002H      ;SELECCIONA EL PUERTO PB
        MOVX    @R1,A        ;RS=1,R/W=0,E=0
        MOV     A,#005H
        MOV     R1,#002H      ;SELECCIONA EL PUERTO PB
        NOP
        NOP
        NOP
        MOVX    @R1,A        ;RS=1,R/W=0,E=1
        MOV     A,#004H
        MOV     R1,#002H      ;SELECCIONA EL PUERTO PB
        NOP
        NOP
        NOP
        MOVX    @R1,A        ;RS=1,R/W=0,E=0
        MOV     A,#006H
        MOV     R1,#002H      ;SELECCIONA EL PUERTO PB
        MOVX    @R1,A        ;RS=1,R/W=1,E=0
        RET

```

SECUENCIA MOTO.



;SUBROUTINA MRTO
 ;ESTA SUBROUTINA SE ENCARGA DE GUARDAR EL CODIGO DE LA TECLA
 ;OPRIMIDA ASI COMO DE DESPLEGARLA EN EL DISPLAY.

```

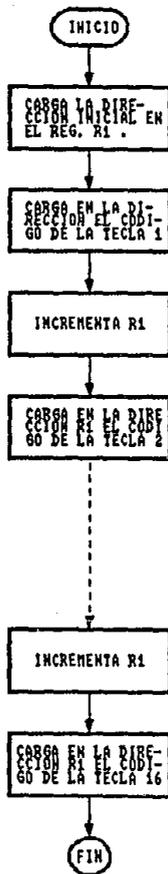
MRTO : CALL   DRAM
      CLR     P1.4      ;HABILITA EL 8155-1
      SETB    P1.6      ;APLICA UN RESET AL 8155-1
      NOP
      NOP
      CLR     A          ;INICIALIZA EL ACUMULADOR
      CLR     P1.6      ;BORRA EL RESET DEL 8155-1
      SETB    P1.5      ;SELEC. LOS PTOS. DEL 8155-1
      MOV     R1,#000H  ;SELECCIONA PROG. DE PUERTOS
      MOV     A,#002H
      MOVX    @R1,A     ;PROGRAMA PC ENT., PB SALIDA
      MOV     R1,#002H  ;SELECCIONA EL PUERTO PB
      MOV     A,#000H
      MOVX    @R1,A     ;HABILITA LAS SAL. DEL 54C922
      MOV     R1,#003H
      MOVX    A,@R1     ;LEE EL CONTENIDO DE PC
      ANL     A,#00FH   ;SALVA EL CODIGO DE LA TECLA
      MOV     R2,A      ;GUARDA EN R2 LA TECLA
      ANL     A,#0FFH
      ORL     A,#0FOH
      ANL     A,#02FH   ;20 MAS EL CODIGO DE LA TECLA
      MOV     R1,A      ;CARGA EL RESULTADO EN R1
      MOV     A,@R1     ;CARGA EL CODIGO EN EL ACC.
      MOV     R4,A
      SETB    RSO       ;SELECCIONA EL BANCO 1
      MOV     A,R5      ;CARGA LA POSICION EN A
      CLR     RSO       ;SELECCIONA EL BANCO 0
      MOV     R3,A      ;CARGA EN R3 LA POSICION
      CLR     P1.5      ;BORRA LA SELECCION DE PTOS.
      SETB    P1.4      ;DESHABILITA EL 8155-1
      CALL    DISPL_
      MOV     R7,#009H  ;CARGA DURACION DEL RETARDO
  
```

CALL
RET

DELAY

;EJECUTA UN RETARDO

SIEMPRE DAM.



```

:SUBROUTINA DRAM
;ESTA SUBROUTINA CARGA LOS CODIGOS DE LAS TECLAS A LA MEMORIA
;RAM DEL MICRO.

```

```

DRAM : MOV      R1,#020H      ;INICIALIZA R1
      MOV      @R1,#030H    ;CARGA EN LA DIR. 20 TECLA 1
      INC      R1
      MOV      @R1,#031H    ;CARGA EN LA DIR. 21 TECLA 2
      INC      R1
      MOV      @R1,#032H    ;CARGA EN LA DIR. 22 TECLA 3
      INC      R1
      MOV      @R1,#033H    ;CARGA EN LA DIR. 23 TECLA 4
      INC      R1
      MOV      @R1,#034H    ;CARGA EN LA DIR. 24 TECLA 5
      INC      R1
      MOV      @R1,#035H    ;CARGA EN LA DIR. 25 TECLA 6
      INC      R1
      MOV      @R1,#036H    ;CARGA EN LA DIR. 26 TECLA 7
      INC      R1
      MOV      @R1,#037H    ;CARGA EN LA DIR. 27 TECLA 8
      INC      R1
      MOV      @R1,#038H    ;CARGA EN LA DIR. 28 TECLA 9
      INC      R1
      MOV      @R1,#039H    ;CARGA EN LA DIR. 29 TECLA 10
      INC      R1
      MOV      @R1,#041H    ;CARGA EN LA DIR. 30 TECLA 11
      INC      R1
      MOV      @R1,#042H    ;CARGA EN LA DIR. 31 TECLA 12
      INC      R1
      MOV      @R1,#043H    ;CARGA EN LA DIR. 32 TECLA 13
      INC      R1
      MOV      @R1,#044H    ;CARGA EN LA DIR. 33 TECLA 14
      INC      R1
      MOV      @R1,#045H    ;CARGA EN LA DIR. 34 TECLA 15
      INC      R1

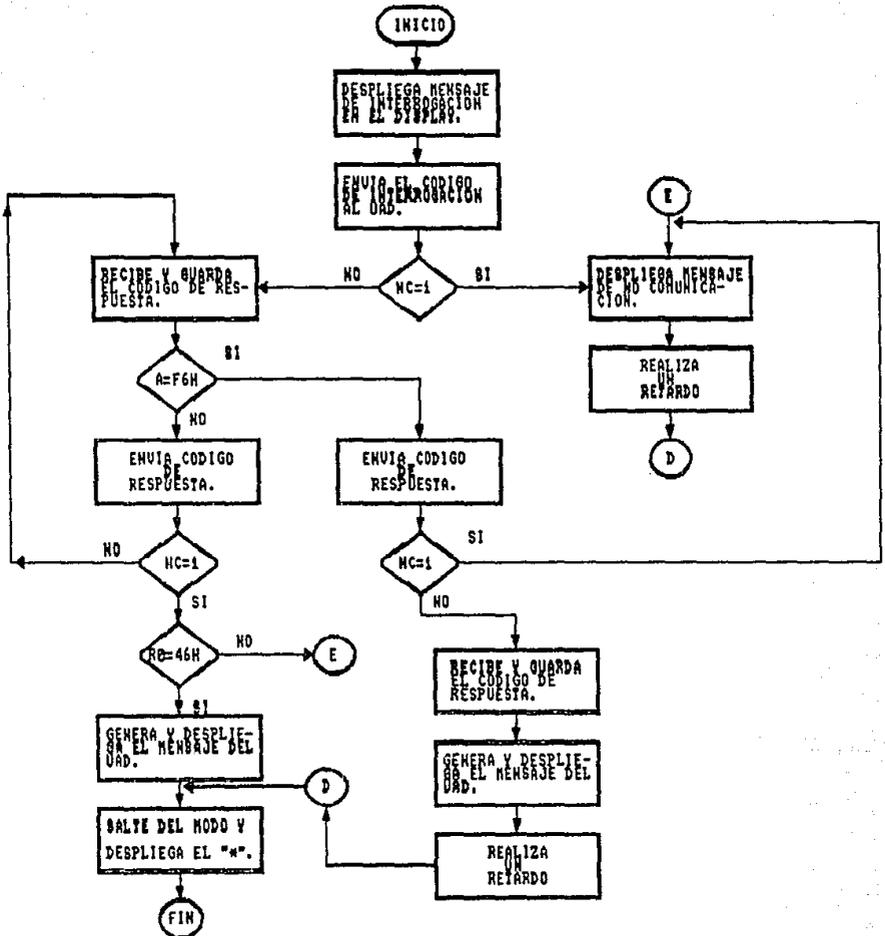
```

NOV
RET

OR1,#046H

;CARGA EN LA DIR. 35 TECLA 16

SUBRUTINA MODINT.



```

;SUBROUTINA MODINT
;ESTA SUBROUTINA REALIZA LA INTERROGACION A UN EQUIPO UAD-1
;-IIE OBTENIENDO DE EL LOS DATOS SIGUIENTES: ESTADO DEL
;RELOJ,FECHA DE INICIO DE CAPTURA DE DATOS ESTADO EN QUE
;SE ENCUENTRA EL EQUIPO Y NUMERO DE DIAS QUE LE RESTAN A
;LA MEMORIA DE ALMACENAMIENTO.

```

```

MODINT: MOV     R7,#005H      ;COLOCA MENS. DE INTERROGACION
        CALL    DISP
        MOV     A,#042H      ;CODIGO DE INTERROGACION
        CALL    COM_SP       ;TRANSMITELO AL UAD
        SETB   RSO           ;SELECCIONA EL BANCO 1
        MOV     A,R2         ;CARGA EN A EL REGISTRO R2'
        JBC    ACC.1,NOCOMO  ;BRINCA SI NC ESTA PUESTA
        CLR    RSO           ;SELECCIONA EL BANCO G
        MOV     RO,#030H     ;INICIALIZA RO
GUARDA: MOV     A,R5
        MOV     @RO,A        ;CARGA A EN LA DIR. RO
        INC    RO            ;INCREMENTA RO
        CJNE   A,#0F6H,CONTI1 ;BRINCA SI A NO ES OF6
        MOV     A,#043H      ;CARGA CODIGO DE CONTESTACION
        CALL    COM_SP       ;ENVIA CONTESTACION AL UAD
        SETB   RSO           ;SELECCINA EL BANCO 1
        MOV     A,R2         ;CARGA EN A EL REG. DE BAND.
        JBC    ACC.1,NOCOM1  ;BRINCA SI NC ESTA LEVANTADA
        CLR    RSO           ;SELECCIONA EL BANCO O
        MOV     A,R5
        MOV     @RO,A        ;GUARDA EL CODIGO DE LA FALLA
        CALL    SELEC_
        MOV     R7,#026H     ;CARGA LA DURACION DEL RETARDO
        CALL    DELAY        ;REALIZA UN RETARDO DE 5 SEG.
        JMP     FIN_O        ;BRINCA A FIN_O
CONTI1: MOV     A,#043H      ;CARGA CODIGO DE RESPUESTA
        CALL    COM_SP       ;RESPONDE AL UAD

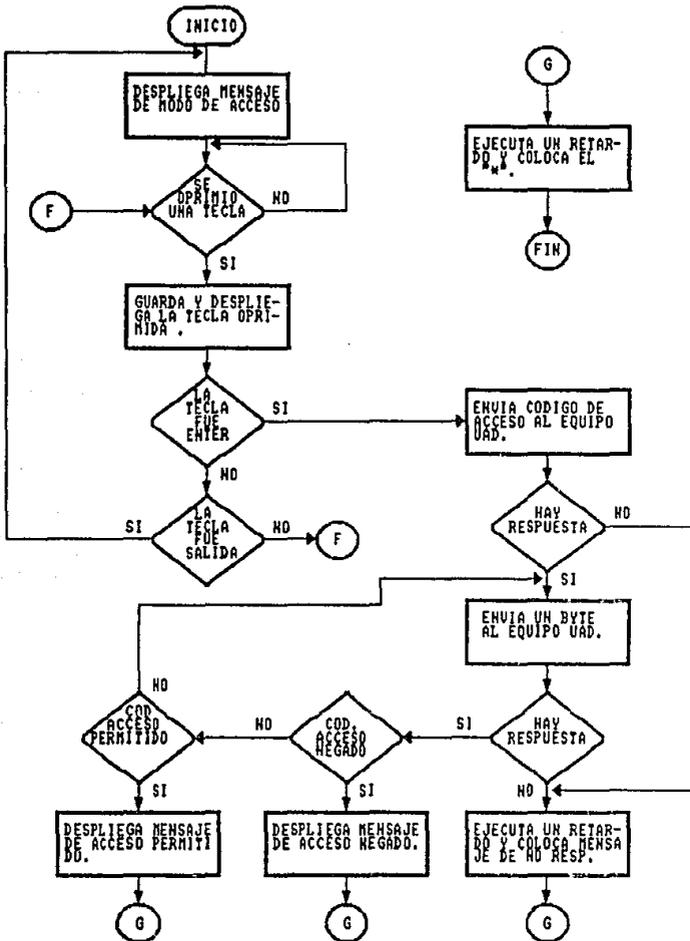
```

```

SETB   RSO           ;SELECCIONA EL BANCO 1
MOV    A,R2         ;CARGA EN A EL REG. DE BAND.
JBC   ACC.1,NOCOM1 ;BRINCA SI NC ESTA LEVANTADA
CLR   RSO           ;SELECCIONA EL BANCO 0
JMP   GUARDA       ;BRINCA A GUARDA
NOCOM0: MOV R2,A     ;BORRA LA BANDERA NC
CLR   RSO           ;SELECCIONA EL BANCO 0
NOCOM8: MOV R7,#00B ;CARGA EL COD. DE NO COMUNI.
CALL  DISP         ;COLOCA MENSAJE DE NO COMUNI.
MOV   R7,#026H    ;CARGA LA DURACION DEL RETARDO
CALL  DELAY       ;REALIZA UN RETARDO DE 5 SEG.
JMP   FIN_0
NOCOM1: MOV R2,A    ;BORRA LA BANDERA NC
CLR   RSO         ;SELECCIONA EL BANCO 0
MOV   A,RO       ;CARGA EN A LA ULTIMA DIREC.
MOV   B,#046H   ;CARGA EN B LA DIR. MINIMA
DIV   AB
JNZ   CONTI2
JMP   NOCOM8
CONTI2: CALL SELEC_ ;COLOCA EL MENSAJE DEL UAD
FIN_0 : MOV R7,#002H ;CARGA EL CODIGO DEL "*"
CALL  DISP      ;DESPLIEGALO EN EL DISPLAY
RET

```

SUBROUTINA MAD .



```

;SUBROUTINA MAD
;ESTA SUBROUTINA SE ENCARGA DE ACCESAR AL EQUIPO UAD-1-IIE,
;ENVIANDOLE UNA COMBINACION DE CUATRO TECLAS, DESPLEGANDO
;UN MENSAJE SI LA COMBINACION ES CORRECTA Y OTRO SI ES
;INCORRECTA.

```

```

MAD   : MOV     R7,#009H      ;DESPLIEGA MODO DE ACCESO
        CALL    DISP
        MOV     RO,#030H     ;INICIALIZA LA DIR. DE LA RAM
        SETB   RSO          ;SELECCIONA EL BANCO 1
        MOV     R5,#0COH    ;CARGA LA POSICION EN R5'
        CLR    RSO          ;SELECCIONA EL BANCO 0
OTRATE: JB     P3.2,$        ;ESPERA UNA TECLA
        CALL    MRT0        ;GUARDA Y DESPLIEGA LA TECLA
        MOV     A,R2        ;CARGA EN LA LA TECLA
        MOV     @RO,A       ;GUARDA.LA TECLA EN LA RAM
        CJNE   R2,#00FH,PIDTEO ;BRINCA SI LA TECLA NO FUE F
        MOV     A,#041H     ;CARGA EL CODIGO DE ACCESO
        CALL    COM_SP      ;ENVIA EL CODIGO AL UAD
        SETB   RSO          ;SELECCIONA EL BANCO 1
        MOV     A,R2        ;CARGA R2' EN A
        JBC   ACC.1,HOCOM2  ;BRINCA SI N.C. ESTA LEVANTADA
        CLR    RSO          ;SELECCIONA EL BANCO 0
        MOV     RO,#030H    ;INICIALIZA RO
ENVI2: MOV     A,@RO        ;CARGA LA TECLA EN A
        CALL    ACOND_      ;SUMA 30 AL DATO
        CALL    COM_SP      ;ENVIA LA TECLA AL UAD
        SETB   RSO          ;SELECCIONA EL BANCO 1
        MOV     A,R2        ;CARGA EN A EL REG. DE BAND.
        JBC   ACC.1,HOCOM2  ;BRINCA SI N.C. ESTA PUESTA
        CLR    RSO          ;SELECCIONA EL BANCO 0
        CJNE   R5,#0F7H,ENVIAO ;BRINCA SI R5 NO ES F7
        MOV     R7,#00AH    ;COLOCA MENS. DE ACCESO NEGADO
        CALL    DISP
        JMP     FIN_4       ;BRINCA A FIN_4

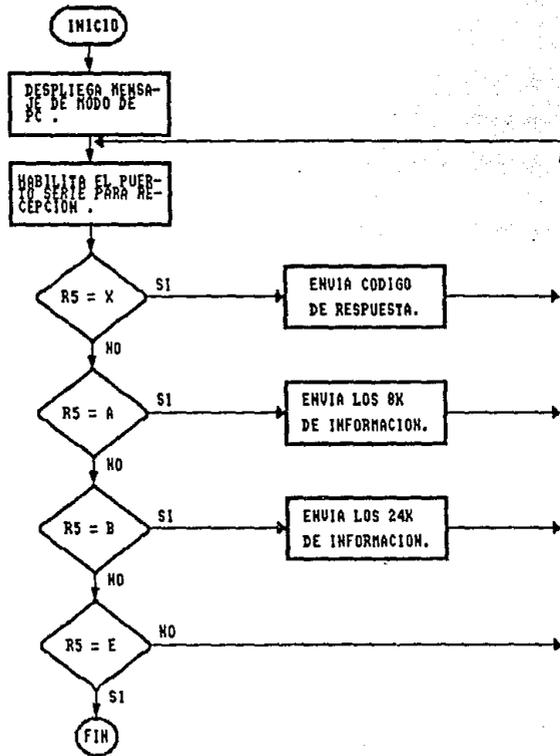
```

```

ENVIAO: CJNE R5,#0F8H,ENVIA1 ;BRINCA SI R5 NO ES F8
        MOV R7,#00BH ;COLOCA MENS. ACCESO PERMITIDO
        CALL DISP
        JMP FIN_4 ;BRINCA A FIN_4
ENVIA1: INC RO ;INCREMENTA RO
        JMP ENVIA2
PIDTEO: CJNE R2,#00EH,PIDTE1 ;BRINCA SI NO ES TECLA DE SAL.
        JMP MAD
PIDTE1: INC RO ;INCREMENTA RO
        SETB R50 ;SELECCIONA EL BANCO 1
        INC R5 ;INCREMENTA R5'
        CLR R50 ;SELECCIONA EL BANCO 0
        JMP OTRATE ;BRINCA A OTRATE
HOCOM2: MOV R2,A ;BORRA LA BANDERA H.C.
        CLR R50 ;SELECCIONA EL BANCO 0
        MOV R7,#008H ;COLOCA MENSAJE DE NO RESP.
        CALL DISP
FIN_4 : MOV R7,#026H
        CALL DELAY ;RETARDO DE 5 SEG.
        MOV R7,#002H ;DESPLIEGA "*"
        CALL DISP
        RET

```

SUBROUTINA MOTPC .



```

;SUBROUTINA MOTPC
;ESTA SUBROUTINA SE ENCARGA DE HACER UNA TRANSFERENCIA DE
;LOS DATOS OBTENIDOS DE UN UAD A UNA COMPUTADORA PERSONAL.

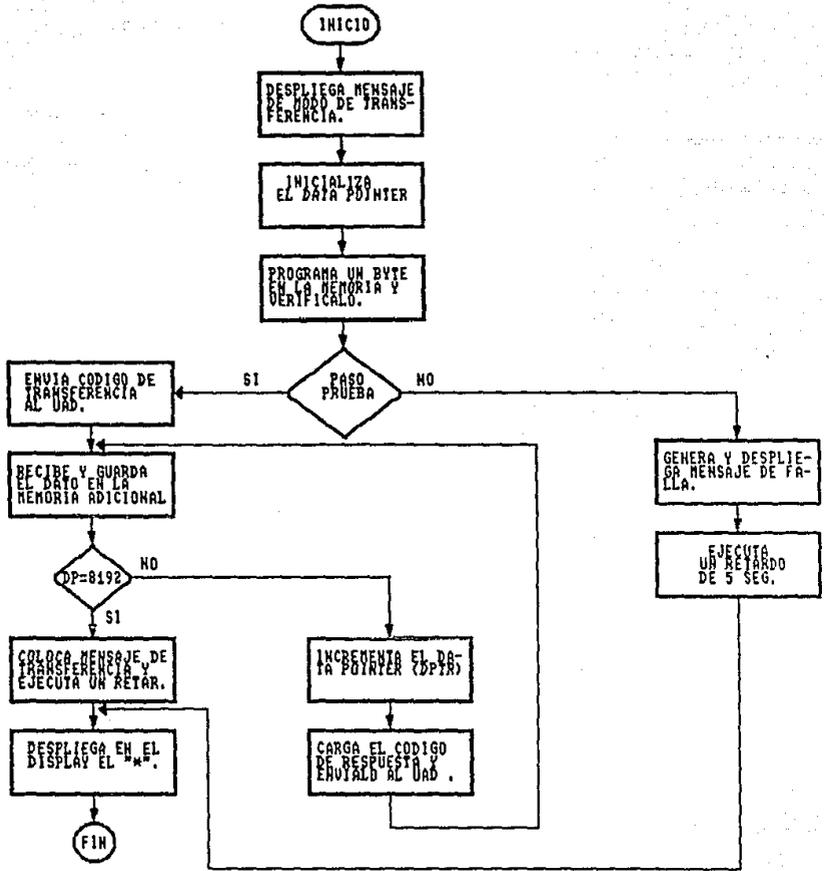
```

```

MOTPC : MOV     R7,#012H      ;COLOCA MENSAJE DE MODO DE PC.
        CALL    DISP
MASLET: CALL    IN_SP        ;HABILITA EL PS PARA RECEPCION
        CJNE   R5,#058H,DATOSO ;BRINCA SI NO ES LA LETRA "X"
        MOV    A,#04CH      ;CARGA EL COD. DE LA LETRA "L"
        CALL   OUT_SP       ;ENVIA EL CODIGO A LA PC.
        JNP    MASLET
DATOSO: CJNE   R5,#041H,DATOS1 ;BRINCA SI NO ES LA LETRA "A"
        MOV    R6,#00CH     ;CARGA EL COD. DE MEMORIA #1
        CALL   LEEME1
        JNP    MASLET
DATOS1: CJNE   R5,#042H,DATOS2 ;BRINCA SI NO ES LA LETRA "B"
        MOV    R6,#000H     ;CARGA EL COD. DE MEMORIA #1
        CALL   LEEME1
        INC    R6           ;LEE LA MEMORIA #2
        CALL   LEEME1
        INC    R6           ;LEE LA MEMORIA #3
        CALL   LEEME1
        JNP    MASLET
DATOS2: CJNE   R5,#045H,MASLET ;BRINCA SI NO ES LA LETRA "E"
        MOV    R7,#002H     ;DESPLIEGA EL "*"
        CALL   DISP
        RET

```

SUBROUTINA MODTRA .



;SUBROUTINA MODTRA
 ;ESTA SUBROUTINA SE ENCARGA DE ESTABLECER UNA COMUNICACION
 ;CON EL EQUIPO UAD-1-IIE CON EL FIN DE RECUPERAR LA INFOR-
 ;MACION QUE HA SIDO ALMACENADA POR DICHO EQUIPO.

```

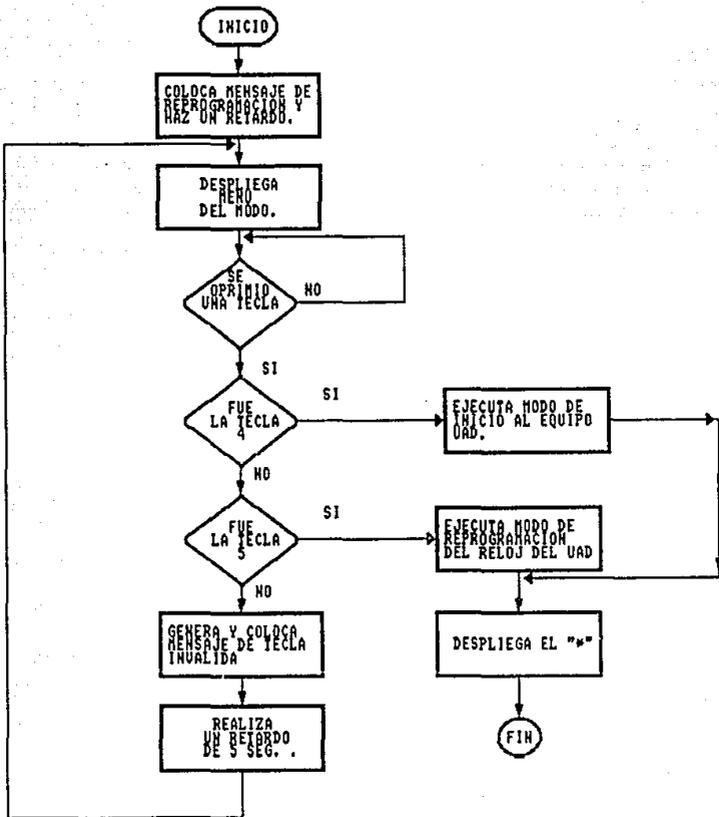
MODTRA: MOV     R7,#00CH      ;COLOCA MENS. MODO DE TRANSF.
        CALL   DISP
        SETB   RS1          ;SELECCIONA EL BANCO 2
        MOV    R1,#055H     ;CARGA EN R1'EL #55H
        CLR    RS1
        MOV    DPTR,#0E00H  ;INICIALIZA EL DATA POINTER
        CALL   ESCRIB       ;ESCRIBE EN LA MEMORIA
        SETB   RSO          ;SELECCIONA EL BANCO 1
        MOV    A,R2         ;CARGA EN A EL REG. DE BAND.
        JBC   ACC.2,MEMAL   ;BRINCA SI N.E. ESTA PUESTA
        CLR    RSO          ;SELECCIONA EL BANCO 0
        MOV    A,#043H     ;CARGA EL CODIGO DE TRANSF.
        CALL   COM_SP       ;ENVIA EL CODIGO AL UAD
        SETB   RSO          ;SELECCIONA EL BANCO 1
        MOV    A,R2         ;CARGA EN A EL REG. DE BAND.
        JBC   ACC.1,NOCOM3  ;BRINCA SI N.C. ESTA PUESTA
        CLR    RSO          ;SELECCIONA EL BANCO 0
        CJNE  R5,#0F2H,SI_ACC ;BRINCA SI R5 NO ES F2
        MOV    R7,#003H    ;DESPLIEGA NO ACCESO PREVIO
        CALL   DISP
        MOV    R7,#026H    ;EJECUTA UN RETARDO DE 6 SEG.
        CALL   DELAY
        JMP    FIN_5
SI_ACC: MOV    A,#055H     ;CARGA EN A EL COD. DE RESP.
        CALL   COM_SP
        SETB   RSO          ;SELECCIONA EL BANCO 1
        MOV    A,R2         ;CARGA EN A EL REG. DE BAND.
        JBC   ACC.1,NOCOM3  ;BRINCA SI N.C. ESTA PUESTA
        CLR    RSO          ;SELECCIONA EL BANCO 0
        MOV    A,R5         ;CARGA EN A EL DATO DEL UAD
        SETB   RS1          ;SELECCIONA EL BANCO 2
  
```

```

MOV      R1,A           ;SALVA EL DATO EN R1''
CLR      RS1           ;SELECCIONA EL BANCO 0
CALL     ESCRIB        ;ESCRIBE EL DATO EN LA MEMORIA
SETB     RSO           ;SELECCIONA EL BANCO 1
MOV      A,R2          ;CARGA EN A EL REG. DE BAND.
JBC      ACC.2,MEMAL   ;BRINCA SI N.E. ESTA PUESTA
CLR      RSO           ;SELECCIONA EL BANCO 0
INC      DPTR          ;INCREMENTA EL DATA POINTER
MOV      A,DPL         ;CARGA EN A EL BYTE BAJO DEL DP.
JNZ      SI_ACC        ;BRINCA SI A NO ES CERO
MOV      A,DPH         ;CARGA EN A EL BYTE ALTO DEL DP.
JNZ      SI_ACC        ;BRINCA SI A NO ES CERO
MOV      R7,#00DH     ;DESPLIEGA FIN DE TRANSF.
CALL     DISP
MOV      R7,#026H     ;RETARDO DE 5 SEG.
CALL     DELAY
JMP      FIN_5
MEMAL : MOV      R2,A           ;BORRA LA BANDERA N.E.
MOV      R4,#031H     ;CARGA EL CODIGO DE FALLA 1
CLR      RSO           ;SELECCIONA EL BANCO 0
MOV      R7,#001H     ;DESPLIEGA FALLA 1
CALL     DISP
MOV      R7,#026H     ;RETARDO DE 5 SEG.
CALL     DELAY
JMP      FIN_5
NOCOM3 : MOV      R2,A           ;BORRA LA BANDERA N.C.
CLR      RSO           ;SELECCIONA EL BANCO 0
MOV      R7,#008H     ;DESPLIEGA NO HAY RESPUESTA
CALL     DISP
MOV      R7,#026H     ;EJECUTA UN RETARDO
CALL     DELAY
FIN_5 : MOV      R7,#002H     ;DESPLIEGA "-"
CALL     DISP
RET

```

SUBROUTINA MODREP .



```

;SUBROUTINA MODREP
;ESTA SUBROUTINA TIENE COMO FUNCION REPROGRAMAR AL EQUIPO
;UAD,EXISTEN DOS FORMAS, LA DE INICIO DE ADQUISICION DE
;DATOS Y LA DE REPROGRAMACION DEL RELOJ DEL UAD-1-IIE.

```

```

MODREP: MOV     R7,#00EH      ;COLOCA MODO DE REPROGRA.
        CALL    DISP
        MOV     R7,#026H     ;EJECUTA UN RETARDO
        CALL    DELAY
MENU1 : MOV     R7,#00FH     ;COLOCA EL MENU DEL MODO
        CALL    DISP
        JB      P3.2,$       ;ESPERA UNA TECLA
        CALL    BORRA        ;BORRA EL DISPLAY
        SETB    RSO          ;SELECCIONA EL BANCO 1
        MOV     R5,#080H
        CLR     RSO          ;SELECCIONA EL BANCO 0
        CALL    MRTD         ;SALVA Y COLOCA LA TECLA
        CJNE   R2,#004H,MENU2 ;BRINCA SI NO FUE LA TECLA 4
        CALL    INIDAT
        JMP     FIN_8
MENU2 : CJNE   R2,#005H,MENU3 ;BRINCA SI NO FUE LA TECLA 5
        CALL    RELOJ_       ;REPROGRAMA EL RELOJ DEL UAD
        JMP     FIN_8
MENU3 : MOV     R7,#004H     ;COLOCA MENS. TECLA INVALIDA
        CALL    DISP
        MOV     R7,#026H     ;EJECUTA UN RETARDO
        CALL    DELAY
        JMP     MENU1
FIN_8  : MOV     R7,#002H     ;DESPLIEGA EL "="
        CALL    DISP
        RET

```

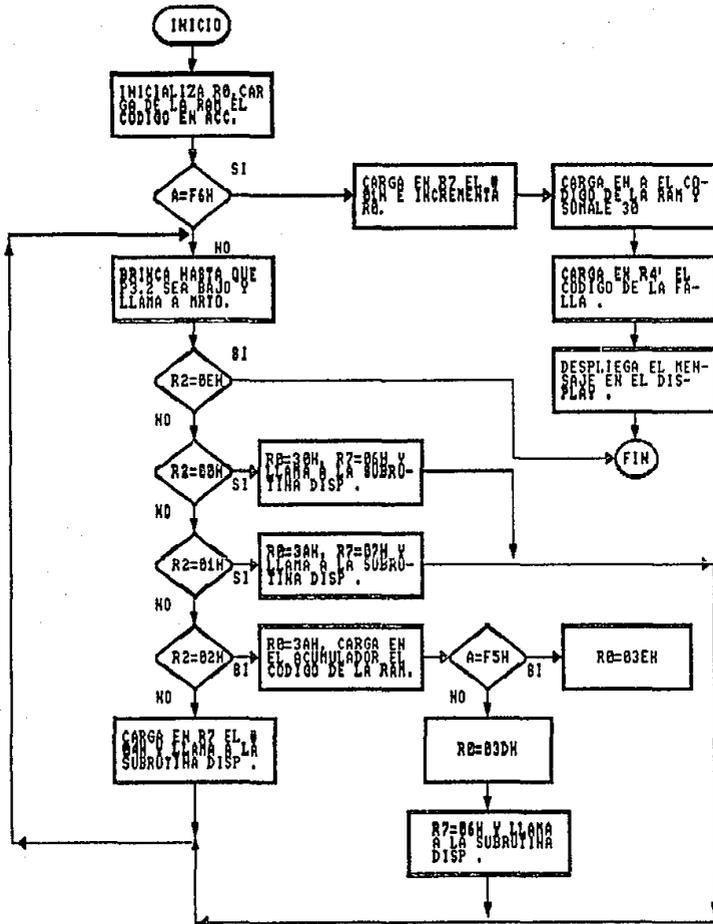
SUBROUTINA AUTO .



;SUBROUTINA AUTO
;ESTA SUBROUTINA SE ENCARGA DE REALIZAR UNA PRUEBA A LOS
;ELEMENTOS DEL EQUIPO IPE-1-IIE .

```
AUTO : MOV      R7,#013H      ;DESPLIEGA "HOLA"  
      CALL     DISP  
      SETB    RSO            ;SELECCIONA EL BANCO 1  
      MOV     R2,#000H      ;LIMPIA EL REG. DE BANDERAS  
      CLR     RSO           ;SELECCIONA EL BANCO 0  
      RET
```

SUBROUTINA SELEC_.



```

;SUBROUTINA SELEC_
;ESTA SUBROUTINA SE ENCARGA DE SELECCIONAR LA INFORMACION
;OBTENIDA DEL EQUIPO UAD Y DESPLEGARLA EN EL DISPLAY .

```

```

SELEC_: CALL BORRA           ;BORRA EL DISPLAY
MOV R3,#080H      ;CARGA LA POSICION
MOV R4,#026H      ;DESPLIEGA "&"
CALL DISPL_
MOV RO,#030H      ;INICIALIZA RO
MOV A,@RO         ;CARGA EN A LA LOC. 30
CJNE A,#0F6H,NOFA_1
MOV R7,#001H      ;CARGA EL CODIGO DE FALLA
INC RO           ;INCREMENTA RO
MOV A,@RO        ;NUM. DE LA FALLA
CALL ACOND_      ;ACONDICIONA EL NUMERO
SETB RSO        ;SELECCIONA EL BANCO 1
MOV R4,A        ;R4' EL NO. DE LA FALLA
CLR RSO        ;SELECCIONA EL BANCO 0
CALL DISP
JMP FIN_1

NOFA_1: JB P3.2,$      ;EJECUTA UN LOOP
CALL BORRA      ;BORRA EL DISPLAY
CALL MRT0      ;LLAMA A MRT0
CJNE R2,#009H,NOFA_2 ;BRINCA SI NO ES EL COD. SAL.
JMP FIN_1

NOFA_2: CJNE R2,#000H,NOFA_3 ;BRINCA SI R2 NO ES 000
MOV RO,#030H      ;CARGA EN RO LA DIR. INICIAL
MOV R7,#006H      ;SELECCIONA EL MENSAJE
CALL DISP
JMP NOFA_1

NOFA_3: CJNE R2,#001H,NOFA_4 ;BRINCA SI R2 NO ES 001
MOV RO,#03AH      ;CARGA LA DIRECCION INICIAL
MOV R7,#007H      ;SELECCIONA EL MENSAJE
CALL DISP

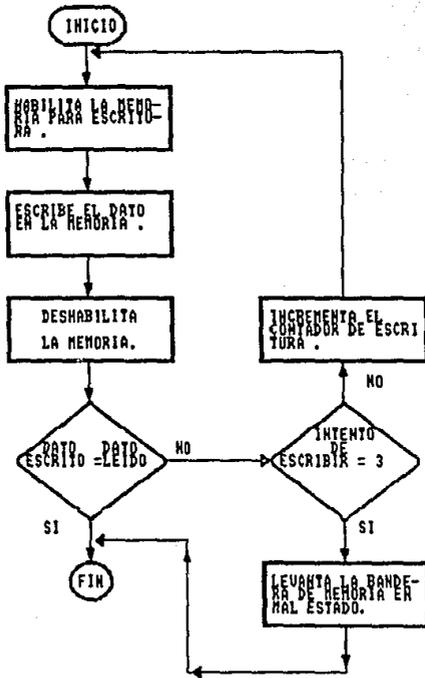
```

```

        JMP      NOFA_1
NOFA_4: CJNE    R2,#002H,NOFA_5 ;BRINCA SI R2 NO ES 002
        MOV     RO,#03AH      ;CARGA LA DIR. INICIAL
        MOV     A,#00
        CJNE    A,#0F5H,SIFA_1 ;BRINCA SI A NO ES IGUAL A 0F5
        MOV     RO,#03EH      ;CARGA LA DIR. INICIAL
SIFA_0: MOV     R7,#006H      ;SELECCIONA EL MENSAJE
        CALL    DISP
        JMP     NOFA_1
SIFA_1: MOV     RO,#03DH      ;CARGA LA DIR. INICIAL
        JMP     SIFA_0
NOFA_5: MOV     R7,#004H      ;SELECCIONA EL MENSAJE
        CALL    DISP
        JMP     NOFA_1
FIN_1:  RET

```

SUBROUTINA ESCRIB .



:SUBROUTINA ESCRIB
 ;ESTA SUBROUTINA SE ENCARGA DE ESCRIBIR UN DATO EN LA MEMORIA
 ;ADICIONAL Y VERIFICAR QUE SEA ESCRITO CORRECTAMENTE.

```

ESCRIB: CALL    DIREC_      ;DIRECCIONA LA MEMORIA
        SETB    RS1        ;SELECCIONA EL BANCO 2
        MOV     RO,#003H   ;RO'' EL NO. DE INTENTOS
        MOV     A,R1       ;DATO A ESCRIBIR EN EL ACC.
        MOV     B,A        ;SALVA EL DATO A ESCRIBIR
DE_NUE: CLR     RS1        ;SELECCIONA EL BANCO 0
        CLR     P1.3       ;HABILITA LA MEMORIA (CE=0)
        MOVX    @R1,A      ;SACA EL DATO POR PO
        NOP
        NOP
        NOP
        SETB    P1.3       ;DESHABILITA LA MEMORIA (CE=1)
        MOV     R6,#014H   ;EJECUTA UN RETARDO
BUENA:  MOV     R4,#OFFH   ;DE 10 MINISEGUNDOS
        DJNZ    R4,$       ;NECESARIO PARA LA
        DJNZ    R6,BUENA   ;MEMORIA
        CALL    LEE        ;LEE EL DATO
        CLR     C          ;BORRA EL CARRY
        SUBB   A,B         ;COMPARA AMBOS DATOS
        JZ     FIN_6       ;BRINCA SI EL ACC. ES CERO
        MOV     A,B        ;CARGA EN A EL DATO A ESCRIBIR
        SETB    RS1        ;SELECCIONA EL BANCO 2
        DJNZ    RO,DE_NUE ;BRINCA SI RO'' NO ES CERO
        CLR     RS1        ;SELECCIONA EL BANCO 0
        SETB    RSO       ;SELECCINA EL BANCO 1
        SETB    ACC.2     ;LEVANTA LA BANDERA H.E.
        MOV     R2,A
        CLR     RSO       ;SELECCIONA EL BANCO 0
FIN_6 : RET
  
```

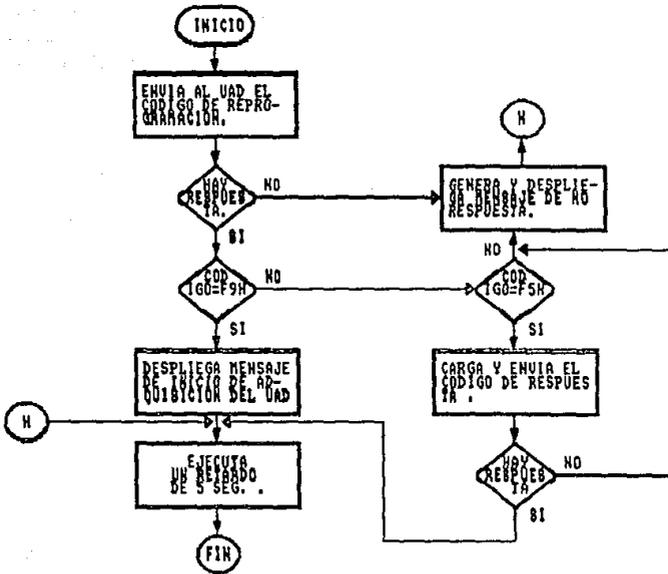
SUBROUTINA LEE .



;SUBROUTINA LEE
;ESTA SUBROUTINA SE ENCARGA DE LEER UN DATO DE LA MEMORIA
;EXTERNA Y SALVARLO EN EL ACUMULADOR.

LEE	:	CALL	DIREC_	:	DIRECCIONA LA MEMORIA
		CLR	P1.3	:	HABILITA LA MEM. PARA LEER
		MOVX	A,0R1	:	LEE EL DATO Y SALVALO EN ACC.
		NOP			
		NOP			
		NOP			
		SETB	P1.3	:	DESHABILITA LA MEMORIA
		RET			

SUBROUTINA INIDAT .



:SUBROUTINA INIDAT

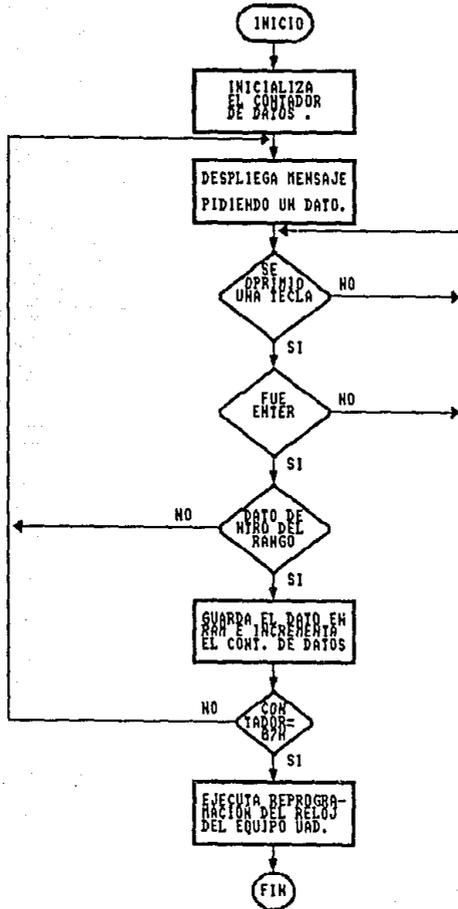
:ESTA SUBROUTINA SE ENCARGA DE INDICARLE AL EQUIPO UAD QUE

:INICIE LA ADQUISICION DE DATOS.

```
INIDAT: MOV     A,#044H      ;ENVIA EL COD. DE REPROGRA.
        CALL    COM_SP
        SETB   RSO        ;SELECCIONA EL BANCO 1
        MOV    A,R2       ;CARGA EN A EL REG. DE BAND.
        JBC   ACC.1,NOCOM4 ;BRINCA SI N.C. ESTA LEVANTADA
        CLR    RSO        ;SELECCIONA EL BANCO 0
        CJNE  R5,#0F2H,SIGUE5 ;BRINCA SI HUBO ACCESO
        MOV    R7,#003H   ;MENSAJE DE NO ACCESO PREVIO
        CALL  DISP
        MOV    R7,#0C6H   ;EJECUTA UN RETARDO DE 5 SEG.
        CALL  DELAY
        JMP   FIN_9
SIGUE5: MOV     A,#031H      ;CARGA Y ENVIA CODIGO DE INICIO
        CALL    COM_SP
        SETB   RSO        ;SELECCIONA EL BANCO 1
        MOV    A,R2       ;CARGA EN A EL REG. DE BAND.
        JBC   ACC.1,NOCOM4 ;BRINCA SI N.C. ESTA LEVANTADA
        CLR    RSO        ;SELECCIONA EL BANCO 0
        CJNE  R5,#0F9H,NO_ADQ ;BRINCA SI R5 NO ES F9
        MOV    R7,#010H   ;DESPLIEGA INICIO
        CALL  DISP
        JMP   RETAR
NO_ADQ: CJNE  R5,#0F5H,NOCOM4 ;BRINCA SI NO ES COD. DE FALLA
        MOV    A,#055H     ;ENVIA CODIGO DE RESPUESTA
        CALL    COM_SP
        SETB   RSO        ;SELECCIONA EL BANCO 1
        MOV    A,R2       ;CARGA EN A EL REG. DE BANDERAS
        JBC   ACC.1,NOCOM4 ;BRINCA SI N.C. ESTA PUESTA
        CLR    RSO        ;SELECCIONA EL BANCO 0
        MOV    A,R5       ;CARGA EN A EL # DE LA FALLA
        CALL  ACOND_     ;SUMALE 30 AL DATO
        SETB   RSO        ;SELECCIONA EL BANCO 1
```

	MOV	R4,A	:R4* EL NO. DE LA FALLA
	CLR	RSO	:SELECCIONA EL BANCO O
	MOV	R7,#001H	:DESPLIEGA MENSAJE DE FALLA
	CALL	DISP	
RETAR :	MOV	R7,#026H	:EJECUTA UN RETARDO DE 5 SEG.
	CALL	DELAY	
	JMP	FIN_9	
NOCOM4:	MOV	R2,A	:BORRA LA BANDERA N.C.
	CLR	RSO	:SELECCIONA EL BANCO O
	MOV	R7,#008H	:DESPLIEGA MENSAJE DE NO COM.
	CALL	DISP	
	JMP	RETAR	
FIN_9 :	RET		

SUBROUTINA RELOJ_ .



```

;SUBROUTINA RELOJ_
;ESTA SUBROUTINA SE ENCARGA DE PEDIR Y GUARDAR LOS DATOS
;NECESARIOS PARA REPROGRAMAR EL RELOJ DEL UAD-1-IIE.

```

```

RELOJ_ : SETB   RSO           ;SELECCIONA EL BANCO 1
        MOV    R6,#06DH     ;CARGA EL CODIGO DE "m"
        CLR   RSO           ;SELECCIONA EL BANCO 0
        CALL  AYUDA2
AGAIN:  CALL  AYUDA3
        CJNE  R2,#00FH,AGAIN ;BRINCA SI NO FUE ENTER
        MOV   RO,#033H
        MOV   @RO,#005H     ;CARGA LAS DECENAS TOPE
        INC   RO            ;INCREMENTA RO
        MOV   @RO,#009H     ;CARGA LAS UNIDADES TOPE
        CALL  CHECA
        JNZ   RELOJ_
        SETB  RSO           ;SELECCIONA EL BANCO 1
        MOV   R1,#035H     ;INICIALIZA R1'
        CLR   RSO           ;SELECCIONA EL BANCO 0
        CALL  MEMORI
HORA   : SETB   RSO           ;SELECCIONA EL BANCO 1
        MOV   R6,#048H     ;CARGA EL CODIGO DE "H"
        CLR   RSO           ;SELECCIONA EL BANCO 0
        CALL  AYUDA2
AGAIN1: CALL  AYUDA3
        CJNE  R2,#00FH,AGAIN1 ;BRINCA SI NO FUE ENTER
        MOV   RO,#033H
        MOV   @RO,#002H     ;CARGA LAS DECENAS TOPE
        INC   RO            ;INCREMENTA RO
        MOV   @RO,#003H     ;CARGA LAS UNIDADES TOPE
        CALL  CHECA
        JNZ   HORA
        CALL  MEMORI
DIA    : SETB   RSO           ;SELECCIONA EL BANCO 1
        MOV   R6,#044H     ;CARGA EL CODIGO "D"

```

```

                CLR      RSO          ;SELECCIONA EL BANCO 0
                CALL     AYUDA2
AGAIN2:        CALL     AYUDA3
                CJNE    R2,#00FH,AGAIN2 ;BRINCA SI NO FUE ENTER
                MOV     RO,#033H
                MOV     @RO,#003H      ;CARGA LAS DECENAS TOPE
                INC     RO            ;INCREMENTA RO
                MOV     @RO,#001H      ;CARGA LAS UNIDADES TOPE
                CALL     CHECA
                JNZ     DIA
                CALL     MEMORI
MES :          SETB    RSO          ;SELECCIONA EL BANCO 1
                MOV     R6,#04DH      ;CARGA EL CODIGO DE "M"
                CLR     RSO          ;SELECCIONA EL BANCO 0
                CALL     AYUDA2
AGAIN3:        CALL     AYUDA3
                CJNE    R2,#00FH,AGAIN3 ;BRINCA SI NO FUE ENTER
                MOV     RO,#033H
                MOV     @RO,#001H      ;CARGA LAS DECENAS TOPE
                INC     RO
                MOV     @RO,#002H      ;CARGA LAS UNIDADES TOPE
                CALL     CHECA
                JNZ     MES
                CALL     MEMORI
YEAR :         SETB    RSO          ;SELECCIONA EL BANCO 1
                MOV     R6,#041H      ;CARGA EL CODIGO DE "A"
                CLR     RSO          ;SELECCIONA EL BANCO 0
                CALL     AYUDA2
AGAIN4:        CALL     AYUDA3
                CJNE    R2,#00FH,AGAIN4 ;BRINCA SI NO FUE ENTER
                MOV     RO,#033H
                MOV     @RO,#009H      ;CARGA LAS DECENAS TOPE
                INC     RO
                MOV     @RO,#009H      ;CARGA LAS UNIDADES TOPE
                CALL     CHECA
                JNZ     YEAR
                CALL     MEMORI
SENANA :       SETB    RSO          ;SELECCIONA EL BANCO 1
                MOV     R6,#053H      ;CARGA EL CODIGO DE "S"

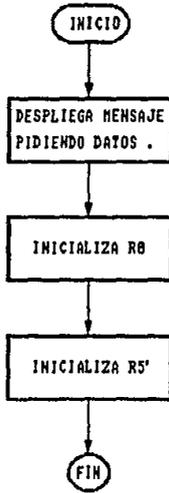
```

```

        CLR      RSO          ;SELECCIONA EL BANCO 0
        CALL     AYUDA2
AGAIN5: CALL     AYUDA3
        CJNE    R2,#00FH,AGAIN5 ;BRINCA SI NO FUE ENTER
        MOV     RO,#033H
        MOV     @RO,#000H      ;CARGA LAS DECENAS TOPES
        INC     RO
        MOV     @RO,#007H     ;CARGA LAS UNIDADES TOPE
        CALL    CHECA
        JNZ     SEMANA
        SETB    RSO          ;SELECCIONA EL BANCO 1
        MOV     RO,#031H
        MOV     A,@RO
        MOV     @R1,A
        INC     R1
        CLR     RSO          ;SELECCIONA EL BANCO 0
BISI_ : SETB    RSO          ;SELECCIONA EL BANCO 1
        MOV     R6,#042H     ;CARGA EL CODIGO DE "B"
        CLR     RSO          ;SELECCIONA EL BANCO 0
        CALL    AYUDA2
AGAIN6: CALL     AYUDA3
        CJNE    R2,#00FH,AGAIN6 ;BRINCA SI NO FUE ENTER
        MOV     RO,#033H
        MOV     @RO,#000H     ;CARGA LAS DECENAS TOPE
        INC     RO
        MOV     @RO,#003H     ;CARGA LAS UNIDADES TOPE
        CALL    CHECA
        JNZ     BISI_
        SETB    RSO          ;SELECCIONA EL BANCO 1
        MOV     RO,#031H
        MOV     A,@RO
        MOV     @R1,A        ;GUARDA EL DATO EN RAM
        CLR     RSO          ;SELECCIONA EL BANCO 0
        CALL    REPRO
        RET

```

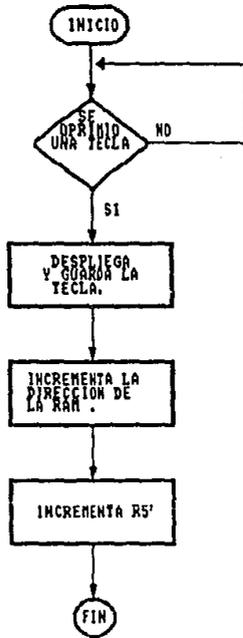
SUBROUTINA AYUDA2 .



;SUBROUTINA AYUDA2
;ESTA SUBROUTINA SE ENCARGA DE DESPLEGAR UN MENSAJE PIDIENDO
;LOS DATOS QUE SE REQUIEREN PARA REPROGRAMAR EL RELOJ.

```
AYUDA2: MOV    R7,#011H      ;COLOCA MENS. PIDIENDO UN DATO
          CALL  DISP
          MOV    RO,#030H    ;INICIALIZA RO
          SETB   RSO        ;SELECCIONA EL BANCO 1
          MOV    R5,#082H    ;INICIALIZA R5'
          CLR    RSO        ;SELECCIONA EL BANCO 0
          RET
```

SUBROUTINA AYUDA3 .



;SUBROUTINA AYUDA3
;ESTA SUBROUTINA SE ENCARGA DE ESPERAR A QUE SE OPRIMA UNA
;TECLA, GUARDA LA TECLA E INCREMENTA LA DIRECCION DE LA RAM
;Y DEL REGISTRO R5'.

AYUDA3:	JB	P3.2,\$;ESPERA UNA TECLA
	CALL	MRTD	;SALVA Y DESPLIEGA LA TECLA
	MOV	A,R2	
	MOV	@RO,A	;GUARDA EN RAM LA TECLA
	INC	RO	;INCREMENTA RO
	SETB	RS0	;SELECCIONA EL BANCO 1
	INC	R5	;INCREMENTA R5'
	CLR	RS0	;SELECCIONA EL BANCO 0
	RET		

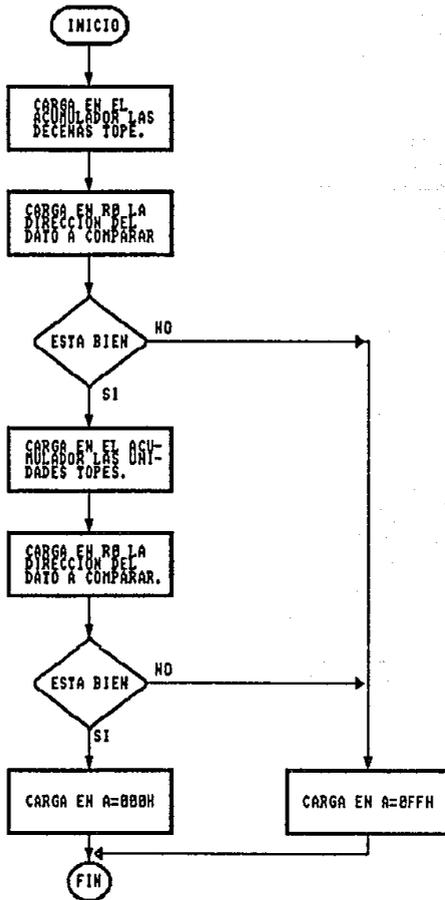
SUBRUTINA MEMORI .



;SUBROUTINA MEMORI
;ESTA SUBROUTINA SE ENCARGA DE GUARDAR LOS DATOS EN LA RAM
;DEL MICRO.

MEMORI:	SETB	RSO	:SELECCIONA EL BANCO 1
	MOV	RO,#031H	:INICIALIZA REG DE DIREC
	MOV	A,@RO	
	MOV	@R1,A	:GUARDA EL DATO EN RAM
	INC	R1	
	DEC	RO	:DECREMENTA RO'
	MOV	A,@RO	
	MOV	@R1,A	:GUARDA EL 2 DATO EN RAM
	INC	R1	
	CLR	RSO	:SELECCIONA EL BANCO 0
	RET		

SUBROUTINA CHECA .



```

;SUBROUTINA CHECA
;LA SIGUIENTE SUBROUTINA TIENE POR OBJETO CHECAR QUE LOS
;DATOS SEAN RAZONABLES TENIENDO COMO DATO DE ENTRADA EL
;ACUMULADOR, SIENDO ESTE EL LIMITE A NO REBASAR, SACANDO
;EL ACUMULADOR IGUAL A CERO SI ESTA DENTRO DEL RANGO 0
;DIFERENTE DE CERO SI ESTA FUERA.

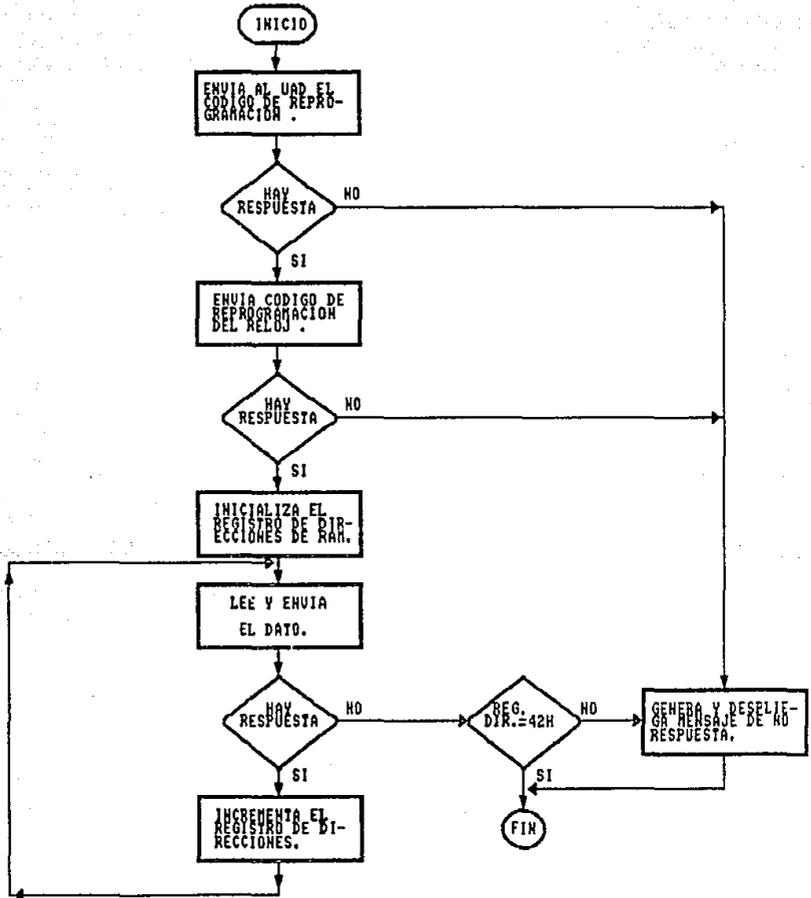
```

```

CHECA : MOV     RO,#033H
        MOV     A,CRO           .CARGA LAS DECENAS TOPE
        MOV     RO,#030H
        CLR     C               ;BORRA EL CARRY
        SUBB    A,CRO           ;COMPARA LAS DECENAS
        JC      MAL            ;BRINCA SI NO SON IGUALES
        JNZ     BIEN           ;BRINCA SI A NO ES CERO
        CLR     C               ;BORRA EL CARRY
        MOV     RO,#034H
        MOV     A,CRO           ;CARGA LAS UNIDADES TOPE
        MOV     RO,#031H
        SUBB    A,CRO           ;COMPARA LAS UNIDADES
        JC      MAL            ;BRINCA SI NO SON IGUALES
        JMP     BIEN
MAL :   CLR     C               ;BORRA EL CARRY
        MOV     A,#OFFH
        JMP     FIN_A
BIEN :  CLR     C               ;BORRA EL CARRY
        CLR     A               ;BORRA EL ACUMULADOR
FIN_A : RET

```

SUBROUTINA REPRO .



```

;SUBROUTINA REPRO
;ESTA SUBROUTINA SE ENCARGA DE ENVIAR AL EQUIPO UAD LOS DATOS
;NECESARIOS PARA LA REPROGRAMACION DEL RELOJ .

```

```

REPRO : MOV     A,#044H           ;CARGA Y ENVIA EL CODIGO DE
        CALL    COM_SP          ;REPROGRAMACION
        SETB   RSO              ;SELECCIONA EL BANCO 1
        MOV    A,R2             ;CARGA EN A EL REG. DE BAND.
        JBC   ACC.1,NOCOM5     ;BRINCA SI N.C. ESTA PUESTA
        CLR   RSO              ;SELECCIONA EL BANCO 0
        CJNE  R5,#0F2H,SIGUE3  ;BRINCA SI HUBO ACCESO
        MOV    R7,#003H        ;COLOCA NO ACC. PREVIO
        CALL  DISP              ;
        MOV    R7,#026H        ;EJECUTA UN RETARDO DE 5 SEG.
        CALL  DELAY             ;
        JMP   FIN_B             ;
SIGUE3: MOV     A,#044H           ;CARGA Y ENVIA CODIGO DE
        CALL    COM_SP          ;REPROGRAMACION DEL RELOJ
        SETB   RSO              ;SELECCIONA EL BANCO 1
        MOV    A,R2             ;CARGA EN A EL REG. DE BAND.
        JBC   ACC.1,NOCOM5     ;BRINCA SI N.C. ESTA PUESTA
        CLR   RSO              ;SELECCIONA EL BANCO 0
        MOV    RO,#035H        ;
SIGUE4: MOV     A,#0RO           ;CARGA EL DATO DE LA RAM
        CALL    COM_SP          ;
        SETB   RSO              ;SELECCIONA EL BANCO 1
        MOV    A,R2             ;CARGA EN A EL REG. DE BAND.
        JBC   ACC.1,NOCOM5     ;BRNCA SI N.C. ESTA PUESTA
        CLR   RSO              ;SELECCIONA EL BANCO 0
        INC   RO                ;
        CJNE  RO,#041H,SIGUE4  ;BRINCA SI YA NO HAY DATOS
        CALL  BORRA             ;BORRA EL DISPLAY
        MOV    R3,#080H        ;COLOCA EN R3 LA POSICION
        MOV    R4,#024H        ;DESPLIEGA "$"
        CALL  DISPL_           ;
        JB    P3.2,$           ;

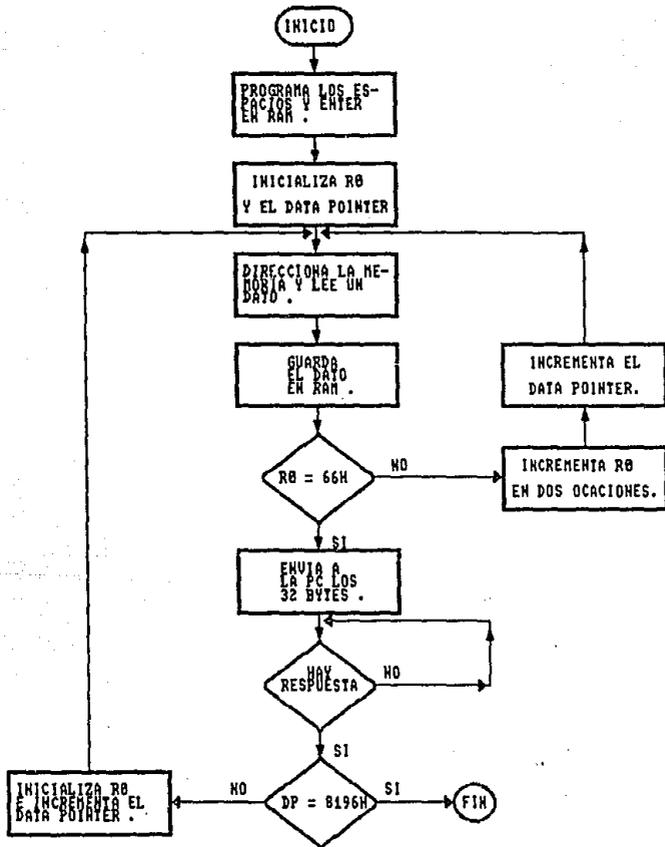
```

```

MOV      A,#031H      ;ENVIA CODIGO DE INICIO
CALL    OUT_SP
SETB    RSO           ;SELECCIONA EL BANCO 1
MOV     R5,#081H     ;CARGA EN R5 LA POSICION
CLR     RSO           ;SELECCIONA EL BANCO 0
CALL    MRTO
JMP     FIN_B
NOCONS: MOV    R2,A    ;BORRA LA BANDERA N.C.
CLR     RSO           ;SELECCIONA EL BANCO 0
MOV     R7,#008H     ;DESPLIEGA MENSAJE DE NO COM.
CALL    DISP
MOV     R7,#026H     ;EJECUTA UN RETARDO DE 5 SEG.
CALL    DELAY
FIN_B : RET

```

SUBROUTINA LEER21 .



;SUBROUTINA LEEME1

;ESTA SUBROUTINA SE ENCARGA DE LEER, GUARDAR EN RAM Y

;TRANSMITIR A LA PC UN TOTAL DE 32 BYTES, ESPERANDO RESPUESTA

;PARA TRANSMITIR LOS SIGUIENTES 32 BYTES.

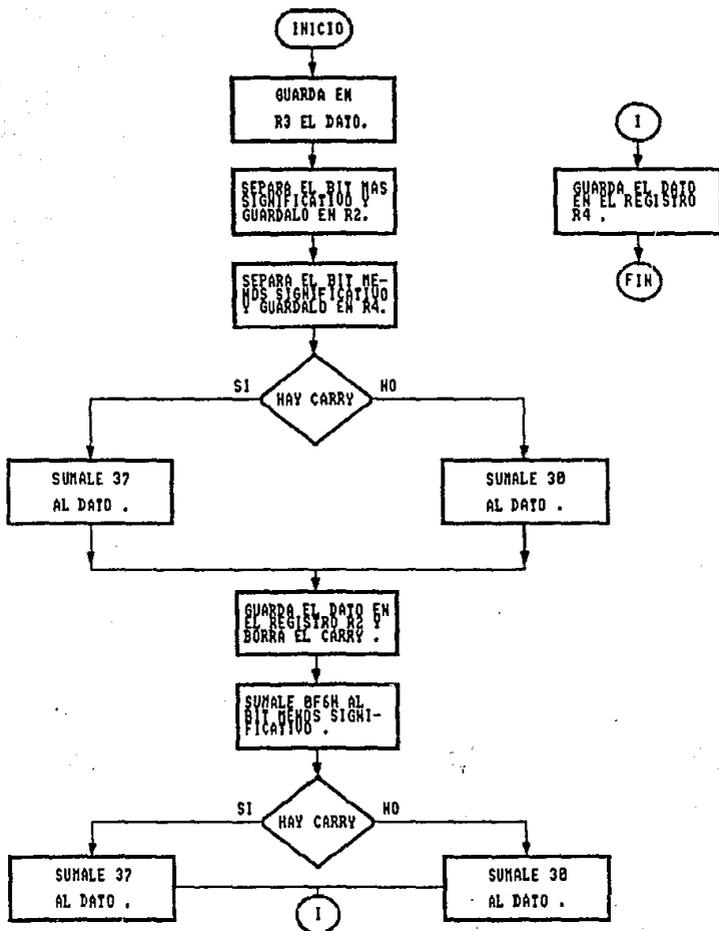
```
LEEME1: MOV     RO,#008H      ;INICIALIZA RO
        MOV     DPTR,#0E000H ;INICIALIZA EL DATA POINTER
LEE_0  : CJNE   RG,#000H,MEM_2 ;BRINCA SI NO ES COD. DE MEM.1
        CALL   LEE
        MOV     @RO,A        ;GUARDA EL DATO EN RAM
        CJNE   RO,#027H,LEE_1 ;BRINCA SI NO ES EL DATO 32
        CALL   LOADPC       ;TRANSFIERE LOS DATOS A LA PC
        INC     DPTR        ;INCREMENTA EL DATA POINTER
        MOV     A,DPL        ;CARGA LA PARTE BAJA DEL DP.
        JNZ    LEE_2        ;BRINCA SI EL ACC. NO ES CERO
        MOV     A,DPH        ;CARGA LA PARTE ALTA DEL DP.
        JNZ    LEE_2        ;BRINCA SI A NO ES CERO
        JMP     FIN_C
LEE_1  : INC     RO          ;INCREMENTA RO
        INC     DPTR        ;INCREMENTA EL DATA POINTER
        JMP     LEE_0
LEE_2  : MOV     RO,#008H    ;INICIALIZA RO
        JMP     LEE_0
MEM_2  : CJNE   RG,#001H,MEM_3 ;BRINCA SI NO ES COD. DE MEM.2
LEE_5  : CALL   LEE1
        MOV     @RO,A        ;SALVA EL DATO EN RAM
        CJNE   RO,#027H,LEE_3 ;BRINCA SI NO ES EL DATO 32
        CALL   LOADPC       ;TRANSFIERE LOS DATOS A LA PC
        INC     DPTR        ;INCREMENTA EL DATA POINTER
        MOV     A,DPL        ;CARGA LA PARTE BAJA DEL DP.
        JNZ    LEE_4        ;BRINCA SI A NO ES CERO
        MOV     A,DPH        ;CARGA LA PARTE ALTA DEL DP.
        JNZ    LEE_4        ;BRINCA SI A NO ES CERO
        JMP     FIN_C
LEE_3  : INC     RO          ;INCREMENTA RO
        INC     DPTR        ;INCREMENTA EL DATA POINTER
```

```

                JNP     MEM_2
LEE_4 : MOV     RO,#008H      ;INICIALIZA RO
                JNP     MEM_2
MEM_3 : CALL    LEE1
                MOV     @RO,A      ;SALVA EL DATO EN RAM
                CJNE   RO,#027H,LEE_6 ;BRINCA SI NO ES EL DATO 32
                CALL   LOADPC      ;TRANSFIERE LOS DATOS A LA PC
                INC     DPTR      ;INCREMENTA EL DATA POINTER
                MOV     A,DPL      ;CARGA LA PARTE BAJA DEL DP.
                JNZ    LEE_7      ;BRINCA SI A NO ES CERC
                MOV     A,DPH      ;CARGA LA PARTE ALTA DEL DP.
                JNZ    LEE_7      ;BRINCA SI A NO ES CERO
                JMP     FIN_C
LEE_6 : INC     RO              ;INCREMENTA RO
                INC     DPTR      ;INCREMENTA EL DATA POINTER
                JNP     MEM_3
LEE_7 : MOV     RO,#008H      ;INICIALIZA RO
                JMP     MEM_3
FIN_C : RET

```

SUBROUTINA TRANSF .

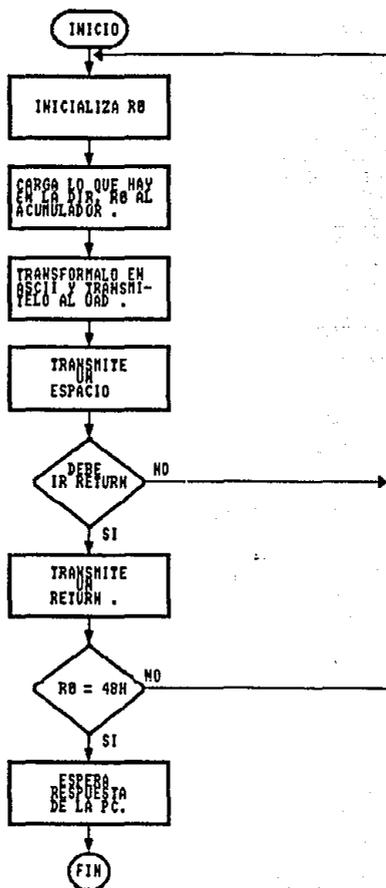


;SUBROUTINA TRANSF

;ESTA SUBROUTINA SE ENCARGA DE HACER UNA TRANSFORMACION DE LOS
;DATOS DE LA MEMORIA DE HEXADECIMAL A CODIGO ASCII.

```
TRANSF: MOV     R3,A           ;SALVA EN EL REG. R3 EL DATO
        ANL     A,#0FOH      ;SEPARA LA PARTE MAS SIGNI.
        SWAP   A
        MOV     R2,A         ;R2 LA PARTE MAS SIGNIFICATIVA
        MOV     A,R3         ;CARGA EL DATO EN A
        ANL     A,#00FH      ;SEPARA LA PARTE MENOS SIGNI.
        MOV     R4,A         ;R4 LA PARTE MENOS SIGNI.
        CLR     C            ;BORRA EL CARRY
        MOV     A,#0F6H
        ADD     A,R2
        JNC     DATNUM       ;BRINCA SI NO HAY CARRY
        MOV     A,#37H
        ADD     A,R2         ;SUMA 37 SI SON A,B,C,D,E
        MOV     R2,A         ;SALVA EL DATO EN R2
        JNP     LSBIT
DATNUM: MOV     A,#030H
        ADD     A,R2         ;SUMA 30 SI SON NUMEROS
        MOV     R2,A         ;SALVA EL DATO EN R2
LSBIT : CLR     C            ;BORRA EL CARRY
        MOV     A,#0F6H
        ADD     A,R4         ;COMPARA SI ES LETRA O NUM.
        JNC     DAT_NO       ;BRINCA SI EL CARRY ESTA PUESTO
        MOV     A,#037H
        ADD     A,R4         ;SUMA 37 SI ES UNA LETRA
        MOV     R4,A         ;SALVA EL DATO EN R4
        JMP     FIN_E
DAT_NO: MOV     A,#030H
        ADD     A,R4         ;SUMA 30 SI ES NUMERO
        MOV     R4,A         ;SALVA EL DATO EN R4
FIN_E : RET
```

SUBROUTINA LOADPC .



```

;SUBROUTINA LOADPC
;ESTA SUBROUTINA SE ENCARGA DE ENVIAR 32 BYTES DE INFORMACION
;A LA PC POR MEDIO DEL PUERTO SERIE.

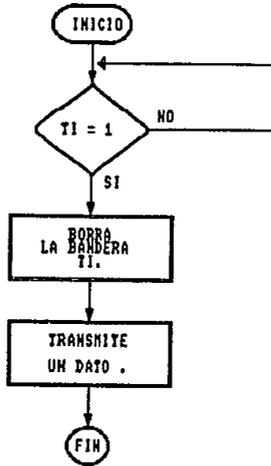
```

```

LOADPC: MOV     RO,#009H      ;INICIALIZA RO
DATOS3: MOV     A,@RO        ;CARGA EN A EL DATO DE RAM
        CALL    TRANSF       ;TRANSFORMA LOS DATOS
        MOV     A,R2         ;ENVIA LA PARTE MAS SIGNI.
        CALL    OUT_SP
        MOV     A,R4         ;ENVIA LA PARTE MENOS SIGNI.
        CALL    OUT_SP
        MOV     A,#020H      ;ENVIA EL COD. DE ESPACIO
        CALL    OUT_SP
        CJNE   RO,#017H,DATOS4 ;BRINCA SI NO HAY ENTER
DATOS6: MOV     A,#00DH      ;ENVIA ENTER
        CALL    OUT_SP
        JMP     DATOS5
DATOS4: CJNE   RO,#027H,DATOS5 ;BRINCA SI NO HAY ENTER
        JMP     DATOS6
DATOS5: CJNE   RO,#027H,DATOS7
        CALL    IN_SP
        RET
DATOS7: INC     RO
        JMP     DATOS3

```

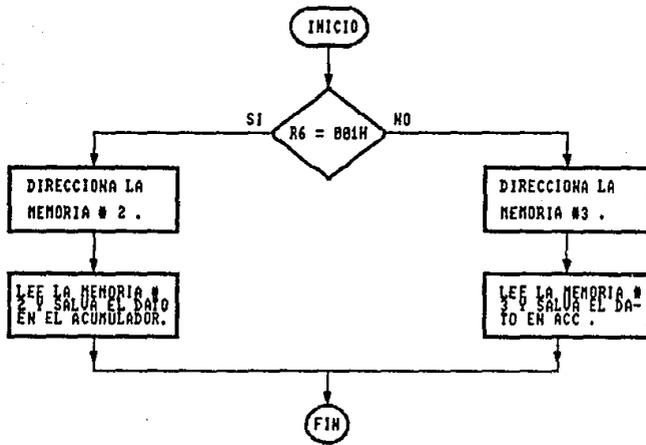
SUBROUTINA OUT_SP .



;SUBROUTINA OUT_SP
;ESTA SUBROUTINA SE ENCARGA DE TRANSMITIR UN DATO POR EL
;PUERTO SERIE.

OUT_SP:	JNB	TI,\$;ESPERA HASTA QUE TI SEA UNO
	CLR	TI	;BORRA TI
	MOV	SBUF,A	;TRANSMITE EL DATO
	RET		

SUBROUTINA LEE_1 .



```

;SUBROUTINA LEE1
;ESTA SUBROUTINA SE ENCARGA DE LEER UN DATO DE LA MEMORIA #2
;O UN DATO DE LA MEMORIA #3, SALVANDO EL DATO EN AMBOS CASOS
;EN EL ACUMULADOR.

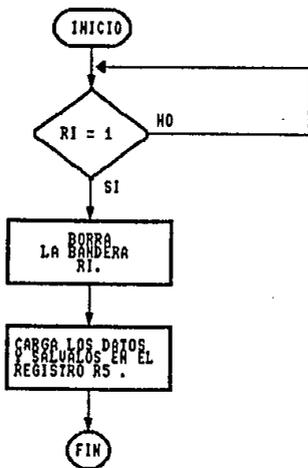
```

```

LEE1 : CJNE R6,#001H,LEE2 ;BRINCA SI YA SE LEYO LA MEM.2
      CALL DIREC_ ;DIRECCIONA LA MEMORIA
      CLR P1.0 ;HABILITA LA MEMORIA 2
      MOVX A,0R1 ;LEE EL DATO Y SALVALO EN ACC.
      NOP
      SETB P1.0 ;DESHABILITA LA MEMORIA 2
      JMP FIN_D
LEE2 : CALL DIREC_ ;DIRECCIONA LA MEMORIA
      CLR P1.1 ;HABILITA LA MEMORIA 3
      MOVX A,0R1 ;LEE EL DATO Y SALVALO EN ACC.
      NOP
      NOP
      SETB P1.1 ;DESHABILITA LA MEMORIA 3
FIN_D: RET

```

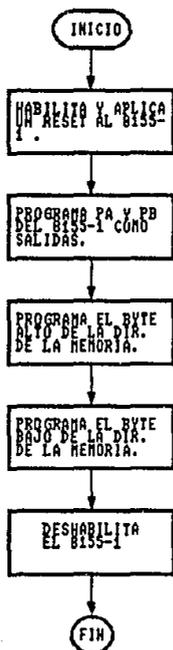
SUBROUTINA IM_SP .



;SUBROUTINA IN_SP
;ESTA SUBROUTINA SE ENCARGA DE HABILITAR LA RECEPCION DEL
;PUERTO SERIE ESPERANDO LA LLEGADA DEL DATO,SALVANDOLO EN
;EL REGISTRO R5.

IN_SP :	JNB	RI,\$;ESPERA UN DATO
	CLR	RI	;BORRA LA BANDERA RI
	MOV	A,SBUF	;CARGA EN A LOS DATOS DE PS.
	MOV	R5,A	;SALVA LOS DATOS EN R5
	RET		

SUBROUTINA DIREC_ .



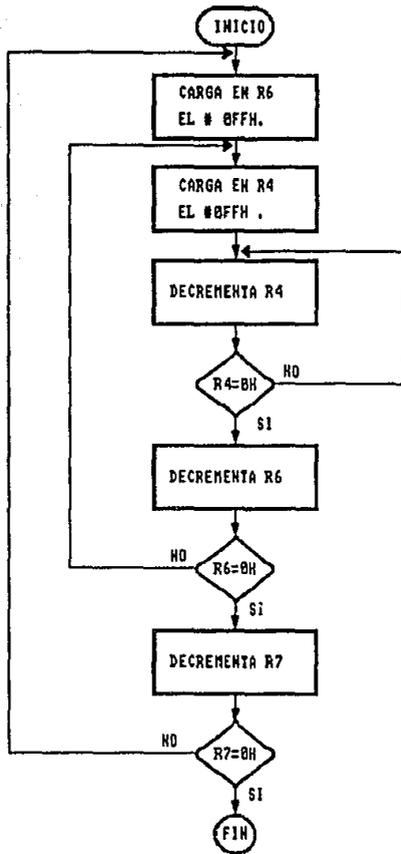
:SUBROUTINA DIREC_

:ESTA SUBROUTINA SE ENCARGA DE DIRECCIONAR A LA MEMORIA

:ADICIONAL PARA UNA LECTURA O UNA ESCRITURA.

```
DIREC_: CLR      P1.4      ;HABILITA EL 8155-1
        SETB     P1.6      ;RESET AL 8155-1
        NOP
        NOP
        NOP
        CLR      P1.6      ;BORRA EL RESET
        SETB     P1.5      ;PUERTOS DEL 8155-1
        NOV      R1,#000H   ;SELECC. PROG. DE PTOS.
        MOV      A,#003H
        MOVX     @R1,A      ;PA Y PB COMO SALIDAS
        MOV      A,DPH      ;CARGA LA PARTE ALTA DEL DP.
        MOV      R1,#002H   ;SELECCIONA EL PUERTO PB
        MOVX     @R1,A
        MOV      A,DPL      ;CARGA LA PARTE BAJA DEL DP.
        MOV      R1,#001H   ;SELECCIONA EL PUERTO PA
        MOVX     @R1,A
        CLR      P1.5      ;BORRA LA SELECCION DE PTOS.
        SETB     P1.4      ;DESHABILITA EL 8155-1
        RET
```

SUBROUTINA DELAY .



;SUBROUTINA DELAY
;ESTA SUBROUTINA SE ENCARGA DE REALIZAR UN RETARDO VARIABLE
;SIENDO LA DURACION DEL RETARGO EL # QUE SE CARGUE EN EL
;REGISTRO R7.

```
DELAY : MOV      R6,#OFFH
DELAY1: MOV      R4,#OFFH
        DJNZ     R4,$           ;EJECUTA UN LOOP FF VECES
        DJNZ     R6,DELAY1      ;DEC.Y BRINCA SI NO ES CERO R6
        DJNZ     R7,DELAY      ;DEC.Y BRINCA SI NO ES CERO R7
        RET
```

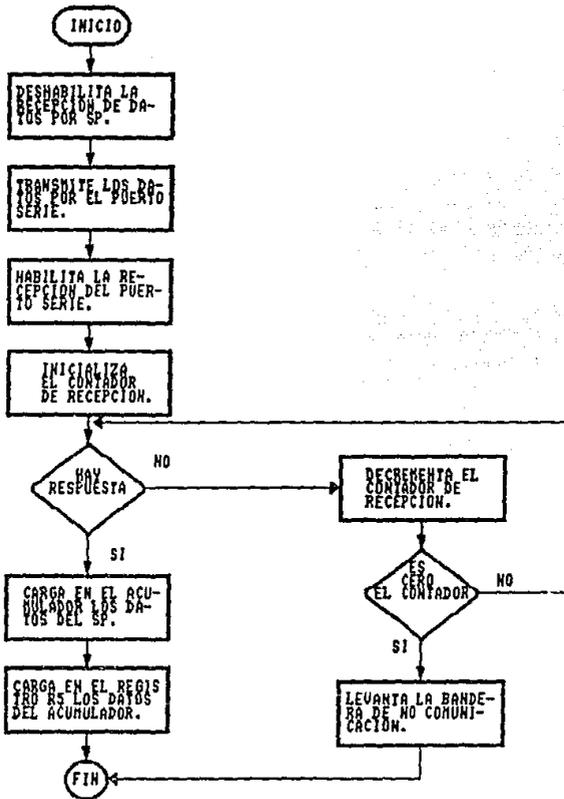
SUBROUTINA AYUDA1 .



;SUBROUTINA AYUDA1
;ESTA SUBROUTINA SE ENCARGA DE REALIZAR UNA SERIE DE PASOS
;QUE SIRVEN PARA DESPLEGAR EL MENSAJE QUE CONTIENE LOS DATOS
;DEL RELOJ.

AYUDA1: DEC R3 ;DECREMENTA R3
INC RO ;INCREMENTA RO
MOV A,0RO
CALL ACOND_ ;ACONDICIONA LOS DATOS
MOV R4,A ;COLOCA EL DATO EN EL DISPLAY
CALL DISPL_
RET

SUBROUTINA COM_SP .



```

;SUBROUTINA COM_SP
;ESTA SUBROUTINA SE ENCARGA DE TRANSMITIR UN MENSAJE DEL
;EQUIPO IPE-1-11E AL EQUIPO UAD-1-11E POR EL PUERTO SERIE
;DE COMUNICACION AGUARDA UN TIEMPO DE 10 SEGUNDOS LA
;RESPUESTA, SI NO HAY CONTESTACION LEVANTA LA BANDERA DE NO
;COMUNICACION NC Y REGRESA .

```

```

COM_SP: CLR      REN      ;DESHABILITA LA RECEP. SERIE
        JNB     TI,$     ;BRINCA HASTA QUE TI SEA 1
        CLR     TI       ;BORRA TI
        MOV     SBUF,A   ;TRANSMITE AL UAD
        NOP
        NOP
        NOP
        SETB   REN      ;HABILITA LA RECEP. SERIE
        MOV     R7,#02CH ;INICIALIZA EL CONT.DE RECEP.
ESPERO: MOV     R6,#OFFH
ESPER1: MOV     R5,#OFFH
ESPERA: JNB     RI,RECIBE ;BRINCA SI RI NO ESTA PUESTA
        CLR     RI       ;BORRA LA BANDERA RI
        MOV     A,SBUF   ;CARGA EN A LOS DATOS DEL PS.
        MOV     R5,A
        JMP     FIN_3
RECIBE: DJNZ    R5,ESPERA ;DEC.Y BRINCA SI R5 NO ES CERO
        DJNZ    R6,ESPER1 ;DEC.Y BRINCA SI R6 NO ES CERO
        DJNZ    R7,ESPERO ;DEC.Y BRINCA SI R7 NO ES CERO
        SETB   RSO      ;SELECCIONA EL BANCO 1
        MOV     R2,#002H ;LEVANTA LA BANDERA DE NO COM.
        CLR     RSO     ;SELECCIONA EL BANCO 0
FIN_3 : RET

```

SUBROUTINA ACCORD .



;SUBROUTINA ACOND_
;ESTA SUBROUTINA SE ENCARGA DE ACONDICIONAR LOS DATOS
;RECIBIDOS DEL EQUIPO UAD-1-11E, PARA PODER SER DESPLEGADOS
;EN EL DISPLAY.

ACOND_: ANL A, #00FH ;BORRA LA PARTE MAS SIGNIFI.
 ORL A, #0FOH
 ANL A, #03FH ;REALIZA A+30
 RET

Capítulo 5

Diseño Mecánico.

5.1 Introducción.

En el proceso de desarrollo de equipo e instrumentación electrónicos, es también de suma importancia que la envolvente mecánica de los equipos a desarrollar, sea diseñada atendiendo a especificaciones técnicas y funcionales que proporcionen al usuario la mejor facilidad de instalación, operación y mantenimiento.

Así mismo, la forma y calidad de los acabados dada a las envolventes, tendrá una repercusión importante en la introducción de los equipos a nivel comercial, ya que estas deberán estar acordes a la actualidad de materiales y formas utilizados en el mercado. Es obvio que un equipo no tendrá una buena aceptación comercial si no obstante que haya sido diseñado con tecnología electrónica moderna, se encierra dentro de una envolvente característica de diseños obsoletos.

Sin embargo, estas características no son las únicas que deben ser tomadas en cuenta para obtener el diseño final, ya que la complejidad de manufactura y el costo de los materiales utilizados en las partes mecánicas que formarán las envolventes, determinará el costo de las mismas.

Si solamente son considerados los aspectos mencionados en los dos primeros párrafos, el resultado podría ser un diseño excelente en cuanto a funcionalidad y calidad, pero ello podría repercutir en un costo de reproducción tal vez demasiado alto para la envolvente mecánica, dados los relativamente bajos costos de los componentes electrónicos en la actualidad.

Obviamente un diseño mecánico, que sea basado únicamente en la obtención de la mayor economía de reproducción, podrá llevar a un resultado pésimo en cuanto a calidad y funcionalidad se refiere.

El objetivo de este proyecto, se enfocó únicamente al diseño electrónico del sistema. Sin embargo con objeto de proporcionar una envolvente mecánica a los prototipos desarrollados, se atacó el diseño mecánico atendiendo principalmente a las características de funcionalidad , y actualidad de diseño y un tanto superficialmente al costo de reproducción de las partes involucradas.

No obstante, los resultados obtenidos aparentan sencillez de reproducción y puede considerarse que los diseños realizados, son el punto de partida de un análisis detallado para la obtención de un diseño óptimo a partir de técnicas de ingeniería de producto utilizadas en procesos industriales.

5.2 Diseño Mecánico de la UAD.

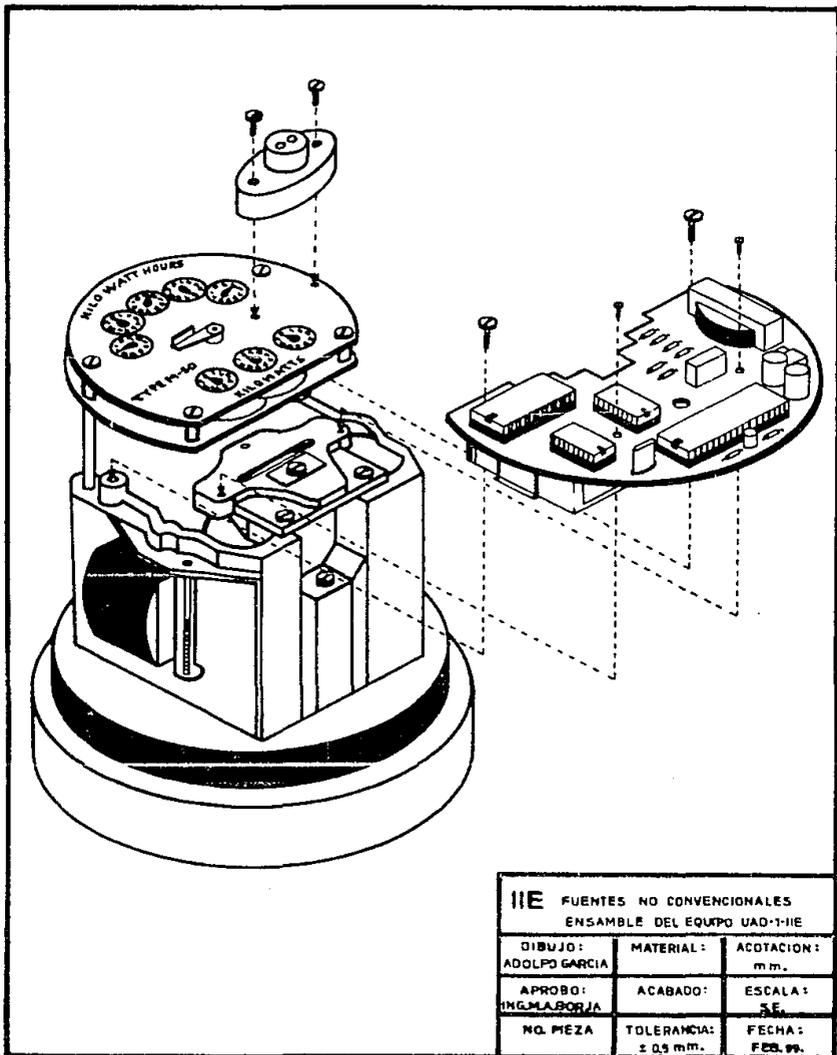
En la presente sección, se documenta el diseño realizado para las UAD.

Tal y como se mencionó en las especificaciones de diseño del capítulo 2, mecánicamente estas unidades consisten en tarjetas de circuito impreso introducibles dentro de Watthorímetros convencionales.

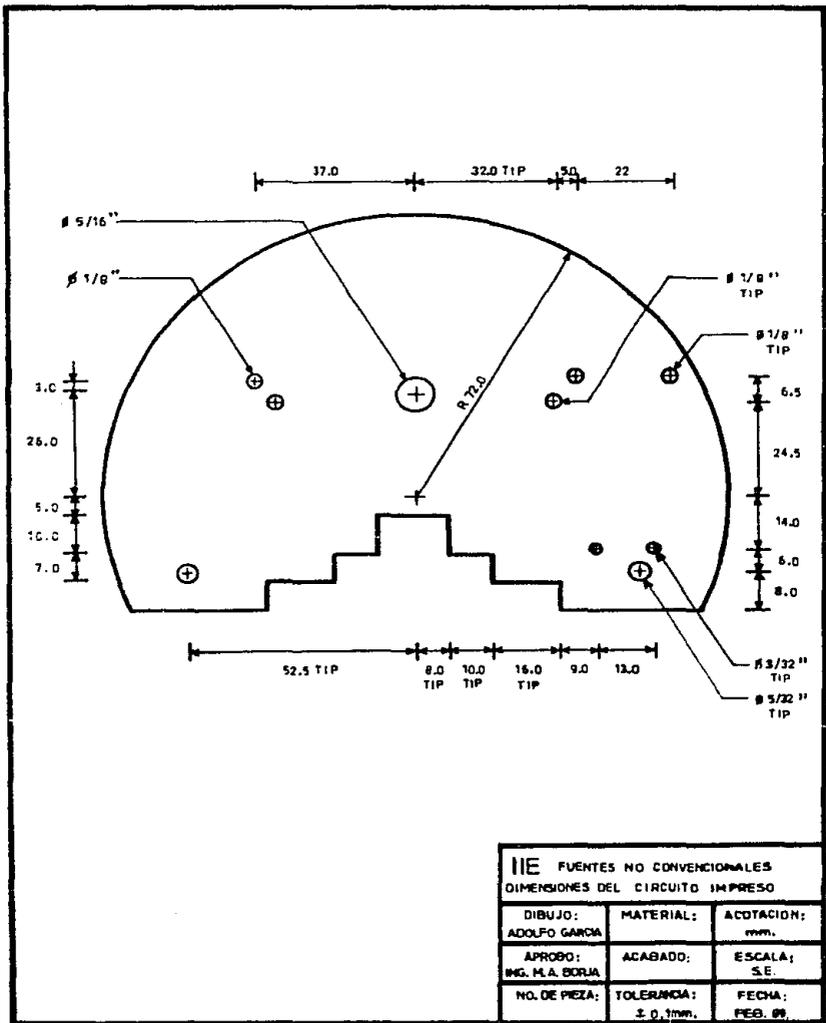
De acuerdo a ello, el problema principal por lo que respecta al diseño mecánico de las mismas, consiste en la minimización del espacio utilizado, de tal manera que la tarjeta desarrollada pueda estar contenida dentro de la cúpula de vidrio del Watthorímetro, y que se proporcione la facilidad de su introducción sin la necesidad de practicar modificación alguna al Watthorímetro utilizado.

Tal y como se mencionó anteriormente, la forma de la UAD, dependerá de la marca del Watthorímetro utilizado y en el presente proyecto, se desarrolló el diseño necesario para su introducción al modelo M-50 de la marca General Electric, utilizado comunmente en mediciones de subestaciones eléctricas y servicios industriales.

En las siguientes páginas, se presenta un bosquejo general del diseño realizado y el diseño de detalle de las partes involucradas.

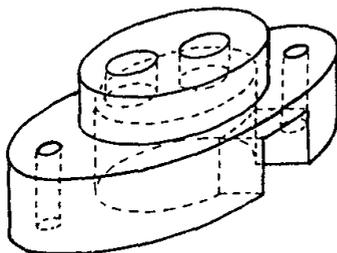
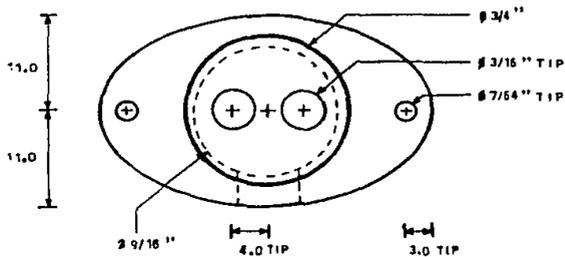
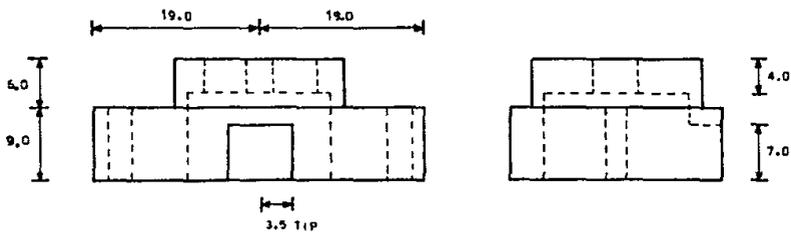


11E FUENTES NO CONVENCIONALES		
ENSAMBLE DEL EQUIPO UAD-1-11E		
DIBUJO: ADOLFO GARCIA	MATERIAL:	ACOTACION: mm.
APROBO: ING. LABORIA	ACABADO:	ESCALA: 5x
NO. MEZA	TOLERANCIA: ± 0.5 mm.	FECHA: FEB. 99.

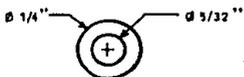
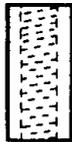
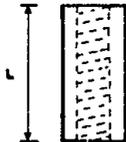


II E FUENTES NO CONVENCIONALES
DIMENSIONES DEL CIRCUITO IMPRESO

DIBUJO: ADOLFO GARCIA	MATERIAL:	ACOTACION: mm.
APROBO: ING. M.A. BORJA	ACABADO:	ESCALA: S.E.
NO. DE PIEZA:	TOLERANCIA: ± 0.1mm.	FECHA: FEB. 88



IIE PUENTES NO CONVENCIONALES PORTA INFRAROJOS		
DIBUJO: ADOLFO GARCIA	MATERIAL: NYLAMID	ACOTACION: mm.
APROBO: ING. M. A. BORJA	ACABADO:	ESCALA: S. E.
NO. DE PIEZA	TOLERANCIA: ± 0.1 mm.	FECHA: FEB. 89.



DOS PIEZAS DE $L = 16 \text{ mm.}$
 UNA PIEZA DE $L = 15 \text{ mm.}$



DOS PIEZAS.

IIE FUENTES NO CONVENCIONALES SEPARADORES		
DIBUJO: ADOLFO GARCIA	MATERIAL: ALUMINIO	ACOTACION: mm.
APROBO: ING. M.A. BORJA	ACABADO:	ESCALA 3E.
NO. PIEZA:	TOLERANCIA: $\pm 0.5 \text{ mm.}$	FECHA: FEB. 99.

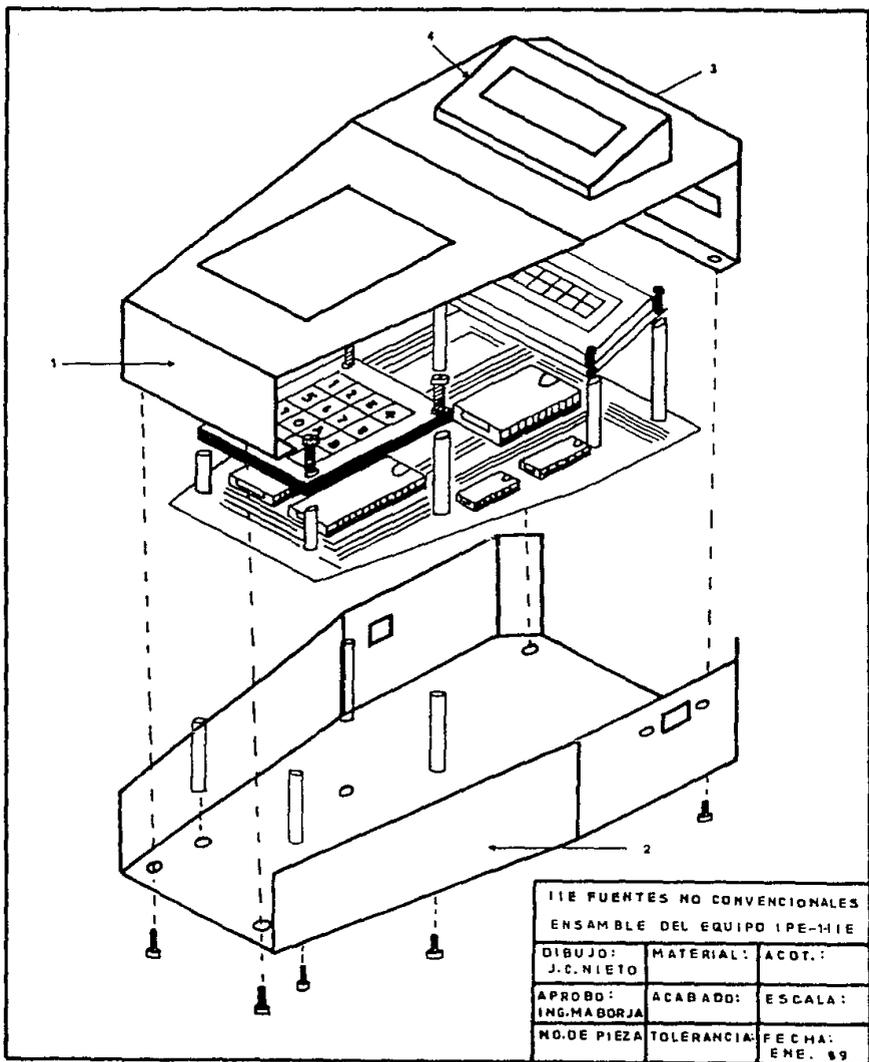
5.3 Diseño Mecánico del IPE.

Tal como se mencionó en las especificaciones de diseño del capítulo 2, el equipo IPE deberá estar contenido en una envolvente compacta y portátil.

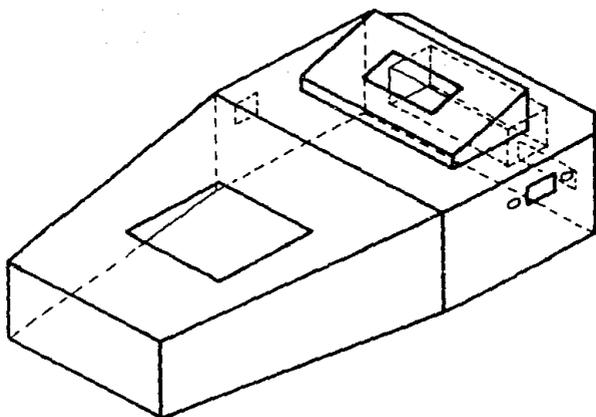
Dadas las características de las instalaciones donde se colocan los Watt-horímetros convencionales, el concepto de portabilidad debe considerarse como "instrumento de mano", refiriéndonos con ello a que el equipo IPE podrá ser sostenido y operado de una manera cómoda, sin la necesidad de colocarlo sobre una mesa o similar.

Para lograr esto, el diseño mecánico del IPE se realizó proporcionando al instrumento el mayor grado de compactación posible, cuidando que se proporcionara al operador un manejo cómodo del teclado y de los interruptores integrados, así como la mejor visualización del monitor alfanumérico de acuerdo al ángulo visual característico del mismo.

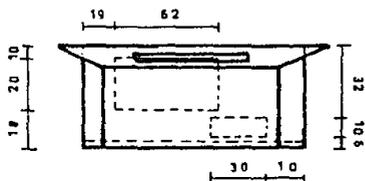
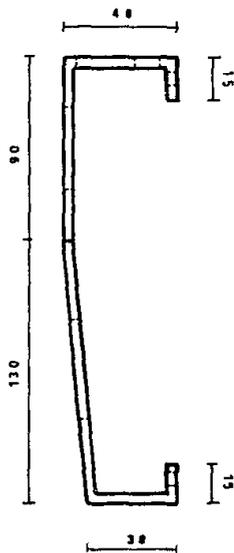
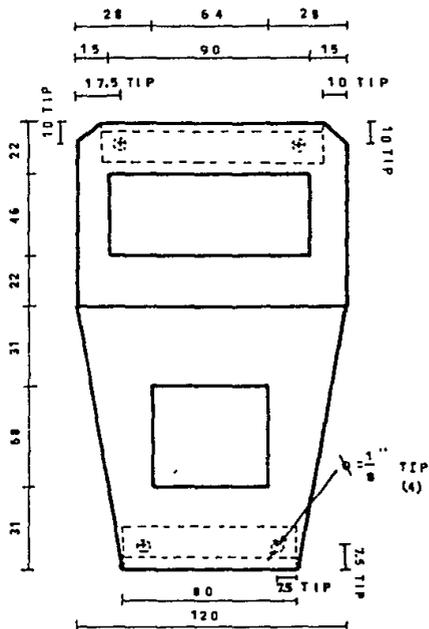
En las siguientes páginas, se presentan dos bosquejos desarrollados para el diseño mecánico conceptual y en seguida se documentan los diagramas correspondientes al diseño de detalle de las partes involucradas.



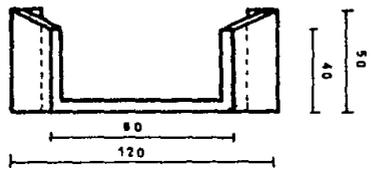
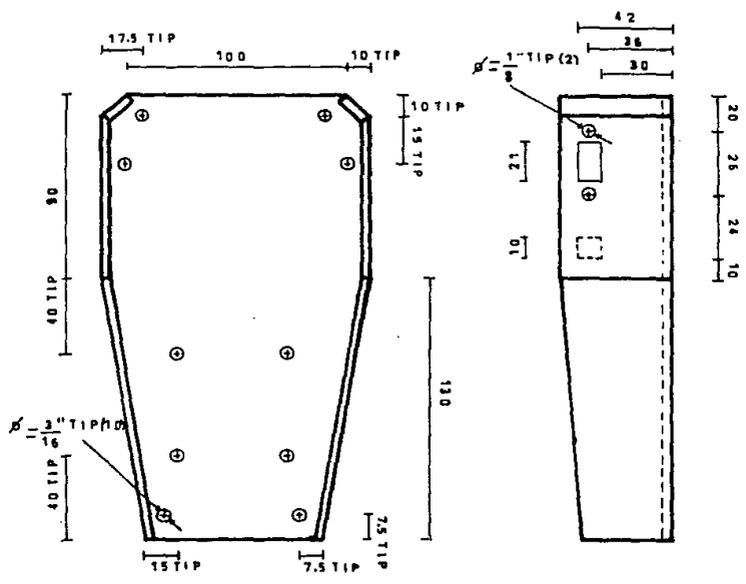
11E FUENTES NO CONVENCIONALES		
ENSAMBLE DEL EQUIPO 1PE-11E		
DIBUJO: J.C. NIETO	MATERIAL:	ACOT.:
APROBO: ING. MABORJA	ACABADO:	ESCALA:
Nº DE PIEZA	TOLERANCIA:	FECHA: ENE. 69



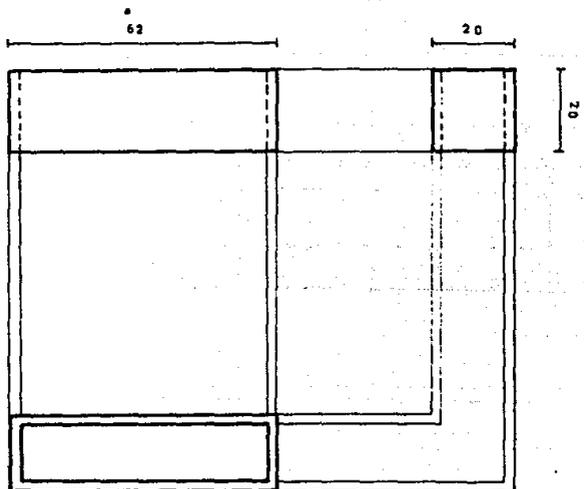
11E FUENTES NO CONVENCIONALES		
ENVOLVENTE DEL IPE-1-11E		
DIBUJO: J.C.NIETO	MATERIAL: LAMINA 78	ACOT: mm.
APROBADO: ING.MABORJA	ACABADO:	ESCALA: 5.E.
NO DE PEZA:	TOLERANCIA:	FECHA: ENE. 69



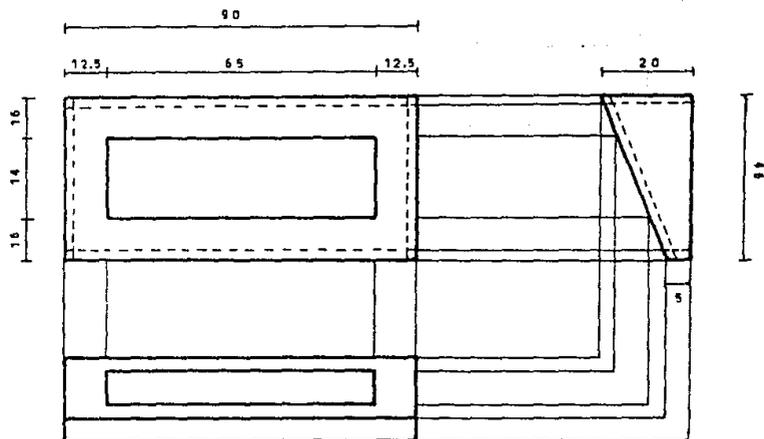
FUENTES NO CONVENCIONALES TAPA SUPERIOR		
DIBUJO: J.C. NIETO	MATERIAL: LAMINA 16	ACOT: m.m.
APROBO: ING. M. BORJA	ACABADO:	ESCALA: S.E.
NO. DE PIEZA: 1	TOLERANCIA: ± 0.5 m.m.	FECHA: ENE. 68



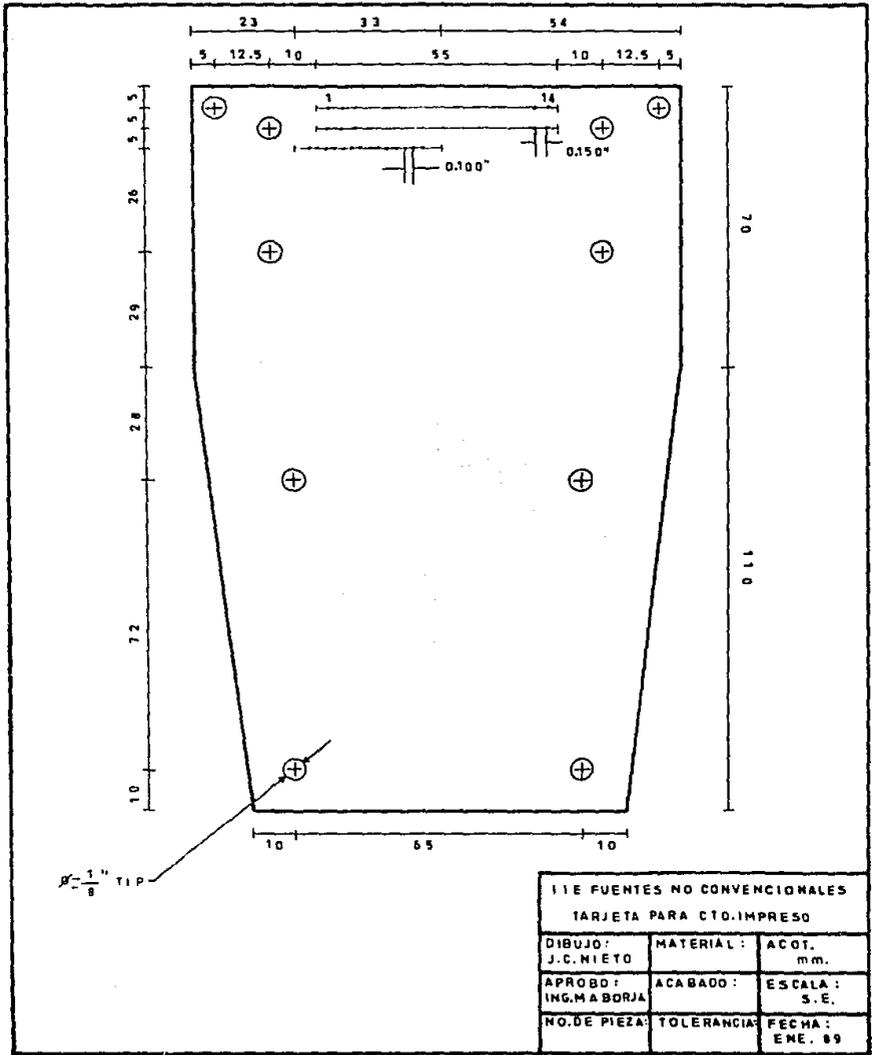
11E FUENTES NO CONVENCIONALES		
TAPA INFERIOR		
DIBUJO: J.C. NIETO	MATERIAL: LAMINA 16	ACOT: mm.
APROBADO: ING. MA BORJA	ACABADO:	ESCALA: 3.E.
NO. DE PIEZA: 2	TOLERANCIA: ± 0.5 mm.	FECHA: ENE. 89

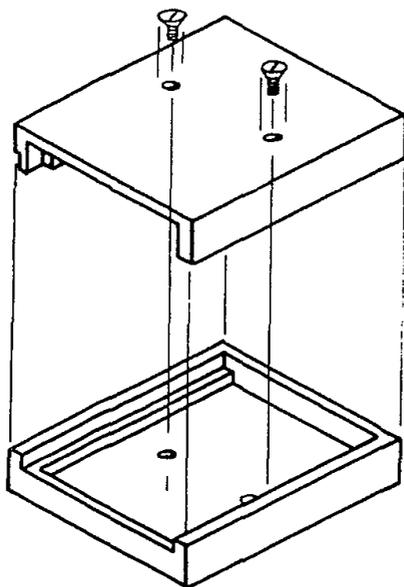


FUENTES NO CONVENCIONALES		
GUIA PARA CARTUCHO		
DIBUJO: J.C.NIETO	MATERIAL: LAMINA 16	ACOT. : mm.
APROBO: ING.MA.BORJA	ACABADO :	ESCALA: S.E.
NO.OE PIEZA: 3	TOLERANCIAS: ± 0.5mm,	FECHA : ENE. 89



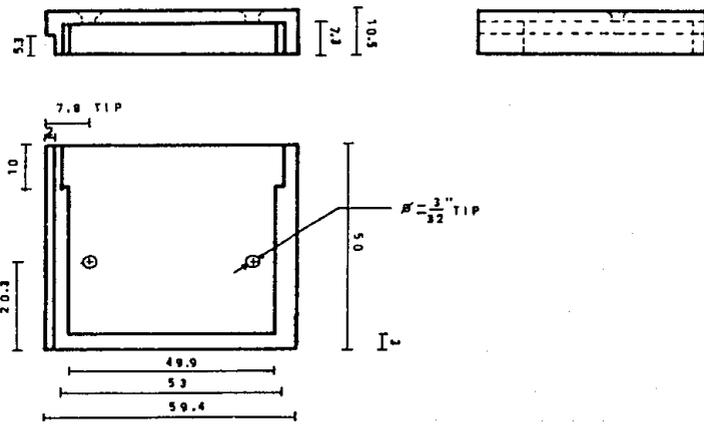
TIE FUENTES NO CONVENCIONALES		
PORTA DISPLAY		
DIBUJO: J C NIETO	MATERIAL: LAMINA 16	ACOT: mm.
APROB O: ING MABORJA	ACABADO:	ESCALA: S.E.
NO DE PIEZA: 4	TOLERANCIAS: ± 0,5 mm.	FECHA: ENE. 89



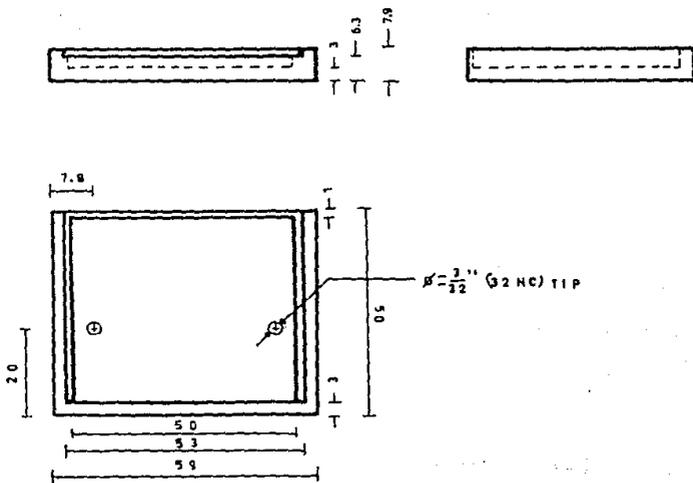


11E FUENTES NO CONVENCIONALES
ENSAMBLE DEL CARTUCHO

DIBUJO: J. C. NIETO	MATERIAL: ALUMINIO	ACOT.:
A PROBO: ING. M. BORJA	ACABADO:	ESCALA:
NO. DE PIEZA	TOLERANCIA	FECHA: ENE. 69

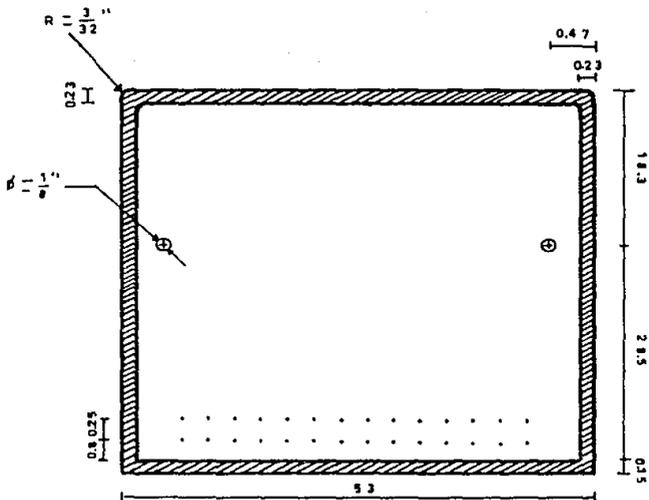


FUEENTES NO CONVENCIONALES TAPA SUPERIOR CARTUCHO		
DIBUJO: J.C. NIETO	MATERIAL: ALUMINIO	ACOT.: m m.
APROBO: ING. MABORJA	ACABADO:	ESCALA: S.E.
NO DE PIEZA	TOLERANCIA $\pm 0.025mm$	FECHA: ENE. 89



11E FUENTES NO CONVENCIONALES
TAPA INFERIOR CARTUCHO

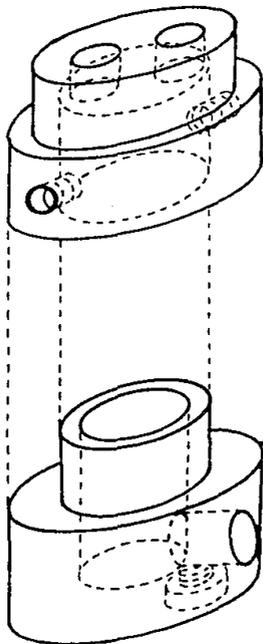
DIBUJO: J.C. NIETO	MATERIAL: ALUMINIO	ACOT.: m m.
APROBO: ING. MATORRA	ACABADO:	ESCALA: 5. E.
NO DE PIEZA	TOLERANCIA ± 0.025 mm	FECHA: ENE. 89



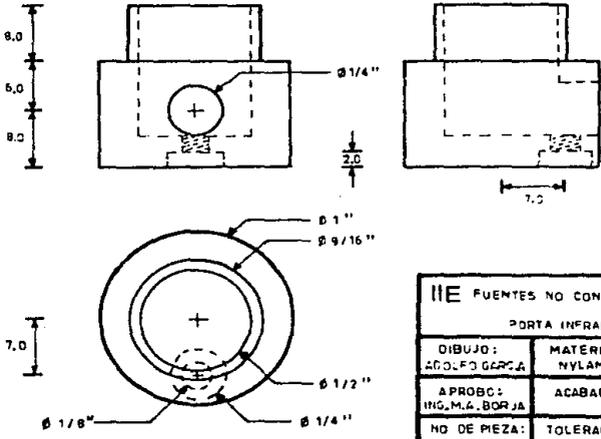
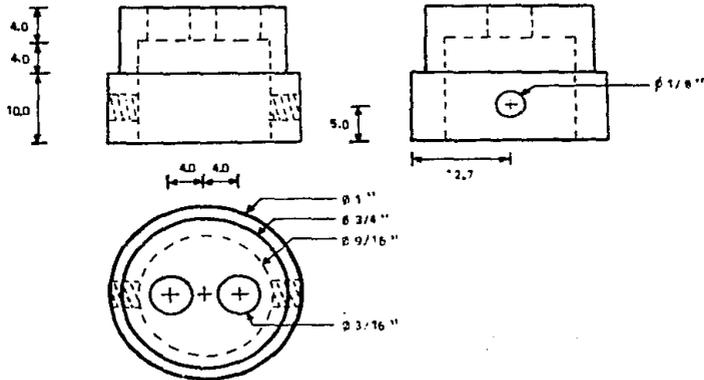
▣ NOTA: AREA LIBRE DE PISTAS

11E FUENTES NO CONVENCIONALES
TARJETA DE IMPRESO CARTUCHO

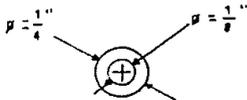
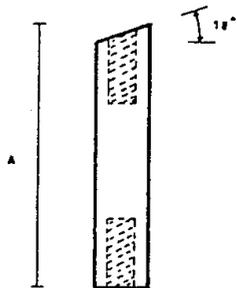
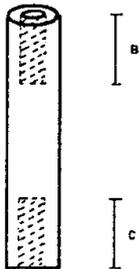
DIBUJO: J.C.NIETO	MATERIAL:	ACOT: m m.
APROBO: ING.MABORJA	ACABADO:	ESCALA: S. E.
NO.DE PIEZA	TOLERANCIA ± 0.025 mm.	FECHA: ENE. 89



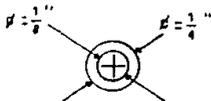
III FUENTES NO CONVENCIONALES PORTA INFRAROJOS		
DIBUJO: ADOLFO GARCIA	MATERIAL: NYLAMID	ACOTACION: mm.
APROBO: ING. M.A. BORJA	ACABADO:	ESCALA: 3.E.
NO. DE PIEZA 1 Y 2	TOLERANCIA: ± 0.5 mm.	FECHA: FEB. 89.



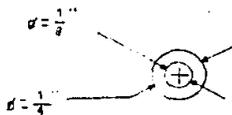
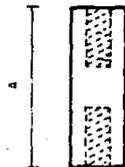
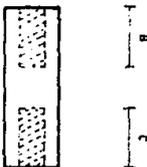
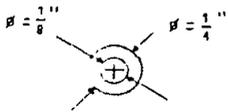
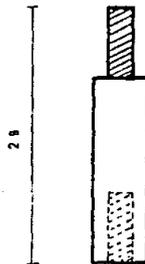
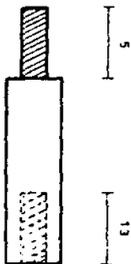
FUEENTES NO CONVENCIONALES		
PORTA INFRAROJOS		
DIBUJO: ACOLFO GARCIA	MATERIAL: NYLAMID	ACOTACION: mm.
A PROBO: ING. M. A. BORJA	ACABADO:	ESCALA: 5.E.
NO DE PIEZA: 1 V 2	TOLERANCIA: ± 0.5 mm.	FECHA: FEB. 89.



PIEZAS	A mm	B mm	C mm
2	38.5	10	10
2	36.5	10	10



11E FUENTES NO CONVENCIONALES SEPARADORES		
DIBUJO: J. C. NIETO	MATERIAL: COLD ROLL	ACOT.: mm.
APROBO: ING. MABORJA	ACABADO:	ESCALA: S. E.
NO DE PIEZA:	TOLERANCIA: ± 0.5 mm	FECHA: ENE. 99



PIEZAS	A mm.	B mm.	C mm.
2	14,5	5	6
2	17	5	6
2	23	10	10

FUEENTES NO CONVENCIONALES SEPARADORES		
DIBUJO: J.C. NIETO	MATERIAL: CGLD ROLL	ACDT: mm.
APROBO: ING. MABORJA	ACABADO:	ESCALA: S.E.
NO. DE PIEZA	TOLERANCIA: ± 0.5 mm.	FECHA: ENE. 89

Capítulo 6

Sistema Operativo.

6.1 Introducción.

Cuando el equipo de Interrogación/ Programación Externa (IPE), se conecta con una Unidad de Adquisición de Datos (UAD), a través de la interfaz de comunicación óptica, ambos equipos funcionan como una sola unidad electrónica que opera bajo un mismo sistema operativo.

Por Sistema Operativo, entendemos el conjunto de aquellas instrucciones que un operador puede introducir al equipo y que siendo reconocidas por este, son ejecutadas de acuerdo a las subrutinas de servicio correspondientes, con que fueron programados los microcontroladores.

Una vez que la interfaz de comunicación óptica ha sido conectada, la unidad electrónica formada por el equipo IPE y por la UAD - a lo que llamaremos SIARDET -, acepta un conjunto de comandos u órdenes que son introducidas por el operador utilizando el teclado instalado en el equipo IPE. Los mensajes generados por el sistema, los resultados de la instrucción dada y el progreso de las mismas, son mostrados en el monitor alfanumérico instalado en el equipo IPE.

El sistema SIARDET, es un sistema electrónico complejo en cuanto a la cantidad de funciones que internamente realiza. sin embargo la mayoría de estas funciones están enfocadas a que su operación sea muy simple. En todos los casos, el sistema operativo guía al operador en forma tutorial, lo que simplifica aún más su operación.

6.2 Modo de Espera de Instrucciones (MEI).

Una vez que el sistema ha sido interconectado a través de la interfaz de comunicación óptica, al prender el equipo IPE, aparece en el monitor una secuencia de la palabra **HOLA**, desplazándose tres veces hacia la derecha. Durante esta secuencia, el equipo IPE realiza subrutinas de autoprueba. Una vez terminada dicha secuencia, en el monitor aparecerá un asterisco (*), que es el signo escogido para indicar al operador que el sistema funciona correctamente y que se encuentra en el MODO DE ESPERA DE INSTRUCCIONES (MEI).

En el estado MEI, SIARDET reconocerá 5 instrucciones que son:

A ACCESO.

B INTERROGACION.

C TRANSFERENCIA.

D REPROGRAMACION.

E MODO PC.

Estas instrucciones serán ejecutadas al introducir la letra correspondiente por medio del teclado del IPE.

En todos los casos, la letra oprimida aparecerá en el monitor durante dos segundos (eco) y posteriormente las palabras correspondientes a la instrucción reconocida, serán mostradas.

6.2.1 Modo de ACCESO (A).

Esta instrucción tiene por objeto proteger al sistema de recuperación de la información almacenada o de procedimientos de reprogramación, por parte de personal no autorizado.

Un procedimiento previo de MODO DE ACCESO, será indispensable para que el operador pueda ejecutar las instrucciones de MODO DE TRANSFERENCIA y de MODO DE REPROGRAMACION.

Al oprimir la tecla **A**, la letra **A** aparecerá en el monitor durante dos segundos y en seguida aparecerá la palabra **ACCESO**.

El operador debe introducir cuatro dígitos, que deben corresponder a la clave o contraseña de la Unidad de Adquisición de Datos a la cual se realizó la interconexión.

En el caso de que el operador cometa un error al introducir alguno de los dígitos, podrá borrarlo oprimiendo la tecla E.

Una vez que los cuatro dígitos han sido introducidos y que son mostrados en el monitor, el operador oprimirá la tecla F, que es reconocida por SIARDET al igual que un computador reconoce la tecla RETURN o ENTER.

Así la instrucción es ejecutada y en el monitor aparecerá uno de los tres mensajes siguientes:

1. ACCESO PERMITIDO.
2. ACCESO NEGADO.
3. NO HAY RESPUESTA.

El primer mensaje indica que la UAD ha reconocido la contraseña introducida por el operador y las instrucciones de MODO DE TRANSFERENCIA y REPROGRAMACION, serán ejecutadas en el momento que el operador lo desee.

El segundo mensaje indica que la UAD NO reconoció la contraseña introducida por el operador, en este caso no se tendrá acceso a los modos de operación mencionados en el párrafo anterior. El acceso puede intentarse nuevamente, sin embargo solo se permiten seis intentos consecutivos y el sistema se deshabilitará automáticamente durante 15 minutos.

El tercer mensaje indica que la UAD no respondió a la instrucción enviada por el IPE, ver sección de MENSAJES DE ERROR.

En cualquiera de los casos, el mensaje aparecerá en el monitor durante 10 segundos y SIARDET regresará al estado MEI.

6.2.2 Modo de INTERROGACION (B).

Este modo de operación, permite al usuario, realizar rutinas de inspección, para verificación del correcto estado de operación del sistema, así como para obtener datos útiles en la programación de operación continua del sistema.

Al oprimir la tecla B, la letra B aparecerá en el monitor durante dos segundos y en seguida se mostrará la palabra INTERROGACION, esta palabra permanecerá en el monitor durante 10 segundos y posteriormente aparecerá uno de los siguientes mensajes:

- El signo &.

• NO HAY RESPUESTA.

El primer mensaje indica al operador que la UAD ha aceptado la interrogación y que la información correspondiente ya se encuentra en el equipo IPE, por lo que podrá ser consultada.

El segundo mensaje, indica que la UAD no respondió a la interrogación, ver sección correspondiente a Mensajes de Error.

Una vez que el signo & aparece en el monitor, el operador puede examinar la información enviada, oprimiendo para ello las teclas 0,1 o 2.

Al oprimir la tecla 0, en el monitor aparecerá la fecha y hora corrientes en el reloj interno de la UAD -en el momento de la interrogación-. esto en el siguiente formato:

 AÑO-MES-DIA HORA:MIN

 AA-MM-DD HH:mm

Estando esta información en el monitor, puede oprimir la tecla 1, y en el monitor aparecerá la información correspondiente al estado de operación del sistema, de acuerdo al siguiente formato:

 ST=XX ND=DD

 Donde: ST indica el estado de operación
 del sistema que puede ser:

 N= OPERACION NORMAL.

 FO= LA UAD NO HA SIDO INICIALIZADA.

 F1= FALLA DE MEMORIA, LA UAD NO PUEDE PROGRAMAR
 DATOS EN SU MEMORIA

 F2= FALLA DE RELOJ, EL RELOJ INTERNO DE LA UAD
 NO ESTA FUNCIONANDO CORRECTAMENTE.

ND=DD , INDICA AL OPERADOR NUMERO DE DIAS
QUE FALTAN PARA QUE LA MEMORIA DE
LA UAD, SEA SATURADA CON INFORMACION
ADQUIRIDA.

Estando presente esta información en el monitor, el operador puede oprimir la tecla 2, lo que ocasionará que aparezca la información correspondiente a la fecha y hora en que se inició el periodo corriente de adquisición de datos. El formato utilizado es el siguiente:

AÑO-MES-DIA HORA :MIN

AA-MM-DD HH:mm

Para abandonar el modo de INTERROGACION, se oprime la tecla 9, lo que causará que el sistema regrese al estado MEI. Mientras esto no se haga, el operador podrá consultar la información enviada por la UAD, cuantas veces lo desee, oprimiendo las teclas 0, 1 o 2, sin que para ello deba cuidarse alguna secuencia.

6.2.3 Modo de TRANSFERENCIA (C).

Esta instrucción esta orientada a la recuperacion PARCIAL o TOTAL de la información adquirida por la UAD.

Se pueden realizar vaciados parciales de la información adquirida, sin que por ello sea necesario reinicializar el periodo de adquisición de datos en proceso.

Por ejemplo: en el caso de que se deseen realizar periodos continuos de medición en una base de tiempo mensual, pero que se desea analizar la información adquirida una semana después de que se inició el periodo mensual.

Para la recuperación de la información, el sistema utiliza un cartucho de memoria que deberá ser instalado en el equipo IPE, en el conector correspondiente.

No se requiere un tratamiento previo a dicho cartucho, tal como sería el borrado de dicha memoria. Si el cartucho instalado contiene información, esta será reemplazada por la nueva información, sin que por ello exista un traslape de la preexistente con la que se graba en la transferencia.

Para iniciar un procedimiento de TRANSFERENCIA, se oprime la tecla C y la letra C aparecerá en el monitor durante dos segundos, en seguida aparecerá uno de los tres mensajes siguientes:

1. TRANSFERENCIA.
2. FALLA 1.
3. NO HAY ACCESO PREVIO.

Con el primer mensaje, el sistema está indicando al operador, que la transferencia de información almacenada en la UAD, se está llevando a cabo y que dicha información se está almacenando en el cartucho de memoria instalado en el IPE. El proceso de transferencia de información toma al sistema aproximadamente 5 minutos.

Si el segundo mensaje aparece en el monitor, SIARDET está indicando que no se colocó un cartucho de memoria en el IPE, o bien que aquel que se colocó está en mal estado.

El tercer mensaje, indicará que la UAD no respondió a la instrucción dada, ver sección de Mensajes de Error.

Al finalizar la transferencia de la información adquirida, en el monitor aparecerá la palabra FIN, durante un intervalo de 10 segundos, y el sistema regresará al estado MEI.

En este momento, el cartucho de memoria puede ser retirado del equipo IPE.

Una vez realizada la secuencia de transferencia, la UAD seguirá adquiriendo datos en forma normal, sin interrumpir el período corriente, esto significa que la información adquirida hasta la fecha sigue presente en la memoria de la UAD y que la fecha y hora de inicio del período de adquisición de datos, no ha variado.

Este procedimiento será el equivalente a una transferencia PARCIAL.

Es importante hacer notar que SIARDET no pierde información al estar transfiriendo datos, al igual que en cualquiera de sus posibles modos de operación.

Para iniciar un nuevo período de adquisición de datos, después de realizada un proceso de transferencia de información, lo que equivale a un proceso de variado TOTAL, se deberá realizar la instrucción correspondiente en el modo de reprogramación, que se detalla en el siguiente inciso.

6.2.4 Modo de REPROGRAMACION (D).

Esta instrucción, es de utilidad cuando después de instalada la UAD, se requiere inicializar su operación, o bien cuando se desea reinicializar un periodo de adquisición de datos.

Esta instrucción, al igual que la de TRANSFERENCIA, no será reconocida por el sistema, si no se ha realizado un procedimiento previo de ACCESO.

Para iniciar una secuencia de REPROGRAMACION, oprima la tecla D, la letra D aparecerá en el monitor y en seguida aparecerá la palabra REPROGRAMACION, la cual permanecerá durante diez segundos y el siguiente menú será mostrado en el monitor:

4=I 5=R

Este pequeño menú, presenta dos opciones que son:

I INICIACION DE UN PERIODO DE ADQUISICION DE DATOS.

R REPROGRAMACION DE RELOJ.

Estos dos modos de operación, se detallan a continuación.

Iniciación de un nuevo período.

Al oprimir la tecla 4, el IPE envía automáticamente a la UAD una instrucción de inicio, y como resultado se pueden obtener los siguientes mensajes:

1. INICIO LA UAD.
2. NO HAY ACCESO PREVIO.
3. NO HAY RESPUESTA.

Si el primer mensaje es mostrado, la UAD reconoció la instrucción y ha iniciado un nuevo periodo de adquisición de datos, lo que significa que:

- A Toda la información almacenada hasta este momento, ha sido borrada.
- B La fecha y hora de inicio del nuevo período de adquisición de datos, han sido almacenadas en la memoria.
- C Un código de identificación de la UAD, ha sido almacenado en la memoria.

En este caso, el mensaje No. 1, aparecerá en el monitor durante 10 segundos y el sistema regresará a su estado de operación MEI.

Si el segundo mensaje aparece, el sistema indica que se está intentado reprogramar la unidad sin haber realizado un procedimiento de ACCESO.

En el caso de aparecer el mensaje No. 3, la UAD no respondió a la instrucción enviada, ver sección de Mensajes de Error.

Programación de reloj.

Cuando una UAD, se instala por primera vez, su reloj interno no está programado y requiere que se le proporcione la hora y fecha actuales, una vez que esto se realiza, el reloj funcionará en forma continua durante aproximadamente 5 años, aún y en el caso de que la UAD sea removida del Watt-horímetro, almacenada sin alimentación principal, etc. Para que dicho funcionamiento sea continuo, solamente se requiere que la pequeña batería de Litio instalada en la UAD, no sea removida.

Para iniciar un procedimiento de reprogramación de reloj de la UAD, oprima la tecla 5. El dígito 5 aparecerá durante 2 segundos en el monitor, y en el posteriormente aparecerá lo siguiente:

m= ?

Aquí se deberá introducir los minutos a los que se desea iniciar la hora del reloj, tecleando dos dígitos que corresponderán a las decenas y unidades.

Para que el dato sea introducido al sistema, se deberá oprimir la tecla F.

En seguida en un proceso repetitivo el sistema solicitará la siguiente información:

H= (hora), (decenas y unidades)

D= (día), (decenas y unidades)
M= (mes), (decenas y unidades)
A= (año), (decenas y unidades)
S= (día de la semana), (dos dígitos ej. 01,02)
B= (ocurrencia de año bisiesto)
00 si el presente año es bisiesto.
01 si el año pasado fue bisiesto.
02 etcetera.

En todos los casos, si se oprime un número fuera de rango correcto, la información no será aceptada y se solicitará nuevamente su introducción.

Una vez que la totalidad de la información, ha sido introducida al sistema, en el monitor aparecerá el signo \$, indicando que el sistema esta en espera de que el operador oprima cualquier tecla para iniciar el conteo de tiempo en el reloj de la UAD. Este hecho tiene por objeto permitir que se introduzca una hora de ocurrencia proxima, para que al momento de su ocurrencia el operador arranque el conteo de tiempo.

Después de que la tecla se oprimió, si todo esta correcto en un intervalo de 10 segundos, aparecerá el asterisco, indicando que el sistema ha regresado a su estado MEI.

Los siguientes mensajes pueden aparecer en el caso de existir un problema:

1. NO HAY ACCESO PREVIO.
2. NO HAY RESPUESTA.

Teniendo estos mensajes el mismo significado que para los modos descritos anteriormente.

6.2.5 Modo PC (E).

Para la recuperacion de la información almacenada en los cartuchos de memoria, por medio de su lectura y transferencia a un computador personal, no se requiere ningún dispositivo adicional, ya que el equipo IPE, tiene integrado la electrónica y software necesarios, para realizar la lectura del cartucho y transferir la información hacia un computador personal, via un puerto de comunicacion serie tipo RS-232

Una vez hechas las conexiones necesarias entre el computador y el equipo IPE, en el equipo IPE se oprime la tecla **E**. La letra **E** aparecerá en el monitor durante dos segundos, y en seguida la leyenda **MODO PC** será mostrada.

En este momento, el equipo IPE, funcionará como un dispositivo periférico a el computador personal, atendiendo a instrucciones enviadas desde este último, vía el puerto de comunicación.

Cuando el IPE reciba una letra **X**, contestará con una letra **L**, lo cual podrá ser interpretado por la computadora como un mensaje de aceptación de comunicación.

Si desde la computadora se envía una letra **A**, el equipo IPE procederá a la lectura de los primeros 32 bytes de la memoria y los transmitirá a la computadora.

Cada nueva recepción de la letra **A**, causará la lectura y envío de los siguiente 32 bytes de memoria, hasta finalizar la lectura del cartucho instalado.

Cuando se desea terminar la comunicación, se envía la letra **E**, lo que causará que el equipo IPE, regrese al estado **MEI**.

La comunicación entre el equipo IPE y la computadora, es realizada a una velocidad de 1200 BAUDS.

El Instituto de Investigaciones Eléctricas, ha realizado programas computacionales apropiados para la operación de este tipo de dispositivos, desde un ambiente tutorial.

6.2.6 Mensajes de Error.

En la operación del sistema, se pueden observar algunos mensajes de error, cuya interpretación se presenta a continuación:

TECLA INVALIDA Este mensaje aparecerá cuando el operador oprima una tecla que en ese momento no es reconocida por el sistema.

NO HAY RESPUESTA Este mensaje será mostrado, cuando el equipo IPE envíe alguna orden a la UAD y no se tenga respuesta, sus causas pueden ser:

1. La interfaz de comunicación óptica no está conectada.
2. La interfaz de comunicación óptica esta incorrectamente conectada.
3. La energía principal para alimentación de la UAD, no está presente (línea medida en baja tensión). En este caso no hay respuesta

pero la UAD continúa con su operación NORMAL. El operador deberá esperar a el reestablecimiento de la energía.

4. Existe una falla en el sistema electrónico de SIARDET, que requiere mantenimiento correctivo.

REFERENCIAS

- REF1.- DESARROLLO DE UN PAQUETE COMPUTACIONAL (ALE.BAS), PARA RECUPERACION, MANEJO Y PROCESAMIENTO DE INFORMACION ADQUIRIDA POR EL EQUIPO ANEMODATA-1-IIE.
Marco Antonio Borja Díaz
Instituto de Investigaciones Electricas
IIE/10/14/2352/104/P
Enero de 1988
México.
- REF2.- PROGRAMA DE COMPUTO PARA ANALISIS DE DEMANDA EN CIRCUITOS DE DISTRIBUCION DE ENERGIA ELECTRICA Y SU CORRELACION CON ENERGIA EOLICA.
Raul Enrique Borja Díaz
Gerardo Perez Juarez
IIE/10/14/2066/105/F/A3/V1/4
Febrero de 1988
México.
- REF3.- INICIADOR DE PULSOS.
Carlos Masallach
Instituto de Investigaciones Electricas
Marzo de 1984
México.
- REF4.- TYPE D-52 PULSE GENERATOR FOR DEMAND METERING.
General Electric
GEH-2786
USA.

REF5.- MICROCONTROLLER HANDBOOK.
Intel Corporation
1985
USA.

REF6.- MEMORIES DATA BOOK
Thomson Semiconducteurs
1987
USA.

REF7.- COMPONENT HANDBOOK
National Semiconductor
1987
USA.

REF8.- LIQUID CRYSTAL DISPLAY MODULE
Seiko Instruments
Cat. No. LCM 690E
Japan.

REF9.- PRODUCT SELECTOR GUIDE
Maxim Integrated Products
1987
USA.

BIBLIOGRAFIA

- 1.- MICROPROCESSORS/MICROCOMPUTERS.
Khambata Adi J.
Jhon Wiley
USA
1982

- 2.- MICROCOMPUTER INTERFACIN
Bruce A Artwick
Prentice Hall
USA
1980

- 3.- HARDWARE/SOFTWARE DESIGN OF DIGITAL SYSTEMS
Bywater, R.E.H
Prentice Hall International
USA
1981

- 4.- MICROPROCESSORS IN INSTRUMENT AND CONTROL
Robert J. Bibbero
Wyley
USA
1977

- 5.- INTRODUCTION TO MICROCOMPUTERS AND MICROPROCESSORS
Arpad Barna
Jhon Wiley
USA
1976

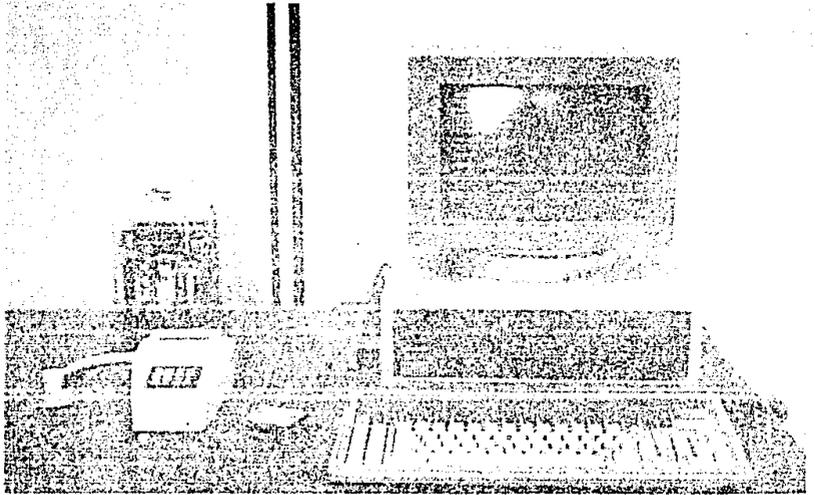
- 6.- GUIDE TO THE ENGINEERING OF MICROPROCESSOR BASED
SYSTEMS FOR INSTRUMENTATION AN CONTROL.
Engeneering Equipment Users Association
USA
1981
- 7.- EL SISTEMA DE MICROCOMPUTADORA
Peter Koehn Klaus
Siemens Aktiengesellschaft
Berlin RFA
1986
- 8.- MICROELECTRONICS: DIGITAL AND ANALOG, CIRCUITS
AND SYSTEMS.
Jacob Millman
Mc Graw Hill
USA
1978
- 9.- THE LOGICAL DESIGN OF MULTIPLE MICROPROCESSOR
SYSTEMS.
B.A. Bowen and R.J.A. Buhr
Prentice Hall
USA
1980
- 10.- DIGITAL CONTROL USING MICROPROCESSORS
Paul Katz
Prentice Hall International
USA
1981

- 11.- MICROPROCESSOR APPLICATIONS HANDBOOK
David F. Stout
Mc Graw Hill
USA
1982
- 12.- DIGITAL CIRCUITS AND MICROCOMPUTERS
Johnson, J.L. Hilburn and P.M. Julich
Prentice Hall
USA
1979
- 13.- MICROPROCESSORS THEIR DEVELOPMENT
AND APPLICATION.
Electric Research Association
Era Technology
1979
- 14.- THE ENCYCLOPEDIA OF ELECTRONIC CIRCUITS
Rudolf F. Graf
Tab Book Inc.
USA
1985
- 15.- INTEGRATED ELECTRONICS
Millman J. and Halkias, C.C.
Mc Graw Hill
USA
1972

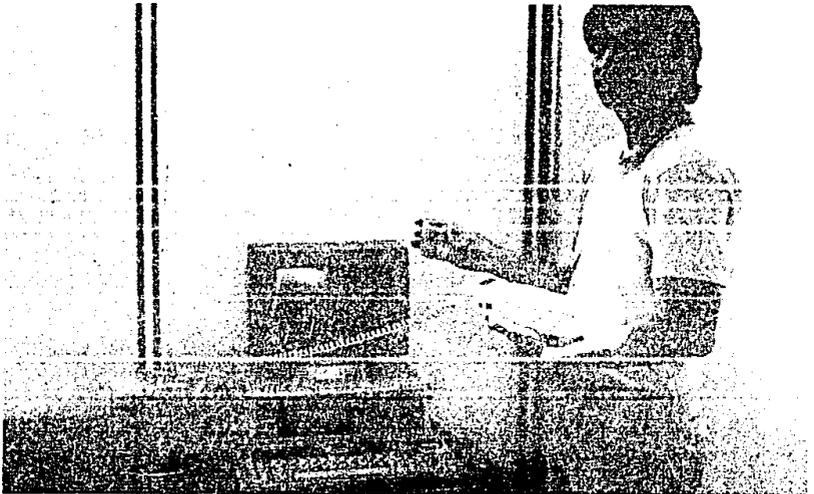
- 16.- DIGITAL SIGNAL PROCESSING: THEORY, DESIGN AND
IMPLEMENTATION
Peled, A. and Liu D.
Wiley
USA
1976
- 17.- DIGITAL INTEGRATED ELECTRONICS
Herbert Taub/ Donald Schilling
Mc Graw Hill
USA
1977
- 18.- ELECTRONICS: BJTs, FETs AND MICROCIRCUITS
E. James Angelo Jr.
Mc Graw Hill
USA
1978
- 19.- DIGITAL DESIGN
M. Morris Mano
Prentice Hall
USA
1984

ANEXO I

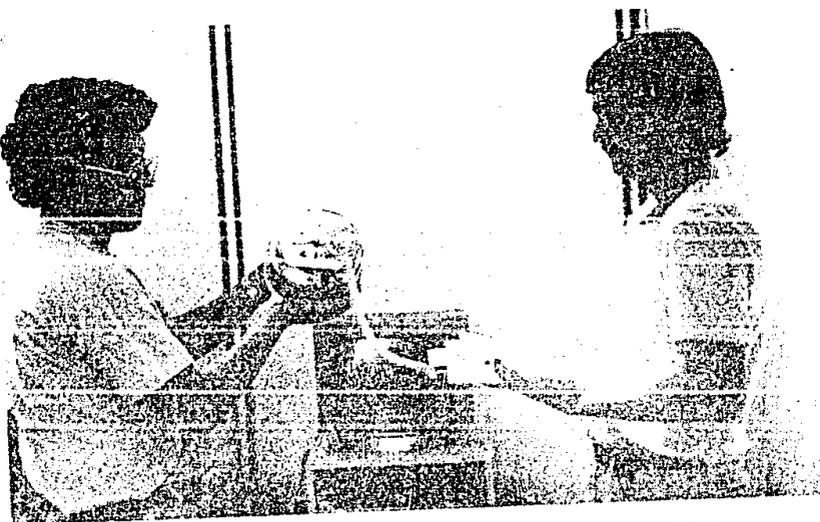
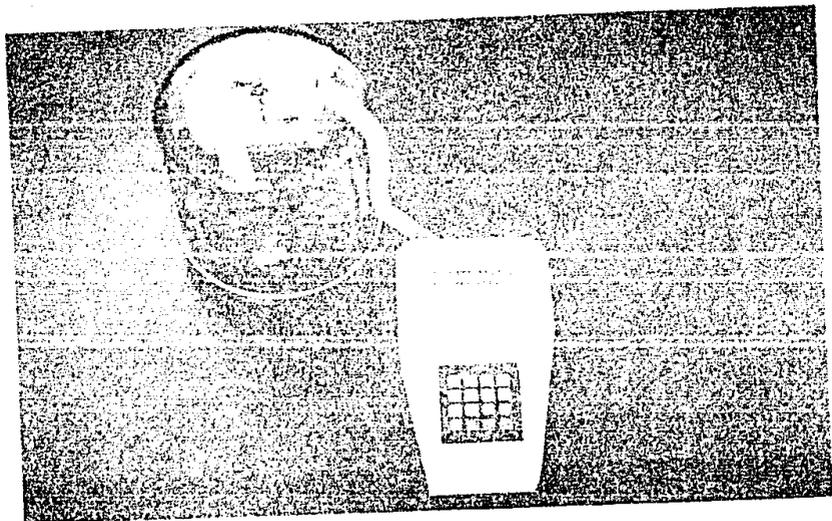
FOTOGRAFIAS



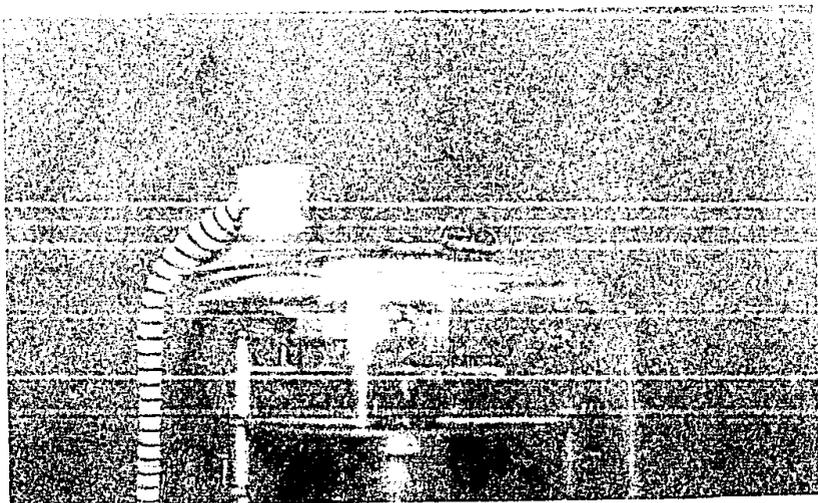
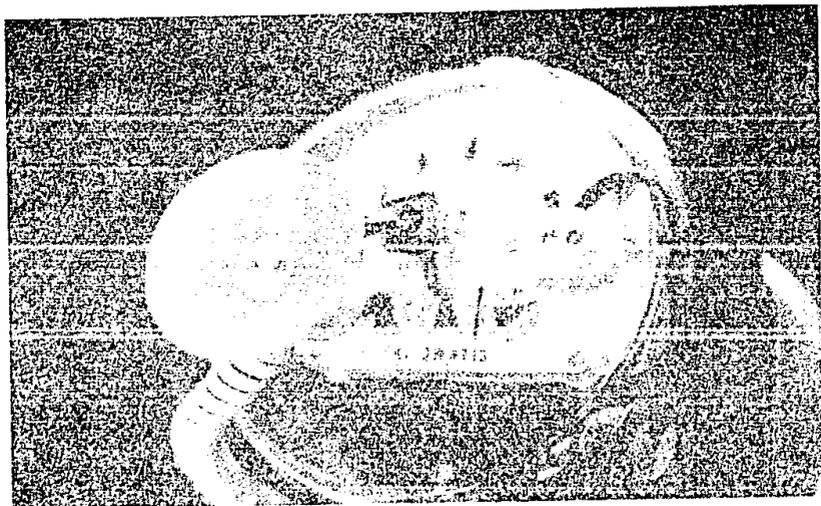
FOTOGRAFIA #1.- UNIDAD DE ADQUISICION DE DATOS INTEGRADA AL MEDIDOR CONVENCIONAL, EQUIPO DE INTERROGACION/PROGRAMACION EXTERNA, CARTUCHO DE MEMORIA Y COMPUTADOR PERSONAL.



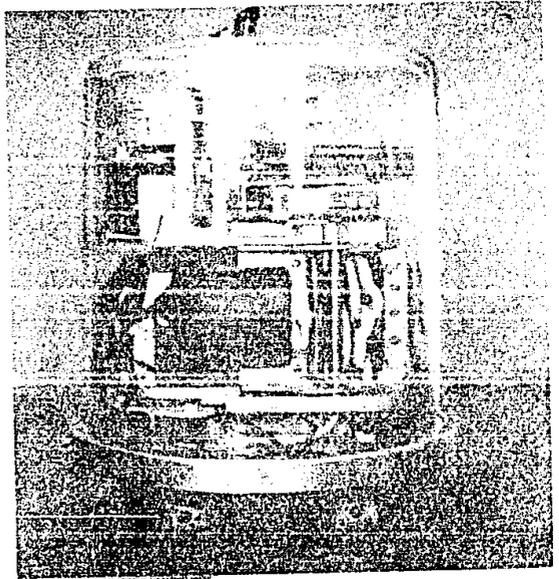
FOTOGRAFIA #2.- EL IPE, UN EQUIPO PORTATIL DE OPERACION SENCILLA.



FOTOGRAFIAS #4 y #5.- INTERCONEXION DE LA UAD Y EL
IPE, PARA RECUPERACION DE LA INFORMACION ADQUIRIDA.



FOTOGRAFIAS #6 y #7.- ACERCAMIENTO DE LA INTERFAZ DE COMUNICACION OPTICA.



FOTOGRAFÍAS #8 Y#9.- ACERCAMIENTO DE LA UNIDAD DE ADQUISICION DE DATOS INTEGRADA AL MEDIDOR CONVENCIONAL Y DEL EQUIPO DE INTERROGACION/PROGRAMACION EXTERNA.