

2  
28j



# Universidad Nacional Autónoma de México

FACULTAD DE ESTUDIOS SUPERIORES  
CUAUTITLAN

DISEÑO DE UNA MAQUINA DE CONTROL NUMERICO  
PARA LA PERFORACION DE CIRCUITOS IMPRESOS

## T E S I S

Que para obtener el Título de  
Ingeniero Mecánico Electricista  
p r e s e n t a n

*CARLOS ANGUIANO ROSILES*  
*ALVARO ESTEBAN IRETA LEON*  
*ALEJANDRO SALDAÑA ROSAS*

Director de Tesis:  
M. C. Juan Antonio Navarro Martínez

Cuautitlán Izcalli, Edo. de México.

1988

TESIS CON  
FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México



## **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

- INTRODUCCION
- OBJETIVOS
- CIRCUITOS IMPRESOS
- MICRO PROFESSOR-I
- PROGRAMACION ( SOFTWARE DEL DISEÑO )
- CIRCUITERIA ( HARDWARE DEL DISEÑO )
- DISEÑO MECANICO
- FUNCIONES DE LA MAQUINA
- CONCLUSIONES
- BIBLIOGRAFIA

## INTRODUCCION

En la actualidad, con el gran avance de la tecnología encontramos que la electrónica ha avanzado de manera impresionante en todas las diferentes áreas como son : la industria, las comunicaciones, la computación y hasta en nuestro hogar. Es increíble la cantidad de artículos de consumo que nos ayudan pensando por nosotros o recibiendo nuestras órdenes complejas, como por ejemplo, los hornos de microondas, la grabación automática de algún programa en nuestra videocassetera, las cámaras electrónicas, los contestadores telefónicos, la selección automática de cualquier estación en una radio digital además de las calculadoras y agendas electrónicas que nos permiten recordar nuestras citas.

Así pues, el funcionamiento de estos artículos como el funcionamiento de los grandes y complejos robots industriales que trabajan en las líneas de producción de alguna fábrica, se logra a través de un SISTEMA BASICO DE COMPUTADORA, este sistema básico esta compuesto en su forma más simple por:

- CPU ,unidad central de proceso comunmente llamada microprocesador; este elemento es el cerebro del sistema.
- MEMORIA ,en este elemento se escribe lo que se desea haga el sistema.

-PUERTOS SALIDA/ENTRADA ,estos Permiten el contacto del sistema con el exterior y viceversa.

- OSCILADOR ,Permite la coordinación entre los elementos del sistema.

- FUENTE DE VOLTAJE ,Permite la alimentación a todos los elementos del sistema.

En este diseño de tesis se trabaja con un sistema básico de computadora aplicado a una máquina que perfora circuitos impresos. En términos de la industria se diría que se ha diseñado una:

MAQUINA DE CONTROL NUMERICO PARA PERFORAR CIRCUITOS IMPRESOS.

## OBJETIVOS

Los objetivos de este trabajo de tesis son:

- Complementar nuestra formación académica aplicando los conocimientos adquiridos durante la carrera.
- Demostrar la funcionalidad de la Programación diseñada Para este trabajo de tesis.
- Construir físicamente la "MAQUINA DE CONTROL NUMERICO PARA LA PERFORACION DE CIRCUITOS IMPRESOS".

## CIRCUITOS IMPRESOS

- QUE ES UN CIRCUITO IMPRESO ?

- TIPOS DE CIRCUITOS IMPRESOS

Circuitos impresos de una cara.

Circuitos impresos doble cara.

Circuitos impresos multicapa.

- PROCESO DE FABRICACION Y GRABADO DE LOS  
CIRCUITOS IMPRESOS

Posibles diseños del layout.

Layout final.

Recopilación de información y

Preparación de documentación.

Preparación de herramienta.

Impresión, grabado y maquinado.

Ensamble y Prueba.

- TECNICAS DE IMPRESION

Positivo.

Negativo.

Laminación de fotoPolímero.

Serigrafía (screen Printing).

## - QUE ES UN CIRCUITO IMPRESO ?

Un circuito impreso es una Placa de aproximadamente 1/16 de Pulgada de grueso, esta Placa está hecha de material aislante (fibra de vidrio o materiales epóxicos) y tiene una capa delgada de material conductor (cobre) en una o en sus dos caras, este material conductor se ataca con elementos químicos, de tal forma que quede grabado el diseño del circuito que se desea. Al material conductor que queda grabado se le conoce con el nombre de LAYOUT.

Las funciones Principales de un circuito impreso son las de servir como soporte físico o medio de ensamble a todos los componentes electrónicos que forman el sistema y asegurar la conexión eléctrica de los componentes a través del layout.

Cabe mencionar que en este trabajo de tesis se darán definiciones sencillas acerca de los circuitos impresos, debido a que es un tema muy amplio y complejo, sin embargo si alguien desea profundizar acerca del mismo la bibliografía anexa sera una excelente herramienta.

## - TIPOS DE CIRCUITOS IMPRESOS

Existen tres grandes tipos de circuitos impresos, esta clasificación se basa en el número de capas conductoras que tenga la Placa aislante.



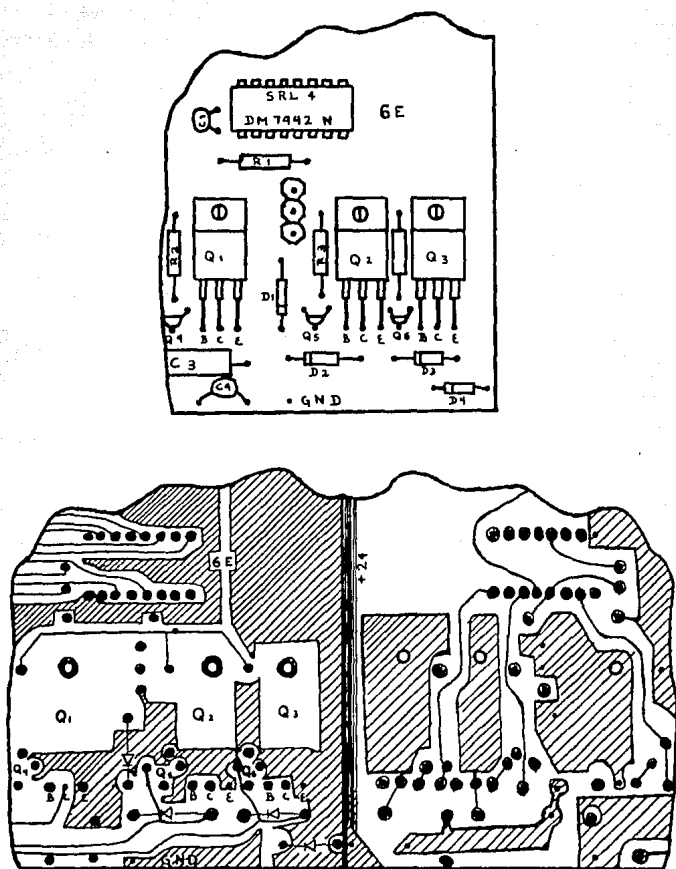


Fig.1. Circuito impresso con su layout.

- CIRCUITOS IMPRESOS DE UNA CARA. Estos circuitos son llamados así, porque sólo tienen una cara conductora, este tipo de circuito es muy usado para diseños no muy sofisticados y relativamente simples; cuando el tipo y velocidad del diseño no demandan características eléctricas especiales.

- CIRCUITOS IMPRESOS DOBLE CARA. Estos circuitos son llamados así, porque tienen dos caras conductoras y su uso es en sistemas que requieren mayor densidad de elementos electrónicos. En este tipo de circuitos aparece la necesidad de interconectar las dos caras conductoras cada vez que el diseño del circuito así lo requiera, esta interconexión se logra a través de una perforación que debe contener material conductor que interconecte eléctricamente ambas caras, a este procedimiento de interconexión de placas en los circuitos impresos de doble cara, se le conoce con el nombre de TROUGH-HOLE ("a través de la perforación").

Los diferentes procesos de trough-hole pueden ser generalizados en trough-hole por grabado y trough-hole mecánico.

TROUGH-HOLE POR GRABADO. Existen dos variantes en esta técnica de trough-hole; el proceso aditivo y el proceso sustractivo. En el primero se parte desde la placa aislante a la cual se aplica un adhesivo, y después de colocar el

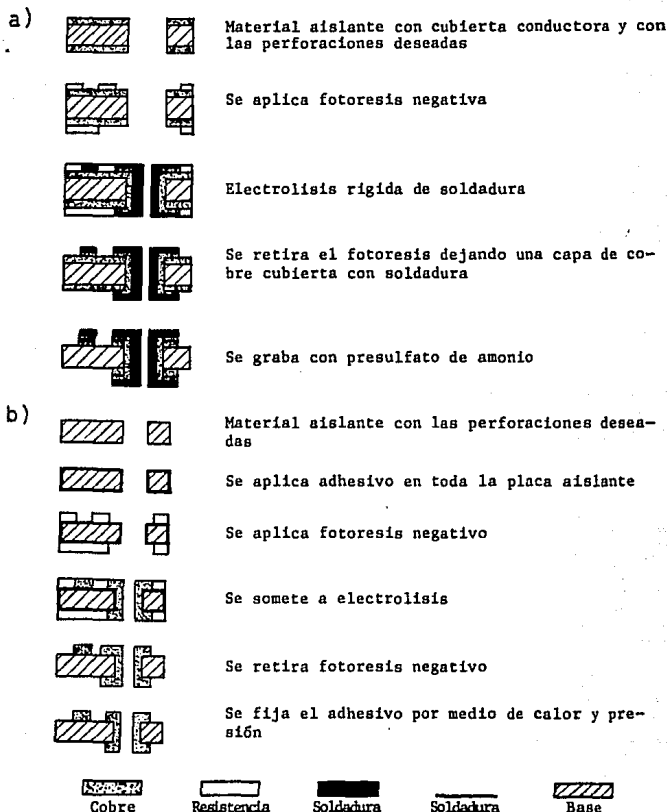


Fig.2. Procesos de through-hole

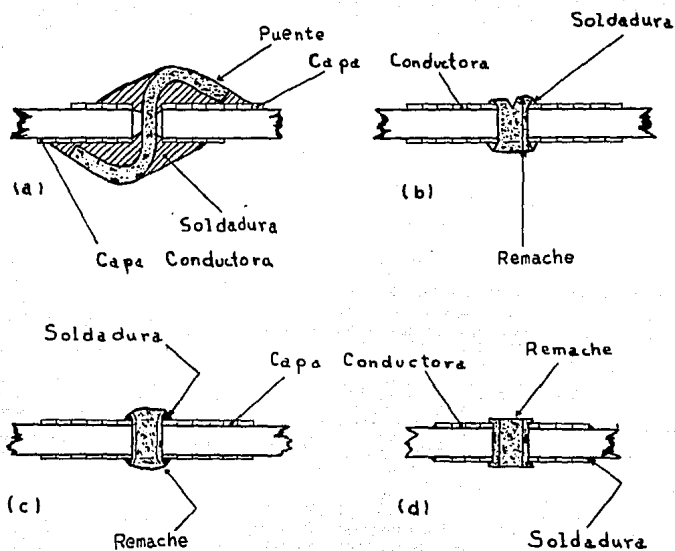
a) Proceso sustractivo

b) Proceso aditivo

fotore sist negativo se somete a electrolisis y finalmente se fija con calor y Presión, en el Proceso sustractivo se trabaja como un C.I. de una cara, sólo que después del grabado se somete al layout a electrolisis y a un baño de soldadura. En ambos casos las Placas deberan estar perforadas.

TROUGH HOLE MECANICO. En esta técnica de trough-hole la interconexión eléctrica entre ambas caras conductoras se lleva a cabo por elementos mecánicos, como puede ser un puente de material conductor que pasa a través de la perforación o por medio de pequeños tubos del diámetro de la perforación y con una altura un poco mayor al grueso de la placa del circuito impreso (estos tubos son muy similares a pequeños remaches). Ver figura # 3.

- CIRCUITOS IMPRESOS MULTICARA. Los circuitos impresos multicara estan formados por más de dos capas de material conductor, estos circuitos aparecen como una solución al problema de interconexión de componentes electrónicos miniatura en sistemas complejos y sus principales usos son en sistemas donde se desea hacer una reducción en peso y volumen, en sistemas que usan circuitos integrados y que tienen problemas en la distribución y disipación de calor, además pueden ser usados en sistemas con necesidad de un plano permanente de tierra.



## Vista Lateral.

Fig.3. Interconexiones mecánicas:

- a) Puente soldado
- b) Remache tabilla de embudo realizado
- c) Remache de embudo realizado
- d) remache fundido

## - PROCESO DE FABRICACION Y GRABADO DE LOS CIRCUITOS IMPRESOS

Durante el capítulo se ha descrito que es un circuito impreso y sus diferentes tipos y aplicaciones, ahora hablar de su fabricación resultaría todo un estudio muy vasto y es que hay que admitir que dentro de nuestra economía no está el proceso de fabricación que además debe cubrir los estándares y las normas que se aplican en la industria de los circuitos impresos a través del INSTITUTO DE CIRCUITOS IMPRESOS (Institute of Printed Circuits). Generalmente si se desea llevar a cabo un diseño que requiera de un circuito impreso, se compra la placa de circuito impreso (sin grabar) y se diseña el layout y se planean todos los procesos que se requieren tanto para el grabado como para el maquinado del circuito impreso. Dentro de esta planeación entra este proyecto de tesis al intentar reducir los tiempos de perforación y mejorar la calidad de la misma.

Así como consideramos un estudio muy vasto el describir los procesos de fabricación de las placas de circuitos impresos, así también consideramos importante para cualquier estudiante de ingeniería el que conozca al menos, el diagrama de bloques del proceso de grabado de los circuitos impresos, este diagrama es mostrado por la figura #4.

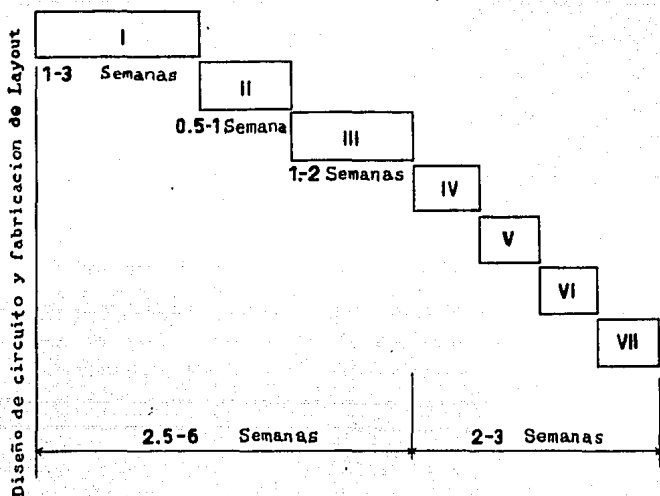


Fig.4. Ciclo de diseño y fabricación de un layout.

Las técnicas automáticas pueden reducir este ciclo en un 50% o más.

I- Probables diseños del layout.

II- Layout final.

III-Recopilación de información y preparación de documentación.

IV- Requerimientos de herramienta.

V- Impresión y grabado.

VI- Maquinado.

VII-Ensamble y Prueba.

A continuación se describen cada uno de los bloques que forman el Proceso de grabado de los circuitos impresos.

- PROBABLES DISEÑOS DEL LAYOUT. El primer Paso en el Proceso de grabado es obtener el diseño de lo que se va a grabar (layout). Este diseño debe asegurar la correcta interconexión de los elementos que formarán el circuito, un buen diseño reducirá espacio y puentes eléctricos; también, dentro de las consideraciones de diseño se deberán tomar en cuenta las condiciones eléctricas, mecánicas y de medio ambiente en las que trabajará el circuito; también se deberán tomar en cuenta las necesidades de mantenimiento del equipo en el que trabajará el circuito impreso y por supuesto se deberán considerar los costos del diseño.

- LAYOUT FINAL. En este Paso se combinan o seleccionan los diferentes diseños del layout hasta lograr el layout definitivo. Se debe tener mucho cuidado en que no existan errores en el layout final, pues estos serían repetitivos.

- RECOPIACION DE INFORMACION Y PREPARACION DE DOCUMENTACION. Este Paso es el más importante, si se toma en cuenta que la fabricación de circuitos impresos es una industria mundial que está regida por normas y estándares



internacionales. Además, la buena documentación permitirá la detección oportuna de errores (que en producción en serie suelen ser muy costosos) y una correcta documentación también facilitará la reproducción y/o modificación de cualquier layout previamente diseñado.

- ENSAMBLE Y PRUEBA. Este último paso del proceso dependerá en su duración, de la automatización con la que se cuenta. Con este paso concluye el proceso, logrando aquí, el circuito impreso con sus componentes electrónicos ya montados. Después de esto, se deberá proceder a las pruebas necesarias para dar al circuito impreso la confiabilidad requerida.

## - TECNICAS DE IMPRESION

A continuación se mencionan los aspectos más importantes de las diferentes técnicas de impresión utilizadas en los circuitos impresos. El Paso inicial de cualquier técnica de grabado, es la generación de un modelo de diseño o film fotográfico, que debiera ser impreso en la cubierta de cobre de la Placa del circuito.

- Positivo. En esta técnica llamada también impresión y grabado (Print & etch) la Placa del circuito impreso es expuesta al Positivo del layout, esta exposición se hace después de haber cubierto la Placa con un material fotoresistivo. El material fotoresistivo se vuelve soluble en las áreas que son expuestas a la luz, después se limpia con una solución especial y se ataca con sustancias como el cloruro férrico, de tal forma que, en las partes donde se disolvió el fotoresist el cobre es disuelto por el cloruro férrico. Siempre que se lleve a cabo este método y en general cualquiera que involucre sustancias fotoresistivas, se debe ser cuidadoso en cuanto a las instrucciones de uso y manejo, especialmente en lo relativo a su exposición a la luz.

- Negativo. Esta técnica es muy similar a la técnica de Positivo, con la diferencia como su nombre lo indica que aquí usamos el negativo de la imagen a grabar (layout). También el fotoreactivo que se debe usar en esta técnica debe tener la característica de endurecer al ser expuesto a la luz, de tal forma que la parte endurecida es la correspondiente a las rutas del layout, la parte endurecida es resistente al cloruro férrico. Tanto con la técnica de Positivo como con la de negativo obtenemos el mismo resultado con algunas variantes en el desarrollo de la técnica y de los elementos (film positivo/negativo y tipo de fotoreactivo).

Hay que mencionar que existen dos tipos fundamentales de material fotoreactivo, cuyas características están absolutamente ligadas al tipo de técnica que se emplee (Positivo o Negativo).

Para aquellas personas que deseen profundizar en cuanto a estas u otras técnicas de impresión, la bibliografía aquí mencionada es un buen punto de partida para comenzar a llevarlas a la práctica, sin olvidar que sólo la experiencia misma sumada a la investigación nos dará óptimos resultados.

Tanto en la técnica positiva como en la técnica negativa, la manera recomendada de aplicación de material fotoresistivo es por medio de spray, más aún si se desea una producción en serie de grandes volúmenes. De otra forma se deberán tomar en cuenta aspectos económicos relacionados con los costos de la maquinaria y de los materiales.

- Laminación de fotopolímero. En esta técnica se unen, por medio de una máquina, la placa de circuito impreso con una capa de film fotoresistivo (dry film fotoresist), este film fotoresistivo tiene un par de cubiertas (algo similar a un sandwich), una de esas cubiertas es retirada por la máquina antes de unirse a la placa de circuito impreso, la unión de la placa de C.I. (circuito impreso) con el film fotoresistivo es como pegar una calcomanía a la placa de C.I. y la otra cubierta del film fotoresistivo es retirada antes de exponer la placa al proceso de impresión.

La capa de film fotoresistivo es aplicada a la placa de C.I. por medio de calor y presión y esto debe hacerse a través de la máquina laminadora de fotopolímero. La figura #5 muestra un corte transversal del film fotoresistivo y un esquema de la máquina laminadora de fotopolímero.

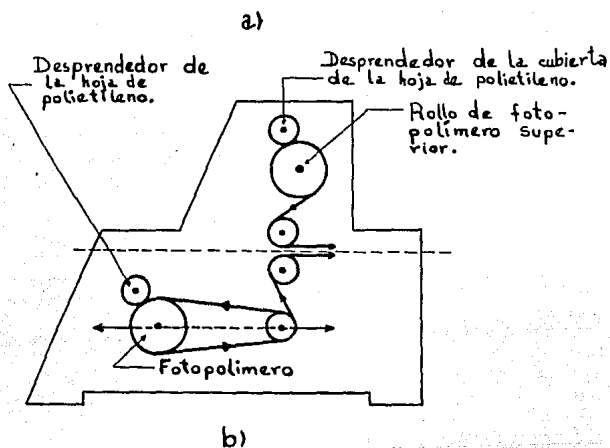
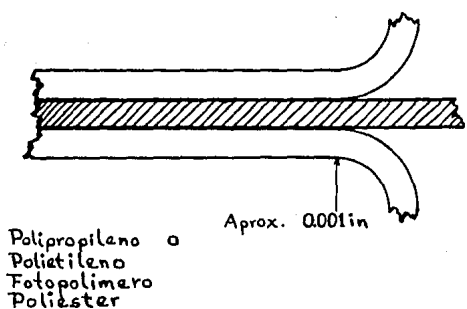


Fig.5. a) Corte transversal del film fotoresistivo (dry-film).  
b) Esquema de la máquina laminadora de fotopolimero.

- Serigrafía (screen printing). Por ser ésta, una de las técnicas de más uso debido a su bajo costo y a su utilidad en la elaboración de circuitos impresos no muy complicados, se describirán a continuación los Pasos más elementales de la misma, con el fin de dar una idea general. Nuevamente se hace hincapié, en que la Práctica de cualquier técnica será lo que nos permita avanzar en nuestras investigaciones enfocadas a mejorar la calidad de impresión.

Como en las técnicas anteriores aquí también se necesita del Positivo del diseño (layout), en esta técnica se usará un bastidor con una malla y todo el equipo y material relacionado con la impresión serigráfica.

A continuación se listan los Pasos más importantes de esta técnica.

-se cubre la malla con emulsión serigráfica (un equivalente a fotoresist negativo) y se Pasa el rasero (instrumento parecido a una espátula) que evita los excesos y además permite una correcta distribución de la emulsión en la malla.

-se coloca el negativo sobre la malla y en cuarto oscuro se expone a la luz.

-se limpia la malla con agua corriente a presión o con el solvente recomendado. Después de estar limpia la malla, ésta quedará lista para la impresión.

-en el proceso de impresión una vez que ya se tiene lista la malla, se coloca debajo de ésta la placa de circuito impreso y se cubre con una tinta resistente al material de grabado (generalmente cloruro férrico).

-se pasa el rasero para tener una correcta distribución de la tinta, después de esto la placa quedará lista para el proceso de grabado.

Debido al uso de un marco con malla esta técnica en inglés es llamada Screen Printing, cuya traducción al español es Impresión en Malla.

# MICRO-PROFESSOR I

- INTRODUCCION
- CARACTERISTICAS DEL HARDWARE
- CARACTERISTICAS DEL SOFTWARE
- RELACION DEL MICROPROFESSOR I CON ESTE PROYECTO DE TESIS



## - INTRODUCCION

Como se mencionó anteriormente, en este Proyecto de tesis se usa un sistema básico de computadora para controlar la máquina Perforadora de circuitos impresos, con esto se obtiene una máquina de control numérico.

El sistema básico de computadora que se utiliza es un kit educacional con nombre de Micro-Professor I (este kit es una pequeña computadora) que puede ser programado en lenguaje de máquina ENSAMBLADOR Z-80.

En este capítulo no se dará toda la descripción del hardware y software del Micro-Professor I, solamente se describirán los elementos de hardware y software que consideramos de importancia para el buen entendimiento de este trabajo de tesis. Nota: Para cualquier tema que se desee profundizar acudir a la bibliografía.

## - CARACTERISTICAS DEL HARDWARE

El kit Micro-Professor I es un sistema básico de computadora lanzado al mercado por Multitech Electronics, es un kit educacional, porque viene acompañado de manuales de funcionamiento de algunos elementos electrónicos que forman el sistema básico de computadora.

A continuación, se listan las especificaciones de hardware del Micro-Professor I. (la figura #6 nos muestra un esquema del kit)

- CPU (Central Processing Unit)

- Zilog Z-80 CPU con 158 instrucciones y trabajando en este kit a una frecuencia de reloj de 1.79 MHz, sin embargo el Z-80A CPU puede trabajar hasta una frecuencia de 4 MHz.

- ROM (Read Only Memory) Memoria de lectura únicamente con voltaje único de trabajo de +5V. Memoria EPROM 2516 2K de memoria, esta memoria se direcciona de 0000-07FF y contiene al Programa monitor del kit.

- RAM (Random Access Memory) Memoria de acceso aleatorio estática 6116 capacidad total de memoria 2K, esta memoria se direcciona de 1800-1FFF, esta área de memoria es para el Programa usuario.

- Área de expansión de memoria. En esta área se puede usar alguna de las siguientes memorias EPROM 2516, 2716, 2532, 2732, o memorias RAM 6116, la dirección para esta expansión de memoria es de 2000-2FFF. (Para cualquier modificación se deberá consultar antes el manual del usuario ya que es necesario hacer cortes y puentes al cambiar de memoria.

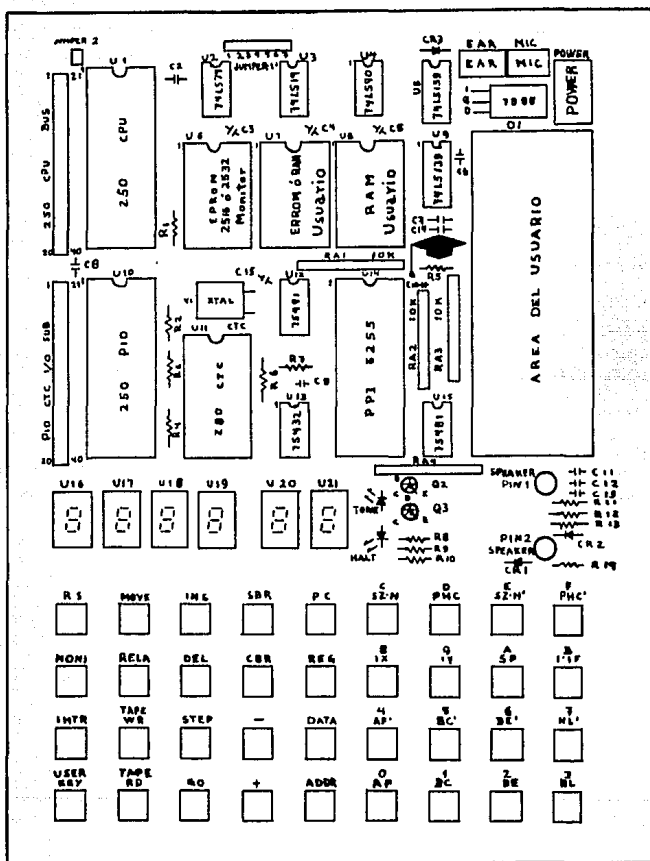


Fig.6. Esquema del Micro-Professor I de Multitech Electronics.

- I/O PORT (Puertos Entrada/Salida). Este Puerto Programable 8255 es Para uso exclusivo de las funciones del sistema básico de computadora, en este caso es usado tanto Para el teclado como Para el display de 7 segmentos, este Puerto tiene un total de 24 líneas Paralelas de entrada/salida.

- I/O ADDRESS (Direcciones de Entrada/Salida) Estas direcciones corresponden al Puerto Programable Entrada/Salida PIO, con un total de 16 líneas Paralelas de entrada/salida, las direcciones de estos Puertos son de 80H a 83H.

También este kit cuenta con la posibilidad de usar un circuito CTC (Counter Timer Clock), este elemento es un dispositivo de la familia Z-80 que nos Permite el conteo de eventos y si así se desea estos eventos pueden generar interrupciones al Programa a través del CPU.

- DISPLAY Este kit cuenta con un display de 6 dígitos, cada uno de los cuales está formado Por 7 segmentos, el display es rojo de 0.5".

- TECLADO El teclado está formado Por 36 teclas que incluyen 19 teclas de función y 16 teclas alfanúmericas Para el código hexadecimal, además de una tecla de usuario.

- BOCINA y Circuitos Controladores de Bocina Este kit incluye una bocina de 2.25" de diámetro que es usada para las subrutinas de tono y también para cargado de y en cinta además de otras posibilidades de uso.

- AREA DE USUARIO Esta área de 3.5" por 1.36" nos permite alambrear cualquier circuito que sea necesario para alguna función específica del kit

- INTERFACE DE CINTA DE AUDIO Esta interface puede ser conectada a cualquier grabadora reproductora de cintas con el fin de salvar o cargar los programas, la velocidad de lectura y cargado es de 165 bps (bits por segundo)

- SEÑAL DE RELOJ Este kit utiliza en el sistema de reloj un cristal de cuarzo que oscila a 3.58 MHz, esta frecuencia es dividida entre dos para lograr la frecuencia real de trabajo del sistema que es de 1.79 MHz, la duración de cada ciclo de la señal de reloj es de 0.56 micro segundos.

- CONSUMO DE ENERGIA Este sistema básico de computadora trabaja con un voltaje único de +5V y consume aproximadamente 500 mA.

- ENTRADA PRINCIPAL DE ENERGIA Este kit tiene su principal alimentación de energía a través de un adaptador que

convierte la corriente alterna 110V a corriente directa 9V/500mA y además el kit está provisto de un regulador de voltaje a +5V.

- CARACTERISTICAS FISICAS DEL MICRO-PROFESSOR I

Ancho	15.75 cm	sin cubierta
Largo	22.33 cm	sin cubierta
Espesor	1.60 cm	sin cubierta
Peso	1.41 lb	

## - CARACTERISTICAS DEL SOFTWARE

El Micro-Professor contiene un Programa de alta funcionalidad que está desarrollado en 2k de memoria. Este Programa monitor está diseñado para responder a las entradas de usuario y mantener siempre un estado transparente al usuario. El Programa monitor comienza su ejecución cuando es conectado el kit. A continuación se listan las diferentes funciones del teclado del Micro-Professor I.

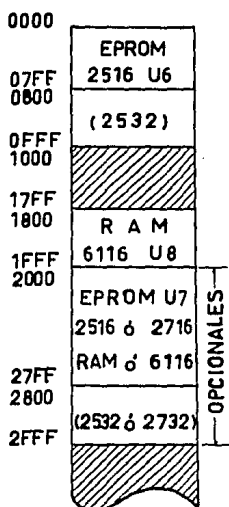
RS :Reset del sistema  
ADDR :Modo de dirección de memoria  
REG :Modo de nombre de registro  
DATA :Permite introducir un dato tanto a memoria como a registro  
PC :LLama al contador de Programa  
+ :Incrementa la dirección de memoria o el registro de uno en uno  
- :Decrementa la dirección de memoria o registro de uno en uno  
STEP :Permite correr el Programa del usuario Paso a Paso  
SBR :Para colocar altos en la ejecución del Programa usuario  
CBR :Para retirar altos del Programa usuario

MONI :Termina la ejecución del Programa  
 usuario y regresa al Programa monitor  
 GO :Comienza la ejecución del Programa en la  
 dirección mostrada en el display  
 INS :Para insertar un byte en memoria  
 DEL :Para borrar un byte de la memoria  
 MOVE :Para mover un bloque de memoria de una  
 área a otra  
 RELA :Para cálculos de distancias relativas  
 TAPE WR :Para escribir datos de memoria a una  
 cinta de audio  
 TAPE RD :Para leer datos de una cinta de audio a  
 memoria  
 INTR : Interrupción mascarable, conectada  
 directamente a la Pata INT del CPU  
 USER KEY:Tecla de uso definido conectada al  
 Puerto de entrada 00, bit 6  
 0 a F : Teclas correspondientes al código  
 hexadecimal

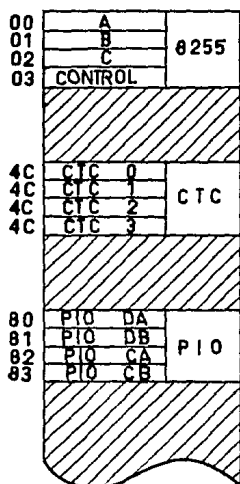
En la figura #7 se muestran los mapas de memoria y de  
 direcciones de los puertos usados por el micro-Professor I.

A continuación se describen las subrutinas más  
 importantes del Micro-Professor I. Estas subrutinas están  
 contenidas en el Programa monitor.





a)



b)

Fig.7. a) Mapa de memorias del MicroProfessor I.

b) Direcciones de Puertos del MicroProfessor I.

DIRECCION	NEMONICO	FUNCION
0624	SCAN1	Buscar en el teclado y colocar en el display la información durante un ciclo.
05FE	SCAN	Buscar en el teclado y colocar en el display la información hasta que una tecla sea pisada.
0689	HEX7	Convertir un dígito hexadecimal en su correspondiente 7 segmentos Para Poder mostrarlo en el display.
0678	HEX7S0	Convertir dos dígitos hexadecimales en sus correspondientes 7 segmentos Para Poder mostrarlos en el display.
05F6	RAMCHK	Comprueba que la dirección de memoria dada sea RAM.
05E4	TONE	Generar un sonido con frecuencia y duración determinada Por el usuario.
05DE	TONE1K	Generar sonido a 1 KHz.

05E2            TONE2K    Generar sonido a 2 KHz.

Las subrutinas que se describen a continuación son software de apoyo a este trabajo de tesis.

- SCAN1

Dirección: 0624

Función : Buscar en el teclado y colocar en el display la información durante un ciclo, esto lo hace de izquierda a derecha. Su tiempo de ejecución es de 9.97 ms.

Entrada : Por el registro IX que sirve como buffer del display.

Salida : Si no hay tecla pisada la bandera de carry=1, si hay tecla pisada la bandera de carry=0 y el código de posición de la tecla pisada en el reg A.

Registros: Destruye AF, A'F', B'C', D'E'

- SCAN

Dirección: 05FE

Función : Similar a SCAN1 excepto que: SCAN1 sólo trabaja durante un ciclo y SCAN lo hace hasta que una nueva tecla es pisada. Además SCAN1 regresa con el código de posición de la tecla pisada mientras SCAN regresa con el código interno.

Entrada : Registro IX como buffer del display.

Salida : Registro A conteniendo el código interno de la tecla pisada.

Registros: Destruye AF, B, HL, AF', BC', DE'.

- HEX7

Dirección: 0609

Función : Convertir un número hexadecimal en su equivalente 7 segmentos Para Poder mostrarlo en el display.

Entrada : El número hexadecimal (0-F) que se desea convertir deberá estar en los 4 bits menos significativos del registro A.

Salida : El resultado es también guardado en el registro A.

Registros: Destruye al registro Par AF unicamente.

- HEX750

Dirección: 0670

Función : Convertir dos números hexadecimales en su equivalente 7 segmentos Para Poder mostrarlos en el display.

Entrada : El Primer número es guardado en la parte derecha del reg. A. El segundo número es guardado en la parte izquierda del reg. A.

Salida : El equivalente del Primer número es guardado en la dirección apuntada por HL, el segundo equivalente es guardado en (HL+1), el reg. HL es incrementado por 2.

Registros: Destruye AF, HL.

- RAMCHK

Dirección: 05F6

Función : Comprueba que la dirección de memoria dada esté en el área de memoria RAM.

Entrada : La dirección a comprobar deberá estar en el registro HL.

Salida : Si la dirección de memoria está en el área RAM entonces la bandera de Zero=1, en caso contrario la bandera de Zero=0.

Registros: Destruye al registro AF.

#### - TONE

Dirección: 05E4

Función : Generar un sonido

Entrada : El registro C controla la frecuencia del sonido. El período es aproximadamente  $(44 \times C \times 13) \times 2 \times 0.56$  micro-seg y la frecuencia es  $200 / (10 + 3 \times C)$  KHz. El registro HL contiene el número de ciclos (el máximo valor es 32768).

Salida : Ninguna

Registros: Destruye AF, B, DE, HL.

#### - TONE1K

Dirección: 05DE

Función : Generar un sonido de 1 KHz.

Entrada : El número de períodos en el reg. HL.

Salida : Ninguna

Registros: Destruye AF, BC, DE, HL.

- TONE2K

Dirección: 05E2

Función : Generar un sonido de 2KHz

Entrada : El número de Períodos en el reg. HL.

Salida : Ninguna.

Registros: Destruye AF, BC, HL, DE.

## - RELACION DEL MICRO-PROFESSOR I CON ESTE PROYECTO

Durante este capítulo se han descrito las diferentes características, tanto de hardware como de software, del Micro-Professor I, y en esta sección se menciona la relación que el sistema básico de computadora guarda con este proyecto.

Para facilitar la comprensión de la relación que guarda el kit MPFI con este diseño, es bueno definir algunas funciones de operación que se desean para esta máquina de control numérico:

Primero - La máquina de control numérico funcionará como una máquina mecánica capaz de realizar movimientos en los ejes coordenados X e Y, y en el eje Z para llevar a cabo la perforación de circuitos impresos, dichas perforaciones serán previamente programadas por el usuario. El medio a través del cual se programará y controlará el movimiento en estos ejes será el kit Micro-Professor I, entonces se define a la máquina de control numérico para perforar circuitos impresos como la conjunción mecánica, eléctrica y electrónica de los diferentes elementos que le permiten realizar su función.

Segundo - Las funciones básicas, que se desea lleve a cabo una máquina de CN son:

a) Permitir el almacenaje en memoria de las rutas a Perforar.

b) Realizar la ejecución de las rutas a Perforar con un mínimo de error

c) Permitir el almacenaje en cinta de las diferentes rutas a utilizar .

d) Poder almacenar en memoria las rutas que deseemos Perforar con tan sólo leerlas de cinta

Así PUES, podemos ya notar la relación tan estrecha que guardan todos los elementos que forman la máquina de CN. (Control Numérico). Por ejemplo; La memoria RAM y el Programa monitor del MPFI hacen posible que la función a) se lleve a cabo, Para b) el control de movimiento en los diferentes ejes se logra con el Puerto PIO y dos motores de Pasos apoyados en los Programas usuario y monitor, Las funciones c) y d), son posibles debido al hardware y software del kit MPFI.

La relación arriba descrita entre este Proyecto y el MPFI podrá parecer muy vaga, Pero el objetivo es Presentar de la manera más clara posible, dicha relación con el fin de no perder al lector en detalles técnicos que por el momento sólo lograrían confundirlo.



# PROGRAMACION

## < SOFTWARE DEL DISEÑO >

-LISTADO DEL PROGRAMA MONITOR DEL MICRO-PROFESSOR I

-PROGRAMACION DE ESTE PROYECTO

Estructura de la Programacion

Programa Principal

Programa METE

Programa SACR

Programa SALE

Programa DATS

-LISTADO DE LA PROGRAMACION

## -LISTADO DEL PROGRAMA MONITOR DEL MICRO-PROFESSOR I

En este capítulo daremos la descripción del software del diseño, sin embargo parte del mismo se apoya en algunas subrutinas del kit; que han quedado descritas en el capítulo anterior (Micro-Professor I).

A continuación se presenta el listado completo del Programa monitor del MPF-I. Este listado se presenta tal como aparece en el manual del usuario del kit Micro-Professor I.

LOC OBJ CODE M SMT SOURCE STATEMENT

```

1 .....
2 .....
3 * * * * *
4 * * * * *
5 * * * * *
6 * * * * *
7 * * * * *
8 * * * * *
9 .....
10 .....
11 .....
12 .....
13 .....
14 .....
15 P255 EQU 03H ;$255 I control port
16 DIGIT EQU 02H ;$255 I port C
17 SEQ7 EQU 01H ;$255 I port B
18 KIN EQU 00H ;$255 I port A
19 PWCODE EQU 0ASH ;Power-up code
20 ISUM EQU 71H ;This will make the sum of all
21 ;monitor codes to be zero.
22 .....
23 ; The following EQUATEs are used for timing. Their values
24 ; depend on the CPU clock frequency. (In this version, the
25 ; crystal frequency is 1.79 MHz.)
26 .....
27 COLDEL EQU 201 ;Column delay time for routine
28 ;SCAN and SCAN1.
29 F1KHZ EQU 65 ;Delay count for 1K Hz square wave,
30 ;used by routine TONE1K.
31 F2KHZ EQU 31 ;Delay count for 2K Hz square wave,
32 ;used by routine TONE2K.
33 MPERIOD EQU 42 ;1K Hz and 2K Hz threshold, used by
34 ;tape input routine PERIOD.
35 .....
36 ; The following EQUATEs are for tape modulation.
37 ; If the quality of tape recorder is good, the user may
38 ; change '4 4 2 8' to '2 2 1 4'. This will double
39 ; the tape data rate.
40 ; If the quality of tape recorder is poor, the user may
41 ; change '4 4 2 8' to '8 8 3 12'. This will improve
42 ; error performance but slow down the data rate.
43 ; Although the data format is changed, the tape is still
44 ; compatible in each case, because only the ratio is
45 ; detected in the Tape-read.
46 .....
47 ONE_1K EQU 4
48 ONE_2K EQU 4
49 ZERO_1K EQU 2
50 ZERO_2K EQU 8
51 .....
52 .....
53 ; I/O port assignment: ($255 I)
54 .....
55 ; port A (address 00H):
56 ; bit 7 -- tape input
57 ; bit 6 -- 'USER KEY' on keyboard, active low
58 ; bit 5-0 row of keyboard matrix input, active low

```

MPP-1  
 LOC OBJ CODE M STMT SOURCE STATEMENT.

```

59 : port B (address 01H): 7 segments of LED, active high
60 : bit 7 -- segment d
61 : bit 6 -- decimal point
62 : bit 5 -- segment c
63 : bit 4 -- segment b
64 : bit 3 -- segment a
65 : bit 2 -- segment f
66 : bit 1 -- segment g
67 : bit 0 -- segment e
68 : port C (address 02H):
69 : bit 7 -- tape & tone output
70 : bit 6 -- BREAK enable. NMI (CPU pin 17) will go to
71 : low 5 M1's (machine cycle one) after this
72 : bit goes to low. (This bit is connected to
73 : the reset input of external counter.)
74 : bit 5 -- columns of keyboard and display matrix,
75 : active high. Bit 5 is the leftmost column.
76 :
77 : ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
78 : -- reset --
79 : There are two cases that will generate a RESET signal:
80 : (i) power-up
81 : (ii) 'RS' key pressed
82 : In both cases, the follow actions will be taken:
83 : a) disable interrupt, set interrupt mode to 0
84 : set I register to 00 and start execution
85 : at address 0000 (by Z80 CPU itself).
86 : b) initial user's PC to the lowest RAM address;
87 : c) set user's SP to 1F9FH;
88 : d) set user's I register to 00 and disable user's
89 : interrupt flip-flop;
90 : In addition, subroutine INI will be called on power-up
91 : reset, which has the following effects:
92 : e) disable BREAK POINT;
93 : f) set the contents of location 1F9FH 1F9FH to 86 and
94 : and 00 respectively. This will make instruction RST
95 : 38H (opcode FF) have the same effect as BREAK.
96 : Memory location POWERUP is used to distinguish power-up
97 : from RS-key. (POWERUP) contains a random data when
98 : power-up and contains PWCODS (0ASH) thereafter.
99 :
0000 0600 100 LD B,0
0002 10FE 101 DJNZ $ ;Power-up delay
102
103 ; Initial 8255 to mode 0 with port A input, port B and C
104 ; output. The control word is 90H.
105
0004 3E90 106 LD A,10010000H
0006 D303 107 OUT (PB255),A
108
109 ; When the control word is sent out to 8255, all output
110 ; ports are cleared to 0. It is necessary to disable
111 ; BREAK and deactivate all I/O by sending 0C0H to
112 ; port C.
113
0008 3E00 114 LD A,0C0H
000A D302 115 OUT (D1GIT),A
000C 31A1F 116 LD SP,SYSSK ;initial system stack

```

MPF-I

LOC	OBJ CODE	M	STMT	SOURCE	STATEMENT
			117		
			118		; If the content of location POWERUP is not equal to
			119		; PPCODE, call subroutine INI. Continue otherwise.
			120		
000F	3AE51F		121	LD	A, (POWERUP)
0012	FEA5		122	CP	PPCODE
0014	C4C103		123	CALL	NZ, INI
			124		
			125		; Determine the lowest RAM address by checking whether
			126		; address 1000H is RAM. If yes, set user's PC to this
			127		; value. Otherwise, set it to 1800H.
			128		
0017	210010		129	LD	HL, 1000H
001A	CD605		130	CALL	RAMCHK
001D	2802		131	JR	Z, PREPC
001F	2818		132	LD	H, 18H
0021	22DC1F		133	PREPC	LD (USERPC), HL
0024	2600		134	LD	H, 0
			135		
			136		; Address 28H and 30H are reserved for BREAK (RST 28H)
			137		; and software BREAK (RST 30H). Skip these areas, monitor
			138		; program resumes at RESET1.
			139		
0026	180A		140	JR	RESET1
			141		
			142		;
0028			143	RST28	ORG 28H
			144		; Address 28H is the entry point of BREAK trap.
			145		; If a location is set as a BREAK point, the monitor
			146		; will change the content of this location to C7 (RST 28H)
			147		; before transferring control to user's program.
			148		; In execution of user's program, a trap will occur if
			149		; user's PC passes this location. The monitor then takes
			150		; over control and the content of BREAK address
			151		; will be restored. Monitor takes care of everything
			152		; and makes the whole mechanism transparent to the user.
			153		; The return address pushed onto stack is the PC after
			154		; executing RST 28H. The original break address should
			155		; be one less than that. The following 3 instructions
			156		; decrease the content of (SP) by one without changing
			157		; HL.
			158		
0028	E3		159	EX	(SP), HL
0029	2B		160	DEC	HL
002A	E3		161	EX	(SP), HL
002B	22E81F		162	LD	(81TEMP), HL
002E	180A		163	JR	CONT28
			164		
			165		;
0030			166	RST30	ORG 30H
			167		
			168		; Instruction RST 30H (opcode F7) is usually used as:
			169		; 1) Software break;
			170		; 1i) Terminator of user's program.
			171		; The effect of this instruction is to save all user's
			172		; registers and return to monitor.
			173		
0030	1834		174	JR	NMI

MPF-1

LOC OBJ CODE M STMT SOURCE STATEMENT

```

175
176
177 ; This is a part of reset routine. Address 0028 and
178 ; 0030 are reserved for break point. Reset routine
179 ; skips this area and resumes here.
180
0032 22D21F 181 RESET1 LD (USERIP),HL ;set user's I register and
182 ; interrupt flip flop to 0
0035 181D 183 JR RESET2 ;monitor resumes at RESET2
184
185 ;
186
187 ; The following byte makes the sum of the monitor
188 ; code in ROM zero. ROMTEST is a self-checking routine.
189 ; This routine requires the sum of ROM to be zero.
190
0037 71 191 DEFB ZSUM
192 ;
193 ;
0038 194 RST38 ORG 38H
195
196 ; Entry point of RST 38H (opcode FF) or mode 1 interrupt.
197 ; Fetch the address stored in location 1FEE and 1FEE
198 ; then jump to this address. Initially, 1FEE and 1FEE
199 ; are set to 0000. So RST 38 will have the same effect
200 ; as software break. By changing the content of 1FEE
201 ; and 1FEE, the user can define his or her own service
202 ; routine.
203 ; The next three instructions push the contents of 1FEE
204 ; and 1FEE to stack without changing any registers.
205
0038 85 206 PUSH HL
0039 2AE1F 207 LD HL,(IMIAD)
003C E3 208 EX (SP),HL
209
210 ; The top of the stack is now the address of user
211 ; defined service routine. Pop out this address then
212 ; branch to it.
213
003D C9 214 RET
215 ;
216 ;
217 ;
218 ; This is a part of break service routine. It continues
219 ; the program at RST28.
220
003E 32E71F 221 LD (ATEMP),A
222
223 ; The monitor has changed the content of user's
224 ; program at break address. The next 3 instructions
225 ; restored the destroyed content. BRAD contains the
226 ; break address, BRDA contains the original data at
227 ; break address.
228
0041 2AE01F 229 LD HL,(BRAD)
0044 3AE21F 230 LD A,(BRDA)
0047 77 231 LD (HL),A
232

```

MPP-I  
 LOC OBJ CODE M STMT SOURCE STATEMENT

```

233 ; Send break enable signal to hardware counter.
234 ; A nonmaskable interrupt will be issued at the 5th MI's.
235
0048 3E80          LD      A,1000000B
004A D302          OUT     (DIGIT),A
004C 3A71F        LD      A,(ATEMP)      ; 1st MI
004F 2A81F        LD      HL,(HLTEMP)   ; 2nd MI
0052 00          NOP                    ; 3rd MI
0053 C9          RET                      ; 4th MI
242
243 ; Return to user's program. Execute the instruction
244 ; at break address. After finishing one instruction,
245 ; a nonmaskable interrupt happens and control is
246 ; transferred to the monitor again.
247 ;
248 RESET2:
0054 219F1F       LD      HL,USERSTK
0057 2D01F       LD      (USERSP),HL   ;set user's SP
005A AF          XOR      A
005B 32E81F      LD      (TEST),A
253
254 ; TEST is a flag for monitor's own use. Illegal key-in
255 ; blanking (bit 7 of TEST) and automatic loading zero
256 ; (bit 0) use this flag. Clear it here.
257
005E DD219F07    LD      IX,MPP_I      ;initial display pattern.
259
260 ; Address 0086 is the address for nonmaskable interrupt.
261 ; Skip this area, monitor resumes at SET3T0
262
0062 C3D000      JP      SET3T0
264 ;
265 ;.....
0066 NMI          ORG      66H
267
268 ; Entry point of nonmaskable interrupt. NMI will occur
269 ; when MONI key is pressed or when user's program is
270 ; broken. The service routine which starts here saves all
271 ; user's registers and status. It also check the validity
272 ; of user's SP.
273
0068 32E71F       LD      (ATEMP),A     ;save A register
0069 3E90         LD      A,1001000B
006B D303         OUT     (P8255),A    ;set 8255 to mode 0.
277 ;Port A input; B,C output.
006D 3E00         LD      A,OCOE
006F D302         OUT     (DIGIT),A   ;disable break and LED's
0071 3A71F       LD      A,(ATEMP)    ;restore A register
0074 22E81F     LD      (HLTEMP),HL  ;save register HL
0077 E1          POP     HL        ;get return address from stack
0078 22D81F     LD      (ADSAVE),HL  ;save return address into
284 ;ADSAVE.
007B 22DC1F     LD      (USERPC),HL ;Set user's PC to return
286 ;address.
007E 2A81F       LD      HL,(HLTEMP) ;restore HL register
0081 ED73D01F   LD      (USERSP),SP ;set user's SP to current SP
0085 31D01F     LD      SP,USERIY*2 ;save other registers by
0088 FDE5       PUSH    IY           ;continuously pushing them
  
```

MPF-I  
 LOC OBJ CODE M STMT SOURCE STATEMENT

```

008A DDE5      281      PUSH   IX           ;onto stack
008C D9        282      RXX
008D E5        283      PUSH   HL
008E D5        284      PUSH   DE
008F C5        285      PUSH   BC
0090 D9        286      RXX
0091 08        287      EX     AF,AF'
0092 F5        288      PUSH   AF
0093 08        289      EX     AF,AF'
0094 E5        300      PUSH   HL
0095 D5        301      PUSH   DE
0096 C5        302      PUSH   BC
0097 F5        303      PUSH   AF
304
305 ; The next two instructions save I register.
306 ; The interrupt flip-flop (IFF2) is copied into
307 ; parity flag (P/V) by instruction LD A,I.
308 ; The interrupt status (enabled or disabled)
309 ; can be determined by testing parity flag.
310
0098 ED57      311      LD     A,I
009A 32D31F    312      LD     (USERIP+1),A
313
314 ; The next four instructions save IFF2 into
315 ; user's IFF.
316
009D 3E00      317      LD     A,0
009F E2A400   318      JP     PO,SETIP      ;PO -- P/V = 0
00A2 3E01      319      LD     A,1
00A4 32D21F    320      SETIP LD     (USERIP),A
321 ;
00A7 31AF1F    322      LD     SP,SYSTK      ;set SP to system stack
323
324 ; The next 8 instructions check user's SP.
325 ; If the user's SP points to a location not
326 ; in RAM, display ERR-SP.
327
00AA 2AD01F    328      LD     HL,(USERSP)
00AD DD21B507   329      LD     IX,ERR_SP
00B1 2B        330      DEC   HL
00B2 CDF605    331      CALL  RAMCHK
00B5 2019      332      JR     NZ,SETSTO
00B7 2B        333      DEC   HL
00B8 CDF605    334      CALL  RAMCHK
00BB 2013      335      JR     NZ,SETSTO
336
337 ; If the user's stack and system stack are
338 ; overlapped, display SYS-SP. This checking
339 ; is done by the following instructions.
340
00BD DD21AF07   341      LD     IX,SYS_SP
00C1 00        342      NOP
00C2 00        343      NOP
344
00C3 1182E0   345      LD     DE,-USERSTK+1
00C6 19        346      ADD   HL,DE
00C7 3807      347      JR     C,SETSTO
00C9 DD21B61F   348      LD     IX,DISPBF
  
```



```

MPP-I
LOC  OBJ CODE M STMT SOURCE STATEMENT
00CD 37      349      SCF          ;set carry flag to indicate
      350          ;the user's SP is legal.
00CE 1804    351      JR          BRRSTO
      352      ;
      353 SETSTO:
      354 ; STATE is a memory location contains the monitor status.
      355 ; It will be described in detail later. STATE 0 stands
      356 ; for fixed display pattern. The initial pattern 'uPP--1'
      357 ; or message 'SYS-SP'... belong to this category. The next
      358 ; two instruction set STATE to zero.
      359
00DD AF      360      XOR          A          ;set A to 0, also clear Carry flag
00D1 32E41F  361      LD          (STATE),A
00D4 3AE21F  362 BRRSTO LD      A,(BRDA) ;restore the data at
      363          ;break address
00D7 2AE01F  364      LD          HL,(BRAD)
00DA 77      365      LD          (HL),A
      366
      367 ; If the user's SP is legal (carry set),
      368 ; display user's PC and the content at PC.
      369 ; Otherwise, display fixed message (ERR-SP
      370 ; or SYS-SP or uPP--1)
00DB DC0B04  371      CALL      C,MENDP2
      372      ;
      373      ;
      374 ;*****
      375 ; Scan the display and keyboard. When a key is
      376 ; detected, take proper action according to the
      377 ; key pressed.
      378
      379 MAIN:
00DE 31AF1F  380      LD          SP,SYSSK ;initial system stack.
00E1 CDFE05  381      CALL      SCAN ;Scan display and input keys.
      382          ;Routine SCAN will not return until
      383          ;any key is pressed.
00E4 CDCB06  384      CALL      BEEP ;After a key is detected, there
      385          ;will be accompanied with a beep
      386          ;sound.
00E7 18F5    387      JR          MAIN ;Back to MAIN, get more keys and
      388          ;execute them.
      389      ;
      390      ;
      391 ;*****
      392 KEYXRC:
      393
      394 ; Input key dispatch routine.
      395 ; This routine uses the key code returned by subroutine
      396 ; SCAN, which is one byte stored in A register. The
      397 ; range of key code is from 00 to 1FH.
      398
      399 ; (1) key code = 00  $\phi$  OFH :
      400 ; These are hexadecimal keys. Branch to routine KHEX.
      401
00E9 FE10    402      CP          10H
00EB 3824    403      JR          C,KHEX
      404
      405 ; If the key entered is not hexadecimal, it must be a
      406 ; function or subfunction key. This means the previous

```

## MFP-I

LOC OBJ CODE M SINT SOURCE STATEMENT

```

407 ; numeric entry has terminated. Bit 0 of TEST flag
408 ; must be set at the beginning of a new numeric entry.
409 ; This is done by the next two instructions. (If bit 0
410 ; of TEST is set, the data buffer will be automatically
411 ; cleared when a hexadecimal key is entered.)
412
00ED 21E61F 413 LD HL,TEST
00FO CBCE 414 SET O,(HL)
415
416 ; (ii) key code = 10H  $\phi$  17H :
417 ; (*, -, GO, STEP, DATA, SBR, INS, DEL)
418 ; There is no state corresponding to these keys.
419 ; The response of them depends on the current
420 ; state and minor-state. (E.g., the response of '+'
421 ; key depends on the current function. It is illegal
422 ; when the display is 'uPF--1', but is legal when the
423 ; display is of 'address-data' form.) In this
424 ; documentation, they are named 'sub-function key'.
425 ; They are all branched by table KSUBFUN and routine
426 ; BRANCH.
427
00F2 D810 428 SUB 10H
00F4 FE08 429 CP 8
00F8 213707 430 LD HL,KSUBFUN
00F9 DAB003 431 JP C,BRANCH
432
433 ; (iii) key code = 18H  $\phi$  1FH
434 ; (PC, Addr, CBR, Reg, Move, Rela, WRTape, RDTape)
435 ; These keys are named 'function key'. They are
436 ; acceptable at any time. When they are hit, the
437 ; monitor will unconditionally enter a new state.
438 ; STMINOR contains the minor-state, which is required
439 ; to dispatch some sub-function keys (e.g. +, -).
440
00FC DD21B61F 441 LD IX,DISPBF
0100 D608 442 SUB 8
0102 21E41F 443 LD HL,STATE
0105 77 444 LD (HL),A
445 ;set STATE to key-code minus 18H
446 ;The STATE is update here. It will
447 ;be modified later by local service
448 ;routines if the function-key is PC,
449 ;Addr or CBR. For other function-
450 ;keys, STATE will not be modified
451 ;later.
0108 21E31F 451 LD HL,STMINOR
0109 3600 452 LD (HL),O ;set STMINOR to 0
010B 214107 453 LD HL,KFUN ;KFUN is the base of branch table
454 ;the offset is stored in A
010E C3B003 455 JP BRANCH
456
457 ;
458 ;.....
459 ;STATE:
460 ; 0=FIX ;Display fixed pattern, e.g. 'uPF--1'.
461 ; 1=AD ;The hex key entered is interpreted as
462 ; ;memory address.
463 ; 2=LM ;The hex key entered is interpreted as
464 ; ;memory data.

```

MPP-I  
 LOC OBJ CODE M STMT SOURCE STATEMENT

```

465 ; 3-RGPIX ;Display fixed pattern: 'Reg-' and
466 ;expect register name to be entered.
467 ; 4-MV ;Expect parameters for 'Move' function.
468 ; 5-RL ;Expect parameters for 'Rela' function.
469 ; 6-WT ;Expect parameters for 'WTape' func.
470 ; 7-RT ;Expect parameters for 'RTape' func.
471 ; 8-RGAD ;Hex-key entered will be interpreted as
472 ;address name for registers.
473 ; 9-RGDA ;Hex-key entered will be interpreted as
474 ;data for registers.
475 ;
476 ; Subroutine name conventions:
477 ; (i) K???? -- K stands for key, ???? is the key name,
478 ; e.g. KING corresponds to key 'INS'. Each
479 ; time a key ???? is entered, the routine
480 ; with name K???? will be executed. All of
481 ; these are branched by table KPUN or KSUBPUN.
482 ; (ii) H???? -- H stands for hexadecimal, ???? is the
483 ; current STATE. For example, routine
484 ; HDA will be executed if the entered
485 ; key is hexadecimal and STATE is DA now.
486 ; These routines are branched by table
487 ; HTAB.
488 ; (iii) I???? -- I stands for increment (+ key), ???? is
489 ; the current STATE. E.g. IMV will be
490 ; executed when STATE is MV and '+' key
491 ; is entered. These routines are branched
492 ; by table ITAB.
493 ; (iv) D???? -- D stands for decrement (- key), ???? is
494 ; the current STATE. These routines are
495 ; branched using table DTAB.
496 ; (v) G???? -- G stands for 'GO' key, ???? is the current
497 ; STATE. These routines are branched using
498 ; table GTAB.
499 ;
500 ;*****
501 ;
502 ; Hexadecimal, '+', '-' and 'GO' key may be entered after
503 ; different function keys. The monitor uses branch tables
504 ; and STATE to determine the current function and branch
505 ; to the proper entry point.
506 ;
507 KHEX:
508 ;Executed when hexadecimal keys are pressed.
509 ;Use HTAB and STATE for further branch.
510 ;
0111 4F LD C,A ;save A register in C
511 ; ;which is the hex key-code.
512 ;
0112 214B07 LD HL,STAB
0115 3A841F BRL LD A,(STATE)
0118 C3B003 JP BRANCH
516 ;
517 ;
518 KINC:
519 ;Branched by KSUBPUN table.
520 ;Executed when '+' key is pressed.
521 ;Use ITAB and STATE for further branch.
522 ;STATE is will be stored in A register at BRL.

```

MPF-I

LOC OBJ CODE M STMT SOURCE STATEMENT

```

523
011B 215707 524 LD HL,ITAB
011R 18F5 525 JR BRI
526
527 ;
528 KDEC:
529 ;Branched by KSUBFUN table. Executed
530 ;when '-' key is pressed. Use DTAB and
531 ;STATE for further branch. STATE will be
532 ;stored in A register at BRI.
533
0120 216307 534 LD HL,DTAB
0123 18F0 535 JR BRI
536
537 ;
538 KGO:
539 ;Branched by KSUBFUN table. Executed
540 ;when 'GO' key is pressed. Use GTAB and
541 ;STATE for further branch. STATE will be
542 ;stored in A register at BRI.
543
0125 216F07 544 LD HL,GTAB
0128 18EB 545 JR BRI
546
547 ;
548 KSTEP:
549 ;Branched by table KSUBFUN. Executed
550 ;when 'STEP' key is pressed.
551
012A CDE503 552 CALL TESTM ;Check if the left 4 digits
553 ;of the display are memory address.
554 ;If not, disable all LED's as
555 ;a warning to the user. This
556 ;is done by routine IGNORE.
012D C2BB03 557 JP NZ,IGNORE
0130 3E80 558 LD A,10000000B ;This data will be output
559 ;to port B to enable
560 ;BREAK. It is done by
561 ;routine PRCOUT.
0132 C3A302 562 JP PRCOUT
563
564 ;
565 KDATA:
566 ;Branched by table KSUBFUN. Executed
567 ;when 'DATA' key is pressed.
568
0135 CDE503 569 CALL TESTM ;Check if the left 4 digits
570 ;of the display are memory address.
0138 2004 571 JR NZ,TESTM ;if not, branch to TESTM
572 ;to check whether the display
573 ;is register or not.
013A C00B04 574 CALL MEMDP2 ;if yes, display the data of
575 ;that address and set STATE
576 ;to 2.
013D C9 577 RET
013E FE08 578 TESTM CP ; ;check if the status is 8 or 9
579 ;(RGAD or RGDA).
0140 DAB803 580 JP C,IGNORE ;if not, ignore this key and

```

MPP-I  
 LOC OBJ CODE M STMT SOURCE STATEMENT

```

581
0143 CD7704 582 CALL REGDPS ;send out a warning message.
583 ;If yes, display register and
0146 C9 584 RET ;set status to 9 (RQDA).
585
586 ;
587 KSRB:
588 ;Branched by table KSUBFON. Executed
589 ;when 'SBR' key (set break point) is
590 ;pressed.
591
0147 CDE503 592 CALL TESTM ;Check if the display is of
593 ;'address-data' form.
014A C2B803 594 JP NZ,IGNORE ;If not, ignore this key and
595 ;send out a warning message.
014D 2ADE1F 596 LD HL,(ADSAVE) ;If yes, get the address
597 ;being display now.
0150 CDF605 598 CALL RAMCHK ;Check if this address is
599 ;in RAM.
0153 C2B803 600 JP NZ,IGNORE ;If not, ignore the 'SBR' key
601 ;and send out a warning message.
0156 22E01F 602 LD (BRAD),HL ;If yes, set this address as
603 ;a break point.
0159 CD0804 604 CALL MEMDP2 ;Display the data of break
605 ;address and set STATE to
606 ;2 (DA).
015C C9 607 RET
608
609 ;
610 KINS:
015D CDE503 611 ;Branched by table KSUBFON. Executed
0160 C2B803 612 ;when 'Ins' key (insert) is pressed.
613
015D CDE503 614 CALL TESTM ;Check if the display is of
615 ;'address-data' form now.
0160 C2B803 616 JP NZ,IGNORE ;If not, ignore the 'INS' key
617 ;and send out a warning message.
0163 2ADE1F 618 LD HL,(ADSAVE) ;If yes, get the address being
619 ;displayed now.
620
0166 00 621 NOP
622
0167 22AF1F 623 LD (STEPBF),HL ; Store this address in
624 ;STEPBF and the next address.
625 ;in STEPBF+4 for later use.
626
016A 23 627 INC HL
016B 22B31F 628 LD (STEPBF+4),HL
016E CDF605 629 CALL RAMCHK ;Check if the address to be
630 ;inserted is in RAM.
0171 C2B803 630 JP NZ,IGNORE ;If not, ignore the 'INS' key
631 ;and send out a warning message.
632 ;If the address to be inserted
633 ;is in 1800-1DFF, store 1DFF into
634 ;STEPBF+2
635 ;Otherwise, ignore the 'INS' key.
636 ;This is done by the following.
637 ;instructions.
0174 11F81D 638 LD DE,1DFF

```

MPP-I

LOC	OBJ CODE	M	STMT	SOURCE STATEMENT
0177	7C		639	LD A,R
0178	FE1E		640	CP 18H
017A	3807		641	JR C,SKIPHI
017C	FE20		642	CP 20H
017E	DAB803		643	JP C,IGNORE
0181	1827		644	LD D,27H
0183	ED53B11F		645	SKIPHI LD (STEPBF+2),DE
			646	
			647	;When one byte is inserted at some
			648	address, all data below this address
			649	will be shifted down one position.
			650	;The last location will be shifted out
			651	and therefore lost.
			652	;The RAM is divided into 3 blocks as
			653	insert is concerned. They are:
			654	;1800-1DFF,1E00-1FFF and 2000-27FF
			655	;The 2 nd block cannot be inserted and
			656	is usually used as data bank. System
			657	data that of course cannot be shifted
			658	are also stored in this bank. Each
			659	block is independent of the other when
			660	shift is performed, i.e. the data
			661	shifted out of the first block will not
			662	be propagated to next block.
			663	;The shift is accomplished by block
			664	transfer, i.e. MOVE. This is the
			665	job of subroutine GMV.
			666	;Routine GMV needs 3 parameters which
			667	are stored in step-buffer (STEPBF):
			668	;STEPBF: starting address (2 bytes);
			669	;STEPBF+2: ending address (2 bytes);
			670	;STEPBF+4: destination address (2 bytes).
			671	
0187	CDK402		672	DMV CALL GMV
018A	AF		673	XOR A
			674	;After the RAM has been shifted down,
			675	the data of the address to be inserted
			676	is cleared to zero. This is done by
			677	the next two instructions. Register
			678	DE contain inserted address after GMV
			679	is performed.
018B	12		680	LD (DE),A
018C	2AB31F		681	HL,(STEPBF+4);Store the data in (STEPBF+4)
018F	22DR1F		682	(ADSAVE),HL;into (ADSAVE).
0192	CD0804		683	CALL MEMDP2;Display the address and data, also
			684	;set STATE to 2.
0195	C9		685	RET
			686	
			687	KDEL;
			688	;Branched by table KSUBFUN. Executed
			689	when 'Del' (delete) key is pressed.
			690	
0196	CDK503		691	CALL TESTM;Check if the display is of
			692	'address-data' form.
0199	C2B803		693	JP NZ,IGNORE;If not, ignore the 'Del' key and
			694	send out a warning message.
			695	'Delete' is quite similar to
			696	'Insert', except that the memory
			697	is shifted up instead of shifted

MPP-I  
 LOC OBJ CODE M STMT SOURCE STATEMENT

```

897                                ;down. See the comments on
898                                ;routine KINS for detail.
019C 2ADE1F          LD          HL,(ADSAVE) ;Get the address being displayed
700                                ;now. This is the address to
701                                ;be deleted.
702
019F 00             NOP
703
01A0 22B31F          LD          (STEPBF+4),HL
01A3 CDF605          CALL       RAMCHK ;Check if the address is in RAM.
01A6 C2B803          JP          NE,IGNORE ;If not, ignore this key and
709                                ;send out a warning message.
710                                ;Following instructions prepare the
711                                ;parameters for routine GMY in step-
712                                ;buffer. Refer to routine KINS for
713                                ;detail.
01A9 11001E          LD          DE,1E00H
01AC 7C             LD          A,H
01AD FE1E          CP          1EH
01AF 3807          JR          C,SKIP2
01B1 FE20          CP          20H
01B3 DAB803          JP          C,IGNORE
01B6 1628          LD          D,28H
01B8 ED53B11F       SKIP2 LD      (STEPBF+2),DE
01BC 23             INC          HL
01BD 22AF1F          LD          (STEPBF),HL
01C0 18C5          JR          DDMV
725                                ;
726                                ;
727                                ; KPC:
728                                ; Branched by table KPUN. Executed when
729                                ; 'PC' key is pressed.
730
01C2 2ADC1F          LD          HL,(USERPC) ;Store the user's program
01C5 22DE1F          LD          (ADSAVE),HL ;counter into (ADSAVE)
01C8 C0B004          CALL       MEMDP2 ;Routine MEMDP2 displays the address
734                                ;in (ADSAVE) and its data. It also
735                                ;set the STATE to 2.
01CB C9             RET
736
737                                ;
738                                ; KCBR:
739                                ; Branched by table KPUN. Executed when
740                                ; 'CBR' (clear break point) key is pressed.
741
01CC C0DE03          CALL       CLRBR ;Call subroutine CBRBR to clear
742                                ;break point. When returned, the HL
743                                ;register will contain FFFF.
744
01CF 22DE1F          LD          (ADSAVE),HL ;Store FFFF into (ADSAVE)
01D2 C0B004          CALL       MEMDP2 ;Display address and its data. Also
747                                ;set STATE to 2.
01D5 C9             RET
748
749                                ;
750                                ; KREG:
751                                ; Branched by table KPUN. Executed when
752                                ; 'Reg' key is pressed.
01D6 DD21CA07       LD          IX,REG ;Routine SCAN uses IX as a pointer
753                                ;for display buffer. Set IX to REG_
754

```

MPP-I

LOC	OBJ CODE	M	STMT	SOURCE	STATEMENT
01DA	CDC404		755		;will make SCAN displays 'Reg-
			756	CALL	PCONV ;Decode user's flag F and P' to
			757		;binary display format. This
			758		;format will be used later, when
			759		;user requires the monitor to
			760		;display decoded flag by pressing
			761		;keys 'SZXH', 'XPMC',...
01DD	C9		762	RET	
			763		;
			764	KADDR:	
			765		; Branched by KFUN table. Executed when
			766		; 'Addr' key is pressed.
			767		
01DE	CD0204		768	CALL	MEMDPI ;Display the address stored in
			769		; (ADSAVE) and its data. Set STATE
			770		; to 1 (AD).
01E1	C9		771	RET	
			772		;
			773		; Function Move, Relative, Read-tape and
			774		; Write-tape requires from one to three
			775		; parameters. They are stored in STEPB
			776		; (step buffer). STMINOR (minor status)
			777		; contains the number of parameters has been
			778		; entered. For Move and Relative, the
			779		; default value of the first parameter is
			780		; the address stored in (ADSAVE). There
			781		; is no default value for the first parameter
			782		; (filename) of Read- and Write-tape. When the
			783		; function keys are pressed, STMINOR is automatically
			784		; reset to 0.
			785		;
			786		
			787	KMV:	
			788		; Branched by table KFUN. Executed when
			789		; 'Move' key is pressed.
			790		
			791	KRL:	
			792		; Branched by table KFUN. Executed when
			793		; 'Rel' (relative) key is pressed.
01E2	2ADRIF		794	LD	HL,(ADSAVE) ;Store the contents of ADSAVE
			795		; into STEPB as default value
			796		; of first parameter.
01E5	22AFIF		797	LD	(STEPB),HL
			798		
			799	KWT:	
			800		; Branched by table KFUN. Executed
			801		; when 'WRtape' key is pressed.
			802		
			803	KRT:	
			804		; Branched by table KFUN. Executed when
			805		; 'RDtape' key is pressed.
01E6	CD3A04		806	CALL	STRPOP ;Display the parameter that
			807		; is being entered now by calling
			808		; subroutine STRPOP.
01E8	C9		809	RET	
			810		;
			811		; .....
			812		; The following subroutines with name R???
					; are the service routine for hexadecimal



LOC OBJ CODE M STMT SOURCE STATEMENT

MPP-I

```

813 ; keys corresponding to each STATE. They
814 ; are all branched by table HTAB and STATE.
815
018C C3B803 816 HFIX JP IGNORE ;When the display is fixed pattern
817 ;hexadecimal keys are illegal.
818 ;Disable all LED's as a warning
819 ;message to the user. This is what
820 ;routine IGNORE does.
821 ;
018F 2ADE1F 822 HDA LD HL,(ADSAVE) ;Get the address being displayed
823 ;now from (ADSAVE)
01F2 CDF805 824 CALL RANCHK ;Check if it is in RAM.
01F5 C3B803 825 JP NE,IGNORE ;If not, ignore this key and
826 ;send out a warning message.
01F8 CD8E03 827 CALL PRECLI ;If this is the first hexadecimal
828 ;key entered after function or sub-
829 ;function key, reset the data of that
830 ;address to 0. (by routine PRECLI)
01FB 79 831 LD A,C ;The key-code is saved in C at
832 ;routine KHEX. Restore it to A.
01FC 8D6F 833 RLD ;Rotate the key-code (4 bits) into
834 ;the address obtained above. (in HL)
01FE CD0804 835 CALL MEMDP2 ;Display the address and data,
836 ;then set STATE to 2 (DA).
0201 C9 837 RET
838 ;
0202 21DE1F 839 RAD: LD HL,ADSAVE
0205 CDFA03 840 CALL PRECLS ;If this is the first hexadecimal
841 ;key after function key is entered,
842 ;set the contents of ADSAVE to 0.
0208 79 843 LD A,C ;The key-code is saved in C
844 ;by routine KHEX.
845 ;The next three instructions shift
846 ;the address being displayed by
847 ;one digit.
0209 ED8F 848 RLD
020B 23 849 INC HL
020C ED6F 850 RLD
020E CD0204 851 CALL MEMDPI ;Display the address and its
852 ;data. Also, set STATE to 1.
0211 C9 853 RET
854 ;
855 ;RGAD:
856 ;HROFIX:
0212 79 857 LD A,C
0213 DD21B61F 858 LD IX,DISPBF
0217 21E31F 859 LD HL,STMINOR
021A 87 860 ADD A,A ;The key-code is the register
861 ;name. Double it and store it
862 ;into STMINOR.
021B 77 863 LD (HL),A
021C CD7304 864 CALL REWDPS ;Display register and set
865 ;STATE to 8. (RGAD)
021F C9 866 RET
867 ;
868 HRT:
869 HWT:
870 HRL:

```

LOC	OBJ CODE	M	STMT	SOURCE	MFT-1	STATEMENT
0220	CD5504		871	HMV:	CALL	LOCSTBF ;Use STMINOR and STEPSF
			872			;to calculate the address
			873			;of current parameter in
			874			;step buffer.
0223	CDFA03		875		CALL	PRECL2 ;If this is the first hex
			876			;key entered, cleared the
			877			;parameter (2 bytes) by
			878			;PRECL2.
0226	79		879		LD	A,C ;C contains the key-code.
			880			;Rotate the parameter (2 bytes)
			881			;1 digit left with the key-code.
0227	ED6F		882		RLD	
0229	23		883		INC	HL
022A	ED6F		884		RLD	
022C	CD3A04		885		CALL	STEPDP ;Display the parameter.
022F	C9		886		RET	
			887			
0230	CD8B04		888	REGDA	CALL	LOCRESBF ;Calculate the address of
			889			;the register being modified.
0233	CD8E03		890		CALL	PRECL1 ;If this is the first hex
			891			;key entered. Clear the register
			892			; (1 byte) by PRECL1.
0236	79		893		LD	A,C ;Rotate user's register (1 byte)
			894			;1 digit left with the key-code
			895			;stored in C.
0237	ED6F		896		RLD	
0239	CD7704		897		CALL	REGDP9 ;Display the register and set
			898			;STATE to 9 (RGDA).
023C	C9		899		RET	
			900			
			901			*****
			902			;The following routines with name
			903			;I7?? are the service routines for
			904			; '+' key corresponding to each STATE.
			905			; They are all branched by table ITAB
			906			; and STATE.
			907			
			908		IFIX:	
			909		IROFIX:	
023D	C38B03		910		JP	IGNORE ; '+' key is illegal for state
			911			; FIX or IROFIX, ignore it.
			912			
			913		IAD:	
0240	2ADR1F		914	IDA:	LD	HL,(ADSAVE) ;Increase the address being
			915			; displayed now (in ADSAVE)
			916			; by 1.
0243	23		917		INC	HL
0244	22DE1F		918		LD	(ADSAVE),HL
0247	CD0B04		919		CALL	MEMDP2 ;Display the address and data,
			920			; then set the STATE to 2.
024A	C9		921		RET	
			922			
			923		INT:	
			924		IWT:	
			925		IRL:	
024E	21R31F		926	HMV:	LD	HL,STMINOR ;STMINOR contains the
			927			; parameter count, increment
			928			; it by one.

MPP-I

LOC	OBJ CODE M	STMT	SOURCE	STATEMENT
024E	34	829		INC (HL)
024F	CD5F04	830		CALL LOCSTNA ;Check if the count is
		831		;overflowed.
0252	2004	832	JR	KZ, ISTEP ;If not overflowed, continue
		833		;at ISTEP.
0254	35	834	DEC	(HL) ;Otherwise, restore the count
		835		;and ignore the '-' key.
0255	C3BB03	836	JP	IGNORE
0258	CD3A04	837	CALL	STEPDP ;Display the parameter at
		838		;step buffer.
025B	C9	839	RET	
		840		;
		841	IRGAD:	
025C	21E31F	842	LD	HL, STMINOR ;In these states, the STMINOR
		843		;contains the register name.
		844		;increase it by 1. If it
		845		;reaches the last one, reset
		846		it to the first one (0).
025F	34	847	INC	(HL)
0260	3E1F	848	LD	A, 1FH
0262	BE	849	CP	(HL)
0263	3002	850	JR	NC, IRGNA
0265	3800	851	LD	(HL), 0
0267	CD7704	852	CALL	REDDP9 ;Display the register and
		853		;set STATE to 9.
026A	C9	854	RET	
		855		;
		856		;*****
		857		;The following routines with name
		858		;D7??? are the service routines for
		859		; '-' key corresponding to each state.
		860		;They are all branched by table DTAB
		861		;and STATE.
		862		;
		863	DPIX:	
		864	DROPIX:	
026B	C3BB03	865	JP	IGNORE ; '-' key is illegal for
		866		;these states. Ignore it.
		867		;
		868	DAD:	
026E	2ADE1F	869	LD	HL, (ADSAVE) ;Decrease the address being
		870		;displayed now (in ADSAVE)
		871		;by one.
0271	2B	872	DEC	HL
0272	22DE1F	873	LD	(ADSAVE), HL
0275	CD0B04	874	CALL	MEMDP2 ;Display the address and data,
		875		;set STATE to 2 (DA).
0278	C9	876	RET	
		877		;
		878	DRT:	
		879	DRT:	
		880	DRL:	
0279	21E31F	881	LD	HL, STMINOR ;In these states, STMINOR
		882		;contains the parameter count.
		883		;Decrease it by one. If overflow
		884		;occurs, restore STMINOR and
		885		;ignore the '-' key. Otherwise
		886		;continue at DSTEP.

MPP-1

LOC	OBJ CODE	M	STMT	SOURCE	STATEMENT
027C	35		987	DRC	(HL)
027D	CD5F04		988	CALL	LOCSTNA
0280	2004		989	JR	NE, DSTEP
0282	34		990	INC	(HL)
0283	C3BB03		991	JP	IGNORE
0286	CD3A04		992	DSTEP	CALL
0289	C9		993	RET	STREPOP ;Display the parameter.
			994		
			996	DRGAD:	
028A	21E31F		998	DRGDA: LD	HL, STMINOR ;In these states, STMINOR
			997		;contains the register name:
			998		;Decrease it by one. If it
			999		;goes below zero, set it to
			1000		;the highest value (1P).
028D	J5		1001	DRC	(HL)
028E	3E1F		1002	LD	A, OFFH
0290	BE		1003	CP	(HL)
0291	3002		1004	JR	NC, DRGNA
0293	3E1F		1006	LD	(HL), 1FH
0295	CD7704		1008	DRGNA CALL	REGDP9 ;Display the register and
			1007		;set STATE to 9.
0298	C9		1008	RET	
			1009		
			1010		;
			1011		*****
			1012		;The following routines with name
			1013		;G7?? are the service routines for
			1014		;'GO' key corresponding to each
			1015		;state. They are all branched by
			1016		;table GTAB and STATE.
			1017	GPIX:	
			1018	GRGPIX:	
			1019	GRGAD:	
0299	C3BB03		1020	GRGDA: JP	IGNORE ;'GO' key is illegal for
			1021		;these states. Ignore it.
			1022		;
			1023	GAD:	
029C	2AK01F		1024	GDA: LD	HL, (BRAD) ;Get the address of break
			1025		;point.
029F	3E1F		1026	LD	(HL), OFFH ;Instruction RST 26H.
			1027		;The content of break address
			1028		;is changed to RST 26H before
			1029		;the control is transferred to
			1030		;user's program. This
			1031		;will cause a trap when user's
			1032		;PC passes this point.
02A1	3E1F		1033	LD	A, OFFH ;Save FF into TEMP. This data
			1034		;will be output to port B later.
			1035		;FF is used to disable break point.
02A3	32EA1F		1036	PROOT LD	(TEMP), A ;Store A into TEMP.
02A6	3AD21F		1037	LD	A, (USERIF) ;Save two instructions into
			1038		;TEMP and TEMP+1. These two
			1039		;instructions will be executed
			1040		;later. If the user's IFF
			1041		;(interrupt flip-flop) is 1,
			1042		;the instructions are 'EI RET'.
			1043		;Otherwise, they are 'DI RET'.
02A9	CB47		1044	BIT	0,A

MPP-I				
LOC	OBJ CODE	M	STMT	SOURCE STATEMENT
02AB	21FBC9		1045	LD HL,0C9FBH ;'BI','RET'
02AE	2002		1046	JR NZ,RIDI
02B0	2EF3		1047	LD L,0F3H ;'DI'
02B2	22EB1F		1048	LD (TEMP+1),HL
02B5	31BC1F		1049	LD SP,REGBF ;Restore user's registers by
			1050	;setting SP to REGBF (register
			1051	;buffer) and continuously popping
			1052	;the stack.
02B8	F1		1053	POP AF
02B9	C1		1054	POP BC
02BA	D1		1055	POP DE
02BB	R1		1056	POP HL
02BC	08		1057	EX AF,AF'
02BD	F1		1058	POP AF
02BE	08		1059	EX AF,AF'
02BF	D9		1060	EXX
02C0	C1		1061	POP BC
02C1	D1		1062	POP DE
02C2	R1		1063	POP HL
02C3	D9		1064	EXX
02C4	DDE1		1065	POP IX
02C6	FDE1		1066	POP IY
02C8	ED7BD1F		1067	LD SP,(USERSP) ;Restore user's SP.
02CC	32BD1F		1068	(USERAP+1),A ;Temporarily save A
02CF	3AD31F		1069	LD A,(USERIF+1) ;Restore user's I
02D2	ED47		1070	LD I,A
02D4	R5		1071	PUSH HL ;The next 3 instructions
			1072	;push the address being
			1073	;displayed now (in ADGAVE)
			1074	;onto stack without changing
			1075	;HL register. This address will be
			1076	;treated as user's new PC.
02D5	2ADE1F		1077	LD HL,(ADGAVE)
02D8	R3		1078	EX (SP),HL
02D9	3AEA1F		1079	LD A,(TEMP) ;Output the data stored in
			1080	;TEMP to port B of 8255.
			1081	;This data is prepared by
			1082	;routine ESTEP or GAD or
			1083	;GDA. In first case, it is
			1084	;10111111 and will enable
			1085	;break point. In other
			1086	;cases, it is FF and will
			1087	;disable break point.
			1088	;If break is enabled, non-
			1089	;maskable interrupt will occur
			1090	;5 M's after the OUT instruction.
02DC	D302		1091	OUT (DIGIT),A
02DE	3ABD1F		1092	LD A,(USERAP+1) ;1st M1,
			1093	;Restore A register.
02E1	C3EB1F		1094	JP TEMP+1 ;2nd M1,
			1096	;Execute the two instructions
			1097	;stored in RAM. They are:
			1098	; ; E1 (or DI) ;3rd M1
			1099	; ; RET ;4th M1
			1100	;The starting address of user's
			1101	;program has been pushed onto
			1102	;the top of the stack. RET pops
				;out this address and transfers

## MPP-1-

LOC OBJ CODE M SMT SOURCE STATEMENT

```

1103                                     ;control to it. The first M1
1104                                     ;of user's program will be the
1105                                     ;5th M1 after OUT. If break point
1106                                     ;is enabled, M1 will occur after
1107                                     ;this instruction is completed.
1108                                     ;This is the mechanism of single
1109                                     ;step.
1110                                     ;
1111                                     ;.....
02E4 21AF1F 1112 GMV LD HL,STEPBF
02E7 CD3D05 1113 CALL GETP ;Load parameters from
1114                                     ;step buffer into registers.
1115                                     ;Also check if the parameters
1116                                     ;are legal. After GETP,
1117                                     ;HL = start address of source
1118                                     ;BC = length to MOVE.
02EA 3867 1119 JR C,ERROR ;Jump to ERROR if the
1120                                     ;parameters are illegal. (I.e., Ending
1121                                     ;address < starting address.)
02EC ED5BB31F 1122 LD DE,(STEPBF+4) ;Load destination
1123                                     ;address into DE.
02F0 ED52 1124 SBC HL,DE ;Compare HL and DE to
1125                                     ;determine move up or down.
02F2 300C 1126 JR NC,MVUP
1127                                     ;Move down:
02F4 EB 1128 EX DE,HL ;HL = destination address
02F5 09 1129 ADD HL,BC ;HL = dest. address + length
02F6 28 1130 DEC HL ;HL = end address of dest.
02F7 EB 1131 EX DE,HL ;DE = end address of dest.
02F8 2AB11F 1132 LD HL,(STEPBF*2) ;HL = end address of source
02FB ED88 1133 LDDR ;block transfer instruction
02FD 13 1134 INC DE ;DE = last address moved
02FE 8B1C 1135 JR ENDFUN ;Continue at ENDFUN.
1136                                     ;Move up:
0300 19 1137 ADD HL,DE ;HL is destroyed by
1138                                     ;SBC HL,DE. Restore HL.
0301 EDB0 1139 LDIR ;block transfer
0303 18 1140 DEC DE ;DE = last address moved
0304 1616 1141 JR ENDFUN ;Continue at ENDFUN.
1142                                     ;
1143                                     ;.....
0306 ED5BAF1F 1144 GRL LD DE,(STEPBF) ;Load starting address
1145                                     ;into DE.
030A 13 1146 INC DE ;Increase this address by 2.
1147                                     ;Relative address is used in
1148                                     ;instruction JE or DJNE.
1149                                     ;The codes for them are 2 bytes.
1150                                     ;The PC is increased by 2 after
1151                                     ;opcode is fetched.
030B 13 1152 INC DE
030C 2AB11F 1153 LD HL,(STEPBF*2) ;Load destination
1154                                     ;address into HL.
030F B7 1156 OR A
0310 ED52 1157 SBC HL,DE ;Calculate difference.
0312 7D 1158 LD A,L ;Check if the offset is between
1159                                     ;+127 (007FH) and -128 (F80H).
1160                                     ;if the offset is positive, both H
1161                                     ;and bit 7 of L must be zero; if it

```

MPP-1

LOC	OBJ CODE	M	STMT	SOURCE	STATEMENT
			1181		;is negative, H and bit 7 of L must
			1182		;be FF and 1. In both cases, adding
			1183		;H with bit 7 of L results in 0.
			1184		;Rotate bit 7 of L into carry flag.
0313	17		1184	RLA	
0314	7C		1185	LD	A,H
0315	CE00		1186	ADC	A,0 ;ADD H and bit 7 of L.
0317	203A		1187	JR	NZ,ERROR ;Branch to ERROR if
			1188		;the result is nonzero.
0319	7D		1189	LD	A,L
031A	1B		1170	DEC	DE
031B	12		1171	LD	(DE),A ;Save the offset into
			1172		;the next byte of opcode.
			1173		;(DJNZ or JR)
			1174		
			1175	ENDPUN:	
031C	ED53DE1F		1178	LD	(ADSAVE),DE ;Save DE into ADSAVE.
0320	CD0B04		1177	CALL	MEMDP2 ;Display this address and
			1178		;its data. Set STATE to 2.
0323	C9		1179	RET	
			1180		
			1181		;*****
			1182	GWT:	
0324	CD2D05		1183	CALL	SUM1 ;Load parameters from
			1184		;step buffer into registers.
			1185		;Check if the parameters
			1186		;are legal. If legal, calculate
			1187		;the sum of all data to be output
			1188		;to tape.
0327	382A		1189	JR	C,ERROR ;Branch to ERROR if the
			1190		;parameters are illegal. (length is
			1191		;negative)
0329	32B51F		1192	LD	(STEPBF*6),A ;Store the checksum into
			1193		;STEPBF*6.
032C	21A00F		1194	LD	HL,4000 ;Output 1k Hz square
			1195		;wave for 4000 cycles.
			1196		;Leading sync. signal.
032F	CDDE05		1197	CALL	TONE1K
0332	21AF1F		1198	LD	HL,STEPBF ;Output 7 bytes starting
			1199		;at STEPBF. (Include:
			1200		;filename, starting, ending
			1201		;address and checksum)
0335	010700		1202	LD	BC,7
0336	CDA705		1203	CALL	TAPEOUT
033B	21A00F		1204	LD	HL,4000 ;Output 2k Hz square
			1205		;wave for 4000 cycles.
			1206		;Middle sync. The file name of the
			1207		;file being read will be displayed
			1208		;in this interval.
033E	CDE205		1209	CALL	TONE2K
0341	CD3A05		1210	CALL	GETPTR ;Load parameters into
			1211		;registers. (Starting, ending and
			1212		;length).
0344	CDA705		1213	CALL	TAPEOUT ;Output user's data.
0347	21A00F		1214	LD	HL,4000 ;Output 4000 cycles of
			1215		;2k Hz square wave.
			1216		;(Tail sync.)
034A	CDE205		1217	CALL	TONE2K
034D	ED5BB31F		1218	RNDTAP: LD	DE,(STEPBF*4) ;DE = last address

MPP-I					
LOC	OBJ CODE	M	STMT	SOURCE	STATEMENT
0351	18C9		1219	JR	ENDPUN ;Continue at ENDPUN.
			1220		
0353	DD21A907		1221	ERROR LD	IX,ERR ;IX points to '-Err
0357	C3D000		1222	JP	SRTSTO ;Set STATE to 0 by
			1223		;branching to SRTSTO.
			1224		
			1225		;.....
			1226	GRT:	
035A	2AAP1P		1227	LD	HL,(STEPBF) ;Temporarily save filename.
035D	22EA1P		1228	LD	(TEMP),HL
0360	3E40		1229	LEAD LD	A,01000000B ;decimal point
0362	D301		1230	OUT	(SEC7),A ;When searching for filename,
			1231		;the display is blank initially.
			1232		;If the data read from MIC is
			1233		;acceptable 0 or 1, the display
			1234		;becomes '.....'.
0364	21E803		1235	LD	HL,1000
0367	CD8C05		1236	LEAD1 CALL	PERIOD ;The return of PERIOD
			1237		;is in flag:
			1238		; HC -- tape input is 1k Hz;
			1239		; C -- otherwise.
036A	38F4		1240	JR	C,LEAD ;Loop until leading sync.
			1241		;is detected.
036C	2B		1242	DEC	HL ;Decrease HL by one when
			1243		;one period is detected.
036D	7C		1244	LD	A,B
036E	B5		1245	OR	L ;Check if both H and L are 0.
036F	20F6		1246	JR	NZ,LEAD1 ;Wait for 1000 periods.
			1247		;The leading sync. is accepted
			1248		;if it is longer than 1000
			1249		;cycles.(1 second).
0371	CD8C05		1250	LEAD2 CALL	PERIOD
0374	30FB		1251	JR	NC,LEAD2 ;Wait all leading sync. to
			1252		;pass over.
			1253		
0376	21AP1P		1254	LD	HL,STEPBF ;Load 7 bytes from
			1255		;tape into STEPBF.
0379	010700		1256	LD	BC,7
037C	CD4D05		1257	CALL	TAPIN
037F	38DF		1258	JR	C,LEAD ;Jump to LEAD if input
			1259		;is not successful.
0381	ED5BAP1P		1260	LD	DE,(STEPBF) ;Get filename from
			1261		;step buffer.
0385	CD8506		1262	CALL	ADDRDP ;Convert it to display
			1263		;format.
0388	0696		1264	LD	B,150 ;Display it for 1.5 sec.
038A	CD2408		1265	CALL	SCAN1
038D	18FB		1266	DJNZ	FILEDP
038F	2AA1P		1267	LD	HL,(TEMP) ;Check if the input
			1268		;filename equals to the
			1269		;specified filename.
0392	B7		1270	OR	A
0393	AD52		1271	SBC	HL,DE
0396	20C9		1272	JR	NZ,LEAD ;if not, find the leading
			1273		;sync. of next file.
			1274		
			1275		;if filename is found,
0397	3E02		1276	LD	A,0000010B ;segment '-'



MPF-I

LOC	OBJ.CODE	M	STMT	SOURCE	STATEMENT
0399	D301		1277	OUT	(SEG7),A ;Display '-----'.
039B	CD3A05		1278	CALL	GETPTR ;The parameters (starting
			1279		;ending address and check-
			1280		;sum) have been load into
			1281		;STEPB. Load them into
			1282		;registers, calculate the block
			1283		;length and check if they are
			1284		;legal.
039E	38B3		1285	JR	C,ERROR ;Jump to ERROR if the
			1286		;parameters are illegal.
03A0	CD4D05		1287	CALL	TAPIN ;input user's data.
03A3	38AE		1288	JR	C,ERROR ;Jump to ERROR if input
			1289		;is not successful.
03A5	CD2D05		1290	CALL	SUM1 ;Calculate the sum of all
			1291		;input data.
03A8	21B51F		1292	LD	HL,STEPB*4
03AB	BE		1293	CP	(HL) ;Compare it with the
			1294		;checksum calculated by and stored
			1295		;in 'WTape'.
03AC	20A5		1296	JR	NZ,ERROR ;Jump to ERROR if not
			1297		;matched.
03AE	189D		1298	JR	ENDTAPE ;Continue at ENDTAPE.
			1299		
			1300		;*****
			1301		BRANCH:
			1302		;Branch table format:
			1303		; byte 1,2 : address of the 1st routine in
			1304		; each group.
			1305		; byte 3 : difference between the address
			1306		; of 1st and 1st routine, which is
			1307		; of course 0.
			1308		; byte 4 : difference between the address
			1309		; of 2nd and 1st routine
			1310		; byte 5 : difference between the address
			1311		; of 3rd and 1st routine
			1312		; ...
			1313		; ...
			1314		; ...
			1315		; HL : address of branch table
			1316		; A : the routine number in its group
			1317		; Such branch table can have table length and avoid page
			1318		; (256 bytes) boundary problem.
			1319		
03B0	5E		1320	LD	E,(HL) ;Load the address of 1st
			1321		;routine in the group into
			1322		;DE register.
03B1	23		1323	INC	HL
03B2	56		1324	LD	D,(HL)
03B3	23		1325	INC	HL
			1326		;Locate the pointer of difference
			1327		;table.
03B4	85		1328	ADD	A,L
03B5	8F		1329	LD	L,A
03B6	8E		1330	LD	L,(HL) ;Load the address
			1331		;difference into L.
03B7	260D		1332	LD	H,0
03B9	19		1333	ADD	HL,DE ;Get routine's real address
03BA	89		1334	JP	(HL) ;Jump to it.
			1334		

```

MPP-I
LOC  OBJ CODE M STMT SOURCE STATEMENT
1335 ;.....
1336 IGNORE:
03BB 21E61F 1337 LD HL,TEST
03BE CBFE 1338 SET 7,(HL) ;Routine SCAN will check bit
;7 of TEST. If it is set,
;all LEDs will be disabled.
;This is a warning message to
;the user when a illegal key
;is entered.
03C0 C9 1344 RET
1345 ;
1346 ;.....
1347 INI:
03C1 DD21A507 1348 ; Power-up initialization.
1349 LD IX,BLANK ;BLANK is the initial pattern
1350
1351 ;Display the following
1352 ;patterns sequence, each 0.16
1353 ;seconds:
1354 ;
1355 ; ' u'
1356 ; ' uP'
1357 ; ' uPP'
1358 ; ' uPP-'
1359 ; ' uPP--'
1360 ; ' uPP--1'
1361
03C5 0807 1362 LD C,7 ;pattern count
03C7 0810 1363 INI1 LD B,10H ;Display 0.16 second.
03C9 CD2406 1364 CALL SCAN1
03CC 10FB 1365 DJNZ INI2
03CE DD2B 1366 DEC IX ;next pattern
03D0 00 1367 DEC C
03D1 20F4 1368 JR NZ,INI1
1369 ;
03D3 3EA5 1370 LD A,PWCODE
03D5 C9B308 1371 JP INI3
03D8 218600 1372 INI4 LD HL,NMI
03DB 228E1F 1373 LD (INI4D),HL ;Set the service routine
;of RST 3EH to NMI, which is the
;nonmaskable interrupt service
;routine for break point and
;single step.
1374
1375
1376
1377
1378 CLBR:
1379 ; Clear break point by setting
1380 ; the break point address to
1381 ; FFFF. This is a non-existent
1382 ; address, so break can never
1383 ; happen.
1384
03DE 21FFFF 1385 LD HL,0FFFFH
03E1 22801F 1386 LD (BRAD),HL
03E4 C9 1387 RET
1388 ;
1389 TESTM:
1390 ; Check if the display is of 'address-data'
1391 ; form, i.e. STATE 1 or 2.
1392 ; The result is stored in zero flag.

```

```

MPP-I
LOC  OBJ CODE M STMT SOURCE STATEMENT
                                1393 ; Z: yes
                                1394 ; NZ: no
                                1395
03E5  3AE41F 1396 LD A,(STATE)
03E8  FE01 1397 CP 1
03EA  C8 1398 RET Z
03EB  FE02 1399 CP 2
03ED  C9 1400 RET
                                1401 ;
                                1402 PRECL1:
                                1403 ; Pre-clear 1 byte.
                                1404 ; If bit 0 of TEST is not 0, load 0-into (HL). Bit 0 of
                                1405 ; TEST is cleared after check.
                                1406 ; Only AF register are destroyed.
                                1407
03EE  3AE61F 1408 LD A,(TEST)
03F1  B7 1409 OR A ;Is bit 0 of TEST zero?
03F2  C8 1410 RET Z
03F3  3E00 1411 LD A,0
03F5  77 1412 LD (HL),A ;Clear (HL)
03F6  32E61F 1413 LD (TEST),A ;Clear TEST too.
03F9  C9 1414 RET
                                1415 ;
                                1416 PRECL2:
                                1417 ; Pre-clear 2 bytes.
                                1418 ; If bit 0 of TEST is nonzero, clear (HL)
                                1419 ; and (HL+1).
                                1420 ; Only AF register are destroyed.
                                1421
03FA  CDBE03 1422 CALL PRECL1
03FD  C8 1423 RET Z
03FE  23 1424 INC HL
03FF  77 1425 LD (HL),A
0400  2B 1426 DEC HL
0401  C9 1427 RET
                                1428 ;
                                1429 ;*****
                                1430 ; Memory display format: (address-data)
                                1431
                                1432 ; i) A.A.A.A. D D -- State is AD. four decimal points
                                1433 ; under the address field indicate
                                1434 ; that the numeric key entered will
                                1435 ; be interpreted as memory address.
                                1436 ; ii) A A A A D.D.-- State is DA. Two decimal points
                                1437 ; under the data field indicate
                                1438 ; the monitor is expecting user to
                                1439 ; enter memory data.
                                1440 ; iii) A.A.A.A. D.D.-- Six decimal points indicate the
                                1441 ; address being displayed is set
                                1442 ; as a break point.
                                1443 ;
                                1444 MEMDPI:
0402  3E01 1445 LD A,1 ;Next STATE =1
0404  0E04 1446 LD B,4 ;4 decimal points active
0406  218E1F 1447 LD HL,DISPBF*2 ;The first active decimal
                                1448 ;point is in DISPBF*2, the
                                1449 ;last in DISPBF*3.
0409  1E07 1450 JA SAV12 ;Continue at SAV12.

```

```

MPP-I
LOC  OBJ CODE M STMT SOURCE STATEMENT

                                1451 MEMDP2:
040B 3E04 1452 LD A,2 ;Next STATE = 2
040D 0802 1453 LD B,2 ;2 active decimal points
040F 21B61F 1454 LD HL,DISPBF ;1st decimal point is in
                                1455 ;DISPBF, 2nd in DISPBF+1.
0412 32E41F 1456 SAV12 LD (STATE),A ;Update STATE
0415 D9 1457 EXX ;Save register HL,BC,DE.
0418 EDSBDE1F 1458 LD DE,(ADSAVE) ;The address to be
                                1459 ;displayed is stored in
                                1460 ;(ADSAVE). Load it into
                                1461 ;DE register.
041A CD6508 1462 CALL ADDRDP ;Convert this address to
                                1463 ;display format and store it
                                1464 ;into DISPBF*2 + DISPBF+5.
041D 1A 1465 LD A,(DE) ;Load the data of this
                                1466 ;address into A register.
041E CD7108 1467 CALL DATADP ;Convert this data to
                                1468 ;display format and store it
                                1469 ;into DISPBF + DISPBF+1.
                                1470 BRTEST:
0421 2AE01F 1471 ; The next 3 instructions serve to refresh the
0424 7E 1472 ; data at break address every time memory is
                                1473 ; displayed.
                                1474 LD HL,(BRAD) ;Get break point address.
0425 32E21F 1475 LD A,(HL) ;Get the data of this
                                1476 ;address into A register.
0428 B7 1477 LD (BRDA),A ;Store it into BRDA (break data).
0429 ED52 1478 OR A
                                1479 SBC HL,DE ;Check if the address to
                                1480 ;be displayed is break point.
042B 2008 1481 JR NZ,SETPT1 ;if not, jump to SETPT1.
042D 0806 1482 LD B,8 ;8 active decimal points.
042F 21B61F 1483 LD HL,DISPBF ;1st decimal point is in
                                1484 ;DISPBF; 5th in DISPBF+5.
0432 D9 1485 EXX ;Restore HL,BC,DE.
0433 D9 1486 SETPT1 EXX
0434 CBFB 1487 SETPT SET G,(HL) ;Set decimal points.
                                1488 ;Count in B, first address
                                1489 ;in HL register.
0436 23 1490 INC HL
0437 10FB 1491 DJNE SETPT
0439 C9 1492 RET
                                1493 ;
0494 ;*****
0495 ; Step display format: (this format is used when user is
0496 ; entering parameters for Move, Rela, WRTape, RDtape.)
0497 ;
0498 ; P.P.P.P. - N
0499 ;
0500 ; 'P' is the digit of parameter. Four decimal points
0501 ; indicate P's are being modified now. N is the mnemonic of
0502 ; the parameter:
0503 ; i) Move S -- starting address
0504 ; E -- ending address
0505 ; D -- destination address
0506 ; ii) Rela S -- source address
0507 ; D -- destination address
0508 ; iii) WRTape F -- file name

```

```

MFP-1
LOC OBJ CODE M STMT SOURCE STATEMENT
1509 ; S -- starting address
1510 ; E -- ending address
1511 ; I=) RDtape F -- file name
1512
1513 STEPDP:
1514 ;Display step buffer and its parameter name.
1515 ;Input: STATE
1516 ; STMINOR (parameter count)
1517 ;register destroyed: AF,BC,DE,HL
1518
043A CD5504 1519 CALL LOCSTBF ;Get parameter address
043D 5E 1520 LD E,(HL) ;Load parameter into DE
043E 23 1521 INC HL
043F 56 1522 LD D,(HL)
0440 CD6508 1523 CALL ADDRDP ;Convert this parameter to
;display format (4 digits)
;and store it into DISPBF+2
; DISPBF+3.
0443 21881F 1527 LD HL,DISPBF+2 ;Set 4 decimal points.
;From DISPBF+2 to DISPBF+5.
0446 0804 1529 LD B,4
0448 CD3404 1530 CALL SKPTP
044E 6F 1532 CALL LOCSTNA ;Get parameter name.
044F 2602 1533 LD L,A
;Pattern '-' for 2nd rightmost
;digit.
0451 22861F 1535 LD (DISPBF),HL
0454 C9 1536 RET
1537
1538 LOCSTBF:
1539 ;Get the location of parameter.
1540 ; address = STEPBF + STMINOR*2
1541 ;register destroyed: AF,HL
1542
0455 3A831F 1543 LD A,(STMINOR) ;Get parameter count.
0456 87 1544 ADD A,A ;Each parameter has 2 bytes.
0459 21AF1F 1545 LD HL,STEPBF ;Get base address.
045C 85 1546 ADD A,L
045D 6F 1547 LD L,A
045E C9 1548 RET
1549
1550 LOCSTNA:
1551 ;Get parameter name.
1552 ;Input: STATE, STMINOR
1553 ;Output: parameter name in A, and I flag.
1554
1555 ;register destroyed: AF,DE
045F 3A841F 1556 LD A,(STATE) ;Get STATE.
1557 ;Possible states are:
1558 ;4,5,6,7. (Move, Rel.
1559 ;RTape, RDtape)
0462 D604 1560 SUB 4 ;Change 4,5,6,7 to
;0,1,2,3.
0464 87 1562 ADD A,A ;Each state has 4 bytes for names.
0465 87 1563 ADD A,A
0466 11BC07 1564 LD DE,ST$TAS
0469 83 1565 ADD A,E
046A 5F 1566 LD E,A ;Now, DE contains the

```

```

MPP-I
LOC  OBJ CODE M  STMT SOURCE STATEMENT
                                1567                                ;address of 1st name
                                1568                                ;for each state.
046B 3AE31F 1569 LD A,(STMINOR) ;Get parameter count
046E 83 1570 ADD A,E ;DE <--- DE + A
046F 5F 1571 LD E,A
0470 1A 1572 LD A,(DE) ;Get parameter name.
0471 87 1573 OR A ;Change zero flag. If the
                                1574 ;returned pattern (in A) is
                                1575 ;zero, the '+' or '-' must
                                1576 ;have been pressed beyond legal
                                1577 ;parameter boundary. (Check if
                                1578 ;parameter name got from STEPTAB
                                1579 ;is zero)
0472 C9 1580 RET
                                1581 ;
                                1582 ;*****
                                1583 ; Register display format:
                                1584 ;
                                1585 ; i) X X X X Y Y -- State is READ. The numeric data
                                1586 ; entered is interpreted as
                                1587 ; register name.
                                1588 ; YY is the register name, the
                                1589 ; data of that register pair is
                                1590 ; XXXX.
                                1591 ;
                                1592 ; ii) X X X.X. Y Y or
                                1593 ; iii) X.X.X X Y Y -- State is REGDA. The unit of
                                1594 ; register modification is byte.
                                1595 ; The numeric data entered will
                                1596 ; change the byte with decimal
                                1597 ; points under it. Decimal points
                                1598 ; can be moved by '+' of '-' keys.
                                1599 ;
0473 3E08 1600 REGDP8:
0475 1602 1601 ; Display register and set STATE to 8.
                                1602 ;
                                1603 LD A,8 ;Next state = 8
                                1604 JR RGSTIN
                                1605 ;
                                1606 REGDP9:
                                1607 ; Display register and set STATE to 9.
                                1608 ;
0477 3E09 1609 LD A,9 ;Next state = 9
                                1610 ;
                                1611 RGSTIN:
                                1612 ; Update STATE by register A.
                                1613 ; Display user's register (count
                                1614 ; contained in STMINOR).
                                1615 ; register destroyed: AF,BC,DE,HL
                                1616 ;
0479 32E41F 1617 LD (STATE),A ;Update STATE.
047C 3AE31F 1618 LD A,(STMINOR) ;Get register count.
047F CB87 1619 RES G,A ;Registers are displayed by
                                1620 ;pair. Find the count
                                1621 ;of pair leader. (count of
                                1622 ;the lower one)
0481 47 1623 LD B,A ;Temporarily save A.
0482 CDAE04 1624 CALL RGNADP ;Find register count.

```

MPP-1				
LOC	OBJ CODE	M	STMT	SOURCE STATEMENT
			1625	;Store them into DISPF
			1626	;and DISPF+1.
0485	78		1627	LD A,B ;Restore A (register pair leader).
0486	CD8E04		1628	CALL LOCRG ;Get the address of
			1629	;user's register.
0489	5E		1630	LD E,(HL) ;Get register data. (2 bytes)
048A	23		1631	INC HL
048B	58		1632	LD D,(HL)
048C	ED53DE1F		1633	LD (ADSAVE),DE ;Convert them to display
			1634	;format and store into
			1635	;display buffer.
0490	CD6506		1636	CALL ADDRDP
0493	3AE41F		1637	LD A,(STATE)
0496	FE09		1638	CP 9 ;If STATE equals to 9 (RGDA),
			1639	;set 2 decimal points.
			1640	;Otherwise return here.
0498-	CO		1641	RET NE
0499	218B1F		1642	LD HL,DISPF+2
049C	3AE31F		1643	LD A,(STMINOR) ;Get register name.
049F	CB47		1644	BIT 0,A ;If this register is
			1645	;group leader, set decimal
			1646	;points of two central digits.
			1647	;Otherwise set two left digits.
04A1	2802		1648	JR Z,LOCPT
04A3	23		1649	INC HL
04A4	23		1650	INC HL
04A5	CBF0		1651	LOCPT SET 6,(HL) ;Set decimal points of
			1652	; (HL) and (HL+1)
04A7	23		1653	INC HL
04A8	CBF0		1654	SET 6,(HL)
04AA	CDC404		1655	CALL FCONV ;Convert user's flag (F,F')
			1656	;to binary display format.
04AD	C9		1657	RET
			1658	;
			1659	RGNADP:
			1660	; Get the patterns of register names and
			1661	; store them into DISPF and DISPF+1.
			1662	; Input: A contains register count of
			1663	; pair leader.
			1664	; register destroyed: AF,DE,HL
			1665	;
04AE	21D007		1666	LD HL,ROTAB ;Get address of pattern
			1667	;table.
04B1	85		1668	ADD A,L
04B2	8F		1669	LD L,A
04B3	5E		1670	LD E,(HL) ;Get first pattern.
04B4	23		1671	INC HL
04B5	56		1672	LD D,(HL) ;Get 2nd pattern.
04B6	ED33B61F		1673	LD (DISPF),DE
04BA	C9		1674	RET
			1675	;
			1676	LOCRGBF:
			1677	; Get the address of user's register.
			1678	; Register name contained in STMINOR.
			1679	; Destroys HL, AF.
			1680	;
04BB	3AE31F		1681	LD A,(STMINOR)
04BE	218C1F		1682	LOCRG LD HL,RGBF

MPF-I				
LOC	OBJ CODE	M	STMT	SOURCE STATEMENT
04C1	85		1683	ADD A,L
04C2	8F		1684	LD L,A
04C3	C9		1685	RET
			1686	
			1687	;FCOINV:
			1688	;Encode or decode user's flag register.
			1689	;STMINOR contains the name of the flag
			1690	;being displayed now.
			1691	;register destroyed: AF,BC,HL.
			1692	
04C4	3AE31F		1693	LD A,(STMINOR) ;Get register name.
04C7	B7		1694	OR A ;Clear carry flag.
04C8	1F		1695	RRA ;name of I register: 17H,
			1696	;name of IFF: 18H.
			1697	;Rotate right one bit, both
			1698	;become 0BH.
04C9	FE0B		1699	CP 0BH
04CB	2809		1700	JR Z,FLAGX ;Jump to FLAGX if
			1701	;I or IFF is being
			1702	;displayed now.
04CD	4F		1703	LD C,A ;Otherwise, mask out bit
			1704	;1 to bit 7 of user's IFF.
			1705	;IFF is only 1 bit, monitor
			1706	;use one byte to store it.
			1707	;masking out bit 1 <sup>st</sup> is to
			1708	;ignore the useless bits.
			1709	;This is done only when the
			1710	;user is not modifying IFF.
			1711	;if user is modifying IFF,
			1712	;monitor will display whatever
			1713	;he enters, even if bit 1 <sup>st</sup>
			1714	;are not all zero.
			1715	;A register is not changed
			1716	;after doing this.
04CE	21D21F		1717	LD HL,USERIF
04D1	7E		1718	LD A,(HL)
04D2	E801		1719	AND 0000001B
04D4	77		1720	LD (HL),A
04D5	79		1721	LD A,C
04D8	FE0C	FLAGX	1722	CP 0CH ;if STMINOR contains
			1723	;the name of SIXH, KPNC,
			1724	;SEKH' or KPNC', after
			1725	;rotating right one bit
			1726	;it will be greater than
			1727	;or equal to 0CH.
			1728	;Decode user's flag if it
			1729	;is not being modified now,
			1730	;encode it otherwise.
04DB	301F		1731	JR NC,FCOINV2
04DA	3ABC1F	FCOINV1	1732	LD A,(USERAF) ;Get user's F register.
04DD	CD1805		1733	CALL DECODE ;Decode upper 4 bits.
04E0	22D41F		1734	LD (FLAGH),HL
04E3	CD1805		1735	CALL DECODE ;Decode lower 4 bits.
04E6	22D81F		1736	LD (FLAGL),HL
04E9	3AC41F		1737	LD A,(USERF) ;Get user's F' register.
04EC	CD1805		1738	CALL DECODE
04EF	22D81F		1739	LD (FLAGRP),HL
04F2	CD1805		1740	CALL DECODE



MPP-I

LOC	OBJ CODE	M	STMT	SOURCE	STATEMENT
04F5	22DA1P		1741	LD	(FLAGLP),HL
04F8	C9		1742	RET	
04F9	2AD41P		1743	FCONV2 LD	HL,(FLAGH);Get the binary form
			1744		;of 4 upper bits of
			1745		;user's F register.
04FC	CD2305		1746	CALL	ENCODE ;Encode it.
04FF	2AD81P		1747	LD	HL,(FLAGL);Encode 4 lower bits.
0502	CD2305		1748	CALL	ENCODE
0505	32BC1P		1749	LD	(USERAF),A ;Save the encoded
			1750		;result into USERAF.
0508	2AD81P		1751	LD	HL,(FLAGHP);Encode P' register.
050B	CD2305		1752	CALL	ENCODE
050E	2ADA1P		1753	LD	HL,(FLAGLP)
0511	CD2305		1754	CALL	ENCODE
0514	32C41P		1755	LD	(UAFF),A
0517	C9		1756	RET	
			1757		
			1758		; DECODE:
			1759		; Decode bit 7 $\frac{1}{2}$ of A register.
			1760		; Each bit is extended to 4 bits.
			1761		; 0 becomes 0000, 1 becomes 0001.
			1762		; The output is stored in HL, which
			1763		; is 16 bits in length. Also, after
			1764		; execution, bit 7 $\frac{1}{2}$ of A register are
			1765		; bit 3 $\frac{1}{2}$ of A before execution.
			1766		; Register AF,B,HL are destroyed.
			1767		
0518	0604		1768	LD	B,4 ;Loop 4 times.
051A	29		1769	DRL4 ADD	HL,HL ;Clear rightmost 3
			1770		;bits of HL.
051B	29		1771	ADD	HL,HL
051C	29		1772	ADD	HL,HL
051D	07		1773	RLCA	
051E	ED6A		1774	ADC	HL,HL ;The 4th bit of HL
			1775		;is determined by carry
			1776		;flag, which is the MSB
			1777		;of A register.
0520	10F8		1778	DJNZ	DRL4
0522	C9		1779	RET	
			1780		
			1781		; ENCODE:
			1782		; Encode HL register. Each 4 bits of HL
			1783		; are encoded to 1 bit. 0000 become 0,
			1784		; 0001 become 1. The result is stored
			1785		; in bit 3 $\frac{1}{2}$ of A register. Also, after
			1786		; execution, bit 7 $\frac{1}{2}$ of A are bit 3 $\frac{1}{2}$
			1787		; before execution.
			1788		; Registers AF,B,HL are destroyed.
			1789		
0523	0604		1790	LD	B,4 ;Loop 4 times.
0525	29		1791	RHL4 ADD	HL,HL ;Shift HL left 4 bits.
			1792		;Bit 12 of HL will be
			1793		;shifted into carry flag.
0526	29		1794	ADD	HL,HL
0527	29		1796	ADD	HL,HL
0528	29		1796	ADD	HL,HL
0529	17		1797	RLA	
			1798		;Rotate carry flag into
					;A register.

MPP-I

LOC	OBJ CODE	M	STMT	SOURCE	STATEMENT
052A	10F9		1799	DJNZ	ERL4
052C	C9		1800	RET	
			1801		
			1802		.....
			1803	SUM1:	
			1804		; Calculate the sum of the data in a memory
			1805		; block. The starting and ending address
			1806		; of this block are stored in STEPB*2 & STEPB*4.
			1807		; Registers AF,BC,DE,HL are destroyed.
			1808		
052D	CD3A05		1809	CALL	GETPTR ;Get parameters from
			1810		;step buffer.
0530	D8		1811	RET	C ;Return if the parameters
			1812		;are illegal.
			1813	SUM:	
			1814		; Calculate the sum of a memory block.
			1815		; HL contains the starting address of
			1816		; this block, BC contains the length.
			1817		; The result is stored in A. Registers
			1818		; AF,BC,HL are destroyed.
			1819		
0531	AF		1820	XOR	A ;Clear A.
0532	86		1821	SUMCAL	ADD A,(HL) ;Add
0533	EDA1		1822	CPI	
0535	EA3205		1823	JP	PE,SUMCAL
0536	B7		1824	OR	A ;Clear flags.
0539	C9		1825	RET	
			1826		
			1827	GETPTR:	
			1828		; Get parameters from step buffer.
			1829		; Input: (STEPB*2) and (STEPB*3) contain
			1830		; starting address.
			1831		; (STEPB*4) and (STEPB*5) contain
			1832		; ending address.
			1833		; Output: HL register contains the starting
			1834		; address.
			1835		; BC register contains the length.
			1836		; Carry flag 0 -- BC positive
			1837		; 1 -- BC negative
			1838		; Destroyed reg.: AF,BC,DE,HL.
			1839		
053A	21B11P		1840	LD	HL,STEPB*2
053D	5E		1841	GETP	LD B,(HL) ;Load starting address
			1842		;into DE.
053E	23		1843	INC	HL
053F	56		1844	LD	D,(HL)
0540	23		1845	INC	HL
0541	4E		1846	LD	C,(HL)
0542	23		1847	INC	HL ;Load ending address
			1848		;into HL.
0543	66		1849	LD	H,(HL)
0544	69		1850	LD	L,C
0545	B7		1851	OR	A ;Clear carry flag.
0546	ED52		1852	SBC	HL,DE ;Find difference.
			1853		;Carry flag is changed here.
054B	4D		1854	LD	C,L
0549	44		1855	LD	B,H
054A	03		1856	INC	BC ;Now BC contains the

```

MPP-1
LOC OBJ CODE M STMT SOURCE STATEMENT
054B EB 1857 ;length.
1858 EX DE,HL ;Now HL contains the
1859 ;starting address.
054C C9 1860 RET
1861 ;
1862 TAPIN:
1863 ; Load a memory block from tape.
1864 ; Input: HL -- starting address of the block
1865 ; BC -- length of the block
1866 ; Output: Carry flag, 1 -- reading error
1867 ; 0 -- no error
1868 ; Destroyed reg. -- AF,BC,DE,HL,AF',BC',DE',HL'
1869
054D AF 1870 XOR A ;Clear carry flag.
1871 ;At beginning, the reading is
1872 ;no error.
054E 08 1873 EX AF,AF'
054F CDSA05 1874 TLOOP CALL GETBYTE ;Read 1 byte from tape.
0552 73 1875 LD (HL),E ;Store it into memory.
0553 EDA1 1876 CPI ;
0555 EA4F05 1877 JP PE,TLOOP ;Loop until length
;is zero.
0558 08 1878
0559 C9 1879 EX AF,AF'
1880 RET
1881 ;
1882 GETBYTE:
1883 ; Read one byte from tape.
1884 ; Output: E -- data read
1885 ; Carry of F', 1 -- reading error
1886 ; 0 -- no error
1887 ; Destroy reg. -- AF,DE,AF',BC',DE',HL'
1888 ; Byte format:
1889 ; start bit bit bit bit bit bit bit bit stop
1890 ; bit 0 1 2 3 4 5 6 7 bit
1891 ;
1892
055A CD8B05 1893 CALL GETBIT ;Get start bit.
055D 1608 1894 LD D,B ;Loop 8 times.
055F CD8B05 1895 BLOOP CALL GETBIT ;Get one data bit.
;Result in carry flag.
0562 CB18 1896 RR E ;Rotate it into E.
0564 15 1897 DEC D
0565 20F8 1898 JR NZ,BLOOP
0567 CD8B05 1899 CALL GETBIT ;Get stop bit.
056A C9 1900 RET
1901 ;
1902 ;
1903 ;
1904 GETBIT:
1905 ; Read one bit from tape.
1906 ; Output: Carry of F', 0 -- this bit is 0
1907 ; 1 -- this bit is 1
1908 ; Carry of F', 1 -- reading error
1909 ; 0 -- no error
1910 ; Destroyed reg. -- AF,AF',BC',DE',HL'
1911 ; Bit format:
1912 ;
1913 ; 0 -- 2K Hz 8 cycles + 1K Hz 2 cycles.
1914 ; 1 -- 2K Hz 4 cycles + 1K Hz 4 cycles.

```

MPF-I

LOC	OBJ CODE	M	SIMT	SOURCE	STATEMENT
					1915
056B	D9			EKK	;Save HL,BC,DR registers
					1917
					1918 ; The tape-bit format of both 0 and 1 are
					1919 ; of the same form: high freq part + low freq part.
					1920 ; The difference between 0 and 1 is the
					1921 ; number high freq cycles and low freq
					1922 ; cycles. Thus, a high freq period may has
					1923 ; two meanings:
					1924 ; 1) It is used to count the number of high
					1925 ; freq cycles of the current tape-bit;
					1926 ; 1) If a high freq period is detected
					1927 ; immediately after a low freq period, then
					1928 ; this period is the first cycle of next
					1929 ; tape-bit and is used as a terminator of the
					1930 ; last tape-bit.
					1931
					1932 ; Bit 0 of H register is used to indicate the usage
					1933 ; of a high freq period. If this bit is zero, high
					1934 ; freq period causes counter increment for the current
					1935 ; tape-bit. If the high freq part has passed, bit 0
					1936 ; of H is set and the next high freq period will be used
					1937 ; as a terminator.
					1938 ; L register is used to up/down count the number of periods.
					1939 ; when a high freq period is read, L is increased by
					1940 ; 1; when a low freq period is read, L is decreased
					1941 ; by 2. (The time duration for each count is 0.5 ms.)
					1942 ; At the end of a tape-bit, positive and negative L
					1943 ; stand for 0 and 1 respectively.
					1944
056C	210000			LD	HL,0 ;Clear bit 0 of H,
					1946 ;Set L to 0.
056F	CD8C05			CALL	PERIOD ;Read one period.
0572	14			INC	D ;The next 2 instructions
					1949 ;check if D is zero. Carry
					1950 ;flag is not affected.
0573	15			DEC	D
0574	2011			JR	NZ,TERR ;If D is not zero, jump
					1953 ;to error routine TERR.
					1954 ;(Because the period is too
					1955 ;much longer than that of 1K Hz.)
0576	3806			JR	C,SHORTP ;If the period is short
					1957 ;(2K Hz), jump to SHORTP.
0578	2D			DEC	L ;The period is 1K Hz,
					1959 ;decrease L by 2. And set
					1960 ;bit 0 of H to indicate this
					1961 ;tape-bit has passed high freq
					1962 ;part and reaches its low freq part.
0579	2D			DEC	L
057A	C8C4			SET	0,H
057C	18F1			JR	COUNT
057E	2C			SHORTP	INC
					1966 ;The period is 2 K Hz,
					1967 ;increase L by 1.
057F	CB44			BIT	0,H
					1968 ;If the tape-bit has passed
					1969 ;its high freq part, high frequency
					1970 ;means this bit is all over and
					1971 ;next bit has started.
0581	28EC			JR	I,COUNT
					1972

```

MPP I
LOC OBJ CODE M STMT SOURCE STATEMENT
0583 CB15 1973 ;L = (# of 2K period) - 2*(# of 1K period)
1974 RL L
1975 ; 0 --- NCarry (L positive)
1976 ; 1 --- Carry (L negative)
1977 ;The positive or negative sign of
1978 ;L corresponds to the tape-bit data.
1979 ;'RL L' will shift the sign bit of
1980 ;L into carry flag. After this
1981 ;instruction, the carry flag
1982 ;contains the tape-bit.
0585 D9 1983 EXX ;Restore BC',DE',HL'
0586 C9 1984 RET
0587 08 1985 TERR EX AF,AF'
0588 37 1986 SCP ;Set carry flag of P' to indicate error.
0589 08 1987 EX AF,AF'
058A D9 1988 EXX
058B C9 1989 RET
1990 ;
1991 PERIOD:
1992 ; Wait the tape to pass one period.
1993 ; The time duration is stored in DE. The
1994 ; unit is loop count. Typical value for
1995 ; 2K Hz is 28, for 1K Hz is 56.
1996 ; Use (56+28)/2 as threshold. The returned
1997 ; result is in carry flag. (1K --- NC, 2K --- C)
1998 ; Register AF and DE are destroyed.
1999
058C 110000 2000 LD DE,0
058F D800 2001 LOOPH IN A,(KIN) ;Bit 7 of port A is Tapein.
0591 13 2002 INC DE
0592 17 2003 RLA
0593 38FA 2004 JR C,LOOPH ;Loop until input goes low.
0595 3E7F 2005 LD A,11111111B ;Echo the tape input to
2006 ;speaker on MPP-I.
0597 D302 2007 OUT (DIGIT),A
0598 D800 2008 LOOPL IN A,(KIN)
059B 13 2009 INC DE
059C 17 2010 RLA
059D 30FA 2011 JR NC,LOOPL ;Loop until input goes high.
059F 3E7F 2012 LD A,01111111B ;Echo the tape input to
2013 ;speaker on MPP-I.
05A1 D302 2014 OUT (DIGIT),A
05A3 7B 2015 LD A,E ;Compare the result with
2016 ;the threshold.
05A4 FE2A 2017 CP MPERIOD
05A6 C9 2018 RET
2019 ;
2020 ;*****
2021 TAPEOUT:
2022 ; Output a memory block to tape.
2023 ; Input: HL -- starting address of the block
2024 ; BC -- length of the block
2025 ; Destroyed reg. -- AF,BC,DE,HL,BC',DE',HL'
2026
05A7 5E 2027 LD E,(HL) ;Get the data.
05A8 CDB103 2028 CALL OUTBYTE ;Output to tape.
05AD EDA1 2029 CPI
05AD EAA705 2030 JP PE,TAPEOUT ;Loop until finished.

```

```

MPP-I
LOC  OBJ CODE M  STMT SOURCE STATEMENT
05B0  C9          2031      RET
          2032      ;
          2033      OUTBYTE:
          2034      ; Output one byte to tape. For tape-byte
          2035      ; format, see comments on GETBYTE.
          2036      ; Input: E -- data
          2037      ; Destroyed reg. -- AF,DE,BC',DE',HL'
          2038
05B1  1608        2039      LD   D,8      ;Loop 8 times.
05B3  B7          2040      OR   A        ;Clear carry flag.
05B4  CDC405      2041      CALL OUTBIT  ;Output start bit.
05B7  CB1B       2042      OLOOP RR   E      ;Rotate data into carry
05B9  CDC405      2043      CALL OUTBIT  ;Output the carry
05BC  15         2044      DEC   D
05BD  20F8       2045      JR    NZ,OLOOP
05BF  37         2046      SCF        ;Set carry flag.
05C0  CDC405      2047      CALL OUTBIT  ;Output stop bit
05C3  C9          2048      RET
          2049      ;
          2050      OUTBIT:
          2051      ; Output one bit to tape.
          2052      ; Input: data in carry flag.
          2053      ; Destroyed reg. -- AF,BC',DE',HL'
05C4  D9          2054      EXX        ;Save BC,DE,HL.
05C5  2800        2055      LD   H,0
05C7  3809        2056      JR   C,OUT1  ;If data=1, output 1.
          2057      OUT0: ;2K 8 cycles, 1K 2 cycles.
05C9  2E08        2058      LD   L,ZERO 2K
05CB  CDE205      2059      CALL TONE2K
05CE  2E02        2060      LD   L,ZERO 1K
05D0  1807        2061      JR   BITEND
          2062      ;
          2063      OUT1: ;2K 4 cycles, 1K 4 cycles.
05D2  2E04        2064      LD   L,ONE 2K
05D4  CDE205      2065      CALL TONE2K
05D7  2E04        2066      LD   L,ONE 1K
05D9  CDDE05      2067      BITEND CALL TONE1K
05DB  D9          2068      EXX        ;Restore registers.
05DD  C9          2069      RET
          2070      ;
          2071      ;.....
          2072      ;
          2073      ;          UTILITY SUBROUTINE
          2074      ;
          2075      ;.....
          2076      ;
          2077      ; Function: Generate square wave to the MIC & speaker
          2078      ; on MPP--1.
          2079      ; Input : C -- period = 2*(44+13*C) clock states.
          2080      ; HL -- number of periods.
          2081      ; Output: none.
          2082      ; Destroyed reg.: AF, B(C), DE, HL.
          2083      ; Call: none.
          2084
          2085      TONE1K:
05DE  0E41        2086      LD   C,F16HZ
05E0  1802        2087      JR   TONE
          2088      TONE2K:

```

```

MPP-I
LOC  OBJ CODE M  STMT SOURCE STATEMENT
05E2  081F      2089      LD      C,P2KHZ
05E4  29        2090      TONE:   LD      C,P2KHZ ;Half period: 44*13°C states
05E5  110100     2091      ADD     HL,HL ;Double for half-cycle count
05E6  3E7F      2092      LD      DE,1
05E7  0302      2093      LD      A,OPFH
05E8  D302      2094      SQWAVE  OUT     (DIGIT),A ;Bit-7 tapeout
05E9  41        2095      LD      B,C
05ED  10FE      2096      DJNZ   $ ;Half period delay
05EF  EE80      2097      XOR    80H ;Toggle output
05F1  ED52      2098      SBC    HL,DE ;Decrement one count
05F3  20F5      2099      JR     NZ,SQWAVE
05F5  C9        2100      RET
2101      ;
2102      ;
2103      ;Function: check if a memory address is in RAM.
2104      ;Input: HL -- address to be check.
2105      ;Output: Zero flag -- 0, ROM or nonexistant;
2106      ;Destroyed reg.: AP, 1, RAM.
2107      ;Destroyed reg.: AP.
2108      ;Call: none
2109      ;
2110  RANCHK:
2111      LD      A,(HL)
2112      CPL
2113      LD      (HL),A
2114      LD      A,(HL)
2115      CPL
2116      LD      (HL),A
2117      CP      (HL)
2118      RET
2119      ;
2120      ;
2121      ;Function: Scan the keyboard and display. Loop until
2122      ;a key is detected. If the some key is already
2123      ;pressed when this routine starts execution,
2124      ;return when next key is entered.
2125      ;Input: IX points to the buffer contains display patterns.
2126      ;6 LEDs require 6 byte data. (IX) contains the
2127      ;pattern for rightmost LED. (IX+5) contains the
2128      ;pattern for leftmost LED.
2129      ;Output: internal code of the key pressed.
2130      ;Destroyed reg. : AP, B, HL, AP', BC', DE'.
2131      ;All other registers except IY are also
2132      ;changed during execution, but they are
2133      ;restored before return.
2134      ;Call: SCAN1
2135      ;
2136  SCAN:
2137      PUSH  IX ;Save IX.
2138      LD      HL,TEST
2139      BIT    7,(HL) ;This bit is set if the use
2140      ;has entered illegal key. The
2141      ;display will be disabled as
2142      ;a warning to the user. This
2143      ;is done by replacing the display
2144      ;buffer pointer IX by BLANK.
2145      JR     Z,SCPR
2146      LD      IX,BLANK

```

```

MPP-I
LOC  OBJ CODE M STMT SOURCE STATEMENT
      2147
      2148 ; Wait until all keys are released for 40 ms.
      2149 ; (The execution time of SCAN1 is 10 ms,
      2150 ; 40 = 10 * 4.)
      2151
0608 0604 2152 SCPRB LD B,4
0600 CD2406 2153 SCNK CALL SCAN1
0610 30F9 2154 JR NC,SCPRB ;If any key is pressed, re-load
      2155 ;the debounce counter B by 4.
0612 10FB 2158 DJNZ SCNK
0614 CBBE 2157 RES 7,(HL) ;Clear error-flag.
0616 DDE1 2158 POP IX ;Restore original IX.
      2159
      2160 ; Loop until any key is pressed.
      2161
0618 CD2406 2162 SCLOOP CALL SCAN1
061B 38FB 2163 JR C,SCLOOP
      2164
      2165 ; Convert the key-position-code returned by SCAN1 to
      2166 ; key-internal-code. This is done by table-lookup.
      2167 ; The table used is KEYTAB.
      2168
061D 217B07 2169 KEYMAP LD HL,KEYTAB
0620 85 2170 ADD A,L
0621 6F 2171 LD L,A
0622 7E 2172 LD A,(HL)
0623 C9 2173 RET
      2174
      2175 ;*****
      2176 ; Function: Scan keyboard and display one cycle.
      2177 ; Total execution time is about 10 ms (exactly
      2178 ; 9.95 ms, 17812 clock states @ 1.79 MHz).
      2179 ; Input: Same as SCAN.
      2180 ; Output: 1) no key during one scan
      2181 ; Carry flag -- 1
      2182 ; 2) key pressed during one scan
      2183 ; Carry flag -- 0,
      2184 ; A -- position code of the key pressed.
      2185 ; If more than one key is pressed, A
      2186 ; contains the largest position-code.
      2187 ; (This key is the last key scanned.)
      2188 ; Destroyed reg: AP, AP', BC', DE'. (see comments on SCAN)
      2189 ; Call: none.
      2190
      2191 SCAN1:
      2192 ;In hardware, the display and keyboard are
      2193 ;arranged as a 6 by 6 matrix. Each column
      2194 ;corresponds to one LED and six key buttons.
      2195 ;in normal operation, at most one column is
      2196 ;active. The pattern of the active LED is the
      2197 ;data output on port C of 8255 I. The data input
      2198 ;from bit 0p5 of port A are the status of key
      2199 ;buttons in the active column. All signals on
      2200 ;I/O port are active low.
      2201
      2202 SCF ;Set carry flag.
0624 37 2202 . SK AP,AP'
0625 08 2203 . SK
0626 D9 2204 . SKX

```



MPP I

LOC OBJ CODE M STMT SOURCE STATEMENT

```

2205
2206 ;Carry flag of P' is used to return the status of
2207 ;the keyboard. If any key is pressed during one
2208 ;scan, the flag is reset; otherwise, it is set.
2209 ;Initially, this flag is set. A' register is used
2210 ;to store the position-code of the key pressed.
2211 ;In this routine, 36 key positions are checked one
2212 ;by one. C register contains the code of the key
2213 ;being checked. The value of C is 0 at the beginning.
2214 ;and is increased by 1 after each check. So the code
2215 ;ranges from 0 to 23H (total 36 positions). On each
2216 ;check, if the input bit is 0 (key pressed), C register
2217 ;is copied into A'. The carry flag of P' is set also.
2218 ;When some key is detected, the key positions after
2219 ;this key will still be checked. So if more than
2220 ;one key are pressed during one scan, the code of the
2221 ;last one will be returned.
2222
0627 0E00 2223 LD C,0 ;Initial position code
0629 1EC1 2224 LD E,11000001B ;Scan from rightmost digit.
062B 2608 2225 LD H,6
;to the active column.
062D 7B 2226 KCOL LD A,E
062E D302 2228 OUT (DIGIT),A ;Activate one column.
0630 DD07E00 2229 LD A,(IX)
0633 D301 2230 OUT (SEG7),A
0635 05C9 2231 LD B,COLDEL
0637 10FE 2232 DJNZ $ ;Delay 1.5 ms per digit.
0639 AF 2233 XOR A ;Deactivate all display segments
063A D301 2234 OUT (SEG7),A
063C 7B 2235 LD A,E
063D 2F 2236 CPL
063E F6C0 2237 OR 11000000B
0640 D302 2238 OUT (DIGIT),A
0642 0608 2239 LD B,6 ;Each column has 6 keys.
0644 D800 2240 IN A,(KIN) ;Now, bit 0p3 of A contain
;the status of the 6 keys
;in the active column.
0646 57 2241 LD D,A
0647 CB1A 2243 KROW RR D ;Store A into D.
;Rotate D 1 bit right, bit 0
;of D will be rotated into
;carry flag.
0649 3802 2247 JR C,NOKEY ;Skip next 2 instructions
;if the key is not pressed.
;The next 2 instructions
;store the current position-code
;into A' and reset carry flag
;of P' register.
064B 79 2253 LD A,C ;Key-in, get key position.
064C 08 2254 BK AF,AF' ;Save A & Carry in AF'.
064D 0C 2255 INC C ;Increase current key-code by 1.
064E 10F7 2258 DJNZ KROW ;loop until 6 keys in the
;active column are all checked.
2257
0650 DD23 2258 INC IX
0652 7B 2259 LD A,E
0653 863F 2260 AND 00111111B
0655 C907 2281 RLC A
0657 F6C0 2282 OR 11000000B

```

MPP-I

LOC	OBJ CODE	M	STMT	SOURCE	STATEMENT
0659	5F		2263	LD	R,A
065A	25		2264	DEC	H
065B	20D0		2265	JR	NZ,KCOL
065D	11FAPF		2266	LD	DE,-6
0660	DD19		2267	ADD	IX,DE ;Get original IX.
0662	D9		2268	RXK	
0663	08		2269	RK	AF,AF'
0664	C9		2270	RET	
			2271	:	;
			2272	:	Function: Convert the 2 byte data stored in DE to
			2273	:	7-segment display format. The output is stored
			2274	:	in the address field of DISPF (display buffer),
			2275	:	most significant digit in DISPF+5.
			2276	:	This routine is usually used by monitor only.
			2277	:	Destroyed reg: AF, HL.
			2278	:	Call: HEX7SG
			2280	:	
			2281	ADDMDP:	
0665	21881F		2282	LD	HL,DISPF+2
0668	7B		2283	LD	A,R
0669	CD7806		2284	CALL	HEX7SG
066C	7A		2285	LD	A,D
066D	CD7806		2286	CALL	HEX7SG
0670	C9		2287	RET	
			2288	:	;
			2289	:	Function: Convert the data stored in A to 7-segment
			2290	:	display format. 1 byte is converted to 2
			2291	:	digits. The result is stored in the data
			2292	:	field of display buffer (DISPF).
			2293	:	This routine is usually used by monitor only.
			2294	:	Destroyed reg: AF, HL.
			2295	:	Call: HEX7SG
			2297	:	
			2298	DATADP:	
0671	21881F		2299	LD	HL,DISPF
0674	CD7806		2300	CALL	HEX7SG
0677	C9		2301	RET	
			2302	:	;
			2303	:	Function: Convert binary data to 7-segment display
			2304	:	format.
			2305	:	Input: 1 byte in A register.
			2306	:	HL points to the result buffer.
			2307	:	Output: Pattern for 2 digits. Low order digit in (HL),
			2308	:	high order digit in (HL+1).
			2309	:	HL becomes HL+2.
			2310	:	Destroyed reg: AF, HL.
			2311	:	Call: HEX7
			2312	:	
			2313	:	
			2314	HEX7SG:	
0678	F5		2315	PUSH	AF
0679	CD8906		2316	CALL	HEX7
067C	77		2317	LD	(HL),A
067D	23		2318	INC	HL
067E	F1		2319	POP	AF
067F	0F		2320	RRCA	

```

MPP-I
LOC OBJ CODE M STMT SOURCE STATEMENT
0680 OF 2321 BRCA
0681 OF 2322 BRCA
0682 OF 2323 BRCA
0683 CDS906 2324 CALL HEX7
0688 77 2325 LD (HL),A
0687 23 2326 INC HL
0686 C9 2327 RET
2328
2329 ;
2330 ; Function: Convert binary data to 7-segment display
2331 ; format.
2332 ; Input: A -- LSB 4 bits contains the binary data
2333 ; Output: A -- display pattern for 1 digit.
2334 ; Destroyed reg: AF
2335 ; Call: none
2336
2337 HEX7:
0689 E5 2338 PUSH HL
068A 21F007 2339 LD HL,SEG7AB
068D E80F 2340 AND OFH
068F 85 2341 ADD A,L
0690 8F 2342 LD L,A
0691 7E 2343 LD A,(HL)
0692 E1 2344 POP HL
0693 C9 2345 RET
2346 ;
2347 ;
2348 ;
2349 ; Function: RAM 1800-1FFF self-check.
2350 ; Input: none
2351 ; Output: none
2352 ; Destroyed reg: AF, BC, HL
2353 ; Call: RAMCHK
2354
2355 RAMTEST:
0694 210018 2356 LD HL,1800H
0697 010008 2357 LD BC,800H
069A CDF805 2358 RAMT CALL RAMCHK
069E 2801 2359 JR $,TWRAT ;If error.
069F 78 2360 HALT
06A0 KDA1 2361 TNEXT CPI
06A2 E8A006 2362 JP PE,RAMT
06A5 C7 2363 RST 0 ;Display 'uPF--1'.
2364 ;
2365 ;
2366 ; Monitor ROM self-check. Add the data of address
2367 ; 0000 to 0800. If the sum equals to 0. Heat the monitor
2368 ; and display 'uPF--1'. If the sum is not 0, which
2369 ; indicates error, HALT.
2370 ; Input: none.
2371 ; Output: none.
2372 ; Destroyed registers: AF, BC, HL.
2373 ; Call: SUM
2374
2375 ROMTEST:
06A6 210000 2376 LD HL,0
06A9 010008 2377 LD BC,800H
06AC CD3105 2378 CALL SUM

```

MPP-I

LOC	OBJ CODE	M	STMT	SOURCE	STATEMENT
06AF	2801		2379	JR	Z,SUMOK
06B1	76		2380	HALT	;if error.
06B2	C7		2381	SUMOK RST	0 ;Display 'uPF-1'.
06B3	32E51F		2382	INI3 LD	(POWERUP);A ;Load power-code into
			2383		;(POWERUP). The monitor
			2384		;uses the location to decide
			2385		;whether a reset signal is
			2386		;on power-up.
06B6	3E55		2387	LD	A,55H
06B6	32F01F		2388	LD	(BEEPSET),A
06B8	3E44		2389	LD	A,44H
06BD	32F11F		2390	LD	(FBEEP),A ;Beep frequency when key is
			2391		;pressed.
06C0	21F21F		2392	LD	HL,TBEEP
06C3	362F		2393	LD	(HL),2FH ;Time duration of beep when
06C5	23		2394	INC	HL
06C6	3600		2395	LD	(HL),0
			2396		;key is pressed.
06C8	C3D803		2397	JP	INI4
			2398		
06CB	F5		2399	BEEP PUSH	AP
06CC	21F11F		2400	LD	HL,FBEEP
06CF	4E		2401	LD	C,(HL)
06D0	2AF21F		2402	LD	HL,(TBEEP)
06D3	3AF01F		2403	LD	A,(BEEPSET)
06D6	FE55		2404	CP	55H
06D8	2003		2405	JR	NZ,NOTONE ;There is no beep sound when
			2406		;the key is pressed if data
			2407		;of (BEEPSET) is not 55H
06DA	CDE405		2408	CALL	STONE
			2409	NOTONE:	
06DD	F1		2410	POP	AP
06DE	C3E900		2411	JP	KEYEXEC ;After a key is detected,determine
			2412		;what action should the monitor take.
			2413		;KEYEXEC uses the next 3 factors
			2414		;to get the entry point of proper
			2415		;service routine:key-code, STATE
			2416		;and STMINOR (Minor-State).
			2417		; Below are the branch tables for each key and
			2418		; state. The first entry of each table is
			2419		; a base address, other entries are the offset to
			2420		; this address. Offset is only one byte long.
			2421		; which is much shorter than the 2-byte address.
			2422		; This can save the monitor code space.
			2423		
0737			2424	KSUBFON ORG	0737H
0737	1B01		2425	DEFW	KINC
0739	00		2426	DEFB	-KINC+KINC
073A	05		2427	DEFB	-KINC+KDEC
073B	0A		2428	DEFB	-KINC+KGO
073C	0F		2429	DEFB	-KINC+KSTEP
073D	1A		2430	DEFB	-KINC+KDATA
073E	2C		2431	DEFB	-KINC+KSHR
073F	42		2432	DEFB	-KINC+KINS
0740	7B		2433	DEFB	-KINC+KDEL
0741	C201		2434	KFUN DEFW	KPC
0743	00		2435	DEFB	-KPC+KPC
0744	1C		2436	DEFB	-KPC+KADDR

MPP-I  
LOC OBJ CODE M STMT SOURCE STATEMENT

0745	0A		2437	DEFB	-EPC+ECBR		
0746	14		2438	DEFB	-EPC+ERBS		
0747	20		2439	DEFB	-EPC+EMV		
0748	20		2440	DEFB	-EPC+ERL		
0749	26		2441	DEFB	-EPC+ERT		
074A	28		2442	DEFB	-EPC+ERT		
074B	EC01		2443	HTAB	DEFW	HFIX	
074D	00		2444	DEFB	-HFIX+HFIX		
074E	18		2445	DEFB	-HFIX+HAD		
074F	03		2446	DEFB	-HFIX+HDA		
0750	26		2447	DEFB	-HFIX+HRGFIX		
0751	34		2448	DEFB	-HFIX+IMV		
0752	34		2449	DEFB	-HFIX+HRL		
0753	34		2450	DEFB	-HFIX+HWT		
0754	34		2451	DEFB	-HFIX+HRT		
0755	26		2452	DEFB	-HFIX+HRGAD		
0756	44		2453	DEFB	-HFIX+HRGDA		
0757	3D02		2454	ITAB	DEFW	IFIX	
0759	00		2455	DEFB	-IFIX+IFIX		
075A	03		2456	DEFB	-IFIX+IAD		
075B	03		2457	DEFB	-IFIX+IDA		
075C	00		2458	DEFB	-IFIX+IRGFIX		
075D	0E		2459	DEFB	-IFIX+IMV		
075E	0E		2460	DEFB	-IFIX+IRL		
075F	0E		2461	DEFB	-IFIX+IWT		
0760	0E		2462	DEFB	-IFIX+IRT		
0761	1F		2463	DEFB	-IFIX+IRGAD		
0762	1F		2464	DEFB	-IFIX+IRGDA		
0763	5B02		2465	DTAB	DEFW	DFIX	
0765	00		2466	DEFB	-DFIX+DFIX		
0766	03		2467	DEFB	-DFIX+DAD		
0767	03		2468	DEFB	-DFIX+DDA		
0768	00		2469	DEFB	-DFIX+DRGFIX		
0769	0E		2470	DEFB	-DFIX+DMV		
076A	0E		2471	DEFB	-DFIX+DBL		
076B	0E		2472	DEFB	-DFIX+DBT		
076C	0E		2473	DEFB	-DFIX+DBT		
076D	1F		2474	DEFB	-DFIX+DRGAD		
076E	1F		2475	DEFB	-DFIX+DRGDA		
076F	9902		2476	GTAB	DEFW	GFIX	
0771	00		2477	DEFB	-GFIX+GFIX		
0772	03		2478	DEFB	-GFIX+GAD		
0773	03		2479	DEFB	-GFIX+GDA		
0774	00		2480	DEFB	-GFIX+GRGFIX		
0775	4B		2481	DEFB	-GFIX+GMV		
0776	8D		2482	DEFB	-GFIX+GRL		
0777	8B		2483	DEFB	-GFIX+GWT		
0778	C1		2484	DEFB	-GFIX+GRT		
0779	00		2485	DEFB	-GFIX+GRGAD		
077A	00		2486	DEFB	-GFIX+GRGDA		
			2487				
			2488				
			2489				
			2490	KEYTAB:			
077B	03		2491	K0	DEFB	03H	HEX 3
077C	07		2492	K1	DEFB	07H	HEX 7
077D	0B		2493	K2	DEFB	0BH	HEX B
077E	0F		2494	K3	DEFB	0FH	HEX F

; Key-position-code to key-internal-code conversion table.

MPP-I							
LOC	OBJ CODE	M	STMT	SOURCE	STATEMENT		
077F	20		2485	K4	DEFB	20H	:NOT USED
0780	21		2486	K5	DEFB	21H	:NOT USED
0781	02		2487	K6	DEFB	02H	:HEX 2
0782	06		2488	K7	DEFB	06H	:HEX 6
0783	0A		2489	K8	DEFB	0AH	:HEX A
0784	0E		2500	K9	DEFB	0EH	:HEX E
0785	22		2501	K0A	DEFB	22H	:NOT USED
0786	23		2502	K0B	DEFB	23H	:NOT USED
0787	01		2503	K0C	DEFB	01H	:HEX 1
0788	05		2504	K0D	DEFB	05H	:HEX 5
0789	09		2505	K0E	DEFB	09H	:HEX 9
078A	0D		2506	K0F	DEFB	0DH	:HEX D
078B	13		2507	K10	DEFB	13H	:STEP
078C	1F		2508	K11	DEFB	1FH	:TAPERD
078D	00		2509	K12	DEFB	00H	:HEX 0
078E	04		2510	K13	DEFB	04H	:HEX 4
078F	08		2511	K14	DEFB	08H	:HEX 8
0790	0C		2512	K15	DEFB	0CH	:HEX C
0791	12		2513	K16	DEFB	12H	:D0
0792	1E		2514	K17	DEFB	1EH	:TAPEWR
0793	1A		2515	K18	DEFB	1AH	:CBR
0794	18		2516	K19	DEFB	18H	:PC
0795	1B		2517	K1A	DEFB	1BH	:R00
0796	19		2518	K1B	DEFB	19H	:ADDR
0797	17		2519	K1C	DEFB	17H	:DEL
0798	1D		2520	K1D	DEFB	1DH	:RELA
0799	15		2521	K1E	DEFB	15H	:SBR
079A	11		2522	K1F	DEFB	11H	:-
079B	14		2523	K20	DEFB	14H	:DATA
079C	10		2524	K21	DEFB	10H	:+
079D	16		2525	K22	DEFB	16H	:INS
079E	1C		2526	K23	DEFB	1CH	:MOVE
			2527	:	:	:	:
			2528	:	:	:	:
			2529	:	:	:	:
			2530	:	:	:	:
079F	30		2531	MPP_I	DEFB	030H	: 'I'
07A0	02		2532		DEFB	002H	: '1'
07A1	02		2533		DEFB	002H	: '1'
07A2	0F		2534		DEFB	0FH	: 'F'
07A3	1F		2535		DEFB	1FH	: 'F'
07A4	A1		2536		DEFB	0A1H	: 'U'
07A5	00		2537	BLANK	DEFB	0	:
07A6	00		2538		DEFB	0	:
07A7	00		2539		DEFB	0	:
07A8	00		2540		DEFB	0	:
07A9	00		2541	ERR	DEFB	0	:
07AA	00		2542		DEFB	0	:
07AB	03		2543		DEFB	3	: 'R'
07AC	03		2544		DEFB	3	: 'R'
07AD	8F		2545		DEFB	8FH	: 'E'
07AE	02		2546		DEFB	2	: '1'
07AF	1F		2547	SYS_SP	DEFB	1FH	: 'P'
07B0	AE		2548		DEFB	0AEH	: 'S'
07B1	02		2549		DEFB	02H	: '1'
07B2	AE		2550		DEFB	0AEH	: 'S'
07B3	B6		2551		DEFB	0B6H	: 'Y'
07B4	AE		2552		DEFB	0AEH	: 'S'

MPP-I						
LOC	OBJ	CODE	M	STMT	SOURCE	STATEMENT
07B5	1F			2553	ERR_SP	DEFB 1PH : 'P'
07B6	AE			2554	DEFB	OAEH : 'S'
07B7	02			2555	DEFB	O2 : '-'
07B8	03			2556	DEFB	O3 : 'R'
07B9	03			2557	DEFB	O3 : 'R'
07BA	8F			2558	DEFB	8FH : 'E'
07BB	00			2559	DEFB	O
07BC	AE			2560	SEPTAB	DEFB OAEH : 'G'
07BD	8F			2561	DEFB	O8FH : 'E'
07BE	B3			2562	DEFB	OB3H : 'D'
07BF	00			2563	DEFB	O
07C0	AE			2564	DEFB	OAEH : 'G'
07C1	B3			2565	DEFB	OB3H : 'D'
07C2	00			2566	DEFB	O
07C3	00			2567	DEFB	O
07C4	0F			2568	DEFB	OFH : 'P'
07C5	AE			2569	DEFB	OAEH : 'S'
07C6	8F			2570	DEFB	O8FH : 'E'
07C7	00			2571	DEFB	O
07C8	0F			2572	DEFB	OFH : 'P'
07C9	00			2573	DEFB	O
07CA	00			2574	REQ	DEFB O
07CB	00			2575	DEFB	O
07CC	02			2576	DEFB	O2H : '-'
07CD	8E			2577	DEFB	O8EH : 'G'
07CE	8F			2578	DEFB	O8FH : 'E'
07CF	03			2579	DEFB	O3H : 'R'
07D0	0F3F			2580	RGTAB	DEFW 3F0FH : 'AF'
07D1	8DA7			2581	DEFW	0A76DH : 'BC'
07D4	8FB3			2582	DEFW	0B36FH : 'DE'
07D6	853F			2583	DEFW	3785H : 'HL'
07D8	4F3F			2584	DEFW	3F4FH : 'AF'
07DA	8DA7			2585	DEFW	0A76DH : 'BC'
07DC	CFB3			2586	DEFW	0B3CFH : 'DE'
07DE	C53F			2587	DEFW	37C5H : 'HL'
07E0	0730			2588	DEFW	3007H : 'IX'
07E2	B630			2589	DEFW	30B6H : 'Y'
07E4	1FAE			2590	DEFW	0AE1FH : 'BP'
07E6	0F3D			2591	DEFW	300FH : 'IF'
07E8	370F			2592	DEFW	0F3FH : 'PB'
07EA	850F			2593	DEFW	0F85H : 'VL'
07EC	770F			2594	DEFW	0F77H : 'PH'
07EE	C50F			2595	DEFW	0FCSH : 'PL'
07F0	8D			2596	SEGTAB	DEFB 08DH : 'O'
07F1	30			2597	DEFB	30H : '1'
07F2	9B			2598	DEFB	09BH : '2'
07F3	BA			2599	DEFB	0BAH : '3'
07F4	36			2600	DEFB	36H : '4'
07F5	AE			2601	DEFB	OAEH : '5'
07F6	AF			2602	DEFB	OAFH : '6'
07F7	38			2603	DEFB	38H : '7'
07F8	BF			2604	DEFB	0BFH : '8'
07F9	BE			2605	DEFB	0BEH : '9'
07FA	3F			2606	DEFB	3FH : 'A'
07FB	A7			2607	DEFB	0A7H : 'B'
07FC	8D			2608	DEFB	08DH : 'C'
07FD	B3			2609	DEFB	OB3H : 'D'
07FE	8F			2610	DEFB	O8FH : 'E'

MFP-I					
LOC	OBJ CODE	M	STMT	SOURCE	STATEMENT
07FF	OF		2611	DEFS	OFH ;'F'
			2612	:	
			2613	:	.....
			2614	:	SYSTEM RAM AREA:
1F9F			2615	USERSTK	ORG 1F9FH
1F9F			2616	DEFS	16
1F9F			2617	SYSSTK:	ORG 1F9FH
1F9F			2618	STEPDF	DEFS 7
1F9E			2619	DISPBF	DEFS 6
			2620	RECBF:	
1F8C			2621	USERAF	DEFS 2
1F8E			2622	USERBC	DEFS 2
1F80			2623	USERDR	DEFS 2
1FC2			2624	USERHL	DEFS 2
1FC4			2625	UAPP	DEFS 2
1FCS			2626	UBCP	DEFS 2
1FCS			2627	UDRP	DEFS 2
1FCA			2628	DHLP	DEFS 2
1FCC			2629	USERIX	DEFS 2
1F8E			2630	USERIV	DEFS 2
1FD0			2631	USERSP	DEFS 2
1FD2			2632	USERIF	DEFS 2
1FD4			2633	FLAGH	DEFS 2
1FD6			2634	FLAGL	DEFS 2
1FD8			2635	FLAGHP	DEFS 2
1FDA			2636	FLAGLP	DEFS 2
1FDC			2637	USERPC	DEFS 2
			2638	:	
1FDE			2639	ADSAVE	DEFS 2 ;Contains the address being
			2640		displayed now.
1FE0			2641	BRAD	DEFS 2 ;Break point address
1FE2			2642	BRDA	DEFS 1 ;Data of break point address
1FE3			2643	STMINOR	DEFS 1 ;Minor state
1FE4			2644	STATE	DEFS 1 ;State
1FE5			2645	POWERUP	DEFS 1 ;Power-up initialization
1FE6			2646	TRST	DEFS 1 ;Flag, bit 0 -- set when function
			2647		or subfunction key is hit.
			2648		bit 7 -- set when illegal key
			2649		is entered.
1FE7			2650	ATEMP	DEFS 1 ;Temporary storage
1FE8			2651	HLTEMP	DEFS 2 ;Temporary storage
1FEA			2652	TEMP	DEFS 4 ;See comments on routine GDA.
1FE8			2653	IMIAD	DEFS 2 ;Contains the address of Opcode 'FF'
			2654		service routine. (RST 38H, mode
			2655		! interrupt, etc.)
1FF0			2656	BEERGET	DEFS 1 ;Default value is 55H
1FF1			2657	FBEEP	DEFS 1 ;Beep frequency
1FF2			2658	TBEEP	DEFS 2 ;Time duration of beep
			2659	END	



CROSS REFERENCE  
SYMBOL VAL M DEFN REFS

MPP-I

ADDRDP 0665	2281	1262	1462	1523	1636														
ADSAVE 1PDE	2639	283	596	618	681	699	732	745	793	822	839								
		914	918	969	973	1077	1176	1458	1633										
ATEMP 1PE7	2650	221	238	274	280														
BEEP 06CB	2399	384																	
BEEPSE 1PFO	2656	2388	2403																
BITEND 05D9	2087	2081																	
BLANK 07A5	2537	1349	2146																
BLOOP 055F	1895	1899																	
BR1 0115	514	525	535	545															
BRAD 1P80	2641	229	364	802	1024	1386	1474												
BRANCH 03B0	1301	431	455	515															
BRDA 1PE2	2642	230	362	1477															
BRSTO 0004	362	351																	
BRTEST 0421	1470																		
CLRR 03DE	1378	742																	
COLDEL 00C9	27	2231																	
CONT28 003E	217	163																	
COUNT 056F	1947	1965	1972																
DAD 026E	966	2467																	
DATADP 0671	2298	1467																	
DDA 026E	969	2468																	
DKCODE 0518	1758	1733	1735	1738	1740														
DPX 026B	963	2465	2466	2466	2467	2468	2469	2470	2471	2472	2473								
		2474	2475																
DIGIT 0002	16	115	237	279	1091	2007	2014	2094	2230	2238									
DISPWF 1P86	2619	346	441	858	1447	1454	1483	1527	1535	1642	1673								
		2282	2299																
DMV 0279	981	2470																	
DMV 0187	872	724																	
DRGAD 028A	995	2474																	
DRGDA 028A	996	2475																	
DRGFX 026B	964	2469																	
DRGNA 0295	1006	1004																	
DRL 0279	980	2471																	
DRL4 031A	1789	1778																	
DRT 0279	978	2473																	
DSTEP 0266	992	969																	
DTAB 0763	2465	534																	
DWT 0279	979	2472																	
KIDI 02B2	1048	1046																	
ENCODE 0523	1761	1746	1748	1752	1754														
ENDFUN 031C	1175	1135	1141	1219															
ENDTAP 0340	1218	1298																	
ERL4 0525	1791	1799																	
ERROR 0353	1221	1119	1167	1169	1285	1286	1296												
ERR 07A9	2541	1221																	
ERRBP 07B5	2553	329																	
FIKE 0041	29	2086																	
F2KE 001F	31	2089																	
FBEEP 1P71	2657	2390	2400																
FCOBY 04C4	1687	756	1655																
FCOBY1 04DA	1732																		
FCOBY2 04F9	1743	1731																	
FILEDP 038A	1285	1266																	
FLAG 1PD4	2633	1734	1743																
FLAGP 1PD6	2635	1739	1751																
FLAGL 1PD6	2634	1736	1747																

CROSS REFERENCE		MPV-I																			
SYMBOL	VAL	M	DEFN	REPS																	
FLAGLP	1FDA			2838	1741	1753															
FLAGK	04D8			1722	1700																
GAD	029C			1023	2478																
GDA	029C			1024	2479																
GETBIT	0588			1904	1893	1895	1900														
GETBIT	055A			1882	1874																
GETP	053D			1841	1113																
GETPTR	053A			1827	1210	1278	1809														
GPIX	0289			1017	2478	2477	2477	2478	2479	2480	2481	2482	2483	2484							
GMV	0284			1112	872	2481															
GROAD	0289			1019	2485																
GRGDA	0289			1020	2486																
GRGPIX	0289			1018	2480																
GRL	0308			1144	2482																
GRT	035A			1228	2484																
GTAB	078P			2478	544																
GWT	0324			1182	2483																
HAD	0202			839	2445																
HDA	01EF			822	2446																
HEX7	0689			2337	2316	2324															
HEX790	0678			2314	2284	2288	2300														
HPFX	01EC			818	2443	2444	2444	2445	2448	2448	2447	2448	2449	2450	2451						
HLTEMP	1FE8			2651	182	239	281	287													
HMV	0220			871	2448																
HROAD	0212			855	2452																
HRODA	0230			888	2453																
HROPIX	0212			858	2447																
HRL	0220			870	2449																
HRT	0220			868	2451																
HTAB	074B			2443	513																
HWT	0220			869	2450																
IAD	0240			913	2456																
IDA	0240			914	2457																
IPFX	023D			908	2454	2455	2455	2455	2457	2458	2459	2460	2461	2462							
IGNORE	03BB			1338	557	580	594	600	618	630	643	662	708	719							
IMIAD	1FER			2653	207	1373															
IMV	024B			926	2459																
INI	03C1			1347	123																
INI1	03C7			1383	1388																
INI2	03C9			1384	1385																
INI3	06B3			2382	1371																
INI4	03D8			1372	2397																
INCAD	025C			941	2463																
IRGDA	025C			942	2464																
IRGPIX	023D			909	2458																
IRGNA	0287			952	890																
IRL	024B			925	2460																
IRT	024B			923	2462																
ISTRP	0258			937	932																
ITAB	0787			2454	524																
IWT	024B			924	2461																
KO	077B			2491																	
KOA	0785			2501																	
KOB	0786			2502																	

CROSS REFERENCE  
SYMBOL VAL M DEPN REFS

MPY-I

KOC	0787	2503	
KOD	0788	2504	
KOE	0789	2505	
KOF	078A	2506	
K1	077C	2482	
K10	078B	2507	
K11	078C	2508	
K12	078D	2509	
K13	078E	2510	
K14	078F	2511	
K15	0790	2512	
K16	0791	2513	
K17	0792	2514	
K18	0793	2515	
K19	0794	2516	
K1A	0795	2517	
K1B	0796	2518	
K1C	0797	2519	
K1D	0798	2520	
K1E	0799	2521	
K1F	079A	2522	
K2	077D	2493	
K20	079B	2523	
K21	079C	2524	
K22	079D	2525	
K23	079E	2526	
K3	077E	2494	
K4	077F	2495	
K5	0780	2496	
K6	0781	2497	
K7	0782	2498	
K8	0783	2499	
K9	0784	2500	
KADDN	01DE	764	2436
KCBR	01CC	738	2437
KCOL	062D	2227	2265
KDATA	0135	565	2430
KDEC	0120	528	2427
KDEL	0196	686	2433
KEYEXX	00E9	392	2411
KEYMAP	061D	2169	
KEYTAB	077B	2490	2169
KFDM	0741	2434	453
KGO	0125	538	2428
KHEK	0111	507	403
KIF	0000	18	2001 2008 2240
KINC	011B	518	2425 2426 2427 2428 2429 2430 2431 2432 2433
KINS	015D	810	2432
KMW	01E2	787	2439
KPC	01C2	727	2434 2435 2436 2437 2438 2439 2440 2441 2442
KRBO	01D6	750	2438
KRL	01E2	790	2440
KROW	0647	2244	2256
KRT	01E8	801	2442
KSBR	0147	587	2431
KSTEP	012A	548	2429
KSUBPU	0737	2424	430
KWT	01E6	797	2441

CROSS REFERENCE				MPP-I																
SYMBOL	VAL	M	DEFN	REFS																
LEAD	0360			1229 1240 1258 1272																
LEAD1	0367			1238 1246																
LEAD2	0371			1250 1251																
LOCPT	0445			1651 1648																
LOCMB	048E			1882 1828																
LOCMB	048B			1678 888																
LOCSTB	0455			1838 871 1519																
LOCSTN	045F			1550 830 988 1531																
LOOPR	058F			2001 2004																
LOOPL	0589			2008 2011																
MAIN	000E			379 387																
MEMDF1	0402			1444 768 851																
MEMDF2	040B			1451 371 574 604 682 733 746 835 819 974 1177																
MPERIO	002A			33 2017																
MPP I	079F			2531 258																
MVDP	0300			1138 1126																
MWI	0066			266 174 1372																
NOKET	064D			2255 2247																
NOTOWE	06DD			2409 2405																
OLOOP	05B7			2042 2045																
ONE 1K	0004			47 2066																
ONE 2K	0004			48 2064																
OUT0	05C9			2057																
OUT1	05D2			2063 2056																
OUTBIT	05C4			2050 2041 2043 2047																
OUTBIT	05B1			2033 2028																
P255	0003			15 107 276																
PERIOD	058C			1991 1236 1250 1847																
POWERU	1F85			2645 121 2382																
PRECL1	038E			1402 827 890 1422																
PRECL2	03FA			1416 840 875																
PRECOI	02A3			1036 862																
PREPC	0021			133 131																
PRCODE	00A5			19 122 1370																
RANCHK	03F6			2110 130 331 334 596 628 707 824 2358																
RANT	0694			2358 2362																
RANT3	0694			2355																
RBOFP	1F8C			2620 1049 1682																
REDDP6	0473			1600 864																
REDDP9	0477			1606 882 897 952 1006																
REG	07CA			2574 763																
RESET1	0032			181 140																
RESET2	0054			248 183																
RGNADP	04AE			1659 1824																
RGSAVE	0074			281																
RGSTIN	0479			1611 1604																
RGTAN	07D0			2580 1868																
RCMTE3	06A6			2375																
RST28	0028			143																
RST30	0030			166																
RST38	0038			194																
SAV12	0412			1456 1450																
SCAN	05F8			2136 381																
SCAN1	0624			2191 1265 1364 2153 2162																
SCLOOP	0616			2162 2163																
SCNK	0600			2153 2158																
SCPR3	0608			2152 2145 2154																
SD7	0001			17 1230 1277 2228 2234																

CROSS REFERENCE		MPT-I									
SYMBOL	VAL	M	DEFN	REFS							
SEDTAB	07FO	2596	2339								
SETIP	00A4	320	318								
SETPT	0434	1487	1481	1530							
SETPT1	0433	1486	1481								
SETSTO	0000	353	263	332	335	347	1222				
SHORTP	057E	1986	1956								
SKIPR1	0183	645	641								
SKIPR2	0188	721	717								
SQWAVE	05EA	2094	2099								
STATE	1PR4	2644	381	443	514	1396	1456	1556	1617	1637	
STPDPF	1PAF	2618	623	627	645	680	706	721	723	796	1112 1122
			1132	1144	1153	1192	1198	1218	1227	1254	1260 1292
			1545	1840							
STPDPD	043A	1513	805	885	937	992					
STEPTA	07BC	2560	1584								
STMINO	1PB3	2643	451	859	926	942	981	996	1543	1569	1616 1643
			1681	1693							
SUM	0531	1813	2378								
SUM1	052D	1803	1183	1290							
SUMCAL	0532	1821	1823								
SUMOK	06B2	2381	2379								
SYSSTK	1PAF	2617	116	322	360						
SYS SP	07AF	2547	341								
TAPEIN	054D	1862	1257	1287							
TAPEOU	05A7	2021	1203	1213	2030						
TBRFP	1PF2	2658	2392	2402							
TEMP	1FEA	2652	1036	1048	1079	1094	1228	1287			
TERR	0587	1985	1952								
TEST	1FE6	2646	252	413	1337	1408	1413	2138			
TESTM	03E5	1389	552	569	592	614	690				
TESTRG	013E	578	571								
TLOOP	054F	1874	1877								
TNEXT	06AO	2361	2359								
TOHE	05EA	2090	2087	2408							
TONEIK	08DE	2085	1197	2067							
TONEIK	05E2	2088	1209	1217	2059	2065					
UAFF	1PC4	2625	1737	1755							
UBCP	1FC6	2626									
UDBP	1FC6	2627									
UHLP	1FCA	2628									
USERAF	1FBC	2621	1068	1092	1732	1749					
USERBC	1FBE	2622									
USERDE	1FCD	2623									
USEREL	1FC2	2624									
USERIF	1FD2	2625	181	312	320	1037	1069	1717			
USERIK	1FCC	2626									
USERIY	1FCE	2630	269								
USERPC	1FDC	2637	133	285	731						
USERSP	1FDD	2631	250	268	326	1087					
USERST	1FDF	2615	249	345							
ZERO 1	0002	49	2060								
ZERO 2	0008	50	2058								
ZSUM	0071	20	191								

## -PROGRAMACION DE ESTE PROYECTO

La Programación Para esta máquina Perforadora tiene como objetivo el cumplir con cuatro funciones Primarias que son:

- Cargar los datos de los Puntos a Perforar.
- Una vez cargados los datos, Poderlos Guardar en cinta.
- Siempre que se requiera, Poder cargar los datos de cinta.
- Ejecutar las Perforaciones, tomando los datos de cinta o de la misma máquina.

### Estructura de la Programación.

Con base a estas necesidades de funcionamiento, se dividió la Programación, de tal forma que cumpliera con las mismas de la manera más Práctica y efectiva Posible. La división es como sigue:

Programa Principal :Llama a cualquiera de los siguientes

#### Programas

- Programa DATS : Carga los datos de los Puntos a Perforar.
- Programa SACR : Guarda los datos en cinta.
- Programa METE : Carga los datos de cinta.
- Programa SALE : Ejecuta las Perforaciones de los datos

A continuación se describirán cada uno de los Programas enseudocódigo y en Programación estructurada.

El mostrar un Programa enseudocódigo significa que se puede saber cual es la función del mismo sin tener que mostrar todo su listado. Y el mostrarlo en Programación estructurada significa que usando algunas instrucciones básicas podemos describir su funcionamiento de manera clara y simple. Muchas veces se dice que un Programa hecho con Programación estructurada está enseudocódigo. Después de mostrar losseudocódigos, se muestran los listados de cada uno de los Programas que forman nuestra Programación.

Hay que recordar que esta Programación está sujeta a las limitaciones de memoria del Micro-Professor I y que el funcionamiento de esta máquina Perforadora puede ser mejorado y sofisticado si se usa una computadora de mayor capacidad.

Programa: PRINCIPAL (PERF-1)

Nombre: Principal

Descripción: Este Programa Permite el acceso a cualquiera de los cuatro programas de función; DATS, SACA, METE, SALE.

Para facilitar la selección de cualquiera de los programas de función, este programa cuenta con un menú de selección con las diferentes opciones.

También dentro de esas opciones está la opción de escape (ESC), con la cual podemos salir del Programa Principal y regresar al Programa monitor del kit Micro-Professor I.



Programa PRINCIPAL (PERF-1)

Funcionamiento Pseudocódigo.

REPITE

REPITE

Despliega mensaje "PERF-1"

Despliega mensaje "METE00"

Despliega mensaje "SACA01"

Despliega mensaje "SALE02"

Despliega mensaje "DATS03"

Despliega mensaje "ESC-04"

Despliega mensaje "CUAL ?"

Lee teclado hasta "A"=<4

FIN REPITE

CLASIFICA "A" entre:

00-EJECUTA METE

01-EJECUTA SACA

02-EJECUTA SALE

03-EJECUTA DATS

FIN CLASIFICA

HASTA que "A"=4 (tecla ESC)

FIN REPITE

### Programa: SACRA

**Descripción:** Este Programa Permite cargar una tabla de datos en cinta. ( Estos datos deben ser Previamente cargados en la función DATS ). Fundamentalmente este Programa SACRA, se apoya en la subrutina TAPEWRITE del kit Micro-Professor-I.

Para cargar los datos a la cinta este Programa requiere que se le asigne un nombre a la tabla. Esto nos permitirá identificar en el futuro dicha tabla y llamarle cuando sea necesario.

La subrutina TAPE WRITE como su nombre lo indica escribe en cinta cualquier bloque de memoria RAM del kit. Para eso es necesario especificar el nombre del archivo o tabla y las direcciones inicial y final de la misma, estas direcciones son asignadas por el Programa SACRA.

Programa SACR

Funcionamiento Pseudocódigo.

DESPLIEGA mensaje "----F" (Pide nombre de tabla)

REPITE

LEE teclado esperando nombre de la tabla

HASTA tecla "GO" oprimida

FIN REPITE

EJECUTA subrutina TAPEWRITE del kit

REGRESA al Programa Principal

## Programa: METE

Nombre: METE

Descripción: Este programa permite cargar una tabla de datos de la cinta al kit. ( Una vez cargada la tabla de datos puede llamarse a la función SALE ). Fundamentalmente este programa METE, se apoya en la subrutina TAPERAD del kit Micro-Professor I.

Para cargar los datos de la cinta, este programa requiere que se le de el nombre de la tabla que se desea cargar. En este caso no son necesarias las direcciones de inicio y fin de tabla.

Programa. METE

Funcionamiento Pseudocódigo.

DESPLIEGA mensaje "----F" (Pide nombre de la tabla)

REPITE

LEE teclado esperando nombre de la tabla.

HASTA tecla "GO" oprimida

FIN REPITE

EJECUTA subrutina TAPE READ del kit

REGRESA al Programa Principal

Programa: SALE

Nombre: SALE

Descripción: Este Programa clasifica la información almacenada en la memoria de la máquina, y dependiendo de dicha clasificación, realiza los Procedimientos correspondientes para la ejecución correcta de los datos analizados.

Los datos almacenados en memoria se clasifican según las siguientes alternativas:

- Inicio de datos en memoria
- Perforación
- Inicio de corte
- Punto intermedio de corte
- Fin de corte
- Integrado horizontal
- Integrado vertical
- Perforaciones en línea recta horizontal
- Perforaciones en línea recta vertical
- Fin de datos en memoria

Este Programa tiene la opción de detener el trabajo realizado por la máquina, presionando la tecla "alto" una vez

hecho esto, se puede indicar otra posición anterior o posterior donde se continuará con el trabajo hasta la terminación de todos los datos.

El control que ejerce este programa consiste en proporcionar la información necesaria para el funcionamiento de los motores de pasos y el taladro; los motores de pasos moverán el circuito impreso de trabajo bajo el taladro, el cual permanece fijo. El movimiento será en un plano, llamado X-Y.

Los datos son almacenados en memoria por medio del programa DATS o del programa METE.

Programa SALE

Funcionamiento Pseudocódigo.

LLAMA origen

REPITE

DESPLIEGA mensaje "SALE"

LEE teclado

HASTA tecla=corre

FIN REPITE

TRAE dato

SI dato=inicio (CD) ENTONCES:

TRAE siguiente dato

MIENTRAS dato sea válido y diferente de FIN

CLASIFICA dato entre:

BD -- Ejecuta Perfora

BX -- Ejecuta línea vertical

CC -- Ejecuta corte

CF -- Ejecuta fin de corte

DX -- Ejecuta integrado horizontal

EX -- Ejecuta integrado vertical

FX -- Ejecuta línea horizontal

FIN CLASIFICA

FIN MIENTRAS



SI último dato fue no válido ENTONCES:

DesPliega mensaje "NO DAT"

FIN SI

DE LO CONTRARIO:

DesPliega mensaje "NO DAT"

FIN SI

REGRESA a Programa Principal PERF-1

## Programa: DATS

Nombre: DATS

Descripción: Con este Programa se Puede hacer el cargado uno a uno de los Puntos a Perforar, empleando Para esto las teclas de movimiento Para los ejes X e Y, así como la tecla DAT que reconoce a esa coordenada como un Punto a Perforar; también se Puede utilizar la tecla CORTE Para indicar un corte entre Puntos, así como las teclas Para integrados y líneas tanto verticales como horizontales.

Una vez que este Programa es llamado, debe Posicionar los ejes en su origen (subrutina origen) y el usuario deberá cargar los datos de los Puntos a Perforar. Cuando ya se han cargado los datos estos Pueden ser Guardados en cinta usando el Programa SACA.

## Programa DATS

### Funcionamiento Pseudocódigo.

LLAMA origen

DESPLIEGA mensaje "DATS--"

REPITE

LEE teclado

CLASIFICA "A" entre:

X → ; Ejecuta MOVX

X ← ; Ejecuta MOVLX

Y → ; Ejecuta MOVY

Y ← ; Ejecuta MOVLY

DATO ; Carga dato

CORTE ; Carga dato corte

INTE ; Carga dato integrado ver/hor

LINEA ; Carga dato línea ver/hor

FIN CLASIFICA

HASTA "A"=Fin

FIN REPITE

INDICA fin de tabla

REGRESA a Programa Principal

## - LISTADO DE LA PROGRAMACION

A continuación se muestra el listado de la programación de este Proyecto de tesis. Este listado deberá ser grabado en una memoria EPROM 2732 de 4 Kb con dirección 2000H - 2FFFH en el kit.

+++ ALVARO +++

ADDRESS	OPERAND	OPERAND	ORG	COMMENT
2000	<0001>	0001	ORG 2000H	
2001	<0001>	0002	ETICORT. DEFS 1	
2002	<0001>	0003	DACORT. DEFS 1	
2003	<0001>	0004	SALVA. DEFS 1	
2004	<0001>	0005	FULSOX. DEFS 1	
2005	<0001>	0006	FULSOY. DEFS 1	
2006	<0001>	0007	FATSEP. DEFS 1	
2007	<0001>	0008	NUPER. DEFS 1	
2008	<0001>	0009	SEPA. DEFS 1	
2009	<0001>	0010	LINE. DEFS 1	
200A	<0001>	0011	BAJA. DEFS 1	
200B	<0001>	0012	CUALRA. DEFS 1	
200C	<0001>	0013	QUEPRO. DEFS 1	
200D	<0002>	0014	POSITX. DEFS 2	
200E	<0002>	0015	POSITV. DEFS 2	
2010	<0002>	0016	AFUNTA. DEFS 2	
2012	<0002>	0017	INITAB. DEFS 2	
2014	<0002>	0018	FINTAB. DEFS 2	
2016	<0002>	0019	MUESXY. DEFS 2	
2018	<0002>	0020	MUESY. DEFS 2	
201A	0230	0021	TABINI. DEFW 3002H	, "PERF-1"
201C	030F	0022	DEFW 0F03H	
201E	1F6F	0023	DEFW 6F1FH	
2020	0202	0024	TABMENU. DEFW 0202H	, "MENU--"
2022	2365	0025	DEFW 0523H	
2024	D3F0	0026	DEFW 8F0D3H	
2026	6000	0027	TABNETE. DEFW 06006DH	, "NETE00"
2028	676F	0028	DEFW 6F67H	
202A	D3F0	0029	DEFW 8F0D3H	
202C	6000	0030	TABSACA. DEFW 3006DH	, "SACA01"
202E	8D3F	0031	DEFW 3F6DH	
2030	AEF0	0032	DEFW 3F0AEH	
2032	6060	0033	TABSALE. DEFW 3606DH	, "SALE02"
2034	656F	0034	DEFW 6F65H	
2036	AEF0	0035	DEFW 3F0AEH	
2038	6DA0	0036	TABDATS. DEFW 06DA6DH	, "DATS03"
203A	67AE	0037	DEFW 0AE67H	
203C	63F0	0038	DEFW 3F063H	
203E	6060	0039	TABESC. DEFW 3606DH	, "ESC-04"
2040	6060	0040	DEFW 026DH	
2042	6FAE	0041	DEFW 0AE6FH	
2044	0236	0042	TABCUAL. DEFW 3602H	, "CUAL-?"
2046	3F65	0043	DEFW 653FH	
2048	6065	0044	DEFW 0656DH	
204A	0202	0045	HENDATS. DEFW 0202H	, "DATS--"
204C	67AE	0046	DEFW 0AE67H	
204E	63F0	0047	DEFW 3F063H	
2050	6063	0048	HENINI. DEFW 0636DH	, "INI-CD"
2052	3062	0049	DEFW 0236H	

+++ ALVARO +++

2054	3023	0050	DEFW 2330H	
2056	3FA7	0051	TABCOR6.DEFW 0A73FH	, "COR. -AS"
2058	4302	0052	DEFW 0243H	
205A	83A3	0053	DEFW 0A383H	
205C	8D8F	0054	TABCOCE.DEFW 8F6DH	, "COR. -CE"
205E	4302	0055	DEFW 0243H	
2060	83A3	0056	DEFW 0A383H	
2062	0302	0057	TABINTE.DEFW 0202H	, "INTE--"
2064	878F	0058	DEFW 8F87H	
2066	3023	0059	DEFW 2330H	
2068	3736	0060	VERHOR.DEFW 3837H	, "V4--H7"
206A	0202	0061	DEFW 0202H	
206C	8536	0062	DEFW 3685H	
206E	8D30	0063	TABFIN.DEFW 308DH	, "FIN-CL"
2070	2302	0064	DEFW 0223H	
2072	0F30	0065	DEFW 300FH	
2074	3F02	0066	TABLIN.DEFW 023FH	, "LINEA--"
2076	2D8F	0067	DEFW 8F3DH	
2078	8530	0068	DEFW 3065H	
207A	0202	0069	HENHETE.DEFW 0202H	, "HETE--"
207C	878F	0070	DEFW 8F87H	
207E	3D8F	0071	DEFW 8F3DH	
2080	0202	0072	HENSACA.DEFW 0202H	, "SACA--"
2082	8D3F	0073	DEFW 3F8DH	
2084	AEF0	0074	DEFW 3F0EH	
2086	3F87	0075	TABENDAT.DEFW 873FH	, "NO DAT"
2088	0083	0076	DEFW 0B300H	
208A	23A3	0077	DEFW 0A323H	
208C	038F	0078	TABERR1.DEFW 8F05H	, "ERR0R1"
208E	A302	0079	DEFW 03A2H	
2090	3003	0080	DEFW 0330H	
<05FE>		0081	SCAN.	EQU 05FEH
<0624>		0082	SCAN1.	EQU 0624H
<1F66>		0083	DISFBUF.	EQU 1F66H
<1FAF>		0084	STEFBUF.	EQU 1FAFH
<0665>		0085	ADDRDP.	EQU 0665H
<0671>		0086	DATADP.	EQU 0671H
<05E4>		0087	TOHE.	EQU 05E4H
<0605>		0088	HAXX.	EQU 0605H
<0660>		0089	HAXY.	EQU 0660H
<0300>		0090	VALRET.	EQU 0300H
<1860>		0091	DAHEX.	EQU 1860H
<1800>		0092	INICIO.	EQU 1800H
<1870>		0093	DAHEY.	EQU 1870H
<06CB>		0094	BEEP.	EQU 06CBH
<0324>		0095	GHT.	EQU 0324H
<035A>		0096	GRT.	EQU 035AH

```

0036 ,+++++
0039 , PROGRAMA. PRINCIPAL. *
0100 , ESTE PROGRAMA DA EL MENU Y ADEMAS SELECCIONA ALGUNA *
0101 , DE LAS DIFERENTES OPCIONES. *
0102 , *
0103 , FRO. TECLA FUNCION *
0104 , *
0105 , HETE "0" CARGA UNA TABLA DE CINTA *
0106 , SACA "1" CARGA UNA TABLA EN CINTA *
0107 , SALE "2" EJECUTA LA TABLA DEL KIT *
0108 , DATS "3" FERNITE PROGRAMAR UNA TABLA *
0109 , ESC "4" SALIDA DEL PROGRAMA PRINCIPAL *
0110 , *
0111 , ESTE PROGRAMA LLAMA A LAS SIGUIENTES SUBROUTINAS DEL *
0112 , HICO PROFESSOR-I . *
0113 , SCAN, SCANL, TONE, ADCDP, DATADF. *
0114 ,+++++
2092 DD211A20 0115 FRIN. LD IX, TABINI , DESPLIEGA MENSAJE
2096 CD4B21 0116 CALL SCANINO , "FERF-1".
2099 DD212020 0117 MENU. LD IX, TABMENU , DESPLIEGA MENSAJE
209D CD4B21 0118 CALL SCANINO , "MENU--".
20A0 DD212620 0119 LD IX, TABHETE , DESPLIEGA MENSAJE
20A4 CD4B21 0120 CALL SCANINO , "HETE00".
20A7 DD212C20 0121 LD IX, TABSACA , DESPLIEGA MENSAJE
20AB CD4B21 0122 CALL SCANINO , "SACA01".
20AE DD213220 0123 LD IX, TABSALE , DESPLIEGA MENSAJE
20B2 CD4B21 0124 CALL SCANINO , "SALE02".
20B5 DD213620 0125 LD IX, TABDATS , DESPLIEGA MENSAJE
20B9 CD4B21 0126 CALL SCANINO , "DATS03".
20BC DD213E20 0127 LD IX, TABESC , DESPLIEGA MENSAJE
20C0 CD4B21 0128 CALL SCANINO , "ESC 04".
20C3 DD214420 0129 DCUAL. LD IX, TABDCUAL , DESPLIEGA MENSAJE
20C7 CDFE05 0130 CALL SCAN , "CUAL-?".
20CA FE00 0131 CF A, 00H , HACE LAS COMPA-
20CC CA1C25 0132 JF Z, HETE , RACIONES Y SALTA
20CF FE01 0133 CF A, 01H , SEGUN LA TECLA
20D1 CA1325 0134 JF Z, SACA , OPRIMIDA.
20D4 FE02 0135 CF A, 02H ,
20D6 C8E022 0136 JF Z, SALE ,
20D9 FE03 0137 CF A, 03H ,
20DB CAE420 R 0138 JF Z, DATS ,
20DE FE04 0139 CF A, 04H , SI SE ELIGE ESCAPE
20E0 C29920 R 0140 JF NZ, MENU , REGRESA AL PROGRAMA
20E3 C7 0141 RST 00H , MONITOR DEL KIT.

```

\*\*\* ALVARO \*\*\*

```

0143 , ++++++
0144 ,PROGRAMA. DATS +
0145 ,ESTE PROGRAMA PERMITE CARGAR LOS DATOS A PERFORAR. +
0146 ,PARA HACERLO SE DEBE POSICIONAR LA PLACA FISICAMENTE+
0147 ,EN EL LUGAR QUE SE DESEA PERFORAR, ESTO SE LOGRA A +
0148 ,TRAVES DE LOS MOVIMIENTOS DE LA PLACA EN LOS EJES +
0149 ,X E Y. +
0150 ,EN ESTE PROGRAMA TAMBIEN SE PUEDEN CARGAR LAS OP- +
0151 ,CIONES DE CORTE, LINEA, Y CIRCUITO INTEGRADO. +
0152 ,LAS DIFERENTES FUNCIONES DE ESTE PROGRAMA SE DES- +
0153 ,CRIBEN A CONTINUACION. +
0154 , +
0155 , +
0156 ,PROGRAMA          TECLA          FUNCION +
0157 , +
0158 ,NOVX              "0"          HUEVE A LA PLACA EN EL EJE X +
0159 ,NOV-X             "1"          HUEVE A LA PLACA EN EL EJE X +
0160 ,NOVY              "2"          HUEVE A LA PLACA EN EL EJE Y +
0161 ,NOV-Y             "3"          HUEVE A LA PLACA EN EL EJE Y +
0162 ,HARDAT           "D"          CARGA EN LA TABLA LOS VALORES+
0163 ,HARCORT          "C"          INDICA CORTE ABIERTO/CERRADO +
0164 ,LINEA             "F"          INDICA LINEA DE PERFORACIONES+
0165 ,                 " "          VERT. O HORZ. DE 3 O 5 PERF. +
0166 ,INTE              "E"          INDICA PERFORACION DE CIR- +
0167 ,                 " "          CUITO INTEGRADO. +
0168 ,ERROR1           "-"          BORRA EL ULTIMO DATO DE LA +
0169 ,                 " "          TABLA. UNICAMENTE UNO. +
0170 , +
0171 ,ESTE PROGRAMA LLAMA A LAS SIGUIENTES SUBROUTINAS. +
0172 ,ORIGEN, CHECATE, SCANLHO Y TONOBEF. +
0173 , +
0174 , ++++++
0175 , +
20E4 DD214A20 0176 DATS. LD IX, HENDATS , DESPLIEGA MENSAJE
20E6 CD4B21 0177 CALL SCANLHO , "DATS--".
20E8 3E7F 0178 LD A, 7FH , CARGA VALOR INICIAL
20ED 320020 0179 LD <ETICORT>,A , EN VARIABLES DE CORTE
20F0 320920 0180 LD <BAJA>, A , E INDICADOR BAJAVEL
20F3 3EFE 0181 LD A, 0FEH , CARGA VALOR INICIAL
20F5 320120 0182 LD <DRCORT>,A , ETIQUETA DATO/CORTE.
20F8 210010 0183 LD HL, INICIO , CARGA VALOR INICIAL
20FB 221020 0184 LD <AFUNTA>,HL , DE VARIABLE APUNTA
20FE 221220 0185 LD <INITAB>,HL , E INICIO DE TABLA
2101 CD4121 0186 CALL ORIGEN , LLAMA SUB. ORIGEN.
2104 CDF003 0187 HASTAIN.CALL SCAN , ESPERA TECLA INICIO
2107 FE0A 0188 CF A, 0A0H , DE TABLA. TECLA "A".
2109 20F9 0189 JR NZ, HASTAIN ,
210B 2A1020 0190 LD HL, <AFUNTA> ,
210E 221220 0191 LD <INITAB>,HL ,
    
```



+++ ALVARO +++

2111	360D	0192	LD	(HL),0CDH	,CARGA DATO INICIO DE
2112	23	0193	INC	HL	,TABLA.
2114	221020	0194	LD	(AFUNTA),HL	,
2117	00215020	0195	LD	IX,HENINI	,DESPLIEGA MENSAJE
2118	002166	0196	DENUEVO.CALL	SCAN1	,INICIO DE TABLA.
211E	FE02	0197	CF	A,02H	,
2120	2805	0198	JR	Z,FIN	,REVISAR SI TECLA FIN
2122	005021	0199	CALL	CHECATE	,DE TABLA TECLA "B".
2125	16F4	0200	JR	DENUEVO	,
		0201	,		
2127	3A0020	0202	FIN. LD	A,(ETICORT)	,COMPRUEBA SI NO HAY
212A	FE00	0203	CF	A,00H	,CORTE ABIERTO,ANTES
212C	C6	0204	RET	Z	,DE FIN.
212D	2A1020	0205	LD	HL,(AFUNTA)	,CARGA DATO DE FIN DE
2130	36C1	0206	LD	(HL),0C1H	,TABLA.
2132	221420	0207	LD	(FINTAB),HL	,
2135	00216E20	0208	LD	IX,TABFIN	,DESPLIEGA MENSAJE
2139	0D4221	0209	CALL	TONOSEP	,FIN DE TABLA "FIN-C1"
213C	0D4021	0210	CALL	SCAN1H0	,INDICACION SONORA.
213F	030320	R 0211	JF	DCUAL	,
		0212	,		
2142	0E0F	0213	TONOSEP.LD	C,0FH	,CON UN VALOR DE
2144	21000A	0214	LD	HL,000H	,FRECUENCIA Y NUMERO
2147	0DE405	0215	CALL	TONO	,DE CICLOS, EJECUTA
214A	C9	0216	RET		,SUB. DE TONO.

+++ ALVARO +++

```

0216 ,-----
0219 ,SUBROUTINA. SCANING                                +
0220 ,                                                    +
0221 , ESTA SUBROUTINA PERMITE LA EJECUCION DE LA SUB.  +
0222 , SCAN1 DEL KIT POR UN NUMERO DE VECES DEFINIDO POR +
0223 , EL REGISTRO HL. SU USO ES PRIMORDIALMENTE PARA  +
0224 , LOGRAR DESPLEGADOS MOMENTANEOS EN EL DISPLAY Y  +
0225 , LECTURA DEL TECLADO.                            +
0226 ,                                                    +
0227 ,-----
0228 ,
0229 SCANING.LD      HL,0250H      ,CARGA EL # DE VECES
214E 110000 0230 LD      DE,0000H  ,QUE VA HA REPETIR
2151 26      0231 ORA.  DEC      HL      ,LA SUB. SCAN1, Y
2152 ED52    0232 SBC      HL,DE   ,REALIZA DECREMENTO
2154 CD2406 0233 CALL     SCAN1    ,HASTA HL - 0000H.
2157 F20121 0234 JF      F,ORA   ,
215A C9      0235 RET      ,
0236 ,
0237 ,-----
0238 ,SUBROUTINA. CHECATE                                +
0239 ,                                                    +
0240 , ESTA SUBROUTINA ES PARTE DEL PROGRAMA DATS, Y    +
0241 , SIRVE PARA LEER EL TECLADO Y SEGUN LA TECLA FISADA +
0242 , SALTA A LA OPCION ELEGIDA, COMO PROTECCION ESTA  +
0243 , SUBROUTINA NO PERMITE EL CARGADO DE FUNCIONES LINEA +
0244 , E INTEGRADO DURANTE UN CORTE.                    +
0245 , SU ENTRADA ES A TRAVES DEL TECLADO Y SU SALIDA  +
0246 , ES SALTANDO A CUALQUIERA DE LAS OPCIONES DEL PRO- +
0247 , GRAMA DATS.                                       +
0248 ,                                                    +
0249 ,-----
0250 ,
215B FE12    0251 CHECATE.CF      A,12H      , TECLA "0" PARA
215D CA7322 0252 JF      Z,H0VX   , H0VX
2160 FE0C    0253 CF      A,0CH   , TECLA "1" PARA
2162 C83F22 0254 JF      Z,H0VX   , H0V-X
2163 FE06    0255 CF      A,06H   , TECLA "2" PARA
2167 CA9D22 0256 JF      Z,H0VY   , H0VY.
216A FE00    0257 CF      A,00H   , TECLA "3" PARA
216C CA5A22 0258 JF      Z,H0VY   , H0V-Y.
216F FE0F    0259 CF      A,0FH   , TECLA "D" PARA
2171 C86F22 0260 JF      Z,HARDAT , HARDAT.
2174 FE13    0261 CF      A,13H   , TECLA "C" PARA
2176 CA2223 0262 JF      Z,HARCORT , HARCORT.
2179 FE14    0263 CF      A,14H   , TECLA "S" PARA
217B CA5621 R 0264 JF      Z,BAJVEL , BAJA VELOCIDAD.
217E FE1F    0265 CF      A,1FH   , TECLA "-" PARA
2180 C85C21 0266 JF      Z,ERR0R1 , ERROR1.

```

\*\*\* ALVARO \*\*\*

2183	06	0267	EX	AF, AF'	,
2184	3A0020	0268	LD	A, <ETICORT>	,
2187	FE00	0269	CF	A, 06H	,
2189	C6	0270	RET	Z	,
218A	06	0271	EX	AF, AF'	,
218B	FE03	0272	CF	A, 03H	, TECLA "E" FARA
218D	CA6C23	0273	JF	Z, INTE	, INTEGRADO.
2190	FE03	0274	CF	A, 03H	, TECLA "F" FARA
2192	CA6C23	0275	JF	Z, INTE	, LINEA.
2195	C9	0276	RET		, REGRESA.
		0277			

+++ ALVARO +++

```

0279 , -----
0280 ,
0281 , SUBROUTINA. BAJAVEL
0282 ,
0283 ,   ESTA SUBROUTINA MODIFICA EL MOVIMIENTO CONTINUO
0284 , DE LOS MOTORES A UN MOVIMIENTO PULSO A PULSO, ESTA
0285 , MODIFICACION OCURRE SOLO CUANDO EL MOVIMIENTO DE
0286 , LOS MOTORES SE DA POR LAS TECLAS DE MOVIMIENTO EN
0287 , EL PROGRAMA DE CARGADO DE DATOS.
0288 ,   POR CADA VEZ QUE SE PRESIONE CUALQUIERA DE LAS
0289 , TECLAS DE MOVIMIENTO, EL MOTOR CORRESPONDIENTE SOLO
0290 , SE MOVERA UN PULSO, NO IMPORTANDO SI LA TECLA PER-
0291 , HANECE OPRINIDA.
0292 ,   ESTA SUBROUTINA SE LLAMA EN EL PROGRAMA DATS CON
0293 , LA TECLA "8", Y DURANTE ESTA FUNCION DE BAJAVEL
0294 , NO SE PUEDEN CARGAR DATOS, POR LO QUE ES NECESARIO
0295 , SALIRSE DE LA MISHA ANTES DE CARGAR UN DATO.
0296 ,   EL USO DE ESTA SUBROUTINA ES PARA PODER HACER
0297 , APROXIMACIONES MAS EXACTAS.
0298 ,
0299 , -----
0300 ,
2196 3A0920 0301 BAJAVEL.LD A, (BAJA) , ACTUALIZA LA ETIQUETA
2199 2F 0302 CFL , DE BAJA VELOCIDAD.
219A 320920 0303 LD (BAJA), A ,
219D C0FE05 0304 REBAJA. CALL SCAN , LEE EL TECLADO Y HACE
21A0 FE00 0305 CF A, 00H , LAS COMPARACIONES
21A2 CC7322 0306 CALL Z, NOVX , PARA LAS DIFERENTES
21A5 FE01 0307 CF A, 01H , OPCIONES DE MOVI-
21A7 C03F22 0308 CALL Z, NOVX , NIENTO.
21AA FE02 0309 CF A, 02H ,
21AC CC9322 0310 CALL Z, NOVY ,
21AF FE03 0311 CF A, 03H ,
21B1 CC9A22 0312 CALL Z, NOVLY ,
21B4 FE06 0313 CF A, 06H ,
21B6 2802 0314 JR Z, FINBAJA , COMPARA SI ES LA
21B8 18E3 0315 JR REBAJA , SALIDA DE LA SUB.
21BA 3A0920 0316 FINBAJA.LD A, (BAJA) , BAJAVEL.
21BD 2F 0317 CFL , ACTUALIZA LA ETIQUETA
21BE 320920 0318 LD (BAJA), A , ANTES DE SALIR.
21C1 C9 0319 RET ,
0320 ,
21C2 0E0F 0321 ALARM. LD C, 0FH , SUBROUTINA DE ALARMA
21C4 21600A 0322 LD HL, 0A00H , CON DISCRIMINADOR DE
21C7 CDE105 0323 CALL TONE , SUBROUTINA CHECATE O
21CA 3A0920 0324 LD A, (BAJA) , BAJAVEL, REGRESA A
21CD FE00 0325 CF A, 06H , DONDE HAYA SIDO
21CF 28CC 0326 JR Z, REBAJA , LLAMADA.
21D1 C31B21 0327 JF DENUEVO ,

```

+++ ALVARO +++

```

0330 , -----+
0331 , +
0332 , LA FUNCION PRINCIPAL DE ESTA SUBROUTINA ES LLEVAR +
0333 , A LOS MOTORES A SU POSICION DE ORIGEN. PERO TAMBIEN +
0334 , TIENE OTRAS FUNCIONES IMPORTANTES COMO. +
0335 , +
0336 , - INICIALIZAR LOS PUERTOS DEL FIO +
0337 , - EL PUERTO A EN MODO 0, BYTE OUT +
0338 , - EL PUERTO B EN MODO 2, BIT INPUT/OUTPUT +
0339 , - AL ESTAR LOS MOTORES EN POSICION DE ORIGEN +
0340 , INICIALIZA LOS LOS CONTADORES X E Y. +
0341 , - ASIGNA VALORES INICIALES A LOS PULSOS QUE SON +
0342 , APLICADOS A LOS MOTORES DE PASOS. +
0343 , - EMITE SENAL DE ALARMA SI DURANTE EL REGRESO AL +
0344 , ORIGEN EL TALADRO ESTA ABAJO Y SUSPENDE REGRESO. +
0345 , +
0346 , ESTA SUBROUTINA ES LLAMADA POR DATS Y FOR SALE Y +
0347 , DE REFERENCIA PARA EL CONTEO FOR SOFTWARE EN EL +
0348 , QUE SE BASA EL CONTROL DE ESTA MAQUINA. +
0349 , +
0350 , -----+
0351 ,
21D4 3EFF 0352 ORIGEN. LD A, 0FFH , CARGA PUERTO B EN
21D6 D353 0353 OUT 83, A , MODO 2, BIT IN/OUT.
21D8 3EC2 0354 LD A, 0C2H , DEFINE LOS BITS DE
21DA D353 0355 OUT 83, A , ENTRADA Y LOS DE
21DC 3E01 0356 LD A, 01H , SALIDA.
21DE D351 0357 OUT 81, A , LINFIA PUERTO B.
21E0 3E6F 0358 LD A, 0FH , CARGA PUERTO A EN
21E2 D352 0359 OUT 82, A , MODO 0, BYTE OUT.
21E4 3E44 0360 LD A, 44H , CARGA VALORES INI-
21E6 328320 0361 LD <PULSOX>, A , CIALES DE PULSOS
21E9 328420 0362 LD <PULSOY>, A , PARA LOS MOTORES.
0363 ,
21EC CDFC21 0364 ORIX. CALL INTA , REALIZA EL REGRESO
21EF CB7F 0365 BIT 7, A , DE LA FLACA EN EL
21F1 CA0022 R 0366 JF Z, ORIX , EJE X CUANDO
21F4 CD1022 0367 CALL RESCONT , TERMINA PASA A
21F7 CD3F22 0368 CALL MOVIX , SUB REGRESO EJE Y.
21FA 18F0 0369 JR ORIX ,
0370 ,
21FC D651 0371 INTA. IN A, 61 , ANTES DE CADA PULSO
21FE 328220 0372 LD <SALVA>, A , CONFIRMA QUE EL
3201 CB4F 0373 BIT 1, A , TALADRO ESTE ARRIBA.
2203 CA0722 R 0374 JF Z, TALBAJO ,
2206 C9 0375 RET ,
0376 ,
2207 CD4221 0377 TALBAJO. CALL TONOBEF , SUENA UN TONO DE
220A CD3C21 R 0378 JF INTA , ALARMA TALADRO ABAJO.

```

\*\*\* ALVARO \*\*\*

		0379	,			
2200	CDFC21	0380	ORIV.	CALL	INTA	, REALIZA EL REGRESO
2210	C877	0381		BIT	6, A	, DE LA PLACA EN EL
2212	CA3122	R 0382		JP	Z, SAL	, EJEY. CUANDO TERMINA
2215	CD1D22	0383		CALL	RESCONT	, REGRESA AL LUGAR DE
2216	CD3A22	0384		CALL	MOVLY	, DONDE FUE LLAMADO.
2218	16F0	0385		JR	ORIV	,
		0386	,			
		0387	,			
2210	210003	0388	RESCONT.	LD	HL, VALRET	, CREA UN RETARDO EN
2220	2B	0389	RES.	DEC	HL	, PULSOS EN ORIGEN
2221	67	0390		OR	A	, ADEMAS DA UN VALOR
2222	ED02	0391		SBC	HL, DE	, PARA EVITAR EL DETE-
2224	C22022	R 0392		JP	NZ, RES	, DETECTOR DE ORIGEN
2227	210F00	0393		LD	HL, 000FH	, TANTO PARA X, COMO
222A	220C20	0394		LD	(FOSIX), HL	, PARA Y.
222D	220E20	0395		LD	(FOSIV), HL	,
2230	C9	0396		RET		,
		0397	,			
2231	210000	0398	SAL.	LD	HL, 0000H	, ANTES DE HACER EL
2234	220C30	0399		LD	(FOSIX), HL	, RETURN DE ORIGEN.
2237	220E20	0400		LD	(FOSIV), HL	, CARGA LOS VALORES
223A	DD214A20	0401		LD	IX, HENDATS	, INICIALES PARA X E
223E	C9	0402		RET		, Y.
		0403	,			

```

0400 -----*
0405 ,SUBROUTINAS. HOVLX, HOVLX. *
0407 , *
0408 , SON LAS SUBROUTINAS QUE CONCRETAN EL MOVIMIENTO *
0409 , DE LOS MOTORES EN SENTIDO NEGATIVO, AMBAS SUBROUTI- *
0410 , NAS CUENTAN CON DETECTOR DE ORIGEN HABILITADO POR *
0411 , SOFTWARE. *
0412 , *
0413 -----*
0414 ,
223F 110000 0415 HOVLX. LD DE,0000H ,CARGA VALOR DE COM-
2242 B7 0416 OR A ,PARACION PARA
2243 2A0C20 0417 LD HL,(FOSIX) ,DETECTOR DE ORIGEN.
2246 ED52 0418 SBC HL,DE ,SI ES ORIGEN BUENA
2248 CAC221 0419 JF Z,ALARM ,ALARMA, SI NO
224B 2A0C20 0420 LD HL,(FOSIX) ,DA UN PULSO.
224E 2B 0421 DEC HL ,
224F 220C20 0422 LD (FOSIX),HL ,
2252 3E02 0423 LD A,02H ,
2254 320A20 0424 LD (CUADRA),A ,
2257 C35022 R 0425 JF EXIS ,
0426 ,
225A 110000 0427 HOVLX. LD DE,0000H ,LO HIZO QUE HOVLX
225D B7 0428 OR A ,
225E 2A0E20 0429 LD HL,(FOSIX) ,
2261 ED52 0430 SBC HL,DE ,
2263 CAC221 0431 JF Z,ALARM ,
2266 2A0E20 0432 LD HL,(FOSIX) ,
2269 2B 0433 DEC HL ,
226A 220E20 0434 LD (FOSIX),HL ,
226D 3E01 0435 LD A,01 ,
226F 320A20 0436 LD (CUADRA),A ,
2272 C3AA22 R 0437 JF VE ,
0438 ,

```

\*\*\* ALVARO \*\*\*

```

0440 ,-----+
0441 , SUBROUTINAS. NOVX, NOVY. +
0442 , +
0443 , SON LAS SUBROUTINAS QUE CONCRETAN EL MOVIMIENTO +
0444 , DE LOS MOTORES EN SENTIDO POSITIVO, AMBAS SUB- +
0445 , RUTINAS CUENTAN CON UN DETECTOR DE AVANCE MAXIMO +
0446 , HABILITADO POR SOFTWARE. +
0447 , +
0448 ,-----+
0449 ,
2273 116006 0450 NOVX. LD DE, NOVX , CARGA VALOR MAXIMO
2276 B7 0451 OR A , PARA HACER LA COM-
2279 2A0C20 0452 LD HL, (POSIX) , PARACION SI ES
227C ED52 0453 SBC HL, DE , VALOR MAXIMO ENTONCES
227E CAC221 0454 JF Z, ALARM , SUENA ALARMA
2281 2A0C20 0455 LD HL, (POSIX) ,
2284 23 0456 INC HL ,
2285 220C20 0457 LD (POSIX), HL , ACTUALIZA VALOR DE
2288 2E00 0458 LD A, 00H , POSICION
228A 220A20 0459 LD (CUADRA), A ,
228D CD6018 0460 EXIS. CALL DAMEX ,
2290 C3B022 R 0461 JF SALE ,
0462 ,
2293 116006 0463 NOVY. LD DE, NOVY , LO MISMO QUE NOVX.
2296 B7 0464 OR A ,
2297 2A0E20 0465 LD HL, (POSIX) ,
229A ED52 0466 SBC HL, DE ,
229C CAC221 0467 JF Z, ALARM ,
229F 2A0E20 0468 LD HL, (POSIX) ,
22A2 220E20 0469 LD (POSIX), HL ,
22A5 2E00 0470 LD A, 00H ,
22A7 220A20 0471 LD (CUADRA), A ,
22AA CD7018 0472 YE. CALL DAMEY ,
22AD C3B022 R 0473 JF SALE ,
0474 ,
22B0 3A0330 0475 SALE. LD A, (PULSOS) , PREPARA LOS PULSOS
22B3 E6F0 0476 AND A, 0FH , PARA SACARLOS POR
22B5 47 0477 LD B, A , EL PUERTO A, AL
22B6 3A0420 0478 LD A, (PULSOS) , SACARLOS SE MUEVEN
22B9 E60F 0479 AND A, 0FH , LOS MOTORES.
22BB 00 0480 ADD A, B ,
22BC D350 0481 OUT 00, A ,
22BE C9 0482 RET ,
0483 ,

```



\*\*\* ALVARO \*\*\*

```

0485 ,-----+
0486 ,SUBROUTINA. HARDAT. +
0487 , +
0488 , ESTA SUBROUTINA NOS PERHITE CARGAR LOS DATOS DE +
0489 , POSICION DEL PUNTO A PERFORAR, EL FORMATO DE CAR- +
0490 , GADO ES EL SIGUIENTE. +
0491 , +
0492 , PRIMER BYTE - X(H+1)/X(H) +
0493 , SEGUNDO BYTE - X(L) / Y(L) +
0494 , TERCER BYTE - Y(H+1)/Y(H) +
0495 , +
0496 , ESTE FORMATO PERHITE RECONOCER EL DATO COMO UN +
0497 , DATO DE POSICION DE LOS EJES X E Y. Y ES USADO +
0498 , PARA CARGAR TANTO DATOS DE PERFORACION COMO DATOS +
0499 , DE INICIO DE CORTE O PUNTOS INTERMEDIOS O DATOS DE +
0500 , INICIO DE CIRCUITO INTEGRADO O LINEA. +
0501 , EL USO DE ESTA SUBROUTINA ES EXCLUSIVO DEL PRO- +
0502 , GRAMA DATS (CARGADO DE DATOS). +
0503 , +
0504 ,-----+
0505 ,
226F 2A0E20 0506 HARDAT. LD HL, (POSIX) , TOMA LOS VALORES DE
22C2 ED5B0C20 0507 LD DE, (POSIX) , POSICION DE X E Y.
22C6 0604 0508 LD B, 04H , Y LOS TRABAJA PARA
22C8 0625 0509 CORRIYE. SLA L , ACOMODARLOS SEGUN
22CA 0614 0510 RL H , EL FORMATO Y TAMBIEN
22CC 16FA 0511 DJNZ. CORRIYE , LOS PREPARA PARA
22CE 0604 0512 LD B, 04H , MOSTRARLOS EN EL
22D0 0625 0513 CORRIY. SLA L , DISPLAY.
22D2 0613 0514 RL E ,
22D4 0612 0515 RL D ,
22D6 16F6 0516 DJNZ. CORRIY ,
22D8 ED531620 0517 LD (HUESY), DE ,
22DC 221620 0518 LD (HUESY), HL ,
22DF 2A1620 0519 LD HL, (AFUNTA) ,
22E2 72 0520 LD (HL), D ,
22E3 23 0521 INC HL ,
22E4 73 0522 LD (HL), E ,
22E6 23 0523 INC HL ,
22E8 3A1920 0524 LD A, (HUESY+1) ,
22EA 77 0525 LD (HL), A ,
22EC 23 0526 INC HL ,
22EE 221620 0527 LD (AFUNTA), HL ,
22F0 2A1620 0528 HUESDAT. LD A, (HUESY) , AQUI PREPARA DATO
22F1 211920 0529 LD HL, HUESY+1 , PARA PERFORACION.
22F4 ED6F 0530 RLD ,
22F6 0F 0531 LD E, A ,
22F7 0D21B61F 0532 LD IX, DISBUF ,
22F8 0D6306 0533 CALL ADDRDF ,

```

+++ ALVARO +++

```

22FE 2A1920      0534      LD      A, <HUESY+1>      ,
2301 216A1F      0535      LD      HL, DISPBUF+1    ,
2304 007166      0536      CALL   DATADF           ,
2307 3A0136      0537      LD      A, <DACORT>     ,
230A FE01       0538      CP      A, 01H         ,
230C 2604       0539      JR      Z, HARDACO     ,
230E 0DFE05      0540      CALL   SCAN           ,
2311 09         0541      RET                    ,
                0542      ,
2312 21B61F      0543      HARDACO LD      HL, DISPBUF      , AGUI PREPARA DATO
2315 0606       0544      LD      B, 06H        , PARA CORTE
2317 3E40       0545      REPITE LD      A, 40H      ,
2319 86        0546      ADD    A, <HL>        ,
231A 77        0547      LD      <HL>, A      ,
231B 23        0548      INC    HL            ,
231C 16F9      0549      DJNZ, REPITE        ,
231E 0DFE05      0550      CALL   SCAN         ,
2321 09         0551      RET                    ,
                0552      ,

```

+++ ALVARO +++

```

0554 ,-----+
0555 , SUBROUTINA. HARCORT. +
0556 , +
0557 , ESTA SUBROUTINA NOS PERMITE CARGAR LOS DATOS DE +
0558 , CORTE CUANDO SE SELECCIONA ESTA OPCION EN EL PRO- +
0559 , GRAMA DATS. +
0560 , LLAMA A LA SUBROUTINA HARDAT. +
0561 , +
0562 ,-----+
0563 ,
2322 3A0820 0564 HARCORT.LD A, (ETICORT) , INDICADOR DE CORTE
2323 2F 0565 CFL , ABIERTO.
2324 320820 0566 LD (ETICORT), A , ACTUALIZA AL SER
2325 FE00 0567 CF A, 06H , LLAMADA ESTA SUB.
2326 C23F23 R 0568 JF HZ, CIECORT , DECIDE SI ES CORTE
232E 3ECC 0569 LD A, 0CCH , ABIERTO O CORTE
2330 2A1020 0570 LD HL, (AFUNTA) , CERRADO.
2333 77 0571 LD (HL), A ,
2334 23 0572 INC HL ,
2335 221020 0573 LD (AFUNTA), HL ,
2336 CD3323 0574 CALL HUESCOA , MUESTRA INDICADOR
233B CD6F22 0575 CALL HARDAT , DE CORTE ABIERTO.
233E C9 0576 RET ,
0577 ,
233F 2A1020 0578 CIECORT.LD HL, (AFUNTA) , MUESTRA INDICADOR
2342 3ECF 0579 LD A, 0CFH , DE CORTE CERRADO.
2344 77 0580 LD (HL), A ,
2345 23 0581 INC HL ,
2346 221020 0582 LD (AFUNTA), HL ,
2349 CD6123 0583 CALL HUESCOA ,
234C CD6F22 0584 CALL HARDAT ,
234F 3EFE 0585 LD A, 0FEH ,
2351 320120 0586 LD (DACORT), A ,
2354 C9 0587 RET ,
0588 ,
2355 3E01 0589 HUESCOA.LD A, 01H ,
2357 320120 0590 LD (DACORT), A ,
2358 14 0591 ,
2359 0592 JF HUE ,
0593 ,
2361 D0215020 0594 HUESCOA.LD IX, TABCOCE ,
2365 CD4221 0595 HUE. CALL TONOSEP ,
2366 CD4021 0596 CALL SCANLHO ,
236B C9 0597 RET ,
0598 ,

```

+++ ALVARO +++

```

0600 -----+
0601 ,SUBROUTINA. INTE +
0602 , +
0603 , ESTA SUBROUTINA PERMITE EL CARGADO DE LA OPCION +
0604 , DE CIRCUITO INTEGRADO. PARA LOGRARLO SOLO SE REQUI- +
0605 , ERE POSICIONAR LA PLACA EN LA PRIMERA PATA DEL +
0606 , CIRCUITO Y SELECCIONAR ESTA OPCION. EL MISMO PRO- +
0607 , GRAMA PEDIRA EL NUMERO DE PATAS DEL C. I. Y SI SE +
0608 , DESEA VERTICAL U HORIZONTAL. +
0609 , LA CONDICION DE ESTA OPCION ES LOGRAR C. I. EN +
0610 , POSICION HORIZONTAL O HORIZONTAL LO QUE PERMITE UN +
0611 , GRAN AHORRO DE MEMORIA. SOBRE TODO EN LOS CASOS DE +
0612 , CIRCUITOS INTEGRADOS CON UN GRAN NUMERO DE PATAS. +
0613 , +
0614 -----+
0615 ,
236C 00216220 0616 INTE. LD IX, TABINTE , DESPLIEGA MENSAJE
2370 CD4B21 0617 CALL SCANLHO , "INTE--".
2373 00216820 0618 LD IX, VERHOR , DESPLIEGA MENSAJE
2377 CDFE03 0619 CALL SCAN , "V4--H7".
237A FE04 0620 CP A, 04H , Y ESPERA POR
237C C80323 R 0621 JF Z, INVER , OPCION VERTICAL O
237F FE07 0622 CP A, 07H , HORIZONTAL.
2381 C80E23 R 0623 JF Z, INHOR ,
2384 C9 0624 RET ,
0625 ,
2385 210602 0626 INVER. LD HL, 0206H , EJECUTA SI ES
2386 22B61F 0627 LD (DISBUF), HL , VERTICAL.
2388 21C78F 0628 LD HL, 8FC7H , CREA PANTALLA
238E 22B61F 0629 LD (DISBUF+2), HL , PARA FORMATO DE
2391 21B545 0630 LD HL, 4563H , CIRCUITO INTEGRADO.
2394 22B61F 0631 LD (DISBUF+4), HL ,
2397 0021B61F 0632 LD IX, DISBUF ,
239B CDFE03 0633 CALL SCAN ,
239E FE10 0634 CP A, 10H ,
23A0 028523 R 0635 JF NC, INVER ,
23A3 328520 0636 LD (PATSEF), A ,
23A6 CD7106 0637 CALL DATADP ,
23A9 3E02 0638 LD A, 02H ,
23AB 32871F 0639 LD (DISBUF+1), A ,
23AE CDFE03 0640 CALL SCAN ,
23B1 FE12 0641 CP A, 12H ,
23B3 C28523 R 0642 JF NZ, INVER ,
23B6 3A0520 0643 LD A, (PATSEF) ,
23B9 C6E0 0644 ADD A, 06EH ,
23BB C3F423 R 0645 JF SALEINT ,
0646 ,
23BE 210602 0647 INHOR. LD HL, 0206H , EJECUTA SI ES
23C1 22B61F 0648 LD (DISBUF), HL , HORIZONTAL.
    
```

+++ ALVARO +++

23C4	2143A5	0649	LD	HL, 0A543H	, CREA PANTALLA
23C7	22881F	0650	LD	<DISPBUF+2>, HL	, PARA FORMATO DE
23CA	213745	0651	LD	HL, 4537H	, CIRCUITO INTEGRADO.
23CD	228A1F	0652	LD	<DISPBUF+4>, HL	,
23D0	0021861F	0653	LD	IX, DISPBUF	,
23D4	0DFE05	0654	CALL	SCAN	,
23D7	FE10	0655	CF	A, 1GH	,
23D9	D2BE23	R 0656	JP	NZ, INHOR	,
23DC	320520	0657	LD	<PATSEP>, A	,
23DF	0D7106	0658	CALL	DATADP	,
23E2	3E02	0659	LD	A, 02H	,
23E4	32871F	0660	LD	<DISPBUF+1>, A	,
23E7	0DFE05	0661	CALL	SCAN	,
23EA	FE12	0662	CF	A, 12H	,
23EC	02BE23	R 0663	JP	NZ, INHOR	,
23EF	3A0520	0664	LD	A, <PATSEP>	,
23F2	06D0	0665	ADD	A, 06GH	,
23F4	2A1020	0666	SALEINT.LD	HL, <APUNTA>	, CARGA LA OPCION EN
23F7	77	0667	LD	<HL>, A	, LA TABLA DE DATOS.
23F8	23	0668	INC	HL	,
23F9	231020	0669	LD	<APUNTA>, HL	,
23FC	006F22	0670	CALL	HARDAT	,
23FF	C9	0671	RET		,
		0672			,
		0673			,

+++ ALVARO +++

```

0675 ,-----+
0676 ,SUBROUTINA. LINEA. +
0677 , +
0678 , ESTA SUBROUTINA PERMITE EL CARGADO DE LA OPCION +
0679 , LINEA, CON LA SIGUIENTES VARIACIONES. +
0680 , +
0681 , - ELECCION ENTRE TRES O CINCO PERFORACIONES +
0682 , - ELECCION DE LA DISTANCIA DE SEPARACION ENTRE +
0683 , OCHO DIFERENTES VALORES. +
0684 , - ELECCION DE LINEA TANTO HORIZONTAL COMO VERTICAL +
0685 , +
0686 , ESTA OPCION DE CARGADO DE DATOS ES APLICABLE A +
0687 , ESOS CASOS ESPECIALES DE CIRCUITOS INTEGRADOS O DE +
0688 , EN GRAN NUMERO. +
0689 , +
0690 ,-----+
0691 ,
2400 DD217420 0692 LINEA. LD IX, TABLIN , DESPLIEGA MENSAJE
2404 CD4221 0693 CALL TONG6P , "LINEA-".
2407 CD4B21 0694 CALL SCANLN0
240A DD216820 0695 LD IX, VERHOR , NUESTRA DESPLEGADO
240E CDFE03 0696 CALL SCAN , VERTICAL, HORIZONTAL
2411 FE04 0697 CP A, 04H , "V4--H7".
2413 CAlC24 R 0698 JF Z, LINVER ,
2416 FE07 0699 CP A, 07H ,
2418 CA2924 R 0700 JF Z, LINHOR ,
241B C3 0701 RET ,
0702 ,
241C 3E03 0703 LINVER. LD A, 060H , EJECUTA SI ES
241E 32681F 0704 LD <DISFBUF+2>, A , VERTICAL.
2421 3EF0 0705 LD A, 0F0H ,
2423 320024 0706 LD <LINEA>, A , CREA PANTALLA PARA
2426 C33624 R 0707 JF LIN , OPCION DE LINEA V.
0708 ,
2429 3E37 0709 LINHOR. LD A, 37H , EJECUTA SI ES
242B 32681F 0710 LD <DISFBUF+2>, A , HORIZONTAL.
242E 3EF0 0711 LD A, 0F0H ,
2430 320024 0712 LD <LINEA>, A , CREA PANTALLA PARA
2433 C33624 R 0713 JF LIN , OPCION DE LINEA H.
0714 ,
2436 2136EE 0715 LIN. LD HL, 0EE56H , LA SELECCION DEL
2439 22661F 0716 LD <DISFBUF>, HL , # DE PERFORACIONES
243C 3E03 0717 LD A, 03H , SE HACE CON LAS
243E 32691F 0718 LD <DISFBUF+3>, A , TECLAS "+" PARA 3
2441 21360F 0719 NOPER. LD HL, 0F06H , Y "-" PARA 3.
2444 226A1F 0720 LD <DISFBUF+4>, HL ,
2447 DD21B61F 0721 LD IX, DISFBUF ,
244B CDFE03 0722 HUES. CALL SCAN ,
244E FE06 0723 CP A, 06H ,

```

+++ ALVARO +++

2450	DA6624	R	0724	JF	C, SEPARA	,
2453	FE10		0725	CF	A, 10H	,
2455	CA9924	R	0726	JF	Z, CINCO	,
2458	FE11		0727	CF	A, 11H	,
245A	CA6624	R	0728	JF	Z, TRES	,
245D	FE12		0729	CF	A, 12H	, ESPERA SE FISE
245F	C24B24	R	0730	JF	NZ, HUES	, TECLA "GO" PARRA
2462	3A6A1F		0731	LD	A, (DISBUF+4)	, CARGAR DATOS EN LA
2465	FE36		0732	CF	A, 36H	, TABLA.
2467	CA4124	R	0733	JF	Z, NOFER	,
246A	3A661F		0734	LD	A, (DISBUF)	, SI FALTA ALGUN
246D	FE36		0735	CF	A, 36H	, DATO VUELVE A
246F	CA6324	R	0736	JF	Z, NOSEP	, MOSTRAR EL FORNATO
2472	3A6320		0737	LD	A, (PATSEP)	, DE OPCION.
2475	47		0738	LD	B, A	,
2476	3A6620		0739	LD	A, (NUFER)	,
2479	80		0740	ADD	A, B	,
247A	47		0741	LD	B, A	,
247B	3A6624		0742	LD	A, (LINEA)	,
247E	80		0743	ADD	A, B	,
247F	2A1020		0744	LD	HL, (APUNTA)	,
2482	77		0745	LD	(HL), A	,
2483	23		0746	INC	HL	,
2484	221020		0747	LD	(APUNTA), HL	,
2487	CDBF22		0748	CALL	HARDAT	,
248A	C9		0749	RET		,
			0750	,		,
248B	320320		0751	SEPARA. LD	(PATSEP), A	, LA SEPARACION SE DA
248E	CD7106		0752	CALL	DATADP	, COMO UN VALOR HEXA-
2491	3EEE		0753	LD	A, 0EEH	, DECIMAL.
2493	32B71F		0754	LD	(DISBUF+1), A	,
2496	CD4B24	R	0755	JF	HUES	,
			0756	,		,
2499	3E08		0757	CINCO. LD	A, 08H	, EJECUTA SI SE
249B	320620		0758	LD	(NUFER), A	, SELECCIONA OPCION DE
249E	3EAE		0759	LD	A, 0AEH	, CINCO PERFORACIONES.
24A0	32BA1F		0760	LD	(DISBUF+4), A	,
24A3	CD4B24	R	0761	JF	HUES	,
			0762	,		,
24A6	3E00		0763	TRES. LD	A, 00H	, EJECUTA SI SE
24A8	320620		0764	LD	(NUFER), A	, SE SELECCIONA OPCION
24AB	3EBA		0765	LD	A, 0BAH	, TRES PERFORACIONES.
24AD	32BA1F		0766	LD	(DISBUF+4), A	,
24B0	CD4B24	R	0767	JF	HUES	,
			0768	,		,
24B3	2156EE		0769	NOSEP. LD	HL, 0EE36H	,
24B6	22B61F		0770	LD	(DISBUF), HL	,
24B9	CD4B24	R	0771	JF	HUES	,
			0772	,		,

+++ ALVARO +++

```

0774 ,-----+
0775 ,SUBROUTINA. ERROR1 +
0776 , . +
0777 , ESTA SUBROUTINA PERHITE CORREGIR EL ULTIMO DATO +
0778 , QUE FUE CARGADO EN LA TABLA. LA CORRECCION SE HACE +
0779 , DE MANERA TOTAL, ESTO SIGNIFICA QUE NO SOLO SE +
0780 , MODIFICA LA POSICION DE LA OPCION SINO TAMBIEN SE +
0781 , BORRA LA OPCION MISHA POR EJEMPLO SI MARCAMOS UN +
0782 , CORTE Y DESPUES USAMOS ESTA SUBROUTINA, EN NUESTRA +
0783 , TABLA SE BORRARA JUNTO CON LA POSICION DEL CORTE +
0784 , EL INDICADOR DE CORTE. +
0785 , +
0786 ,-----+
0787 ,
246C DD216C20 0788 ERROR1. LD IX, TABERR1 , DESPLIEGA MENSAJE
246D CD4221 0789 CALL TONOBEP , "ERROR1" Y SONORIZA
246E CD4B21 0790 CALL SCANIMO ,
246F 2A1020 0791 LD HL, (AFUNTA) ,
246G 2B 0792 DEC HL , CONFUEBA LOS INDI-
246H 7E 0793 LD A, (HL) , CADORES DE LA TABLA
246I FECD 0794 CF A, 0C0H , Y SEGUN CORRESPONDA
246J CAE824 R 0795 JF Z, INITA , SE HACEN LOS MOVI-
246K 2B 0796 DEC HL , MIENTOS EN LA TABLA.
246L 2B 0797 DEC HL ,
246M 2B 0798 DEC HL ,
246N 7E 0799 LD A, (HL) ,
246O FECF 0800 CF A, 0CFH , SEGUN CORRESPONDA
246P CAF324 R 0801 JF Z, ABRECO , MODIFICA LAS
246Q FECC 0802 CF A, 0C0H , ETIQUETAS.
246R CA3F23 0803 JF Z, CIECOR ,
246S FEAB 0804 CF A, 0A0H ,
246T D20F25 R 0805 JF NC, INDI ,
246U 23 0806 INC HL ,
246V 221020 0807 LD (AFUNTA), HL ,
246W C9 0808 RET ,
0809 ,
246X DD216620 0810 INITA. LD IX, TABENDAT , CONFUEBA SI AL
246Y CD4221 0811 CALL TONOBEP , RETROCEDER NO
246Z CD4B21 0812 CALL SCANIMO , SE LLEGA AL
2470 C9 0813 RET , INICIO DE TABLA.
0814 ,
2471 221020 0815 ABRECO. LD (AFUNTA), HL ,
2472 3E01 0816 LD A, 01H ,
2473 320120 0817 LD (CACORT), A ,
2474 3E00 0818 LD A, 00H ,
2475 320020 0819 LD (ETICORT), A ,
2476 C9 0820 RET ,
0821 ,
2477 221020 0822 CIECOR. LD (AFUNTA), HL ,

```



2504	DEFE	0623	LD	A, 0FEH	,	
2506	220126	0624	LD	(DACORT), A	,	
2509	3E7F	0625	LD	A, 7FH	,	
250B	220026	0626	LD	(ETICORT), A	,	
250E	C9	0627	RET		,	
		0628	,		,	
256F	221026	0629	INDI.	LD	(AFUNTA), HL	,
2512	C9	0630	RET		,	
		0631	,		,	

\*\*\* ALVARO \*\*\*

```

0033 ,+++++
0034 ,SUBROUTINAS. SACA Y HETE.
0035 ,
0036 , ESTAS SUBROUTINAS PERMITEN CARGAR EN CINTA O DE
0037 , CINTA UNA TABLA DE DATOS A PERFORAR. EN AMBOS CASOS+
0038 , SOLO SE REQUIERE LLAMAR A CUALQUIERA DE LAS OPCIO-
0039 , NES ESTANDO EN EL PROGRAMA PRINCIPAL.
0040 ,
0041 , SOLO NECESITAMOS CONOCER EL NOMBRE DEL PROGRAMA
0042 , QUE VANOS A LEER O ASIGNARLE UN NOMBRE A LA TABLA
0043 , QUE VANOS A ESCRIBIR. LAS ASIGNACIONES DE
0044 , DIRECCION INICIAL Y FINAL DE TABLA LAS HACEN LAS
0045 , MISMAS SUBROUTINAS.
0046 ,
0047 ,+++++
0048 ,
2513 3EFF 0049 SACA. LD A,0FFH ,SEGUN LA OPCION
2515 320620 0050 LD <QUEPRO>,A ,SELECCIONADA CARGA
2516 CD2320 0051 CALL TAPE ,VARIABLE DE IDENTIFI-
2518 C9 0052 RET ,CACION.
0053 ,
251C 3E00 0054 HETE. LD A,00H ,SEGUN LA OPCION
251E 320620 0055 LD <QUEPRO>,A ,SELECCIONADA CARGA
2521 CD2320 0056 CALL TAPE ,VARIABLE DE IDENTIFI-
2524 C9 0057 RET ,CACION.
0058 ,
3525 210F02 0059 TAPE. LD HL,020FH ,EJECUTA PARA
3526 22B61F 0060 LD <DISBUF>,HL ,CUALQUIERA DE LAS
3528 D021B61F 0061 LD IN,DISBUF ,OPCIONES.
352F C0FE00 0062 CALL SCAN ,
3532 FE10 0063 CF A,10H ,ESPERA NOMBRE
3534 D24C20 R 0064 JF NC,PUEDEGO ,DE TABLA, YA SEA
3537 21AF1F 0065 LD HL,STEPBUF ,PARA LEER O PARA
353A ED6F 0066 RLD ,ESCRIBIR.
353C 23 0067 INC HL ,
353D ED6F 0068 RLD ,
353F C0C600 0069 CALL BEEP ,
3542 ED06FF1F 0070 LD DE,<STEPBUF> ,
3546 CD6500 0071 CALL ADDRDP ,
3549 C32020 R 0072 JF TAPE ,
0073 ,
254C FE12 0074 PUEDEGO.CF A,12H ,ESPERA TECLA
254E C22020 R 0075 JF NZ,TAPE ,"GO" PARA
3551 3A0620 0076 LD A,<QUEPRO> ,EJECUTAR.
3554 FE00 0077 CF A,00H ,
3556 CA9200 R 0078 JF Z,ESHETE ,
3559 2A1320 0079 ESSACA. LD HL,<INITAB> ,
355C 22B11F 0080 LD <STEPBUF+2>,HL ,
355F 2A1420 0081 LD HL,<FINAB> ,

```

+++ ALVARO +++

```

2562 22631F 0662 LD <STEPSUF+4>,HL ,
2563 CD2103 0663 CALL GMT ,
2566 C9 0664 RET ,
0665 ,
2569 CD5A03 0666 ESHETE. CALL GRT ,
256C C9 0667 RET ,
0668 ,-----+
0669 ,SUBROUTINA. BAJATAL. +
0690 , +
0691 , ESTA SUBROUTINA MANDA LA SENAL DE BAJADA DEL +
0692 , TALADRO A LA MAQUINA, PARA ENVIAR LA SENAL UTILIZA +
0693 , EL BIT 0 DEL PUERTO B DEL PIO. LA DURACION DE ESTA +
0694 , SENAL ES DETERMINADA POR LOS AJUSTES DE LOS POTEN- +
0695 , CIOMETROS . +
0696 , EL EFECTO REAL DE LA BAJADA DEL TALADRO DEFENDE +
0697 , TAMBIEN DE LOS TIEMPOS DE RETARDO QUE EXISTEN +
0698 , TANTO EN ESTA SUBROUTINA COMO EN LA SUBROUTINA SUBE- +
0699 , TAL. +
0900 ,-----+
0901 ,
256D 0601 0902 BAJATAL.LD B,01 , CARGA VALOR DE
256F CD7723 0903 CALL RETA , RETARDO, ANTES DE
2572 3E00 0904 LD A,B0H , LA BAJADA DEL TAL.
2574 D261 0905 OUT 81H,A , SACR PULSO FOR
2576 C9 0906 RET , PUERTO B, DEL PIO.
0907 ,
2577 C5 0908 RETA. PUSH BC , SALVA VALOR DE
2578 CD1221 0909 CALL TONOSEP , REG. B. Y LLAMA
257B C1 0910 POP BC , SUB. TONO.
257C 10F9 0911 DJNZ,RETA , EJECUTA B VECES.
257E C9 0912 RET ,
0913 ,
0914 ,-----+
0915 ,SUBROUTINA. SUBETAL. +
0916 , +
0917 , ESTA SUBROUTINA ES LA QUE SE ENCARGA DE DAR FIN +
0918 , AL PULSO DE BAJADA TALADRO, ADEMAS TIENE OPCION DE +
0919 , CREAR TIEMPOS DE RETARDO ENTRE LA SUBIDA DEL TALA- +
0920 , DRO Y EL SIGUIENTE MOVIMIENTO DE LA PLACA ESTE +
0921 , TIEMPO DE RETARDO ES AJUSTABLE POR SOFTWARE, SEGUN +
0922 , SE REQUIERA. ESTA SUBROUTINA TAMBIEN CUENTA CON UN +
0923 , DETECTOR DE TALADRO ARRIBA QUE ES CENSADO POR EL +
0924 , 1 DEL PUERTO B DEL PIO. LA SUBROUTINA ENTRA EN UN +
0925 , CICLO HASTA QUE EL TALADRO REGRESA A SU POSICION +
0926 , ORIGINAL. +
0927 ,-----+
0928 ,
257F C6C7 0929 SUBETAL.SET 0,A , MANDA PUL SUBE TAL.
2581 D361 0930 OUT 81H,A , POR BIT 0, PUERTO B.
    
```

+++ ALVARO +++

```

2383 0686      0931      LD      B,06H      , CARGA VALOR DE
2385 C07725    0932      CALL   RETA       , RETARDO EJECUTANDO
2386 D681      0933  TALAR. IN   A,01H    , 6 VECES SUB. TONO.
238A C64F      0934      BIT   L,A        , COMPROBEA SI TAL.
238C CA8825    R 0935      JP    Z,TALAR    , ARRIBA SI ESTA
238F C9        0936      RET                    , ARRIBA RET.
                0937      ,
                0938      ,+++++-----+
                0939      ,
                0940      ,
2390 C09220    ✓ 0941      START. CALL  FRIN
2393 C38000    0942      JF    0006H
2396 (2598)   0943      END    START

ERRORS      0
RANGE COUNT 11
    
```

\*\*\* ALVARO \*\*\*

SYMBOL	VALUE	DEFN	REFERENCES
ABRECO	21FD	0015	0001
ADDRESS	0663	0003	0071
ALARM	21C2	0021	0419 0431 0454 0467
APUNTA	2010	0016	0104 0190 0194 0205 0319 0327 0370 0373 0378 0382 0666
			0669 0744 0747 0791 0807 0815 0822 0829
BAJA	2009	0011	0100 0301 0303 0316 0318 0324
BAJATAL	2060	0002	
BAJAYEL	2196	0001	0264
BEEP	060B	0094	0069
CHECATE	2158	0031	0199
CIECOR	2001	0022	
CIECORT	203F	0070	0060 0003
CINCO	2499	0057	0726
CORRIW	2200	0013	0316
CORRIVE	2208	0009	0011
CUADRA	206A	0012	0424 0426 0459 0471
DACORT	2001	0003	0102 0037 0066 0090 0017 0024
DAWEX	1000	0001	0460
DAWEX	1070	0003	0472
DATADP	0071	0006	0036 0037 0050 0702
DATS	20E4	0176	0130
DCUAL	0003	0129	0211
DENUEVO	2110	0196	0200 0007
DISPBUF	1FB6	0003	0032 0035 0043 0627 0629 0631 0632 0639 0640 0650 0652
			0653 0660 0704 0710 0716 0718 0720 0721 0731 0734 0754
			0760 0766 0770 0860 0001
ERROR1	2400	0000	0266
ESMETE	2069	0006	0070
ESSACA	2009	0079	
ETICORT	2000	0002	0179 0202 0260 0064 0066 0019 0026
EXIS	220D	0460	0420
FIN	2127	0002	0190
FINBAJA	210A	0016	0014
FINTAB	2014	0010	0207 0001
GRT	000A	0096	0000
GHT	0001	0090	0000
HASTAIN	2101	0107	0109
INDI	200F	0009	0000
INHOR	200E	0047	0003 0050 0003
INICIO	1000	0002	0100
INITA	24E0	0010	0790
INITAB	2012	0017	0100 0191 0079
INTA	21FC	0071	0064 0070 0000
INTE	2060	0016	0270 0270
INVER	2000	0000	0000 0000
LIN	2400	0710	0707 0710
LINE	2000	0010	
LINEA	2400	0002	0700 0712 0712

\*\*\* ALVARO \*\*\*

SYMBOL	VALUE	DEFN	REFERENCES
LINHOR	2429	0709	0700
LINVER	241C	0702	0698
HARCORT	2322	0564	0262
HARDACO	2312	0543	0539
HARDAT	22BF	0506	0260 0375 0584 0670 0748
HARX	0607	0668	0450
HARY	0660	0669	0463
HENDATS	204A	0045	0176 0401
HENINI	2050	0048	0193
HENHETE	207A	0069	.
HENSACA	2060	0072	
HENU	2099	0117	0140
HETE	251C	0654	0132
HOVLX	223F	0415	0254 0300 0368
HOVLY	225A	0427	0258 0312 0384
HOVX	2275	0450	0252 0306
HOVY	2293	0463	0256 0310
HUE	2365	0595	0592
HUES	2446	0722	0730 0755 0761 0767 0771
HUESCOA	2355	0589	0574
HUESCOC	2361	0594	0583
HUESDAT	22EE	0528	
HUESXY	2016	0019	0517 0526
HUESY	2016	0020	0518 0524 0529 0534
HOPER	2444	0719	0733
HOSEF	2483	0769	0736
HUPER	2006	0008	0729 0756 0764
ORA	2151	0231	0234
ORIGEN	21D4	0352	0186
ORIX	21EC	0364	0369
ORIV	2200	0380	0366 0385
FATSEF	2005	0007	0636 0643 0657 0664 0737 0751
FOSIX	200C	0014	0394 0399 0417 0420 0422 0432 0455 0457 0507
POSIV	200E	0015	0395 0400 0429 0432 0434 0465 0468 0469 0506
PRIN	2092	0115	0941
FUEDEGO	254C	0674	0664
FULSON	2003	0005	0361 0475
FULSOY	2004	0006	0362 0476
QUEPRO	2006	0013	0650 0655 0676
REBAJA	2190	0304	0315 0326
REPITE	2317	0545	0549
RES	2220	0389	0392
RESCONT	2210	0388	0367 0383
RETA	3577	0908	0903 0914 0932
SACA	2513	0649	0134
SAL	2231	0396	0382
SALE	2260	0475	0136 0461 0473
SALEINT	22F4	0666	0645

+++ ALVARO +++

SYMBOL	VALUE	DEFN	REFERENCES
SALVA	2002	0004	0372
SCAN	00FE	0001	0130 0167 0304 0540 0550 0619 0633 0640 0654 0661 0636
			0722 0862
SCANL	0624	0002	0196 0233
SCANLMO	214B	0229	0116 0118 0120 0122 0124 0126 0128 0177 0210 0596 0617
			0694 0790 0812
SEPA	2007	0009	
SEPARA	2468	0751	0724
START	2590	0941	0943
STEPBUF	1FAF	0004	0663 0670 0680 0682
SUBETAL	257F	0929	
TABCOAB	2056	0031	0591
TABCOCE	205C	0034	0594
TABCUAL	2044	0042	0129
TABDAT5	2038	0036	0120
TABERR1	206C	0078	0708
TABESC	203E	0039	0127
TABFIN	206E	0063	0206
TABINI	201A	0021	0115
TABINTE	2062	0037	0616
TABLIN	2074	0066	0692
TABMENU	2020	0024	0117
TABNETE	2026	0027	0119
TABNDAT	2066	0075	0610
TABSACA	202C	0030	0121
TBSALE	0032	0033	0123
TALAR	2588	0933	0933
TALBAJO	2307	0377	0374
TAPE	2535	0859	0851 0856 0872 0873
TONE	05E4	0067	0245 0323
TONGSEP	2142	0213	0309 0377 0593 0693 0769 0811 0909
TRES	2466	0763	0728
VALRET	0300	0090	0368
VERHOR	2068	0060	0618 0693
VE	23AA	0472	0437

+++ PERF-1 +++

```

0001 ,          SUBPROGRAMA +++ S A L E +++
0002
(2700)          0003          ORG 2700H
2700 (0001)     0004 ANCHO. DEFS 1
2701 (0001)     0005 ANGULO. DEFS 1
2702 (0002)     0006 CONTRA. DEFS 2
2704 (0001)     0007 CORTA. DEFS 1
2705 (0001)     0008 CUADRA. DEFS 1
2706 (0001)     0009 CUENTA. DEFS 1
2707 (0001)     0010 CUENTAX. DEFS 1
2708 (0002)     0011 DIRTAS. DEFS 2
2709 (0006)     0012 DISPLAY. DEFS 6
2710 (0002)     0013 DISTAR. DEFS 2
2712 (0001)     0014 DATOV. DEFS 1
(270A)         0015 DATO. EQU DISPLAY
2713 (0002)     0016 EQUIS. DEFS 2
2715 (0001)     0017 INTELIN. DEFS 1
2716 (0001)     0018 LEERTE. DEFS 1
2717 (0001)     0019 OFCION. DEFS 1
2718 (0001)     0020 PATASIN. DEFS 1
2719 (0001)     0021 PERFOR. DEFS 1
271A (0002)     0022 FOSIX. DEFS 2
271C (0002)     0023 FOSIV. DEFS 2
271E (0001)     0024 FULSOW. DEFS 1
271F (0001)     0025 FULSOY. DEFS 1
2720 (0001)     0026 SEPARA. DEFS 1
2721 (0001)     0027 SEPARA1. DEFS 1
2722 (0001)     0028 SIGNOV. DEFS 1
(270B)         0029 SIGNO. EQU DISPLAY+1
2723 (0002)     0030 WENTRE2. DEFS 2
2725 (0002)     0031 YE. DEFS 2

```



+++ PERF-1 +++

2727	0609	0003	TABLA1.	DEFW 0900H, 8 Y 9 PULSOS
2729	1011	0004		DEFW 1110H, 16 Y 17 PULSOS
272B	1110	0005		DEFW 1011H, 17 Y 16 PULSOS
272D	1314	0006		DEFW 1413H, 21 Y 20 PULSOS
272F	1919	0007		DEFW 1919H, 23 Y 23 PULSOS
2731	1A1A	0008		DEFW 1A1AH, 26 Y 26 PULSOS
2733	2122	0009		DEFW 2221H, 33 Y 34 PULSOS
2735	2A2A	0010		DEFW 2A2AH, 42 Y 42 PULSOS
2737	0202	0011	SALEXM.	DEFW 0302H, CODIGO 7 SEGMENTOS
2739	838F	0012		DEFW 8F83H, PARA MENSAJE "SALEXM"
273B	A63F	0013		DEFW 3FA6H, PARA MENSAJE "SALEXM"
273D	2F87	0014	NODAT.	DEFW 872FH, IDEN
273F	0602	0015		DEFW 0602H, PARA MENSAJE "NODAT"
2741	23A3	0016		DEFW A323H, PARA MENSAJE "NODAT"
2743	0230	0017	HIRA.	DEFW 3002H, IDEN MENSAJE "PERF-1"
2745	020F	0018		DEFW 0F02H, PARA MENSAJE "PERF-1"
2747	1F8F	0019		DEFW 8F1FH, PARA MENSAJE "PERF-1"
2749	0202	0020	ALTOXM.	DEFW 0202H, IDEN
274B	878D	0021		DEFW 8D87H, PARA MENSAJE "ALTOXM"
274D	2F85	0022		DEFW 852FH, PARA MENSAJE "ALTOXM"
274F	265B	0023	CTS??.	DEFW 5B26H, IDEN
2751	060E	0024		DEFW 060EH, PARA MENSAJE "CTS??"
2753	878D	0025		DEFW 8D87H, PARA MENSAJE "CTS??"
(1000)		0026	INITAS.	EQU 1000H
(0600)		0027	MOTOR.M.	EQU 0600H
(2104)		0028	ORIGEN.	EQU 2104H
(07F0)		0029	SEGTAS.	EQU 07F0H
(03DE)		0030	TONEIK.	EQU 03DEH
(00E2)		0031	TONEIK.	EQU 00E2H



+++ PERF-1 +++

2794	FE60	0112	CF	A. 060H
2796	2636	0113	JR	Z. LINEAS , LINEA VERTICAL
2798	FEF6	0114	CF	A. 0F0H
279A	2637	0115	JR	Z. LINEAS , LINEA HORIZONTAL
279C	FE60	0116	CF	A. 060H
279E	2640	0117	JR	C. NODATOS , DATO NO VALIDO
27A0	2A0427	0118	LD	A. <CORTA>
27A2	2D	0119	DEC	A
27A4	2666	0120	JR	Z. DENCORI , DENTRO DE CORTE
		0121		
27A6	CDEC27	0122	CALL	PERFORA , PERFORACION
27A9	18D0	0123	JR	HASDAT
		0124		+ + + FIN CLASIFICA + + +
27AB	2D	0125	CORTES. INC	HL
27AC	220827	0126	LD	<DIRTAB>, HL
27AF	18D3	0127	JR	DENCOR
27B1	CD072C	0128	DENCORI. CALL	BAJATAL
27B4	CDF627	0129	DENCOR. CALL	CORTE , EJECUTA CORTE
27B7	18D3	0130	JR	HASDAT
27B9	AF	0131	FINCOR. XOR	A , FINALIZA CORTE
27BA	220427	0132	LD	<CORTA>, A
27BD	2D	0133	INC	HL
27BE	2D	0134	INC	HL
27BF	2D	0135	INC	HL
27C0	2D	0136	INC	HL
27C1	220827	0137	LD	<DIRTAB>, HL, AFUNTA SIGUIENTE DATO
27C4	18D3	0138	JR	HASDAT
27C6	7E	0139	INTEDOS. LD	A. <HL>
27C7	221527	0140	LD	<INTELIN>, A, GUARDA CODIGO
27CA	2D	0141	INC	HL
27CB	220827	0142	LD	<DIRTAB>, HL
27CE	CD1E26	0143	CALL	INTEGRA , EJECUTA INTEGRADO
27D1	18A6	0144	JR	HASDAT
27D3	7E	0145	LINEAS. LD	A. <HL>
27D4	221527	0146	LD	<INTELIN>, A, GUARDA CODIGO
27D7	2D	0147	INC	HL
27D8	220827	0148	LD	<DIRTAB>, HL
27DB	CD0826	0149	CALL	LINEA , EJECUTA LINEA
27DE	1896	0150	JR	HASDAT , CONTINUA
27E0	DD213D27	0151	NODATOS. LD	IX, NODAT , AFUNTA MENSAJE "NO DAT"
27E4	CD052C	0152	TIENPO. CALL	SCAN , MUESTRA
27E7	FE12	0153	CF	A, 12H , HASTA PRESIONAR
27E9	26F9	0154	JR	NZ, TIENPO , TECLA CORRE
27EB	C9	0155	RET	





+++ PERP-1 +++

2856	2883	0256	JR	Z, VERTI	, DETERMINA POSICION DE INTEGRADO
		0257			
		0258			
2858	AF	0259	WOR	A	, INDICADOR INTEGRADO HORIZONTAL
		0260			
2859	1883	0261	JR	POSIIN	
285B	2888	0262	VERTI. LD	A, 08H	, INDICADOR INTEGRADO VERTICAL
		0263			
285D	221527	0264	POSIIN. LD	<INTELIN>, A, EN BIT 3 DE <INTELIN>	
2860	2E11	0265	LD	A, 11H	
2862	222027	0266	LD	<SEPARA>, A	, COLOCA SECUENCIA
2865	2E18	0267	LD	A, 16H	, DE PULSOS PARA
2867	222127	0268	LD	<SEPARA1>, A	, SEPARACION ENTRE PATAS
		0269			
286A	0D6629	0270	CALL	HILERA	, REALIZA PRIMER LADO DE INTEGRADO
		0271			
286D	2A1827	0272	LD	A, <PATASIN>	
2870	2D	0273	DEC	A	
2871	17	0274	LD	B, A	
2872	110000	0275	LD	DE, 0H	
2875	210000	0276	LD	HL, 0H	
2878	2E18	0277	LD	A, 16H	
287A	222127	0278	LD	<SEPARA1>, A	, SECUENCIA DE PULSOS PARA
287D	2C	0279	INC	A	, SEPARACION DE 100" IN
287E	222027	0280	LD	<SEPARA>, A	
2881	2A2027	0281	HASEFA. LD	A, <SEPARA>	, CALCULA DISTANCIA
2884	2F	0282	LD	E, A	
2885	19	0283	ADD	HL, DE	, A RETROCEDER
2886	05	0284	DEC	B	
2887	2888	0285	JR	Z, LISTOSE	, PARA REALIZAR
2889	19	0286	ADD	HL, DE	
288A	05	0287	DEC	B	, EL SEGUNDO LADO
288B	2807	0288	JR	Z, LISTOSE	
288D	2A2127	0289	LD	A, <SEPARA1>	, LA CUAL DEPENDE
2890	2F	0290	LD	E, A	
2891	19	0291	ADD	HL, DE	, DEL NUMERO DE PATAS
2892	18ED	0292	DJNZ, HASEFA		, DEL INTEGRADO
		0293			
2894	ED4B1C27	0294	LISTOSE. LD	BC, <POSIY>	, COORDENADAS DE
2898	ED061A27	0295	LD	DE, <POSIY>	, ULTIMA PERFORACION
289C	2A1527	0296	LD	A, <INTELIN>	
289F	052F	0297	BIT	D, A	, PRUEBA POSICION
28A1	280D	0298	JR	Z, INHOR	, DE INTEGRADO
28A3	05	0299	ADD	HL, BC	
28A4	14	0300	LD	B, H	, INDICA
28A5	1D	0301	LD	C, L	
28A6	2A0027	0302	LD	A, <ANCHO>	, POSICION
28A9	280D	0303	LD	H, 08H	
28AB	6F	0304	LD	L, A	, ARRIBA

+++ PERF-1 +++

26AC	19	0300	ADD	HL, DE	
26AD	EB	0306	EX	DE, HL	, A LA DERECHA
26AE	186E	0307	JR	UNLADO	
26B0	B7	0308	OR	A	
26B1	EB	0309	EX	DE, HL	, INDICA
26B2	ED52	0310	SSC	HL, DE	
26B4	EB	0311	EX	DE, HL	, POSICION
26B5	3A0037	0312	LD	A, (ANCHO)	
26B6	2608	0313	LD	H, 06H	, ARRIBA
26B8	6F	0314	LD	L, A	
26B9	09	0315	ADD	HL, BC	, A LA IZQUIERDA
26BC	44	0316	LD	B, H	
26BD	4D	0317	LD	C, L	
26BE	CD6829	0318	UNLADO. CALL	AVANZAR	, INICIO DE SEGUNDO
		0319	,		LADO DE INTEGRADO
26C1	3A1837	0320	LD	A, (PATASIN)	
26C4	321937	0321	LD	(PERFOR), A	
26C7	CD6629	0322	CALL	HILERA	, REALIZA SEGUNDO
		0323	,		LADO DE INTEGRADO
26CA	C9	0324	RET		





+++ PERF-1 +++

```

0273 ,+++++
0274 ,+SUBROUTINA. "HILERA"
0275 ,+
0276 ,+ DESCRIPCION.
0277 ,+ REALIZA UNA SECUENCIA DE PERFORACIONES EN +
0278 ,+ LINEA RECTA VERTICAL U HORIZONTAL A UNA +
0279 ,+ SEPARACION DEFINIDA, INICIANDO EN LA POSI- +
0280 ,+ CION EN QUE SE ENCUENTRA EL TALADRO. +
0281 ,+
0282 ,+ ENTRADA. NUMERO DE PERFORACIONES EN LOCALIDAD +
0283 ,+ (PERFOR) +
0284 ,+ SEPARACION ENTRE PERFORACIONES EN +
0285 ,+ LOCALIDADES (SEPARA) Y (SEPARA) +
0286 ,+ POSICION DE HILERA EN BIT 3 DE (INTELIN) +
0287 ,+ 0 --) ' HORIZONTAL 1 --) ' VERTICAL +
0288 ,+
0289 ,+ SALIDA. CONTROL DE MOTORES DE PASOS Y TALADRO +
0290 ,+ MODIFICA. AF, BC, DE, HL, AF', BC', DE', IX, IF +
0291 ,+ LLAMA. BAJATAL, SUBETAL Y AVANZAL +
0292 ,+++++
2906 2E03 0293 HILERA. LD A, 02H , PARA CONTROL
2906 220727 0294 LD (CUENTA), A , DE LA SECUENCIA
2906 220627 0295 LD (CUENTA), A , DE PULSOS
0296
290E CD072C 0297 HILERAL. CALL BAJATAL , REALIZA
2911 CD082C 0298 CALL SUBETAL , PERFORACION
2914 2A1327 0299 LD A, (PERFOR)
2917 2D 0400 DEC A , TERMINA SI ES ULTIMA
2918 221327 0401 LD (PERFOR), A
291B C6 0402 RET Z , PERFORACION
291C 6600 0403 LD B, 00H
291E 2A2627 0404 LD A, (SEPARA) , SEPARACION ENTRE
2921 4F 0405 LD C, A , PERFORACIONES
2922 2A1A27 0406 LD HL, (POSIV) , COORDENADAS DE
2925 ED061C27 0407 LD DE, (POSIV) , ULTIMA PERFORACION
2929 2A1527 0408 LD A, (INTELIN)
292C C60F 0409 BIT D, A , DETERMINA POSICION
292E 2004 0410 JR NZ, VERTICA , DE HILERA
292F 69 0411 ADD HL, BC , PONE SEPARACION
2931 E6 0412 EX DE, HL , PARA HILERA
0413 , HORIZONTAL
2932 1604 0414 JR PONYE
2934 E7 0415 VERTICA. EX DE, HL , PONE SEPARACION
2935 67 0416 OR A , PARA HILERA
2936 ED43 0417 SBC HL, BC , VERTICAL
2938 41 0418 PONYE. LD B, H
2939 4D 0419 LD C, L
293A CD6829 0420 CALL AVANZAL , AVANZA A SIGUIENTE
0421 , PERFORACION

```

+++ PERF-1 +++

293D	2A0627	0422	LD	A	<CUENTA>
2940	2D	0423	DEC	A	, PRUEBA SECUENCIA
2941	2005	0424	JR	Z, OTRASEP	, DE PULSOS
2943	2A0627	0425	LD	<CUENTA>, A	
2946	1806	0426	JR	HILERA1	, CONTINUA
2948	2A2127	0427	OTRASEP, LD	A, <SEPARA1>	
294B	17	0428	LD	B, A	
294C	2A2027	0429	LD	A, <SEPARA>	
294F	2A2127	042B	LD	<SEPARA1>, A	, CORRIJE
2952	76	0431	LD	A, B	
2953	2A2027	0432	LD	<SEPARA>, A	, SECUENCIA
2956	2A0727	0433	LD	A, <CUENTA>	
2959	2D	0434	DEC	A	, DE PULSOS
295A	200A	0435	JR	Z, HILERA	
295C	2A0727	0436	LD	<CUENTA>, A	, Y
295F	2A0627	0437	LD	<CUENTA>, A	, REPETICION
2962	180A	0438	JR	HILERA1	

+++ PERF-1 +++

```

0440 ,* * * * *
0441 ,+ SUBPROGRAMA. " AVANZA ", " AVANZA0 ", " AVANZA1 " ,+
0442 ,+ ,+
0443 ,+ DESCRIPCION. ,+
0444 ,+ CONTROLA LOS MOTORES DE PASOS PARA LOGRAR ,+
0445 ,+ EL MOVIMIENTO RELATIVO DEL TALADRO EN LOS ,+
0446 ,+ EJES "X" Y "Y", ESTE MOVIMIENTO RELATIVO ,+
0447 ,+ DEL TALADRO SE REALIZA EN " LINEA RECTA" ,+
0448 ,+ A UNA VELOCIDAD DEFINIDA, DESDE EL LUGAR ,+
0449 ,+ DONDE SE ENCUENTRA EL TALADRO HASTA EL ,+
0450 ,+ PUNTO QUE SE DESEA AVANZAR. ,+
0451 ,+ ,+
0452 ,+ ENTRADA. DIRECCION DEL PUNTO AL QUE SE DESEA ,+
0453 ,+ ,+
0454 ,+ SALIDA. CONTROL DE MOTORES DE PASOS "X" Y "Y" ,+
0455 ,+ Y TALADRO. ,+
0456 ,+ MODIFICA. AF, BC, DE, HL, AF', BC', DE', IN, IV. ,+
0457 ,+ LLAMA. DANEV, DANEV, SCANL, SUBETAL, SCAN Y AVANZA0 ,+
0458 ,+ * * * * *
2964 2E61 0459 AVANZA0 LD A, 01H , AVANZA SIN LEER TECLADO
2966 1807 0460 JR SILEET
2968 AF 0461 AVANZA1 XDR A
2969 321627 0462 LD (LEERTE), A , AVANZA SIN TRAER DATO
0463 , DE MEMORIA
296C 1824 0464 JR OADRANT
0465 ,-----
0466 , PARTE "A". " TRADAT "
0467 ,
0468 , DESCRIPCION. TRAE DE LA MEMORIA DEL KIT LAS COORDENA-
0469 , DAS DEL NUEVO PUNTO A PROCESAR. LOS DATOS
0470 , TRAIDOS ASI COMO SU DISTRIBUCION EN ME-
0471 , MORIA SON DE LA SIGUIENTE MANERA.
0472 ,
0473 , COORDENADA "X" EN EL REGISTRO FAR" DE"
0474 ,
0475 , MH MH-1 XL XL-1
0476 ,
0477 , COORDENADA "Y" EN EL REGISTRO FAR" BC"
0478 , VH VH-1 YL YL-1
0479 ,
0480 , YH: MH 0 PARA TODOS LOS DATOS
0481 ,
0482 , DIRTAB: MH-1 XL
0483 , XL-1 YL-1
0484 , YH-1 YL
0485 ,-----
296E AF 0486 AVANZA. XDR A
296F 321627 0487 SILEET. LD (LEERTE), A
2970 3A6827 0488 LD HL, (DIRTAB), DIRECCION DE NUEVO DATO

```

```

      0189 ,                A TRAER EN "HL"
2975 7E      0190  TRADAT.LD  A, <HL>
2976 07      0191  LD        D, A      , TRAE DATO "X"
2977 23      0192  INC        HL       , EN REGISTRO
2978 7E      0193  LD        A, <HL>   , PAR
2979 0F      0194  LD        E, A      , "DE"
297A 07      0195  RLCA
297B 07      0196  RLCA
297C 07      0197  RLCA
297D 07      0198  RLCA
297E 4F      0199  LD        C, A      , TRAE DATO "Y"
                0200                , EN LOS REGISTROS
297F 23      0201  INC        HL
2980 7E      0202  LD        A, <HL>   , "A" Y "C"
2981 0601    0203  LD        B, 04H
2982 063A    0204  ROTATE, SRL  D      , CORRIGE LOS DATOS
2983 061B    0205  RR        E      , "X" Y "Y"
2984 063F    0206  SRL      A      , HACIENDO CORRIMIENTOS
2985 0619    0207  RR        R      , A LA DERECHA
2986 16F6    0208  DJNZ, ROTAE
298D 47      0209  LD        B, A
298E 23      0210  INC        HL
298F 2C0627  0211  LD        <DIRTAB>, HL, ALHACENA DIRECCION
                0212                ACTUAL
                0213
                -----
                0214  , PARTE "B". " QADRANT "
                0215
                0216  , DESCRIPCION. CALCULA LAS COORDENADAS Y EL CUADRANTE
                0217  , DONDE SE ENCUENTRA EL NUEVO PUNTO CON
                0218  , RESPECTO A LA POSICION ACTUAL
                0219  , LOS RESULTADOS SE ALMACENAN DE LA
                0220  , SIGUIENTE MANERA.
                0221  , NUEVA COORDENADA "X" EN "DE"
                0222  , NUEVA COORDENADA "Y" EN "BC"
                0223  , DATO DE CUADRANTE EN LOCALIDAD (CUADRA)
                0224  ,
                0225  , "X" 0' "Y" - 0 --> EJE POSITIVO
                0226  , "X" 0' "Y" - 1 --> EJE NEGATIVO
                0227  ,
                0228  , CUADRANTE (CUADRA)
                0229  , XY
                0230  , PRIMERO 000000 00
                0231  , SEGUNDO 000000 10
                0232  , TERCERO 000000 11
                0233  , CUARTO 000000 01
                -----
2992 2A1C27  0235  QADRANT.LD  HL, <POSIV>, TRAE COORDENADA "Y" DE
                0236                POSICION ACTUAL
2993 ED131C27 0237  LD        <POSIV>, BC, GUARDA COORDENADA "Y" DE
  
```

+++ PERF-1 +++

					NUEVA POSICION
2999	AF	0538			
		0539	MOR	A	
299A	ED:2	0540	SBC	HL,BC	, ENCUENTRA NUEVA
		0541			COORDENADA "Y"
299C	FAB229	0542	JF	H, COMFLEY	, CORRIGE A VALOR POSITIVO
		0543			
299F	CBC7	0544	SET	G,A	, INDICADOR "Y"
29A1	1868	0545	JR	CARGAY	
29A2	:4	0546	COMFLEY.LD	B,H	, CORRECCION
29A4	:D	0547	LD	C,L	
29A5	218888	0548	LD	HL,GH	, DE
29A8	B7	0549	OR	A	
29A9	ED:2	0550	SBC	HL,BC	, COORDENADA "Y"
29AB	:4	0551	CARGAY.LD	B,H	, CARGA EN "BC" LA NUEVA
		0552			
29AC	:D	0553	LD	C,L	, COORDENADA "Y"
		0554			
29AD	B7	0555	OR	A	
29AE	3A1A27	0556	LD	HL (POSIX),	TRAE COORDENADA "X" DE
		0557			POSICION ACTUAL
29B1	ED32A27	0558	LD	(POSIX),	DE, GUARDA COORDENADA "X" DE
		0559			LA NUEVA POSICION
29B5	ED32	0560	SBC	HL,DE	, ENCUENTRA NUEVA
		0561			COORDENADA "X"
29B7	FAB229	0562	JF	H, COMFLEY	, CORRIGE A VALOR POSITIVO
		0563			
29BA	CBCF	0564	SET	I,A	, INDICADOR "X"
29BC	1887	0565	JR	CARGAY	
29BE	EB	0566	COMPLEX.EX	DE,HL	, CORRECCION
29BF	218888	0567	LD	HL,GH	, DE
29C2	B7	0568	OR	A	
29C3	ED32	0569	SBC	HL,DE	, COORDENADA "X"
		0570			
29C5	EB	0571	CARGAY.EX	DE,HL	, CARGA EN "DE"
		0572			LA COORDENADA "X"
		0573			
29C6	328527	0574	LD	(CUADRA),	A, GUARDA DATO DE CUADRANTE
		0575			
		0576			-----
		0577			, PARTE "C". " ANGULO "
		0578			
		0579			, DESCRIPCION. CALCULA EL ANGULO COMPRENDIDO ENTRE LA
		0580			POSICION ACTUAL Y EL NUEVO PUNTO.
		0581			
		0582			, SI EL ANGULO ES MAYOR DE 15 GRADOS.
		0583			LOCALIDAD (ANGULO) - UNO
		0584			INTERCAMBIA NUEVAS COORDENADAS "X" Y "Y".
		0585			DE LO CONTRARIO.
		0586			LOCALIDAD (ANGULO) - CERO
		0587			-----

+++ PERF-1 +++

29C9	62	0587	ANGULO: LD	H, D	, "X" EN
29CA	68	0588	LD	L, E	, REGISTRO "HL"
29CB	AF	0589	XOR	A	
29CC	ED42	0590	SBC	HL, BC	, COMPARA LAS NUEVAS COORDENADAS
		0591	,		"X" Y "Y"
29CE	F2D729	0592	JF	F, INDICA	SI "X" ES MAYOR O IGUAL A "Y"
		0593	,		FOENE INDICADOR CERO
29D1	60	0594	LD	H, B	, INTERCAMBIA
29D2	69	0595	LD	L, C	
29D3	EB	0596	EX	DE, HL	, COORDENADAS
29D4	44	0597	LD	B, H	
29D5	4D	0598	LD	C, L	, "X" Y "Y"
29D6	3C	0599	INC	A	, INDICADOR UNO
29D7	320137	0600	INDICA. LD	(ANGULO), A, FOENE INDICADOR DE ANGULO	
		0601			
		0602			
		0603			, PARTE "D"
		0604			
		0605	DESCRIPCION. LA SUBROUTINA "AVANZA" TERMINA SI EL TA-		
		0606	LADRO SE ENCUENTRA EN EL PUNTO AL QUE SE		
		0607	DESEA AVANZAR. DE LO CONTRARIO ESTABLECE		
		0608	CONDICIONES INICIALES Y PROSIGUE A EJE-		
		0609	CUTAR EL ALGORITMO "RECTA" HACIA EL NUE-		
		0610	VO PUNTO.		
		0611			
		0612			
		0613			
29DA	21FFFF	0614	LD	HL, 0FFFFH	
29DD	19	0615	ADD	HL, DE	, TERMINA SI
29DE	D0	0616	RET	NC	, DATO "X" ES CERO
		0617			
29DF	ED42327	0618	LD	(YE), BC	
29E3	ED01327	0619	LD	(EQUIS), DE	, GUARDA
29E7	ED030327	0620	LD	(CONTR), DE	, "VARIABLES"
29EB	210000	0621	LD	HL, GH	, EN
29EE	321637	0622	LD	(DSTAR), HL	, MEMORIA
29F1	14	0623	INC	D	
29F2	15	0624	DEC	D	
29F3	2000	0625	JR	NZ, HITAD	, EN ALGORITMO "RECTA"
29F5	1D	0626	DEC	E	, LA HITAD DE "X"
29F6	2001	0627	JR	NZ, SUBE1	, DEBE SER
29F8	1C	0628	INC	E	, DIFERENTE DE CERO
29F9	1C	0629	SUBE1. INC	E	
29FA	CB3A	0630	HITAD. SRL	D	
29FC	CB18	0631	RR	E	, DIVIDE "X" ENTRE DOS
29FE	ED032327	0632	LD	(NENTRE), DE	, Y LO GUARDA

+++ PERF-1 +++

```

0634 /-----
0635 , PARTE "E". ALGORITMO "RECTA"
0636 /
0637 , DESCRIPCION. EL ALGORITMO "RECTA", PROPORCIONA UN INDICADOR
0638 , QUE REPRESENTA EL MOVIMIENTO A REALIZAR PARA OBTENER LA APROXIMACION DE LA
0640 , RECTA QUE SE ENCUENTRA EN EL PRIMER OCTANTE DEL PLANO "X-Y".
0641 /
0642 /
0643 , SI CARRY = 0 --)/ MOVIMIENTO EN EJE "X"
0644 , SI CARRY = 1 --)/ MOVIMIENTO EN EJES "X-Y"
0645 /-----
0646 RECTA. LD BC, (YE)
0647 LD DE, (WENTREZ)
0648 LD HL, (DSTAR)
0649 ADD HL, BC , DISTANCIA
0650 LD (DSTAR), HL , APROXIMACION-REAL
0651 XOR A
0652 SBC HL, DE , SI "WENTREZ" ES MENOR
0653 JP P, FONSET , QUE "DSTAR", CARRY-UNO
0654 / DE LO CONTRARIO
0655 , CARRY = CERO
0656 JR SACAPUL
0657 FONSET. LD HL, (DSTAR), CAMBIA
0658 LD DE, (EQUIS), DISTANCIA
0659 XOR A , APROXIMACION-REAL
0660 SBC HL, DE
0661 LD (DSTAR), HL, Y GUARDA
0662 SCF , CARRY = UNO
0663 /-----
0664 , PARTE "F". " SACAPUL "
0665 /
0666 , DESCRIPCION. INTERPRETA EL INDICADOR DEL ALGORITMO
0667 , "RECTA" PARA AJUSTARLO A LOS OCHO OCTANTES DEL PLANO "X-Y", PROPORCIONANDO A LOS
0668 , MOTORES DE FASOS "X" Y "Y" LOS PULSOS
0669 , APROPIADOS PARA OBTENER LA APROXIMACION
0670 , DE LA RECTA EN DICHO PLANO.
0671 /-----
0672 /
0673 SACAPUL. JR C, EXISYE , C 1 --)/ MOVIMIENTO "XY"
0674 LD A, (ANGULO)
0675 DEC A
0676 JR Z, YES , Z 1 --)/ MOVIMIENTO "Y"
0677 CALL DAHEX , TRAE NUEVO PULSO
0678 , DE MOTOR "X"
0679 LD B, A
0680 LD A, (PULSOY) , TRAE PULSO DE MOTOR "Y"
0681 AND GFH
0682 JR SACA

```

2A2B	CEDECB	0683	YES.	CALL	DAVEY	, TRAE NUEVO PULSO DE MOTOR "Y"
		0684	,			
2A2E	17	0685		LD	B, A	
2A2F	2A1E27	0686		LD	A, (PULSO)	, TRAE PULSO DE MOTOR "X"
2A12	E6F8	0687		AND	BFGH	
2A14	1887	0688		JR	SACA	
2A16	CD03CB	0689	EXISTE.	CALL	DAVEN	, TRAE NUEVO PULSO DE MOTOR "X"
		0690	,			
2A19	17	0691		LD	B, A	
2A1A	CEDECB	0692		CALL	DAVEY	, TRAE NUEVO PULSO DE MOTOR "Y"
		0693	,			
2A1D	B8	0694	SACA.	OR	B	, JUNTA AMBOS DATOS
2A1E	D288	0695		OUT	(MOTORX), A	, SACA DATO-PULSO A MOTORES
2A50	2A8427	0696		LD	A, (CORTA)	
2A52	3D	0697		DEC	A	
2A54	3816	0698		JR	NZ, SIAL	
2A56	21C888	0699		LD	HL, 00C8H	, RETARDO PARA VELOCIDAD
2A59	CD2885	0700		CALL	TOHE2K	
2A5C	181C	0701		JR	TECLADO	
2A5E	ED38827	0702		LD	DE, (CONTAR)	
2A62	18	0703		DEC	DE	
2A63	21FFFF	0704		LD	HL, 0FFFFH	, CONTINUA
2A66	19	0705		ADD	HL, DE	, AVANZANDO
2A67	ED38827	0706		LD	(CONTAR), DE	, HASTA
2A6B	D8	0707		RET	NC	, CONTADOR-CERO
2A6C	2A1627	0708	SIAL.	LD	A, (LEERTE)	
2A6F	3D	0709		DEC	A	, SI (LEERTE) UNO
2A70	3888	0710		JR	NZ, TECLADO	, NO LEE TECLADO
2A72	216488	0711		LD	HL, 0064H	, RETARDO SOLO
2A73	CD2885	0712		CALL	TOHE1K	, PARA PARTE "ALTO "
2A76	1888	0713		JR	RECTA	
2A7A	DDE5	0714		TECLADO.PUSH	IX	
2A7C	DD214227	0715		LD	IX, HIRA	
2A80	CD862C	0716		CALL	SCAN1	, SCAN1, LEE EL TECLADO
2A82	DDE1	0717		POP	IX	
2A85	FE28	0718		CF	A, 26H	, SI TECLA "ALTO" ENTONCES
2A87	2882	0719		JR	Z, ALTO	, DETENER EJECUCION
		0720	,			, DE LO CONTARIO
2A89	C882A	0721		JF	RECTA	, CONTINUA
		0722				
		0723				-----
		0724				, PARTE "G". "ALTO "
		0725				,
		0726				, DESCRIPCION. PERMITE EL DESPLAZAMIENTO RELATIVO DEL
		0727				TALADRO A CUALQUIER PUNTO INDICADO EN LA
		0728				TABLA DE DATOS, ENTRE INTERVALOS DE + - 15
		0729				(F) PUNTOS.
		0730				INDICAR POR TECLADO EL PUNTO AL QUE SE
		0731				DESEA AVANZAR.
		0732				-----



+++ PERF-1 +++

```

0732 ,
0733 , HUESTRA LOS DESPLEGADOS "ALTO" Y "CTS ??"
0734 , INTERPRETA LOS DATOS LEIDOS A TRAVES DEL
0735 , TECLADO, MOSTRANDO EN DISPLAY LOS DATOS
0736 , VALIDOS.
0737
-----
2A6C 2AF627 0738 ALTO. LD A, <CORTE>
2A6F 2D DEC A , PRUEBA SI DENTRO DE CORTE
2A90 2002 0740 JR NZ, NOSUBE , DECIDE SUBIDA DE TALADRO
2A92 CD082C 0741 CALL SUBETAL , SUBE TALADRO
2A93 AF 0742 NOSUBE. MCR A
2A96 221727 0743 LD <OPCION>, A , CONTROL DE SALIDA PARTE
2A99 3C 0744 INC A , ALTO
2A9A DDEG 0745 PUSH IX , SALVA POSICION EJE "X"
2A9C DD211927 0746 LD IX, ALTOX , APUNTA A MENSAJE "ALTO"
2A9D 0560 0747 LD B, CCH
2A9E CD062C 0748 HUESTRA CALL SCAN1 , HUESTRA
2AA0 16FB 0749 DJNZ, HUESTRA UN MOMENTO
2AA7 2A6627 0750 INICIA LD HL, <DIRTAB>
2AAA 2B 0751 DEC HL
2AAB 2B 0752 DEC HL
2AAC 2B 0753 DEC HL
2AAD 2B 0754 DEC HL , PONE DIRECCION DE TABLA
2AAE 7E 0755 LD A, <HL>
0756 , , AL FUNTO QUE SE DIRIGIA
2AAF FECD 0757 CP A, 0CDH
2AB1 2504 0758 JR Z, ININC , PRUEBA INICIO DE TABLA
2AB2 FEB6 0759 CP A, 056H
2AB3 2601 0760 JR NC, ESMAYOR
2AB7 23 0761 ININC. INC HL
2AB8 220827 0762 ESMAYOR LD <DIRTAB>, HL
2ABB 214F27 0763 INICIA LD HL, CTS??
2ABE 110A27 0764 LD DE, DISPLAY , COPIA MENSAJE "CTS ??"
2AC1 010600 0765 LD BC, 0066H , EN ZONA DE DISPLAY
2AC4 ED80 0766 LDIR
2AC6 DD210A27 0767 LD IX, DISPLAY , APUNTA A DISPLAY
2ACA CD052C 0768 MOSTRAR CALL SCAN , HUESTRA MIENTRAS
2ACD FE12 0769 CP A, 12H , DATO SEA NO VALIDO
2ACF F3C0CA 0770 JF F, MOSTRAR
2AD2 FE12 0771 CP A, 12H , PRUEBA SI TECLA CORRE
2AD4 2620 0772 JR Z, CASETA
2AD5 211727 0773 LD HL, OPCION , INDICADOR
2AD9 0BFE 0774 SET 7, <HL> , INICIO DE DATOS
2ADB FE10 0775 CP A, 10H , PRUEBA TIPO DE DATO
2ADD FAF3CA 0776 JF M, HEVADec
2AE0 2207 0777 JR Z, SIGNAS
2AE2 202227 0778 LD <SIGNOV>, A
2AE5 2E62 0779 LD A, 02H , CODIGO 75E0 PARA (-)
2AE7 1803 0780 JR PONSIG

```

+++ PERF-1 +++

2AE9	320227	0761	SIGNAS. LD	(SIGNOV), A	
2AEC	3E23	0762	LD	A, 32H	, CODIGO 7SEG PARA (+)
2AEE	320627	0763	FORNIG. LD	(SIGNO), A	
2AF1	16D7	0764	JR	HOSTRAR	
2AF3	321227	0765	HEXADEC. LD	(DATOV), A	
2AF6	21F067	0766	LD	HL, SEGTAB	
2AF9	0600	0767	LD	B, 06H	, CALCULA
2AFB	4F	0768	LD	C, A	, CODIGO 7SEG
2AFC	09	0769	ADD	HL, EC	, PARA DATO
2AFD	7E	0790	LD	A, (HL)	, HEXADECIMAL
2AFE	320A27	0791	LD	(DATO), A	
2B01	18C7	0792	JR	HOSTRAR	
2B03	3A1727	0793	CASETA. LD	A, (OPCION)	
2B06	17	0794	RLA		
2B07	3803	0795	JR	C, CASETA1	, SI NO SE INICIARON
2B09	C2C92B	0796	JF	SALIDA	, DATOS PUEDE TERMINAR
		0797	,	" ALTO "	
2B0C	3A0A27	0798	CASETA1. LD	A, (DATO)	
2B0F	FE36	0799	CF	A, 36H	, SI HAY TECLA CORRE
2B11	35B7	0800	JR	Z, HOSTRAR	, Y NO SE COMPLETO
2B13	47	0801	LD	B, A	, SIGNO O DATO
2B14	2A0627	0802	LD	A, (SIGNO)	, CONTINUA
2B17	FE36	0803	CF	A, 36H	, HOSTRANDO
2B19	26AF	0804	JR	Z, HOSTRAR	, DISPLAY
2B1B	2E5D	0805	LD	A, 0E5H	
2B1D	66	0806	CF	A, B	, SI DATO ES CERO
2B1E	2667	0807	JR	HL, ACTUA	
2B20	211727	0808	LD	HL, OFCION	
2B23	C65E	0809	RES	7, (HL)	
2B25	1894	0810	JR	INICIA	, INICIA LECTURA
		0811	,	DE DATOS	
		0812			
2B27	DDE1	0813	ACTUA. FOF	IX	, OBTIENE POSICION EJE "X"
2B29	D0221A27	0814	LD	(FOPIX), IX	, ACTUALIZA
2B2D	F0221C27	0815	LD	(FOPIY), IY	, POSICION
2B31	3A1227	0816	LD	A, (DATOV)	
2B34	47	0817	LD	B, A	
2B37	3A2227	0818	LD	A, (SIGNOV)	
2B38	FE10	0819	CF	A, 10H	, DETERMINA MOVIMIENTO
2B3A	2629	0820	JR	Z, ADE	, EN LA TABLA
		0821			
		0822	,		
		0823	,		
		0824	,	PROPORCIONA EL DESPLAZAMIENTO HACIA	
		0825	,	ATRAS EN LA TABLA DE DATOS	
		0826	,		
		0827	,		
2B3C	2A0627	0828	ATRADE. LD	HL, (DIRTAB)	
2B3F	36	0829	DEC	HL	

\*\*\* PERF-1 \*\*\*

2610	7E	0620	LD	A, <HL>	
2611	FEC0	0621	CF	A, 0C0H	, PRUEBA INICIO DE TABLA
2612	266E	0622	JR	Z, EJECUTA	
2613	26	0623	DEC	HL	
2614	26	0624	DEC	HL	
2615	26	0625	DEC	HL	, ATRAS UN DATO
2616	7E	0626	LD	A, <HL>	
2617	FEC0	0627	CF	A, 0C0H	, PRUEBA INICIO DE TABLA
2618	266E	0628	JR	NZ, ATRAS	
2619	23	0629	INC	HL	
261E	200627	0610	LD	<DIRTAB>, HL	
2621	160D	0611	JR	EJECUTA	
2622	FEB0	0612	ATRAS. CF	A, 060H	, AJUSTA INICIO DE DATO
2623	2001	0613	JR	NC, NOUN	
2627	23	0614	INC	HL	
2628	200627	0615	NOUN. LD	<DIRTAB>, HL	
2629	7E	0616	LD	A, <HL>	
262C	FEB0	0617	CF	A, 060H	
262E	F0712B	0618	JF	HL, ESULTI	
2631	FEC0	0619	CF	A, 0C0H	, PRUEBA INICIO DE CORTE
2632	2002	0620	JR	NZ, FINCORT	
2635	AF	0621	WOR	A	, INDICADOR FUERA DE CORTE
2636	1606	0622	JR	FONCORT	
2638	FECF	0623	FINCORT. CF	A, 0CFH	, PRUEBA FIN DE CORTE
263A	2005	0624	JR	NZ, ESULTI	
263C	2E01	0625	LD	A, 01H	, INDICADOR DENTRO DE CORTE
263E	20127	0626	FONCORT. LD	<CORTA>, A	, PONE CONDICION DE CORTE
2671	1003	0627	ESULTI. DJNZ, ATRADE		, HASTA DATO HEXADECIMAL 0
2672	162B	0628	JR	EJECUTA	, 0 INICIO DE TABLA
		0629			
		0660			, PROPORCIONA EL DESPLAZAMIENTO HACIA
		0661			, ADELANTE EN LA TABLA DE DATOS
		0662			
		0663			
2675	05	0664	ADE. DEC	B	
2676	2628	0665	JR	Z, EJECUTA	
2678	2A0627	0666	DELANTE. LD	HL, <DIRTAB>	, TRAE DATO AFUNTADO
267E	7E	0667	LD	A, <HL>	, FOR <DIRTAB>
267C	FEC1	0668	CF	A, 0C1H	, SI ES FIN DE TABLA
267E	261F	0669	JR	Z, DECRE4	, SE COLOCA EN ULTIMO DATO
		0670			
2630	FEB0	0671	CF	A, 060H	
2662	2611	0672	JR	C, INCR3	
2664	FEC0	0673	CF	A, 0C0H	, PRUEBA INICIO DE CORTE
2666	2001	0674	JR	NZ, FINACOR	
2628	2E01	0675	LD	A, 01H	, INDICADOR DENTRO DE CORTE
		0676			
266A	1605	0677	JR	FONCOR	
266C	FECF	0678	FINACOR. CF	A, 0CFH	, PRUEBA FIN DE CORTE

+++ PERF-1 +++

268E	2004	0679	JR	NZ, INCRE4	
2690	AF	0680	MOR	A	, INDICADOR FIN DE
		0681	,		CORTE
2691	220427	0682	FONCOR. LD	(CORTA), A	, INDICA CONDICION DE CORTE
2694	23	0683	INCRE4. INC	HL	
2695	23	0684	INCRE3. INC	HL	
2696	23	0685	INC	HL	
2697	23	0686	INC	HL	, AVANZA AL SIGUIENTE DATO
2698	7E	0687	LD	A, (HL)	
2699	FEC1	0688	CF	A, 0C1H	, PRUEBA FIN DE TABLA
269B	2002	0689	JR	Z, DECRE4	
269D	180C	0690	JR	SIULTI	
269F	26	0691	DECRE4. DEC	HL	
26A0	26	0692	DEC	HL	
26A1	26	0693	DEC	HL	
26A2	26	0694	DEC	HL	, RETROCEDE UN DATO
26A3	7E	0695	LD	A, (HL)	
26A4	FEB0	0696	CF	A, 060H	, AJUSTA INICIO DE DATO
26A6	2001	0697	JR	NC, MAYOR	
26A8	23	0698	INC	HL	
26A9	0601	0699	MAYOR. LD	B, 01H	, TERMINA CON FIN DE TABLA
26AB	220027	0600	SIULTI. LD	(DIRTAB), HL	
26AE	10C8	0901	DJNE, DELANTE		, HASTA DATO HEXADECIMAL 0
		0902	,		O FIN DE TABLA
26B0	2A0027	0903	EJECUTA. LD	HL, (DIRTP)	
26B3	7E	0904	LD	A, (HL)	
26B4	FEB0	0905	CF	A, 060H	, AJUSTA INICIO DE DATO
26B6	2001	0906	JR	C, NOUNO	
26B8	23	0907	INC	HL	
26B9	220027	0908	NOUNO. LD	(DIRTAB), HL	
26BC	006429	0909	CALL	AVANZAS	, AVANZA AL PUNTO DESEADO
26BF	DCE3	0910	PUSH	IX	, GUARDA POSICION EJE "X"
26C1	211727	0911	LD	HL, OPCION	
26C4	068E	0912	RES	Z, (HL)	, LIMPIA PARA OTRO POSIBLE
		0913	,		MOVIMIENTO
26C6	03A72A	0914	JF	INICIA	
26C9	0DE1	0915	SALIDA. POP	IX	, RECUPERA POSICION EJE "X"
26CB	00221A27	0916	LD	(POSIX), IX	, ACTUALIZA
26CF	FD221C27	0917	LD	(POSIX), IX	, POSICION
26D3	E1	0918	POP	HL	, REGRESA A "SALE"
26D4	09	0919	RET		

+++ PERF-1 +++

		0921	+++++			
		0922	+	SUBROUTINA.	" DANEY "	+
		0923	+	DESCRIPCION.		+
		0924	+	ESTA SUBROUTINA PROPORCIONA EL DATO-PULSO		+
		0925	+	PARA EL MOTOR "X". ESTE DATO-PULSO REPRE-		+
		0926	+	SENTA UN MOVIMIENTO HACIA ADELANTE O HA-		+
		0927	+	CIA ATRAS EN EL EJE "X".		+
		0928	+	ENTRADA.	CUADRANTE EN LOCALIDAD <CUADRA>	+
		0929	+	SALIDA.	DATO-PULSO EN REGISTRO "A"	+
		0930	+	MODIFICA.	AF, IN, IY, <PULSOX>	+
		0931	+	LLAMA.	NINGUNA	+
		0932	+++++			+++++
26D7	3A0327	0933	DANEY.	LD	A, <CUADRA>	
26D8	CB4F	0934	BIT	1, A	, PRUEBA INDICADOR EJE "X"	
26D9	3A1E27	0935	LD	A, <PULSOX>	, PULSO ANTERIOR	
26D0	2605	0936	JR	Z, MENY		
26DF	0F	0937	RRCR		, NUEVO PULSO HACIA ADELANTE	
26E0	DD3B	0938	DEC	IX	, POSICION EJE "X"	
26E2	1803	0939	JR	LISTOX		
26E4	07	0940	MENY.	RLCA	, NUEVO PULSO HACIA ATRAS	
26E5	DD33	0941	INC	IX	, POSICION EJE "X"	
26E7	321E27	0942	LISTOX.	LD	<PULSOX>, A, GUARDA PULSO	
26EA	E6F0	0943	AND	A, OFGH	, SEPARA NIBLE "X"	
26EC	C9	0944	RET			
		0945				
		0946	+++++			+++++
		0947	+	SUBROUTINA.	" DANEY "	+
		0948	+	DESCRIPCION.		+
		0949	+	ESTA SUBROUTINA PROPORCIONA EL DATO-PULSO		+
		0950	+	PARA EL MOTOR "Y". ESTE DATO PULSO REPRE-		+
		0951	+	SENTA UN MOVIMIENTO HACIA ADELANTE O HA-		+
		0952	+	CIA ATRAS EN EL EJE "Y".		+
		0953	+	ENTRADA.	CUADRANTE EN LOCALIDAD <CUADRA>	+
		0954	+	SALIDA.	DATO-PULSO EN REGISTRO "A"	+
		0955	+	MODIFICA.	AF, IN, IY, <PULSOY>	+
		0956	+	LLAMA.	NINGUNA	+
		0957	+++++			+++++
26ED	3A0327	0958	DANEY.	LD	A, <CUADRA>	
26F0	CB47	0959	BIT	0, A	, PRUEBA INDICADOR EJE "Y"	
26F2	3A1F27	0960	LD	A, <PULSOY>	, PULSO ANTERIOR	
26F3	2605	0961	JR	Z, MENY		
26F7	0F	0962	RRCR		, NUEVO PULSO HACIA ADELANTE	
26F8	FD3B	0963	DEC	IY	, POSICION EJE "Y"	
26FA	1803	0964	JR	LISTOY		
26FC	07	0965	MENY.	RLCA	, NUEVO PULSO HACIA ATRAS	
26FD	FD33	0966	INC	IY	, POSICION EJE "Y"	
26FF	321F27	0967	LISTOY.	LD	<PULSOY>, A, GUARDA PULSO	
2C02	E66F	0968	AND	A, OFH	, SEPARA NIBLE "Y"	
2C04	C9	0969	RET			

+++ PERF-1 +++

2C05	00	0971	, ++++++		
2C06	00	0972	SCAN.	NOF	
2C07	00	0973	SCANL.	NOF	
2C08	00	0974	BAJATAL.	NOF	
2C09	00	0975	SUBETAL.	NOF	
2C0A	C3	0976	RET		
2C0B	0D5527	0977	START.	CALL	SALE
2C0C	030000	0978	JF		0000H
2C10	(2C0A)	0979	END		START

ERRORS	0
RANGE COUNT	0

+++ PERF-1 +++

SYMBOL	VALUE	DEFN	REFERENCES
ACTUA	2627	0613	0607
ADE	2670	0664	0626
ALTO	2660	0738	0719
ALTOXX	2749	0600	0746
ANCHO	2700	0604	0203 0303 0312
ANGULO	2701	0600	0600 0674
ANGULOX	2909	0507	
ATRADE	2630	0628	0607
ATRAS	2632	0642	0628
AVANZA	296E	0486	0170 0188 0200 0221 0341
AVANZAO	2964	0459	0309
AVANZAI	2968	0461	0318 0420
BAJATAL	2007	0374	0128 0171 0189 0397
CARGAX	2900	0571	0500
CARGAY	29A8	0501	0540
CASETA	2603	0793	0772
CASETAI	260C	0798	0790
COMPLEM	295E	0566	0562
COMPLEY	29AD	0546	0542
CONTAX	2702	0606	0620 0702 0706
CONTIN	2611	0200	0190
CORRE	2760	0607	0609
CORTA	2704	0607	0605 0118 0122 0191 0197 0201 0696 0656 0682
CORTE	27F6	0186	0129 0728
CORTES	27A8	0120	0104
CTS??	274F	0600	0762
CUADRA	2700	0608	0374 0923 0308
CUENTA	2706	0609	0390 0422 0420 0437
CUENTAX	2707	0610	0294 0422 0426
CAMEX	2600	0923	0677 0689
CAMEY	26E0	0908	0682 0692
CATO	270A	0610	0791 0798
CATOV	2712	0614	0700 0816
DECRE1	269F	0691	0669 0669
DEFINAX	264B	0201	0247 0249
DEFPAT	2626	0228	0220 0224
DELANTE	2676	0666	0901
DENCOR	2704	0129	0127
DENCOR1	2701	0128	0120
DIRTAB	2708	0611	0690 0698 0126 0127 0142 0148 0192 0199 0468 0511 0700 0752 0828 0540 0640 0656 0900 0902 0908
DISPLAY	270A	0612	0610 0629 0764 0767
DISTAR	2710	0612	0622 0648 0600 0607 0661
EJECUTA	2680	0902	0820 0641 0656 0660
EQUIS	2710	0616	0649 0658
ESMAYOR	2688	0702	0700
ESULTI	2671	0607	0648 0654
EMISYE	2A46	0629	0672

+++ PERF-1 +++

SYMBOL	VALUE	DEFN	REFERENCES
FINCOR	266C	0676	0674
FINCOR	276F	0131	0106
FINCORT	2668	0652	0650
HEXADEC	26FD	0763	0776
HILERA	2906	0393	0370 0322 0270 0435
HILERAL	290E	0397	0426 0426
HORIZON	26EA	0356	0352
INCR3D	2695	0664	0672
INCR4	2694	0662	0679
INDICA	26D7	0660	0592
INHOR	2660	0360	0258
INICIA	2666	0762	0810
INICIAA	26A7	0750	0914
ININC	26B7	0761	0756
INITAB	1000	0056	0090
INTECOS	2706	0139	0109 0111
INTEGRA	261E	0321	0142
INTELIN	2715	0017	0140 0146 0322 0253 0264 0296 0312 0257 0406
LEERTE	2716	0018	0462 0467 0706
LINEA	26C6	0241	0149
LINEAS	27D2	0145	0112 0115
LISTOSE	2694	0294	0265 0266
LISTOX	26E7	0942	0939
LISTOY	26FF	0367	0364
MASCAT	2776	0096	0122 0120 0138 0144 0150
MASEFA	2661	0361	0292
MAYOR	26A9	0699	0697
MEH:	26E4	0940	0926
MEHY	26FC	0965	0961
MIRA	2742	0617	0745
HITACK	26FA	0620	0620
HOSTRAR	26CA	0766	0770 0764 0732 0300 0804
HOTORAY	0030	0057	0695
HUESTRA	26A2	0748	0749
MODAT	272D	0044	0151
MODATOS	27E0	0151	0692 0117
MOSUBE	2695	0712	0740
NOUN	2658	0645	0642
NOUNO	26B9	0906	0906
NUMPER	26DC	0249	0247
OPCION	2717	0019	0742 0772 0792 0306 0911
ORIGEN	2104	0658	0063
OTRASEP	2948	0427	0424
OTROCOR	2661	0192	0202
OTROIN	2630	0322	0226
PATASIN	2716	0020	0241 0272 0320
PERFOR	2719	0021	0242 0249 0299 0401
PERFORA	27EC	0176	0122



+++ PERF-1 +++

SYMBOL	VALUE	DEFN	REFERENCES
FONCOR	2591	0582	0677
FONCORT	266E	0856	0852
FONSET	2A1A	0507	0653
FONSIG	2AEE	0763	0760
FONVE	2926	0416	0414
FOSIIN	2850	0264	0261
FOSILIN	26E6	0307	0255
FOSIIN	271A	0623	0295 0406 0506 0506 0614 0916
FOSIY	271C	0623	0294 0407 0505 0507 0615 0917
FULSOX	271E	0624	0626 0925 0942
FULSOY	271F	0625	0626 0926 0927
QADRANT	2952	0525	0464
RECTA	2A02	0616	0713 0721
ROTAGE	2983	0504	0506
SACA	2A1D	0624	0622 0626
SACAFUL	2A26	0622	0626
SALE	2755	0623	0977
SALEXM	2737	0611	0626
SALIDA	26C9	0915	0796
SCAN	2C05	0972	0627 0152 0765
SCAN1	2C06	0972	0716 0716
SEGTAB	07F0	0659	0766
SEPARA	2738	0626	0266 0268 0261 0366 0404 0429 0432
SEPARA1	2731	0627	0266 0275 0287 0369 0427 0430
SIAL	2A60	0706	0698
SIGNAS	2AEE	0761	0777
SIGNO	2706	0629	0762 0802
SIGNOV	2722	0626	0776 0761 0616
SILEET	296F	0467	0460
SIULTI	26A6	0900	0650
START	2C0A	0977	0979
SUBE1	29F9	0629	0627
SUBETAL	2C06	0975	0172 0204 0296 0741
TABLA1	2727	0623	0262
TECLADO	2A7A	0714	0701 0710
TIEHFO	27E1	0152	0154
TONELK	05DE	0660	0712
TONELK	05E2	0661	0700
TRAEDAT	2975	0490	
TRES	280A	0246	0245
UNLADO	266E	0216	0207
VERTI	280B	0262	0256
VERTICA	2954	0415	0410
MENTRES	2723	0626	0632 0647
VE	2725	0624	0613 0616
YES	2A56	0623	0676

## CIRCUITERIA

### < HARDWARE DEL DISEÑO >

-IMPORTANCIA DEL HARDWARE DEL DISEÑO

-HARDWARE DE ESTE PROYECTO

Motores ejes X e Y

Bajada taladro eje Z

Hardware Propio del kit MPF-I

-MOTORES PASO A PASO

Principio de funcionamiento

Tipos de motores Paso a Paso

Control de motores Paso a Paso

Características de los motores Paso a Paso

Amortiguamiento de las oscilaciones

Información sobre los motores de Pasos usados en esta tesis

Funcionamiento del motor Paso a Paso

Fuente de alimentación y etapa de amplificación

Para los motores Paso a Paso

-OTROS CIRCUITOS

## -IMPORTANCIA DEL HARDWARE DEL DISEÑO

Tanto el hardware como el software deberán permitir al diseño alcanzar sus objetivos de funcionalidad con un máximo de eficiencia. De ahí la importancia del hardware cuya selección estará relacionada con la capacidad técnica del diseñador, además de la gran consideración en todo diseño, el factor costo.

## -HARDWARE DE ESTE DISEÑO

En nuestro diseño se puede agrupar al hardware en tres diferentes grupos y para facilitar su entendimiento se explicará cada grupo y la relación que guarda con alguna de las funciones de esta máquina.

- Motores ejes X e Y. Esta podría ser una de las partes de hardware más importantes del diseño y se relaciona con la función de posicionamiento en los ejes X e Y de la máquina de control numérico. Se consideró de tal importancia esta etapa de hardware que se ha preparado una sección especial en este capítulo llamada "Motores Paso a Paso", además dicha sección será de gran ayuda para comprender otros proyectos debido al gran uso de este tipo de motores.

- Bajada taladro eje Z. Esta etapa es la que le da nombre a nuestra máquina de control numérico (Perforadora de

Circuitos Impresos ).En esta etapa se usa un motor de corriente alterna además de dos fuentes de voltaje y una válvula de Paso.

Toda esta etapa será explicada con más detalle adelante bajo el nombre de "OTROS CIRCUITOS".

- Hardware Propio del kit MPF-I. También se considera el hardware Propio del Micro-Professor I como hardware de este diseño, dicho hardware ya fue explicado en el segundo capítulo (Micro-Professor I).Nota: Para todos aquellos lectores interesados en los Programas del Micro-Professor-I, Pueden consultar el listado en el capítulo tres de esta tesis.

## -MOTORES PASO A PASO

El avance espectacular en la tecnología de semiconductores, ha propiciado una mayor utilización de los motores Paso a Paso, debido principalmente al desarrollo de circuitos de actuación y control suficientemente rápidos y adecuados. El desarrollo de los motores Paso a Paso, se debe en gran parte a las técnicas digitales y en especial a los secuenciadores programables, ya que estos proporcionan las señales de control que son utilizadas directamente para el funcionamiento del motor Paso a Paso.

Una aplicación fundamental de este tipo de motor, es aquella en la que se precisa un sistema de posicionamiento seguro y fiable, sin tener que recurrir a otros sistemas más complejos del tipo de servomecanismos.

No es pues de extrañar que las aplicaciones aeroespaciales utilicen profusamente estos componentes: son ligeros, fiables de controlar, por lo que son muy aptos para cualquier movimiento telecontrolado, como puede ser el posicionamiento de películas, la apertura de obturadores, el desplegamiento de antenas, la apertura de paneles solares etc. Por lo anterior se puede apreciar la importancia de conocer el control y funcionamiento básico de los motores de

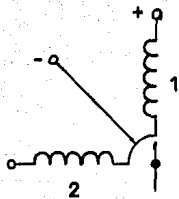
Pasos, que día a día se utilizan en nuevas aplicaciones industriales y científicas.

- Principio de funcionamiento. El Principio de funcionamiento de un motor Paso a Paso tiene una concepción bastante antigua. Es cosa sabida que la aguja imantada de una brújula indica el norte magnético. En realidad la aguja se orienta según las líneas de campo magnético terrestre. Si colocamos una brújula en las proximidades de una bobina eléctrica alimentada con corriente continua, su aguja se orientará paralelamente al eje de la bobina, según las líneas de campo magnético por ella creado.

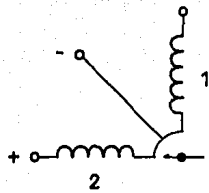
Imaginemos que situamos la brújula en la proximidad de dos bobinas de ejes perpendiculares como se muestra en la figura m1.

Excitando las bobinas según el orden adecuado, se logra el giro de la aguja imantada, es decir, para que la aguja gire el orden de excitación es el siguiente:

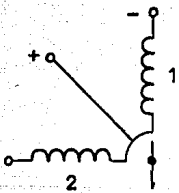
- a) excitación del bobinado 1
- b) excitación del bobinado 2
- c) nueva excitación del 1 con Polaridad invertida
- d) nueva excitación del 2 con Polaridad invertida.



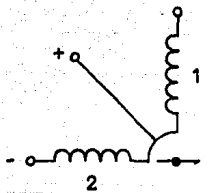
a)



b)



c)



d)

Fig.m1. Ejemplo de excitación.

La aguja efectuará vueltas completas, a saltos de 90 grados, siempre que se alimente a las bobinas 1 y 2 en los sentidos apropiados. De lo dicho, basta pues con sustituir a la aguja de la brújula por una barra imantada montada en un eje perpendicular al plano de la figura para obtener un motor paso a paso. Un motor paso a paso es aquel en el cual los arrollamientos del estator, denominados fases, son alimentados uno tras otro, provocando cada conmutación un desplazamiento angular que se denomina paso angular y que es la principal característica del motor. El sentido de rotación estará definido por la secuencia en que se han excitado los diferentes arrollamientos.

Tal como se ha definido, este motor es denominado de imán permanente, pero se puede sustituir el rotor del imán por un simple rotor de hierro dulce que será atraído por los campos sucesivos de las bobinas: se obtiene entonces un motor paso a paso de reluctancia variable. A partir de este principio hay que hacer dos observaciones:

- el ángulo de avance que era de 90 grados en el esquema de la figura m1, puede ser modificado aumentando el número de bobinas, aumentando el número de imanes o barras de hierro dulce, o combinando ambos elementos.
- El par creado por la atracción imán - bobina o núcleo -



bobina no es constante durante el recorrido angular, Pero sí es el mismo en el cambio de las diferentes Posiciones que experimenta el rotor.

Aumentando el número de bobinas, se reduce en la misma Proporción, el ángulo de cada Paso, Pero ello representa un inconveniente:

Al aumentar el número de bobinas, Para obtener el mismo Par el motor deberá ser más grande, es Por esta razón, que no se ha seguido esta técnica de aumento de bobinas Para la reducción del Paso angular.

La noción de Potencia de un motor Paso a Paso, no tiene Practicamente ningún significado, dado que el motor está constantemente en fase transitoria de arranque o de Parada. Su trabajo útil se descompone en dos elementos:

- Vencer las resistencias Pasivas Para accionar la carga
- Vencer las inercias de su Propio rotor y de la carga accionada.

Como Puede verse, el Principio de funcionamiento de los motores Paso a Paso es bastante sencillo, Pero es importante conocer sus características, como son: Paso angular, Par, velocidad, etc. Para comprender mejor el funcionamiento real de este tipo de motores.

- Tipos de motores Paso a Paso. Como se ha dicho en el punto anterior, teniendo distinto número de fases en combinación con el número de polos del rotor, es posible crear una gran variedad de motores con características de funcionamiento diferentes y así poder cubrir las necesidades requeridas para una aplicación en particular.

- Motor Paso a Paso de imán permanente. Este tipo de motor está formado por un estator de forma cilíndrica con un cierto número de bobinados alimentados en secuencia que crean un campo magnético giratorio de manera discontinua. El rotor concéntrico con el estator y situado sobre el eje contiene un imán fuertemente magnetizado que en cada instante tenderá a alinearse con el campo magnético creado por la correspondiente bobina del estator. Como puede observarse, su modo de operación no podría ser más simple, ya que está basado en las fuerzas de atracción desarrolladas entre dos imanes, uno permanente y el otro temporal.

Normalmente y a diferencia de los motores sincrónicos, el imán del rotor es bipolar y los polos norte y sur se encuentran en los extremos opuestos de uno de los diámetros del cilindro del rotor.

En el diseño de construcción del estator el aumento del número de bobinas impone dos límites a considerar:

- Cuanto mayor es el número de bobinas, más complicada es la electrónica de conmutación necesaria.

- Cuanto mayor es el número de bobinas, mayor es también la cantidad de hilo empleado, por lo que para una armazón de dimensiones fijas el par es menor.

El número de bobinados de que dispone en el estator este tipo concreto suele ser de dos o cuatro, con lo que se obtienen cuatro pasos por vuelta, y el desplazamiento angular será de 90 grados. Las frecuencias de trabajo empleadas suelen ser bajas y el par motor elevado debido a la presencia del imán.

En la figura m2 se indican las polaridades sucesivas de las bobinas en un motor de este tipo. La presencia del imán en el rotor proporciona a estos motores un par relativamente elevado y un ligero par de mantenimiento cuando el motor no está alimentado.

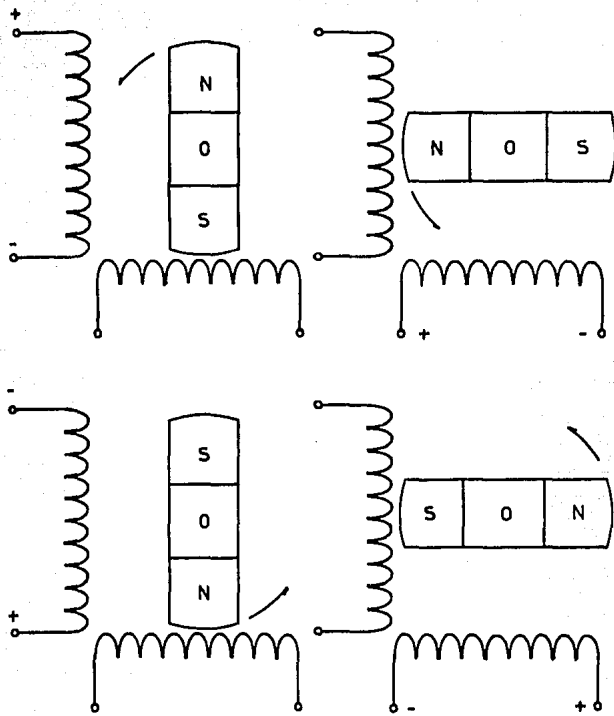


Fig.m2. Polaridades de alimentación sucesivas de las bobinas de un motor Paso a Paso de dos fases de imán Permanente.

- Motor Paso a Paso de reluctancia variable. Este tipo de motor se desarrollo con el objeto de poder conseguir unos desplazamientos angulares más reducidos que para el caso anterior, sin que por este motivo haya de aumentarse considerablemente el número de bobinados. El estator presenta la forma cilíndrica habitual conteniendo generalmente un total de tres devanados distribuidos de tal forma que existirá un ángulo de 120 grados aproximadamente entre cada dos de ellos. El rotor esta formado de un núcleo de hierro dulce de estructura cilíndrica pero con una cierta cantidad de dientes tallados longitudinalmente a lo largo de su superficie lateral.

Para analizar su modo de operación se va a suponer un motor constituido por tres devanados, B1, B2, B3, excitados secuencialmente y un rotor con cuatro dientes, D1, D2, D3 y D4 como se muestra en la figura m3.

Cuando el arrollamiento B1 recibe la alimentación atraerá el rotor hasta que el diente más cercano se alinie con el campo, por ejemplo D1; al llegar la excitación a B2, el diente D2 será el más próximo con lo cual el rotor girará 30 grados, de la misma forma, con el siguiente impulso aplicado a B3 será el diente D3 el alineado, con un segundo paso de 30 grados y al volver la alimentación a B1 será atraído D4, avanzando un ángulo igual a los anteriores.

Como fácilmente puede deducirse, en este caso a diferencia del motor de imán permanente, la polaridad o sentido de circulación de la corriente en cada devanado es indiferente, ya que al no estar imantado el rotor siempre se desplazará hasta la posición en que la reluctancia del circuito magnético del estator sea mínima.

Para obtener desplazamientos angulares de 15 grados será necesario aumentar el número de dientes del rotor hasta ocho y esa cifra se hará tanto mayor cuanto más reducido sea el avance de cada paso. La principal característica de estos motores de reluctancia variable es la elevada velocidad de accionamiento que permiten, siendo normal la cifra de 1200 pasos por segundo.

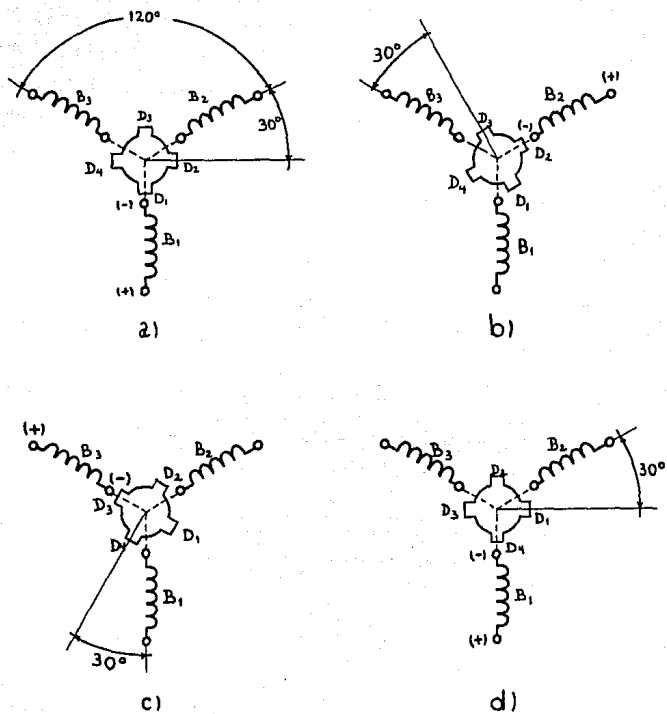


Fig.M3. Secuencia de funcionamiento de un motor de reluctancia variable de cuatro dientes:

- la corriente circula por  $B_1$ .
- alimentación de  $B_2$ .
- alimentación de  $B_3$ .
- nuevamente la corriente circula por  $B_1$ .

- Motores multiestator. Este tipo de motor Paso a Paso tiene un diseño que consiste en un agrupamiento de estatores y de rotores en un mismo eje. Si se desplazan angularmente los estatores manteniendo alineados los rotores o a la inversa, se pueden obtener Pasos muy Pequeños, teóricamente tan Pequeños como se desee. Los Pasos se obtienen al alimentar sucesivamente los estatores. Estos motores existen, pero sus características no son tan buenas a dimensiones fijas comparadas con las de los motores clásicos de reluctancia variable.

En la figura m4 se muestra un esquema de este tipo de motores Paso a Paso.

- Motores híbridos. Se trata de combinaciones de motores de imán Permanente y motores de reluctancia variable. El estator consta de un cierto número de bobinas, en general ocho, y el rotor está constituido por dientes de hierro dulce, de un número ligeramente distinto al de los dientes del estator, montados en un imán axial, de esta manera se logran Pasos angulares de 1.8 a 5 grados como en los motores de reluctancia variable, es decir, en la combinación de ambos diseños se obtienen importantes Pares de accionamiento, un gran número de Pasos Por vuelta y una frecuencia de trabajo bastante elevada.



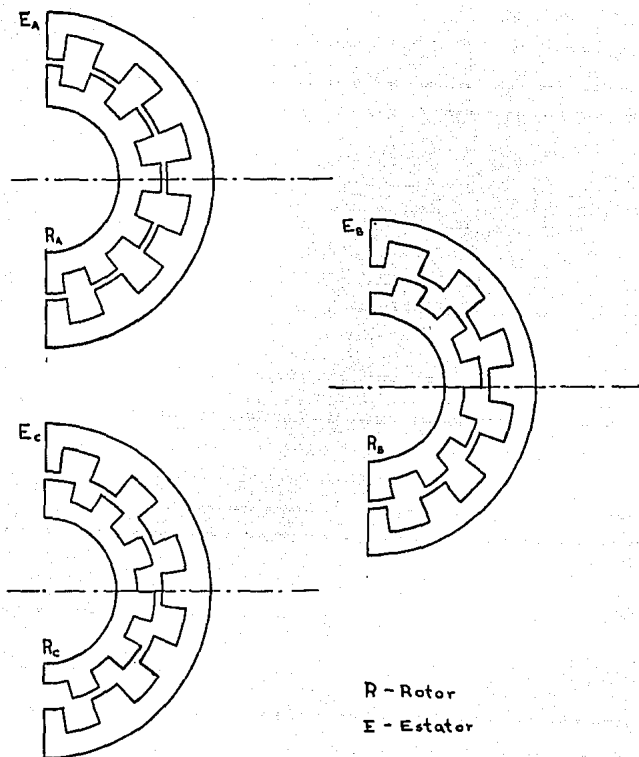


Fig. 14. Motor multiestator por agrupamiento de estatores y de rotores en el mismo eje.

A continuación se muestra una tabla comparativa de los distintos tipos de motor Paso a Paso.

COMPARACION DE LOS DISTINTOS TIPOS DE MOTORES DE PASOS

Tipo de motor	Imán Permanente	Reluctancia Variable	Híbridos con amplificador de:	
			Par	Potencia--
Pasos angulares tip. en grados	30-15	30-15	0.45	2.25
	2.5-1.8	7.5	0.18	1.5
Inercia del rotor	alta	baja	baja	baja
amortiguamiento	bueno	malo	malo	bueno
Velocidad máxima	media	alta	media	alta
Potencia de salida	alta	baja	baja	muy alta
Rendimiento	alto	bajo	bajo	medio a alto

## -CONTROL DE MOTORES PASO A PASO

El principio consiste en enviar a los arrollamientos del estator, según un orden dado, impulsos eléctricos cuya duración y frecuencia condicionarán la reacción del motor.

En general los pulsos proceden de un conmutador electrónico denominado controlador, cuyo esquema de principio se presenta en la figura M5 para un motor de tres fases.

El generador orienta a través de los transistores los pulsos hacia los circuitos de interés.

En este esquema hay que señalar dos características importantes:

- La presencia de las resistencias R, cuya finalidad es la de reducir la constante de tiempo eléctrica  $L/R$  de los arrollamientos, cosa que permite acelerar la llegada de tensión a las bobinas.
- La presencia de tres diodo-resistencias que derivan rápidamente la corriente, cuando cesa la alimentación de la bobina, ello permite ofrecer frecuencias de trabajo elevadas.

En efecto, si analizamos con más detenimiento los fenómenos transitorios en el momento de la aplicación o del corte de la corriente en los arrollamientos del estator, podemos observar en primer lugar que dichos arrollamientos

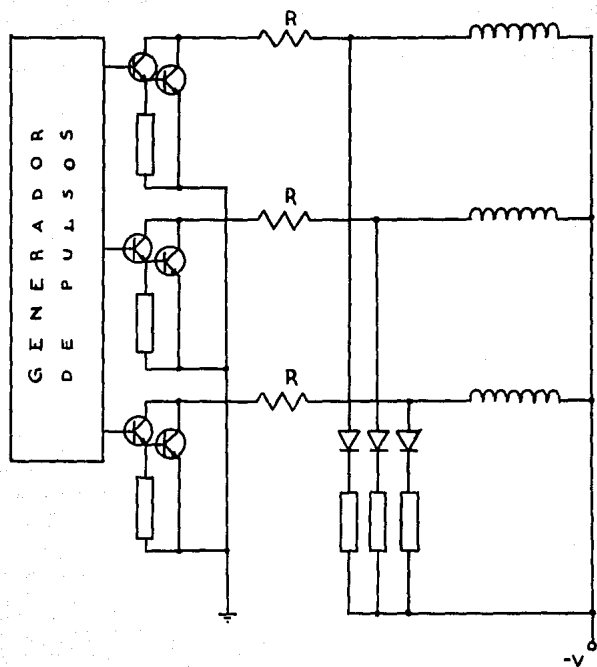


Fig.m5. Esquema de Principio de un conmutador electrónico Para motor Paso a Paso de tres fases.

son equivalentes a una inductancia  $L$  en serie con una resistencia  $r$  (resistencia del arrollamiento).  $L$  varía en función de la posición de los dientes del rotor, pero en promedio se puede admitir que:  $\theta = L/r$  es del orden de 1 a 10 ms.

Aplicando a cada bobina impulsos rectangulares de tensión como lo muestra la figura M6, la corriente  $i$  tomará la forma indicada. Alcanza el 95% de su valor final al cabo de  $3\theta$ , o sea en tres ms. El período de  $6\theta$  es pues de seis ms. lo que corresponde a una frecuencia de 166 Hz., por consiguiente no se pueden superar los 166 pasos por segundo si se quiere alcanzar el 95% de la corriente final en cada bobina. Ello limita la velocidad del motor, por lo tanto, para evitar esta dificultad fue imaginado el montaje de la resistencia  $R$  de la figura M6. La constante de tiempo es, en este caso:  $\theta = L/(R+r)$

Tanto más pequeña con respecto a  $\theta$  cuanto mayor sea  $r$ , el inconveniente de este montaje es la pérdida de potencia en  $R$ , pero que no ha impedido el desarrollo de este procedimiento.

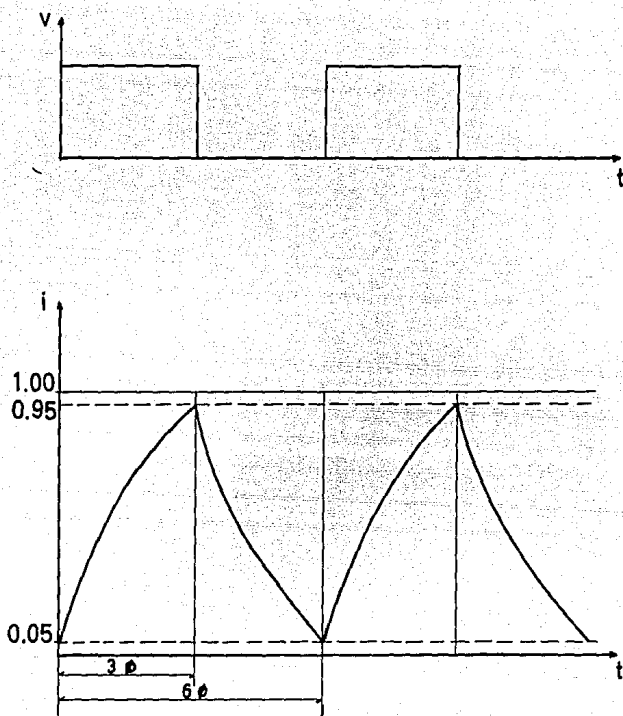


Fig.m6. Curvas  $V$  e  $i$  que caracterizan los impulsos eléctricos de un motor Paso a Paso.

Existen otros métodos como el de conmutación de tensión. Se aplica la tensión  $U$  hasta que alcanza el valor final de  $i.A$  continuación se anula  $U$  durante un tiempo pequeño, volviéndolo después a su valor hasta que dure el pulso al motor.

Otro método, consiste en aplicar una tensión elevada al inicio del impulso y una tensión más pequeña a partir del momento de que la corriente ha alcanzado el valor deseado (control bitensión). Y un cuarto método, denominado de "corriente constante" que emplea una fuente de corriente constante cuya tensión de salida toma automáticamente el valor deseado para alcanzar el objetivo.

Estos tres últimos métodos, menos empleados que el primero, mejoran las características de los motores a costa de una mayor complejidad de los circuitos electrónicos de control.

- Modos de alimentación. A continuación se darán algunos ejemplos de diferentes modos de alimentación. Para esto consideremos a la figura m7 como la representación de un motor de imanes permanentes de cuatro fases.

a) Primer modo de alimentación: Los puntos 2 y 5 se conectan a la masa, y el positivo de la alimentación se aplica sucesivamente a las bornas 1, 4, 3, y 6. El giro del motor en este caso es de 90 grados. La eficiencia de este

sistema puede ser evaluada calculando la relación de los amperios - vuelta con respecto a la potencia consumida, es decir:

$$E_2 = \frac{(V/R) N}{V^2/R} = \frac{N}{V}$$

b) Segundo modo de alimentación: El positivo se aplica sucesivamente a 1 y 4, 4 y 3, 3 y 6, 6 y 1. El rotor girará entonces 90 grados en cada paso, pero sus posiciones de equilibrio se encuentran a 45 grados con respecto a las fases.

la eficiencia es en este caso:

$$E_2 = \frac{N}{V \sqrt{2}}$$

pero al ser el campo mayor, los pares son también más elevados.



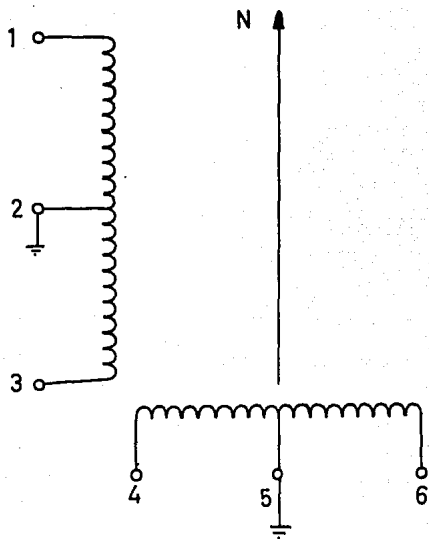


Fig.m7. Esquema de Principio de un motor Paso a Paso de cuatro fases de imán Permanente.

c) tercer modo de alimentación: los puntos medios de los bobinados no se utilizan (no se conectan a masa). Se conmutan sucesivamente el 1 al Positivo (+) y el 3 al negativo (-), a continuación 4+6-, 3+1-, 6+4-. La notación sigue siendo de 90 grados y la eficiencia es:

$$E_3 = \frac{2 N}{V}$$

(alimentación denominada "Bifásica")

El rendimiento es excelente y el calentamiento muy pequeño, pero el circuito de control es más complejo.

d) Control de ocho Posiciones Por vuelta: el Paso es entonces de 45 grados. Los puntos 2 y 5 están a Positivo y se alimenta sucesivamente en negativo: 1; 1+4; 4; 4+3; 3; 3+6; 6; 6+1; 1; etc. Para los motores de reluctancia variable sirve de base el sistema de tres fases con Pasos de 30 grados o de 15 grados. Son aplicables los mismos Principios de conmutación: alimentación simultanea de dos bobinas contiguas, o secuencias A, A+B, B, B+C, etc. (figura m4)

e) La multiplicación del número de Pasos: es teóricamente posible hasta el infinito. Basta con Prever secuencias con tensiones variables en las bobinas. Si las Posiciones de equilibrio coinciden con los dientes del rotor, la Precisión es buena, pero bastante mala en los demás casos.

## -CARACTERISTICAS DE LOS MOTORES PASO A PASO.

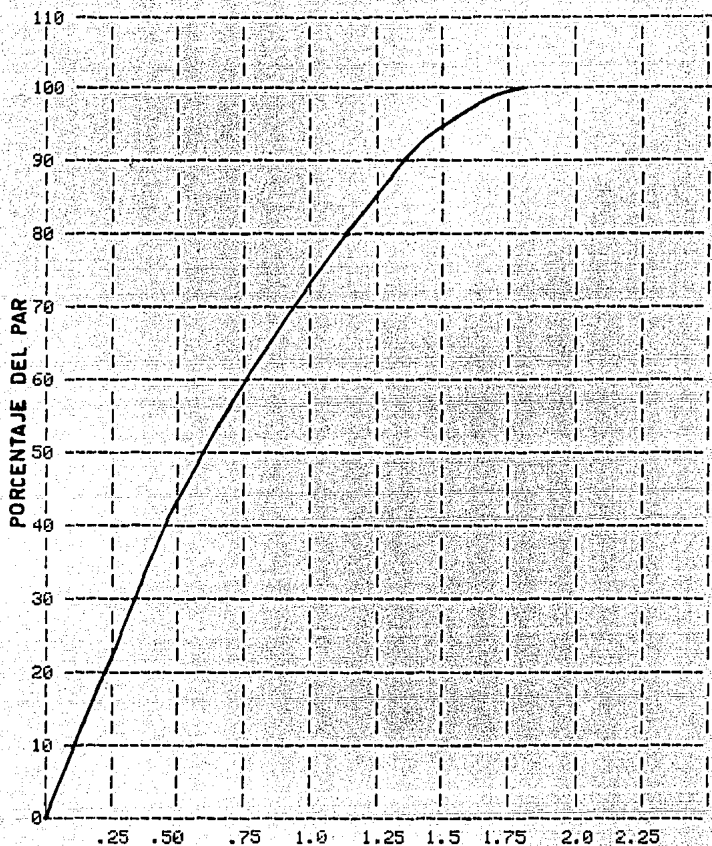
### - Características Estáticas. (motor Parado)

Cuando un motor de reluctancia variable no esta excitado, el rotor gira libremente. Si el motor es de imán permanente, los Polos del rotor son mantenidos Por la atracción de los Polos del estator.

Si se hace girar manualmente, el Par resistente crece hasta un máximo denominado "Par de distensión". a Partir de este punto, gira bruscamente hasta su nueva posición de equilibrio.

Si el motor esta excitado, los Polos del rotor son atraídos Por los Polos del estator con imán permanente a un Par mayor que en el caso anterior, y con reluctancia variable, a un Par dado denominado "Par de mantenimiento" en ambos casos. Este Par Pasa Por un máximo a una posición del rotor equidistante de los puntos de estabilidad, como lo muestra la gráfica Para un motor con Paso angular de 1.8 grados en la figura m8.

El Par de mantenimiento suele evitar el uso de un freno, Pero hay que tener en cuenta que, cuando se aplica un Par al rotor, este gira un cierto ángulo denominado error angular, que es función del Par aplicado.



DESVIACION EN GRADOS ENTRE DOS POSICIONES DEL ROTOR (1.8 GRADOS)

FIG. 88. LA CURVA MUESTRA LA DESVIACION DE LA FLECHA CONTRA EL PORCENTAJE DEL PAR PARA UN MOTOR PASO A PASO CON UN PASO ANGULAR DE 1.8 GRADOS.

- Características dinámicas. En las características dinámicas tiene gran influencia la carga accionada y el dispositivo de conmutación. La carga accionada puede ser representada por resistencias pasivas de rozamiento y una inercia. Las resistencias de rozamiento son prácticamente constantes cuando varía la velocidad en caso de rozamiento sólido, o bien crecen con la velocidad en caso de rozamiento fluido y no plantean problemas particulares mientras sean inferiores al par dinámico del motor, desempeñan incluso un papel favorable en el amortiguamiento de las oscilaciones. No obstante, la inercia es mucho más molesta. Los fabricantes dan la inercia de los rotores o la inercia del conjunto rotor-instrumento de medida.

El problema de la inercia aparece plenamente cuando se examina el movimiento del rotor sometido a un impulso. Partiendo de su posición de equilibrio, se acelera hacia el polo que lo atrae. En ausencia de un dispositivo adecuado, llegará a plena velocidad frente a este nuevo polo, con aceleración nula. Supera su posición de equilibrio con aceleración negativa, alcanza un máximo y vuelve hacia atrás, de nuevo vuelve a rebasar el polo que lo atrae, quedando pues sometido a una serie de oscilaciones amortiguadas alrededor de su posición teórica de equilibrio. La figura M9 representa este fenómeno.

Si la inercia rotor-carga es importante, el amortiguamiento será pequeño, ello limita considerablemente la frecuencia de los impulsos si se espera la detención final del rotor y prohíbe la marcha en continuo.

Si se desea girar a frecuencia rápida habrá que enviar el impulso siguiente mientras el rotor está en plena oscilación. Si el impulso siguiente llega entre los 7 y 9 ms. según la figura M9, el rotor puede ser solicitado por el polo que acaba de abandonar y volver hacia atrás-hay resonancia entre la frecuencia de los impulsos y las oscilaciones del rotor.

El estudio matemático del fenómeno es complejo por lo tanto en el punto siguiente sólo se citan los principales sistemas prácticos de amortiguamiento.

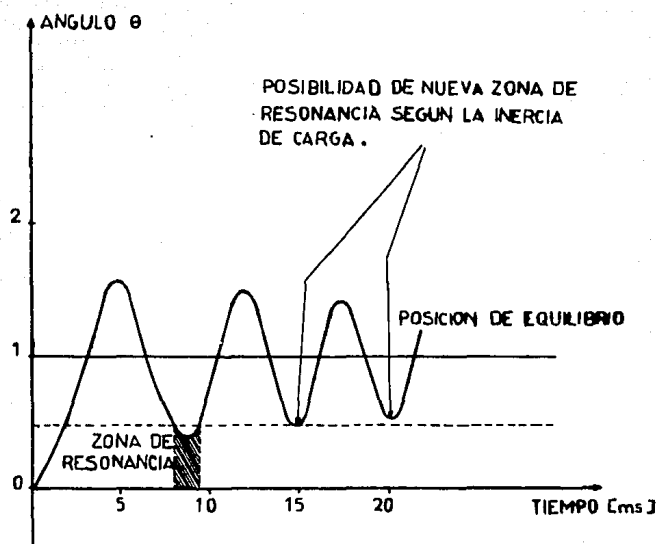


Fig. 109. Oscilaciones del rotor de un motor Paso a Paso alrededor de su posición de equilibrio. Cuando la inercia rotor-carga es elevada, el amortiguamiento es pequeño lo que limita la frecuencia de los impulsos.

#### - AMORTIGUAMIENTO DE LAS OSCILACIONES

- Rozamiento mecánico. El sistema más sencillo emplea el rozamiento mecánico; Por ejemplo, un Piñón montado sobre un dispositivo de fricción cumple esta función. Algunos motores poseen un segundo árbol con un disco prensado por un resorte en una parte fija.

Un sistema derivado del anterior emplea un carter que encierra un líquido viscoso en el que gira una turbina. Estos sistemas reducen ligeramente la frecuencia máxima del motor, además de añadir una carga parasita. Para asegurar eficazmente un amortiguamiento de este tipo habría que emplear un reductor de engranes rectos, o mejor, un reductor de rueda y de tornillo sin fin.

- Amortiguamiento eléctrico. Una solución más elegante consiste en hacer uso del amortiguamiento eléctrico: las pérdidas por histéresis en el rotor reducen las oscilaciones, al tiempo que reducen el rendimiento, estas pérdidas pueden ser aumentadas mediante una elección apropiada de los materiales, o bien aumentando la saturación. El segundo método de alimentación citado en la sección anterior (rotor defasado 45 grados con respecto a las fases), aumenta las pérdidas del rotor, creando un amortiguamiento interesante.



Otro modo de amortiguamiento eléctrico consiste en construir el estator con bobinados auxiliares alimentados por corriente continua, cuya acción se suma a los arrollamientos normales.

- Amortiguamiento electrónico. El amortiguamiento electrónico es, por supuesto, el más satisfactorio, pues perturba muy poco las características par-velocidad. Si la tensión no decrece instantáneamente, durante un momento existirá alimentación simultánea de dos bobinas, lo que reduce la velocidad del rotor en el momento de la llegada a su posición de equilibrio. Este resultado puede obtenerse con un conjunto de conmutación formado por un condensador en paralelo con un tiristor en cada fase o mediante la instalación de simples resistencias entre las fases (en este caso aumentarán las pérdidas por efecto Joule)

Otro sistema consiste en cortar, durante un pequeño intervalo de tiempo, la alimentación de la fase en el momento en que el rotor alcanza, por primera vez, su posición de equilibrio. Dado que el corto período de frenado es seguido por una realimentación, el rotor se bloquea en su posición de equilibrio (fig 110). Finalmente, cuando el motor gira en continuo se puede simplemente retrasar el último impulso. En efecto, ocurre que el impulso en la  $n$ -ésima bobina tiene lugar cuando el rotor todavía no ha alcanzado la posición  $n-1$ , entonces frenará y la llamada de la bobina  $n$  será más débil cuando el rotor llegue frente a ella.

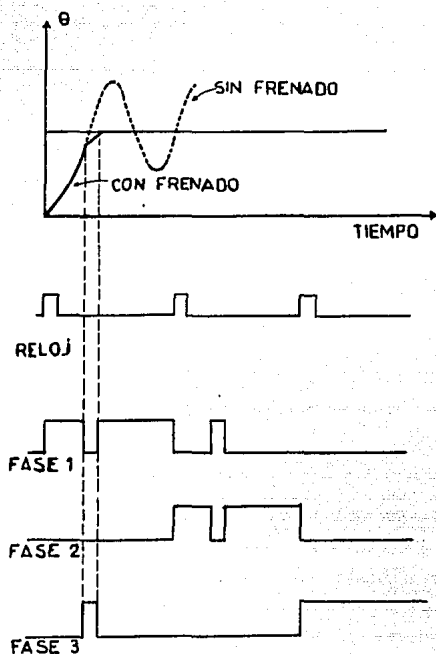


Fig.m10. Amortiguamiento electrónico de las oscilaciones de un motor Paso a Paso .En este sistema, se corta durante un pequeño tiempo la alimentación de la fase en el momento que el rotor alcanza su posición de equilibrio.La pequeña duración de frenado, es seguida por una retroalimentación, y el rotor se bloquea en su posición de equilibrio.

#### -INFORMACION SOBRE LOS MOTORES DE PASOS USADOS EN ESTA TESIS.

El motor empleado en esta tesis, es un motor Paso a Paso híbrido, es decir, su funcionamiento es de forma semejante al motor de reluctancia variable Pero con las características también de un motor Paso a Paso de imán Permanente.

El motor consta, basicamente; de un estator, un rotor, rodamientos y tapas laterales donde se colocan estos. Para entender mejor su funcionamiento, a continuación se da una breve descripción de las Partes que conforman este motor:

- El rotor, es la Parte giratoria del motor, consta de 50 dientes alineados como se muestra en la figura m11; este rotor tiene en su interior un imán Permanente, cuyas secciones del Polo norte y Polo sur se aprecian también en esta figura.

Una vista extendida del rotor se muestra en la figura m12.b, en ella se observa que los dientes del rotor se encuentran alineados, no así, los dientes en las zapatas del estator.

- El estator, consta de ocho zapatas Polares, en ellas estan colocadas las bobinas, las cuales generan los campos magnéticos necesarios Para el funcionamiento del motor; cada zapata Polar consta de cinco dientes, como se muestra en la figura m13. Los dientes en las zapatas del estator se encuentran desalineados el equivalente a un diente del rotor,

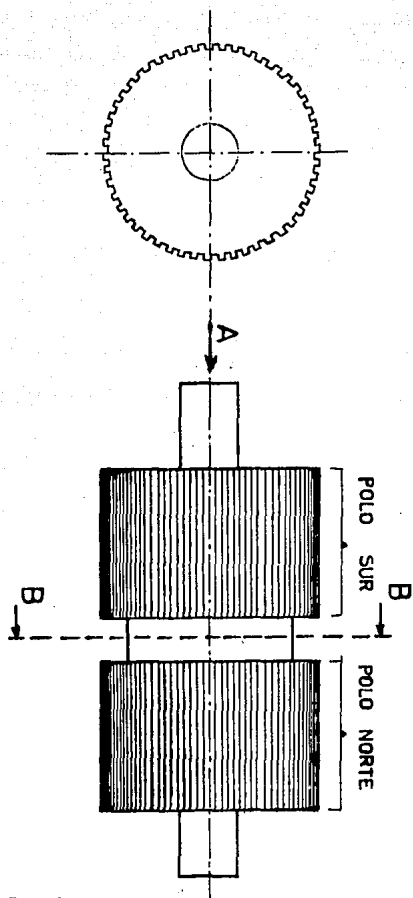


Fig.m11. Rotor del motor Paso a Paso.

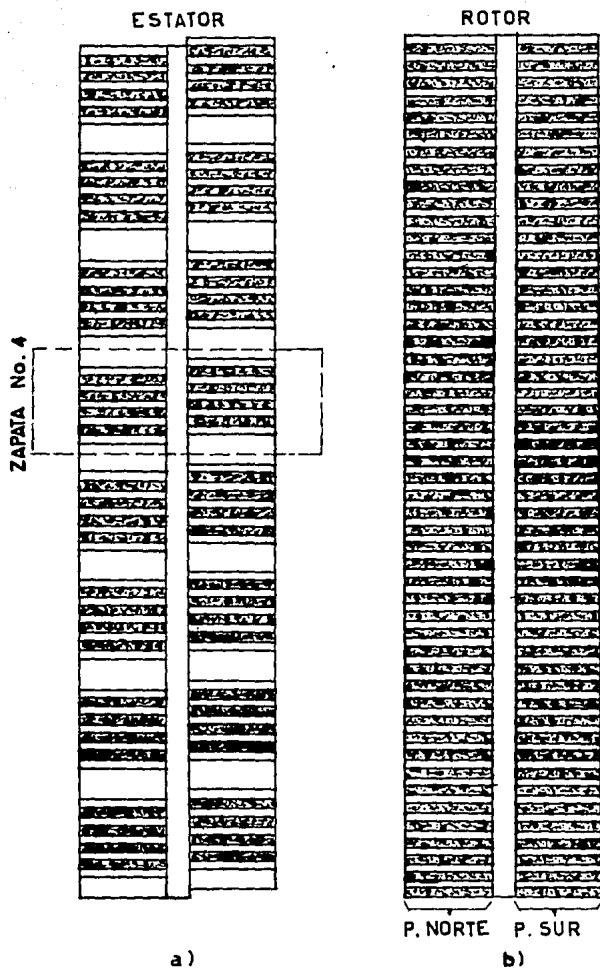


Fig. m12. Vista extendida del rotor y estator del motor Paso a Paso.

lo cual es necesario Para el funcionamiento del motor Paso a Paso, una vista extendida del estator se muestra en la figura m12.a.

- Los rodamientos van colocados en las tapas laterales del motor, en ellos se apoya el rotor Para girar libremente dentro del estator.

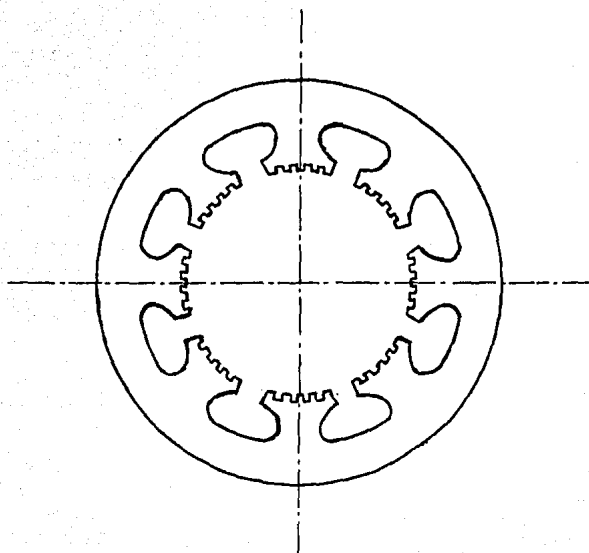


Fig.m13. Vista lateral del estator  
del motor Paso a Paso.

## -FUNCIONAMIENTO DEL MOTOR PASO A PASO.

El motor de Pasos empleado consta de ocho terminales externas, es decir, es un motor de cuatro fases, donde cada par de terminales externas forman una fase; cada fase forma cuatro bobinas en serie, las cuales se montan en las zapatas Polares del estator como se muestran en la figura m17, cada fase crea en el estator campos magnéticos que equivalen a dos polos norte y dos (Polos) sur. Para producir el movimiento giratorio del rotor, es necesario crear un campo magnético giratorio con las polaridades correspondientes como se indica en la figura m14. Los círculos representan las zapatas del estator y las letras "N" Polo norte y "S" Polo sur, creados en ellas.

El campo magnético creado en el estator, interacciona con el campo propio del rotor, provocando efectos de repulsión y atracción según las combinaciones de los polos norte y sur. El campo magnético giratorio, es creado por la adecuada aplicación de una tensión a las diferentes fases del motor, es decir, se necesita de una secuencia de pulsos ordenada para lograr que el rotor gire.

A continuación se analiza el funcionamiento del motor de pasos aplicando la secuencia de pulsos uno por uno, para mayor claridad, sólo se dibuja la fase correspondiente a la cual se aplica el pulso. Antes de iniciar con la aplicación de



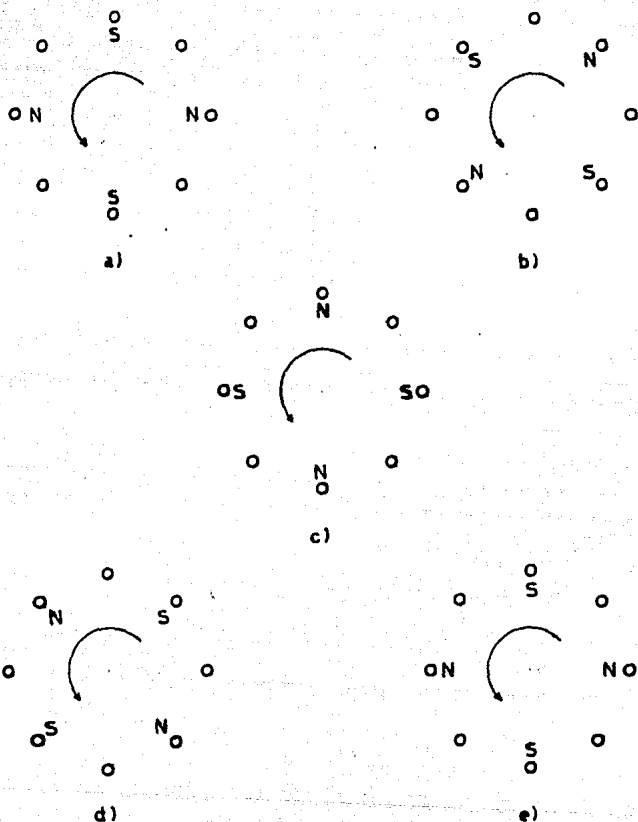


Fig.m14. Polaridades del campo magnetico creado en las zapatas Polares del estator.

los pulsos, es necesario hacer notar que las figuras de referencia para este efecto son obtenidas de la combinación del rotor y el estator (vista A y vista sección BB) como se indica parcialmente en la figura m11, donde no se encuentra el estator.

a) Aplicación del primer pulso.

En la figura m15.a, se muestra la fase B1 a la cual se le aplica el primer pulso, como puede observarse, en las zapatas donde se crea un polo norte corresponden exactamente los dientes de las zapatas con los dientes del rotor, y donde se crea un polo sur corresponden los dientes de las zapatas con el claro entre dientes del rotor, esta observación corresponde a la parte del rotor donde se encuentra el polo sur, ya que en la parte del rotor con polo norte las correspondencias son contrarias.

Como referencia se han dibujado las bobinas que crean un polo norte, en la parte más cercana a los dientes de las zapatas y las que crean un polo sur en la parte más alejada a esta parte. Se hace notar que en las zapatas donde coinciden los dientes de éstas, con los dientes del rotor existe una fuerza de atracción (polos opuestos) mientras que en las zapatas donde sus dientes coinciden con los claros de los dientes del rotor, la fuerza es de repulsión (polos iguales).

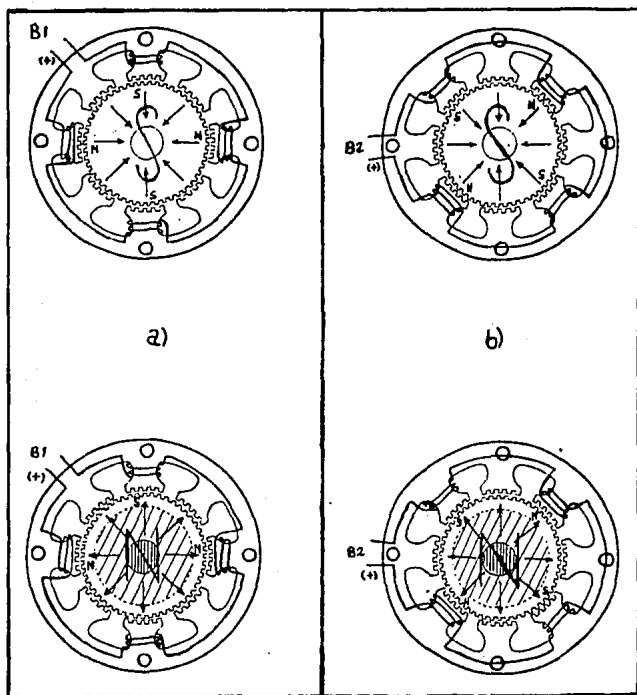


Fig.m15. Secuencias de funcionamiento del motor de imán Permanente de cuatro bobinas.  
(aplicación del Primero y segundo Pulso)

b) Aplicación del segundo Pulso.

En la figura m15.b. se muestra la fase B2 a la cual se aplica el segundo Pulso, como se puede observar el análisis es idéntico que el anterior sólo que ahora el rotor ha girado el equivalente a medio diente en sentido antihorario; cabe recordar que sólo una de las cuatro fases debe tener aplicado un Pulso en un momento dado.

c) Aplicación del tercer Pulso.

En la figura m16.a. se muestra la fase B3 a la cual se aplica el tercer Pulso, en este momento el rotor ha girado el equivalente a un diente del rotor en el mismo sentido, se hace notar que el movimiento del rotor se debe a las fuerzas de atracción y de repulsión de los polos creados, y este movimiento termina cuando se obtiene el camino con menor reluctancia entre los dientes del rotor y los dientes del estator.

d) Aplicación del cuarto Pulso.

La figura m16.b. muestra la fase B4 a la cual se le aplica el cuarto Pulso, en este momento el rotor ha girado un diente y medio, y de aquí en adelante cada secuencia de cuatro Pulsos el rotor girará el equivalente a un diente,

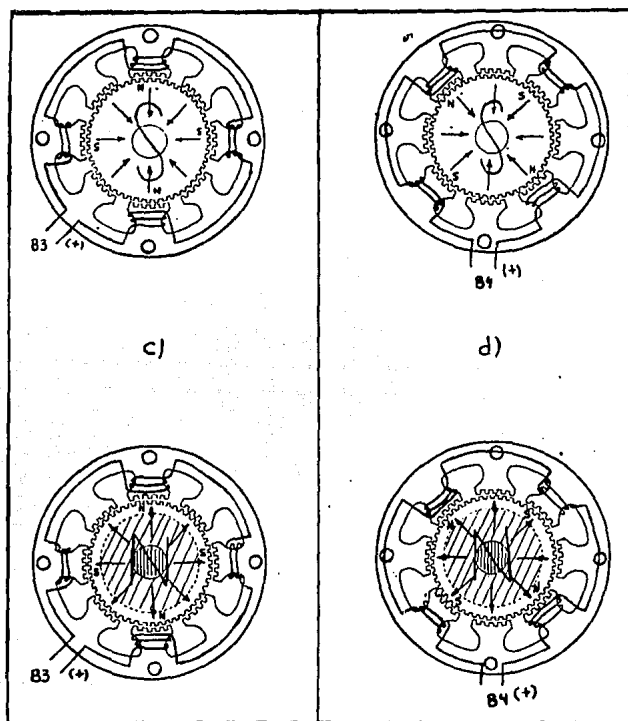


Fig.m16. Secuencias de funcionamiento del motor de imán Permanente de cuatro bobinas.  
(aplicación del tercero y cuarto Pulso)

cabe recordar que un diente va seguido por un espacio de la misma longitud.

De lo anterior se deduce que para que el rotor de una vuelta completa se debe aplicar una secuencia de 200 pulsos, ya que por cada 4 pulsos el avance de giro es de un diente, como la circunferencia del rotor consta de 50 dientes se tiene que:

$$(50 \text{ dientes/vuelta}) \times (4 \text{ pulsos/diente}) = 200 \text{ pulsos/vuelta}$$

En la figura m17, se muestran las cuatro fases juntas para el motor Paso a Paso.

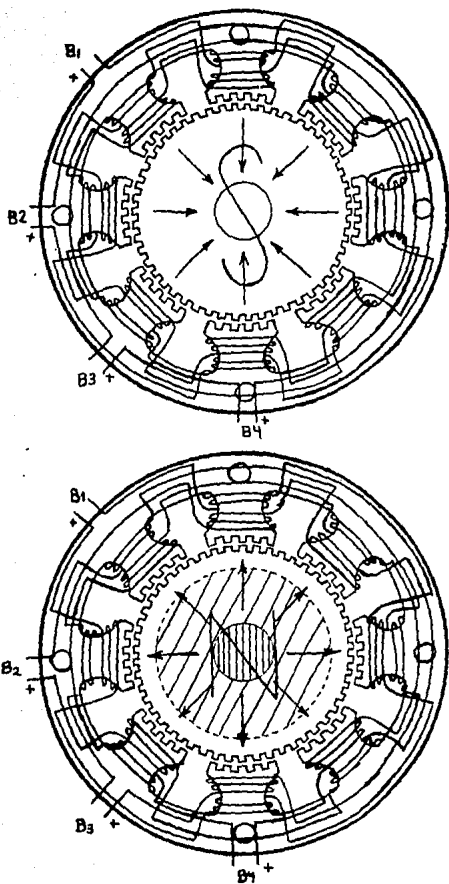


Fig. n17. Vista lateral que muestra las bobinas del motor.

- FUENTE DE ALIMENTACION Y ETAPA DE AMPLIFICACION  
PARA LOS MOTORES PASO A PASO.

Hasta este momento se ha estudiado el funcionamiento de un motor Paso a Paso; se sabe de la secuencia ordenada de Pulsos que se deben aplicar y se conoce también el efecto que estos Pulsos ocasionan. Los datos técnicos del motor Paso a Paso en su alimentación son:

Tensión = 3 volts e Intensidad de corriente = 1.7 AMPERES;  
el equipo de control que proporciona la secuencia de Pulsos, en este caso la microcomputadora MicroProfessor I, no satisface los datos técnicos para los motores ya que los Pulsos enviados son de potencia muy baja, es pues necesario, una etapa de amplificación para obtener las condiciones eléctricas necesarias en los motores.

Antes de pasar al estudio de la etapa de amplificación se hace una observación más sobre el funcionamiento del motor Paso a Paso. En la figura m17, se puede observar que en cada zapata Polar se encuentran dos bobinas de diferente fase, mientras una crea un Polo norte, la otra crea un Polo sur, esto hace posible el reforzamiento del campo magnético en las zapatas donde estos bobinados están juntos, es decir, si una fase crea un Polo norte la otra fase también lo puede hacer si la tensión que se le aplica se cambia de polaridad; este principio de alimentación de las fases con reforzamiento del



campo magnético se ha tomado en cuenta en el diseño de la etapa de amplificación para estos motores.

A continuación se mencionan los puntos que se tomaron en cuenta en el diseño de la fuente de alimentación y la etapa amplificadora.

- Motores de Pasos. Estos motores Paso a Paso tienen la característica de inductancia en sus devanados, esto ocasiona la oposición a las variaciones de la corriente que circula a través de ellos, generándose tensiones que se oponen a la tensión aplicada; la tensión aplicada depende de la microcomputadora, como una medida de protección para ésta, se emplean optoacopladores para mantener en el puerto de salida del MicroProfessor-I una carga eléctrica muy baja e independiente de los efectos mencionados.

- La fuente de alimentación. Empleada para esta etapa, debe ser capaz de proporcionar en un momento dado la corriente necesaria para cuatro fases, a la tensión indicada, ya que se está empleando el reforzamiento de los campos magnéticos alimentando dos fases; en este diseño se utilizan dos motores Paso a Paso, es por eso, que la corriente en un momento es para cuatro fases, dos por cada motor.

- Diodos destelladores. se emplean para indicar la condición de fusible abierto en la fuente de alimentación, y diodos emisores de luz (led's) para indicar la secuencia de pulsos enviados a cada fase de cada motor.

En la figura m18. se muestra la fuente de alimentación bipolar empleada en este diseño; se usa un transformador reductor (T) terminal central con una relación de transformación de 10:1; se observa la etapa rectificadora formada por el puente de diodos DR1-DR4 así como la etapa de filtrado de la señal rectificada formada por los capacitores C1 y C2; los diodos destelladores DF1, DF2 y DFL forman el sistema de alarma e indican la condición de fusible abierto para polaridad positiva (+), polaridad negativa (-) y polaridad en la línea respectivamente. Los diodos D1-D4 sirven para polarizar los transistores que hacen conducir a los diodos destelladores; R1 y R2 son resistencias para la polarización de T1 y T2. El diodo DZL, el capacitor CL, la resistencia RL y el diodo D0, sirven para asegurar las condiciones de funcionamiento del diodo destellador DFL cuando se abra el fusible FL.

La figura m19. nos muestra la electrónica de conmutación para los motores Paso a Paso. A continuación se describen las partes que se muestran en la figura:

Los optoacopladores aíslan eléctricamente el puerto A del MicroProfessor-I (OP1-OP8). Las resistencias R's limitan la corriente en el diodo emisor de luz del optoacoplador.

Los transistores (T1's y T2's) se emplean como conmutadores electrónicos, es decir, trabajan en corte o en saturación.

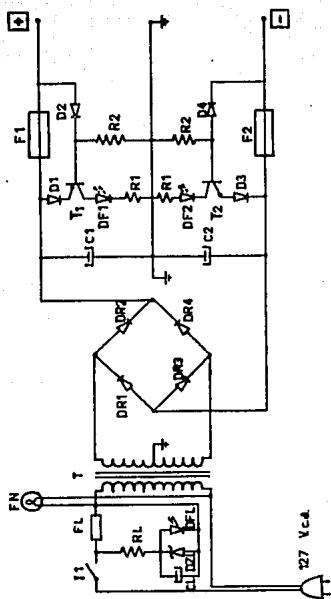


Fig.m18. Fuente de alimentación bipolar.

Las resistencias  $R_P$  y  $R_n$  sirven Para Polarizar los transistores.

Los diodos DL1-DL8 se emplean Para visualizar la secuencia de Pulsos enviados a las fases de los motores de pasos, son controlados Por el interruptor I2. Tienen cuatro resistencias limitadoras de corriente  $R_d$ .

Se muestran las cuatro fases de cada motor conectadas de forma que se Produzca el reforzamiento del campo magnético.

La secuencia de Pulsos Provenientes del Puerto A del MicroProfessor-I Provocan las siguientes Polaridades en las fases, suponiendo que el rotor gira en un sentido.

PULSO #	SECUENCIA	POLARIDAD
1	0 0 0 1	(+) B1 B3
2	0 0 1 0	(+) B2 B4
3	0 1 0 0	(-) B1 B3
4	1 0 0 0	(-) B2 B4
<hr/>		
5	0 0 0 1	(+) B1 B3 se repite

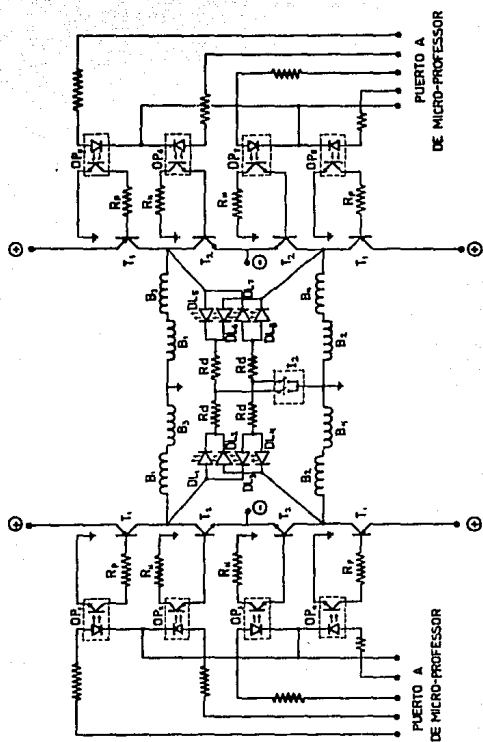


Fig.m19. Electrónica de conmutación Para los motores Paso a Paso

A continuación se muestra una lista con los valores y matriculas de los elementos de las figuras m18 y m19.

FN = Foco de neón 127 V.

I1 = Interruptor general de la fuente 10 AMP.

FL = Fusible de línea 1 AMP.

RL = Resistencia limitadora 20K, 1/2 W.

CL = Capacitor 10 mf. 16 V.

DZL = Diodo zener 6.1 V, 1/2 W.

DFL = Diodo destellador 6 V.

T = Transformador terminal central.

DR1-DR4 = Diodos rectificadores 10 AMP.

C1,C2 = Capacitores 60 000 mf. 25 V.

T1 = Transistor PNP TIP 125.

T2 = Transistor NPN TIP 120.

R1 = Resistencia 150 ohms, 1/2 W.

DF1,DF2 = Diodos destelladores 6 V.

R2 = Resistencia 4.7 k ohms, 1/2 W.

F1,F2 = Fusibles 5 AMP.

R = Resistencia 330 ohms, 1/2 W.

OP1-OP8 = Optoacopladores 4N31.

RP = Resistencia 1 K ohm, 1/2 W.

Rn = Resistencia 1 K ohm, 1/2 W.

B1-B4 = Fases de un motor.

DL1-DL8 = Diodos emisores de luz 2.5 V, 10 m AMP.

Rd = Resistencia 270 ohms.

I2 = Interruptor 1 AMP.

(<+) = Polaridad Positiva + 5.5 V +- 1 V.

(<-) = Polaridad negativa - 5.5 V +- 1 V.

M1,M2 = Motores de Pasos 1 y 2.

#### - OTROS CIRCUITOS.

En este trabajo de tesis, además de los circuitos relacionados con los motores Paso a Paso también se utilizan otros circuitos que permiten:

la interconexión de la microcomputadora con los sensores de origen (ejes X, Y y Z). La creación de los tiempos de trabajo y de retardo para las válvulas de bajada del taladro y el suministro de corriente eléctrica para el funcionamiento de los circuitos arriba mencionados.

Algunos de los circuitos que se utilizan en este trabajo de tesis son:

T1 - Este transformador junto con sus rectificadores forma dos fuentes de voltaje, una de 12 Vdc y la otra a 6 Vdc. Estas fuentes alimentan a las válvulas de bajada del taladro y a toda la circuitería de los módulos MD1 y MD2. En la figura m20, se muestran estas fuentes de voltaje.

MD1 - El módulo 1 es la conexión entre el Puerto B del PIO y los diferentes micro switches (origen X, origen Y, Taladro arriba), también en este módulo se recibe la señal de bajada del taladro proveniente del MicroProfessor-I. La figura m21, muestra al módulo 1.

MD2 - El módulo 2 contiene a los dos timers y a la lógica de control para el tiempo de trabajo de las válvulas



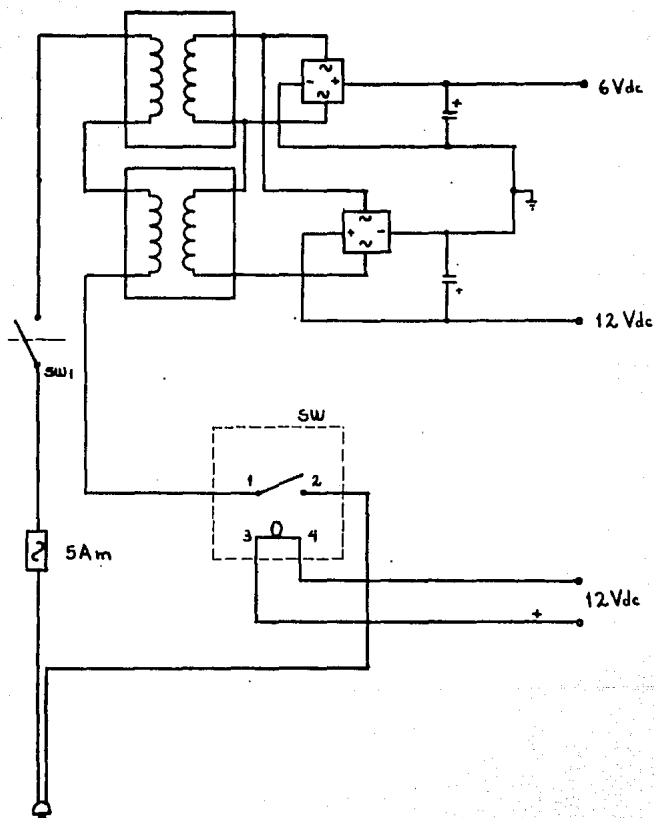


Fig.m20. Diagrama de conexión de T1 y sus rectificadores.

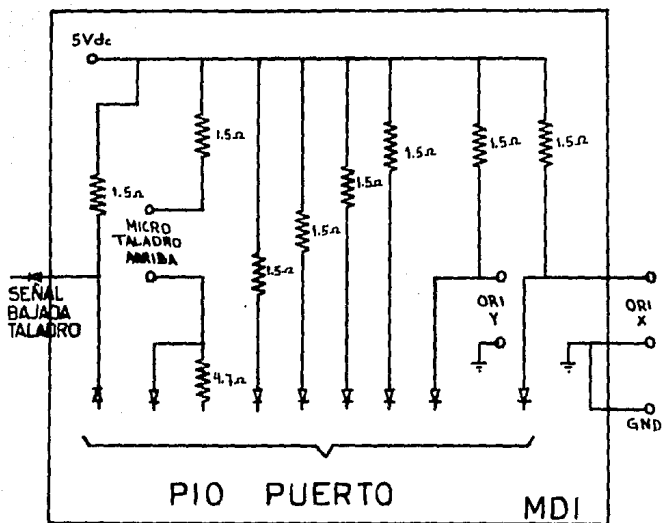


Fig.n21. Módulo de conexión entre el Puerto B y los micro switches de origen X, Y y taladro arriba.

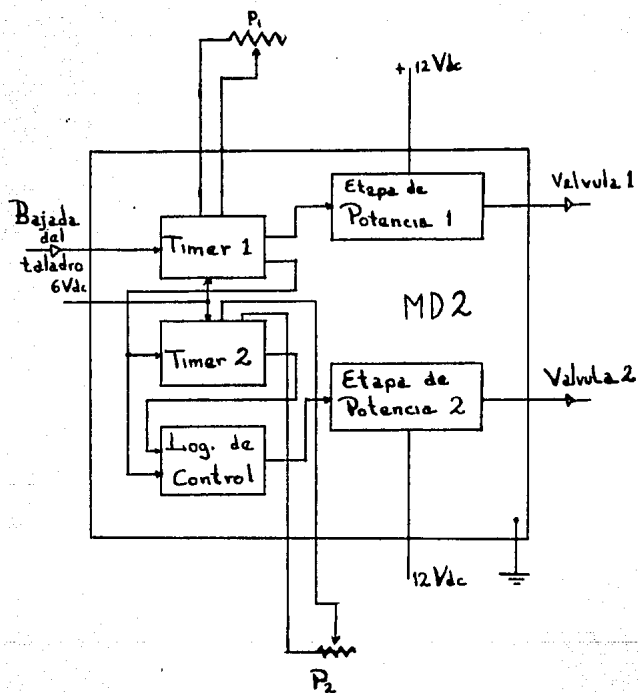


Fig.m22. Módulo 2, este módulo contiene los timers (555), las etapas de Potencia Para las válvulas V1 y V2 y la lógica de control para los tiempos de trabajo.

que Permiten la bajada del taladro. Los timers estan formados esencialmente Por circuitos integrados 555 y la lógica utiliza un arreglo de compuertas NAND. La figura m22<sup>i</sup> muestra al módulo 2.

Estos módulos y su fuente de voltaje se encuentran localizados en un Panel Por debajo de la mesa de trabajo. Este Panel Puede desmontarse Para su revisión o reparación.

Existen otros circuitos formados Por los interruptores del tablero de control, tales interruptores son:

- Interruptor Principal
- Interruptor motor taladro
- Botón de bajada taladro
- Potenciómetros ajuste tiempo bajada taladro

Las funciones de los interruptores del tablero de control se describen con mayor detalle en la sección Manual del usuario.

## DISEÑO MECANICO

- BASES PARA EL DISEÑO DE UNA MAQUINA DE CONTROL HUMERICO PARA PERFORAR CIRCUITOS IMPRESOS.
- CLASIFICACION Y SELECCION DE UNA MAQUINA PERFORADORA DE CIRCUITOS IMPRESOS.

Máquinas Perforadoras de control manual

Máquinas Perforadoras de control numérico

- ESTE DISEÑO MECANICO

- BASES PARA EL DISEÑO DE UNA MÁQUINA DE CONTROL NUMÉRICO PARA PERFORAR CIRCUITOS IMPRESOS.

Ante todo, este diseño parte de una idea de aplicación para los motores Paso a Paso y el sistema básico de computadora (kit Micro-Professor I), sin embargo, no comienza muy claro en cuanto a la forma mecánica de la máquina, esto se debe, principalmente, a la falta de conocimiento de otras máquinas de este tipo, de tal forma que para encontrar el diseño mecánico adecuado recurrimos a la clasificación y selección que el Manual de Circuitos Impresos (Printed Circuits Handbook) hace sobre las máquinas perforadoras de circuitos impresos.

- SELECCIÓN Y CLASIFICACIÓN DE UNA MÁQUINA PERFORADORA DE C.I.

"La selección de una máquina perforadora de C.I. puede ser difícil y puede afectar tanto al costo del proceso de perforado como a la calidad del mismo"

Con el párrafo anterior describe el manual de C.I. la selección de una máquina perforadora, dicha selección se refiere, obviamente, a la existencia en el mercado de este tipo de máquinas, en México no existen estas máquinas, por lo

que hay que importárlas y solamente las poseen aquí, las grandes compañías o los lugares especializados en perforación de circuitos impresos.

Las máquinas perforadoras de circuitos impresos pueden ser clasificadas en dos grandes grupos; las de tipo manual y las de tipo de control numérico. En las manuales, el operador posiciona el trabajo y hace que comience el ciclo de perforación. En las de control numérico, tanto el posicionamiento inicial como el comienzo de ciclo de trabajo, son controlados por un sistema como una lectora de tarjetas perforadas o una computadora. Tanto las máquinas manuales como las de control automático pueden ser de un sólo broquero o de broquero múltiple.

Así pues la selección de una máquina perforadora puede hacerse basándose en los siguientes factores:

- Tamaño de la placa. Cuál es el máximo tamaño de placa a perforar y cuántas vamos a perforar a la vez? Este factor es más aplicable a las máquinas de broquero múltiple que a las máquinas de broquero simple y se refiere, específicamente, a la cantidad de taladros que van a perforar la plantilla de manera simultánea. Esto afecta directamente el tamaño que debe tener la placa base de la máquina.

- Volumen.           Cuál es el volumen de circuitos impresos a Perforar? Cuando las necesidades de Producción requieren de un alto volumen es Preferible usar una máquina de control numérico, si el volumen de Perforación de circuitos impresos es bajo, una máquina manual sera suficiente. Debemos considerar en este factor si deseamos que nuestra máquina Perfore varias Placas una encima de otra, esto con el fin de aumentar el volumen de trabajo, la afectación en el diseño mecánico sera directa sobre la capacidad de sujeción de las cuchillas que detienen las Placas de C.I. a ser Perforadas.

- Exactitud.           Qué exactitud es necesaria? En las máquinas manuales, la mayor exactitud se da en función de la habilidad del operador, una mayor exactitud se logrará con las máquinas de control numérico, en este factor, el diseño tiene una exactitud en desplazamiento de 0.006 in. Por Pulso.

- Costo.           De cuánto se dispone Para comprar una máquina Perforadora? El capital necesario Para comprar una máquina manual de broquero simple Puede variar de 4,000 a 7,000 dólares y Para una máquina manual de broquero multiple, Puede acercarse a los 20,000 dolares; las máquinas de control numerico Pueden costar desde 25,000 dólares a 150,000 dólares. Este factor es, si no el más importante, sí uno de los más importantes que se deben considerar, tanto Para la adquisición como Para la fabricación. Este factor en el



diseño se Presenta como el más importante y lo discutiremos en la sección "ESTE DISEÑO MECANICO".

- Máquinas Perforadoras de Control Manual. En este tipo de máquinas el trabajo es Posicionado Para su Perforación Por alguno de los dos métodos básicos ; el método Óptico o el método de Pantógrafo.

El método Óptico requiere que la localización del punto a Perforar se haga ya sea Por sobretrazo (overlay) o Por reproducción del circuito, definiendo el relieve. El fotoresist es a menudo usado cuando la imagen desarrollada es suficiente Para localizar la Posición de la Perforación. Los circuitos impresos Pueden ser apilados y Perforados Por copia de un modelo maestro.

En la técnica de Pantógrafo la Posición de las Perforaciones se logra a través de la duplicación de un modelo de Pantógrafo, estos modelos generalmente son de aluminio. Los circuitos impresos modelo son recorridos Por un sensor de bastón controlado Por el operador, este sensor se desPlaza sobre el modelo y cuando detecta una Perforación, la máquina recibe una instrucción Para Perforar la Placa de circuito impreso, esta Perforación se hace generalmente de abajo hacia arriba.

Para todos los casos de Posicionamiento Por operador es aconsejable contar con un mapa-ruta de trabajo Para guiar al operador en su recorrido a través del circuito, sin esta guía es muy fácil que el operador omita Perforaciones en el recorrido.

#### - MAQUINAS PERFORADORAS DE CONTROL NUMERICO.

La complejidad y alto costo de las máquinas de control numérico requieren que demos una mirada cercana a los componentes del sistema de Perforación. Dos métodos son usados Para trasladar y controlar la información de una máquina lectora de cinta Perforada o computadora a la máquina Posicionadora (el control numérico y el control numérico Por computadora).

El control numérico es definido como la lógica de control de máquina obtenida de Programas fijos que son diseñados y alambrados en electrónica (el término alambrado en electrónica significa que el Programa se llevará a cabo Por un arreglo especial de componentes electrónicos), si ese Programa se deseara cambiar, se tendría que modificar la circuitería. Este es el mas viejo de los sistemas y cuando es diseñado y ajustado correctamente, este sistema es muy seguro y confiable y las fallas son generalmente imputables a los componentes.

En el control numérico por computadora (CNC) la lógica del control de máquina es obtenida del software (Programas de computación) y la computadora es programada para controlar la máquina. El software está, generalmente, en forma de cinta perforada o en disco magnético, este software entra a la computadora a través de los lectores propios del sistema.

Las ventajas de las máquinas de control numérico por computadora CNC sobre las máquinas de control numérico CN son: flexibilidad, versatilidad y velocidad, estas ventajas son más importantes para los fabricantes que para los usuarios; para los fabricantes estas ventajas les permiten hacer cambios en el funcionamiento de la máquina con un mínimo de costo y esfuerzo. Las ventajas para el usuario son un poco menos definidas, consisten, generalmente, en la forma de almacenaje de modelos, capacidad de repetición, funcionamiento paso a paso y muchos otros procesos, posibles, gracias a la capacidad de memoria de la computadora.

#### - ESTE DISEÑO MECANICO.

Como mencionamos antes, nuestro trabajo de tesis surge de la idea de aplicar tanto los motores paso a paso, como el sistema básico de computadora al diseño de una máquina perforadora de circuitos impresos. Este diseño es afectado, principalmente, por el factor económico, esa es la razón por

la cual la adquisición de los motores Paso a Paso y otras Piezas mecánicas fue hecha en un deshuesadero de computadoras. De otra manera, tal vez no hubiera sido posible el desarrollo físico de este diseño.

El criterio que se utiliza en el diseño de este trabajo de tesis, es el siguiente:

Según las funciones de la máquina, se requiere de ciertos elementos o partes mecánicas que las realicen y así el conjunto de todas estas partes cumpla con la totalidad de funciones que debe realizar la parte mecánica. Un buen funcionamiento de la parte mecánica en combinación con un buen funcionamiento de la parte electrónica, dará como resultado el correcto funcionamiento de esta máquina perforadora.

Para facilitar la descripción del diseño mecánico, se listarán las funciones mecánicas o características que debe de tener una máquina de control numérico para la perforación de circuitos impresos:

- Deberá tener un sistema de mordazas o sistema de sujeción que permita fijar las placas de circuitos impresos a perforar.

- Deberá tener un sistema de Posicionamiento tanto Para el eje X como Para el eje Y, este Posicionamiento se controlará a través de los motores de Pasos.

- Deberá tener un sistema de taladro, que servirá Para Perforar las Placas, después de haber sido Posicionadas. La bajada del taladro Puede hacerse utilizando medios eléctricos, neumáticos o hidráulicos.

Como se Puede ver muchos diseños Podrían cumplir con las funciones antes mencionadas, sin embargo no todos cumplen con los objetivos que se han trazado desde el inicio de este trabajo de tesis; estos objetivos son:

1) Probar la funcionalidad de una máquina de este tipo, Para hacerlo será necesario un modelo de Prueba.

2) Procurar que la máquina sea diseñada lo más simple posible, y dejar las modificaciones y sofisticaciones como probables mejoras al diseño de este trabajo de tesis.

3) Procurar el menor costo en este diseño.

Considerando los objetivos anteriores y Procurando satisfacer las necesidades de funcionalidad, se describe a continuación el Porque de cada una de las diferentes secciones que forman el diseño mecánico.

- Sistema de sujeción.

Este sistema está Pensado Para ser intercambiado Por uno de la misma forma, Pero con aristas más largas, siempre que así lo requiera el tamaño de la Placa a Perforar, en la Práctica será comprobada su funcionalidad al usar levas Para sujetar la Placa, esto fue hecho con la idea de facilitar el cambio de Placa en un trabajo continuo. La figura D1 muestra la Parte inferior de la mordaza, la figura D2 muestra la Parte superior de la mordaza.

-Sistema de Posicionamiento.

El sistema de Posicionamiento de la Placa de circuito impreso, consiste en un Par de ejes en los cuales corre un carro que lleva las mordazas, en realidad el eje de menor tamaño (eje X) se desPlaza sobre el de mayor tamaño (eje Y) y el carro de desPlaza sobre el eje X.

Existen tanto en el eje X, como en el eje Y algunas características que a continuación mencionamos.

El eje X es más corto que el eje Y debido a que todo el eje X se desPlaza sobre el eje Y, entonces X requiere ser lo más ligero posible. El eje Y es más largo que el eje X Para poder aprovechar toda la longitud del tornillo Y que mueve el motor de Pasos, esto nos Permitirá Perforar Placas mas grandes.



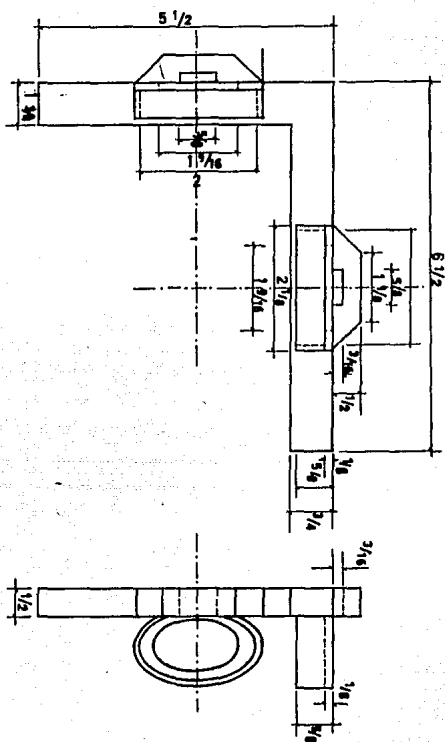


Fig.D2. Sistema de Sujeción.

Parte superior de la mordaza.



El desplazamiento del eje X sobre el eje Y se da sobre un Par de barras Paralelas llamadas barras guías, además el eje X usa bujes de bronce Para Poder desplazarse sobre las barras guías, tanto el eje X como el eje Y usan cojinetes de lubricación Para facilitar su desplazamiento. También en el eje X el sistema de sujeción se desPlaza sobre un Par de barras Paralelas (barras guías), usando bujes y cojinetes de lubricación. Las figuras D3 y D4 nos muestran los ejes X e Y respectivamente.

- Sistema bajada de taladro.

Este sistema esta diseñado Para ser completamente independiente, tanto del sistema de sujeción como del de Posicionamiento y esta formado en nuestro diseño Por una Placa cuadrada de 5"x5" en donde está sujeto un motor de C.A. que gira aProx. a 1600 rpm. La flecha del motor se Prolonga a un cilindro hueco dentro del cual se Puede desPlazar una flecha que hace girar al broquero. La figura D5 muestra este sistema, la bajada del taladro se hace Por medio de un cilindro activado Por aire, el funcionamiento adecuado de este sistema dependerá más de Pruebas que de diseños, no debemos olvidar que al llevar algún diseño a la Práctica generalmente se Presentan Problemas no contenPlados en el diseño, más aún si se está trabajando con la combinación de varios sistemas como el electrónico, eléctrico, mecánico y neumático.

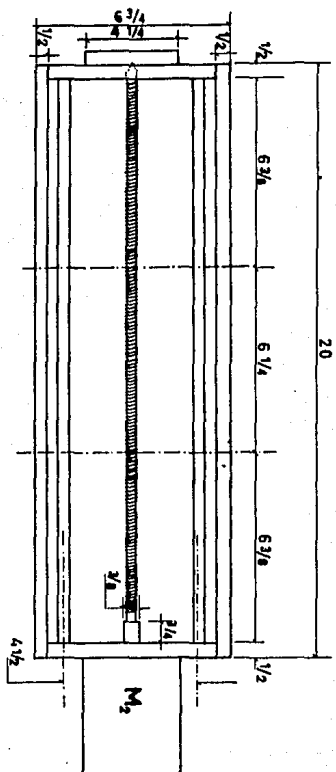


Fig.D3. Sistema de Posicionamiento.

Eje X.

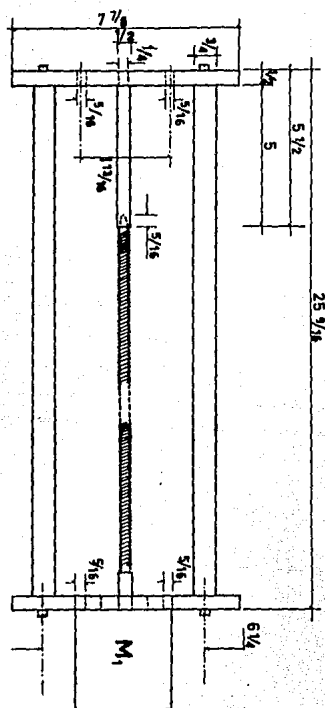


FIG. D4. Sistema de Posicionamiento.

Eje Y.

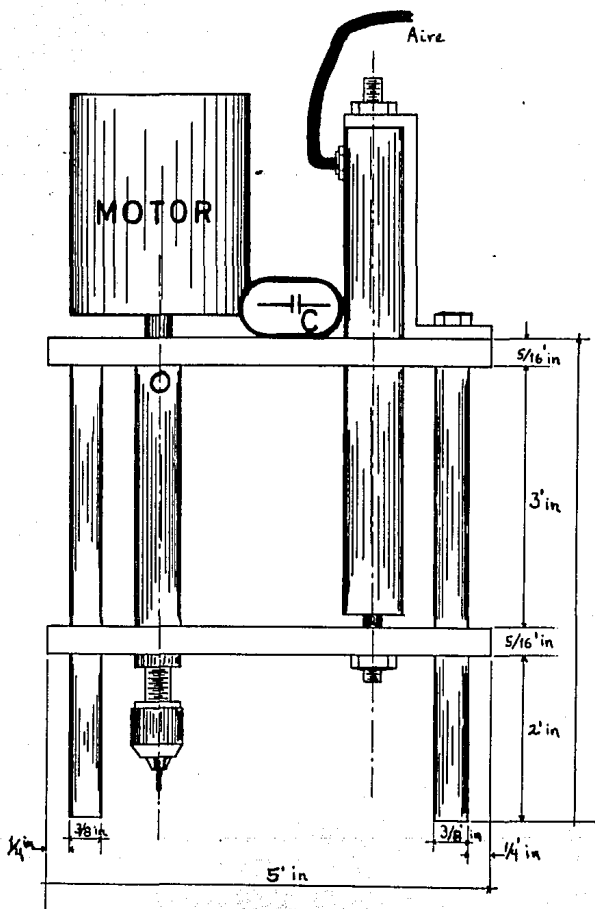


FIG. D5. Sistema de Perforación.

Eje Z.

La Parte correspondiente al diseño de la bajada de taladro, debe tener como máximos objetivos el de evitar el Juego en el sistema de Perforación al bajar el taladro, Pues esto sería contrario a la exactitud del Posicionamiento en los ejes X e Y, y también se deberá buscar una bajada suave del taladro, con el fin de no maltratar ni la Placa, ni la broca.

Durante las Pruebas de esta máquina se observó la bajada suave del taladro (Gracias al diseño del sistema neumático), sin embargo el Juego en el sistema de Perforación no permite un adecuado desarrollo de las Perforaciones, aquí se sugiere, a futuro, cambiar el actual sistema de Perforación por un taladro de banco Profesional; al que se le deberá adaptar el sistema de bajada de taladro, esto redundará en la calidad de las Perforaciones de manera muy favorable.

## \* FUNCIONES DE LA MAQUINA

### - DESCRIPCION GENERAL

Tablero de control

Sistema neumático

Taladro

### - MANUAL DEL USUARIO

Cargado de datos

Perforación

Cargado en cinta

Cargado de cinta

Opción de corte

Opción de circuito integrado

## - DESCRIPCION GENERAL

En este capítulo se mencionan las funciones de la máquina y su relación con el operario. Para facilitar el manejo de la máquina se describirán previamente las diferentes partes que la componen, hay que recordar que el operario deberá familiarizarse con su uso hasta lograr el mejor desempeño de la máquina.

- Tablero de control. En la figura, fl. se muestra el tablero de control de la máquina. A través de este se puede probar y ajustar la bajada del taladro, encender el motor del mismo y ejecutar la función principal, encender todo el sistema. El tablero de control se ve apoyado con el micro kit que está colocado encima de él.

A continuación se listan los elementos que forman el tablero de control y sus diferentes funciones.

SW1 - Switch General, habilita a los sistemas de bajada del taladro y a todos los sistemas alternos.

SW2 - Switch on/off motor taladro, SW2 debe ser habilitado Por SW1.

SW3 - Switch on/off luz de trabajo. (Opcional)

P1, P2 - Potenciómetros de los timers que controlan las válvulas de bajada del taladro, estos potenciómetros deben ser ajustados previamente.

PB1 - Push boton, este botón nos sirve Para ver la bajada de taladro y así Poder ajustar los timers (P1 y P2) antes de comenzar a Perforar la Placa de CI.

CI - Contactor múltiple, este contactor nos sirve Para conectar la fuente de Poder, el Micro-Professor I y temporalmente alguna grabadora de cinta. Este contactor es habilitado Por el switch General SW1 (esta localizado en el costado derecho de la mesa de trabajo)

Existen otros elementos, ademas del tablero de control, sobre los cuales el usuario no debe tener injerencia, sin embargo es bueno saber que están ahí y lo que hacen. Estos elementos están contenidos en un Panel bajo la mesa de trabajo, dicho Panel es desmontable Para su revisión.

T1 - Este transformador y sus rectificadores Proveen 12 Vcd y 6 Vcd que son regulados a 5 Vcd, ambos voltajes necesarios Para el funcionamiento del sistema

MD1 - Módulo de conexión entre el Puerto B del kit y los diferentes micros switches de la máquina (origen X, origen Y, taladro arriba). En este módulo también se tiene la señal Proveniente del kit Para indicar la bajada del taladro.



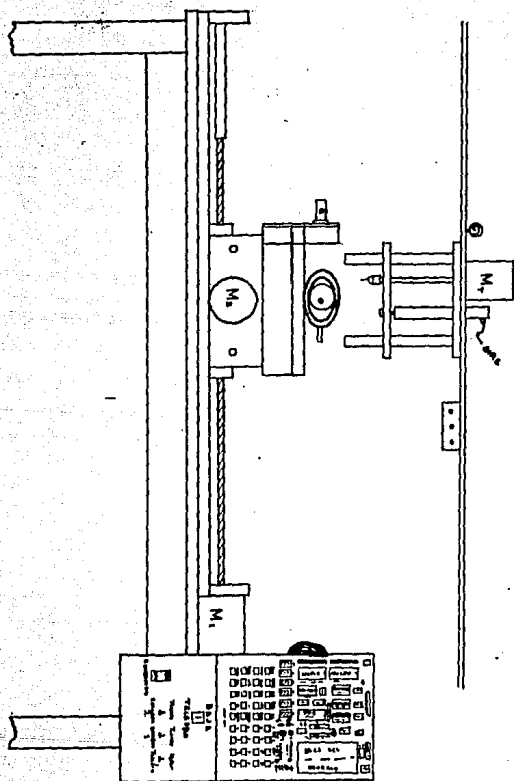


Fig.F1. Tablero de Control.

MD2 - Este módulo contiene los timers de bajada del taladro. El timer 1 corresponde a la válvula Principal y el timer 2 corresponde al tiempo de retardo de la válvula de fuga. Además también en este módulo se encuentra la lógica para este tiempo de retardo (arreglo de compuertas Nand).

- Sistema neumático. En este diseño de tesis se utiliza un sistema neumático que lleva a cabo la bajada del taladro. La figura F2. muestra el diagrama de este sistema neumático, como se puede apreciar, el sistema es de lazo abierto y su función es permitir dos diferentes presiones de trabajo a la salida, una para hacer la bajada del taladro lo más suave posible y otra que dé la presión necesaria para la perforación.

El sistema deberá ser conectado a una compresora con aproximadamente 30 Psi. R1 es el regulador de presión propio de la compresora. Con la válvula principal V1 se controla el paso del aire cuando baja el cilindro y también permite la descarga del mismo, cuando este regresa a su posición original. La válvula de fuga V2 está conectada a una derivación T antes de la válvula reguladora de aguja V3. La función de V2 es la de permitir una fuga de aire en el momento exacto de la bajada del taladro, esta fuga reduce la presión haciendo a la bajada más suave, este tiempo de fuga se controla con la lógica de retardo y el timer 2. La gráfica de la figura F3.

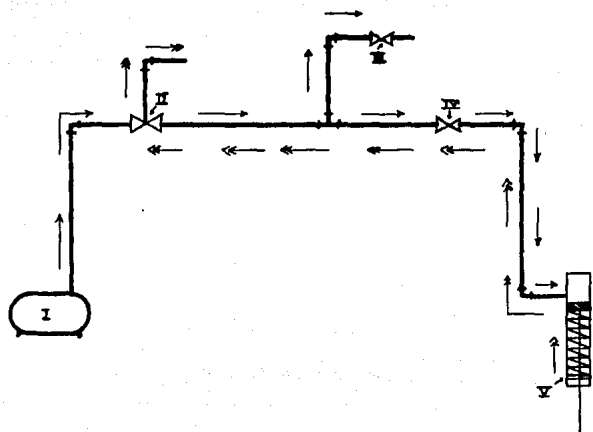


Fig.F2. Diagrama De Circulación De Aire Durante La Bajada Del Taladro.

- I.- Tanque (30 Psi).
- II.- Válvula Principal V1.
- III.- Válvula de fuga V2.
- IV.- Válvula reguladora de aguja V3.
- V.- Cilindro.

- Circulación de aire durante la bajada del cilindro.
- Circulación de aire cuando el cilindro regresa a su posición original.

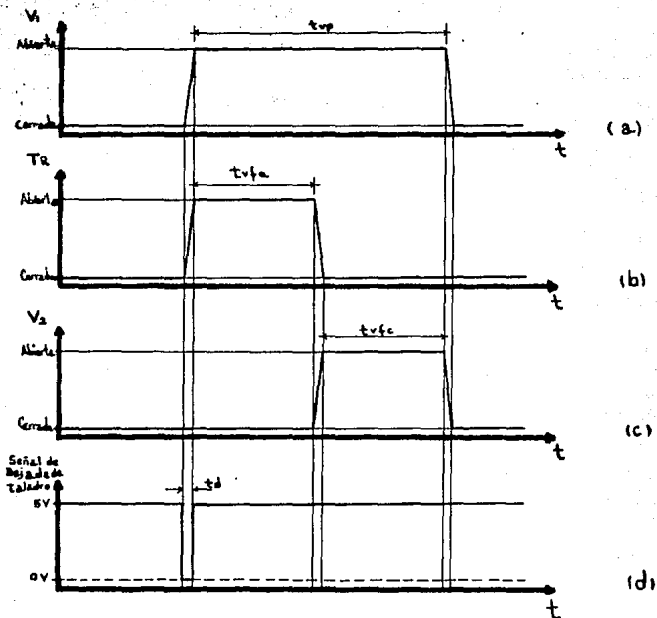


FIG.F3. Relación de los tiempos de operación de las válvulas V1 y V2 de la señal de disparo.

- a) Muestra el tiempo de trabajo de la válvula Principal V1 (tvp).
  - b) Muestra el tiempo de válvula de fuga abierta (tvfa).
  - c) Muestra el tiempo de válvula de fuga cerrada (tvfc).
  - d) Muestra el tiempo de disparo, este disparo lo hace el kit y llega a las válvulas pasando por el Puerto B del PIO.
- Para un correcto funcionamiento la única relación que deberán guardar estos tiempos es:  $(tvp) > (tvfa + tvfc)$ .

muestra la relación entre los tiempos de operación de las válvulas V1 y V2, no debemos olvidar que V2 es permanentemente abierta.

El ajuste de los tiempos de operación de las válvulas V1 y V2 se hace con los Potenciómetros P1 y P2 y con ayuda del botón PB1, el cual al ser oprimido activa la bajada del taladro.

- Taladro. Esta es una parte muy importante de conocer y dominar para el usuario, pues el adecuado funcionamiento del sistema de taladro dependerá, en un futuro, de su adecuado ajuste y de la preparación diaria del mismo. Sin embargo no hay que olvidar que este diseño es un modelo experimental al cual se le pueden hacer muchas mejoras.

Algunos de los factores a considerar en este sistema además de los ajustes de los tiempos son:

- tamaño y longitud de la broca a utilizar
- distancia entre la punta de la broca y la placa a perforar
- que en los puntos extremos a perforar la placa cuente con un apoyo firme

Se deben considerar todos estos puntos antes de comenzar a usar la máquina, sin embargo, aparecerán con el uso, muchos detalles a perfeccionar y que deberán ser considerados para un diseño futuro.

- MANUAL DEL USUARIO.

En esta sección se menciona cómo deben ser llevadas a cabo las diferentes funciones de la máquina perforadora. No hay que olvidar que estas funciones implican el uso simultáneo de más de un sistema, por tal razón se recomienda antes de utilizar la máquina comprender bien su principio de funcionamiento.

- Cargado de datos. Esta función permite cargar los datos de los puntos que se desean perforar, esto se logra posicionando la placa en el punto deseado (moviendo los ejes X e Y) y escogiendo la función adecuada de perforación (dato, integrado, línea, corte). Para mayor facilidad consultar la guía rápida de referencia en la misma máquina.

- Colocar la placa a perforar en la mordaza
- Ajustar la placa según las líneas de referencia
- Encender la fuente de voltaje para los motores
- En el kit, llamar al programa DATS
- El sistema se moverá a su origen
- El kit espera se pise la tecla inicio de tabla para programa DATS tecla "A"
- Presionando las teclas adecuadas se posicionan los puntos que se desean perforar (0-1) para X, (2-3) para Y

- Cuando se Posiciona el Punto, se elige entre:
    - C Para corte
    - D Para dato
    - E Para integrado
    - F Para línea
  - Todas las Posibilidades de selección dentro de una opción son indicadas Por el mismo kit
  - Al terminar de cargar todos los Puntos que se desean Perforar, se debe oprimir la tecla de fin de tabla, tecla "B"
  - El kit vuelve al menú Principal y la máquina queda lista Para cualquiera de las dos funciones siguientes: Cargado en cinta o Perforación.
- Perforación. A esta función se Puede llegar unicamente habiendo Pasado Por cualquiera de las siguientes funciones: Cargado de datos (descrita arriba) o Por cargado de cinta. El Procedimiento de la función Perforación es el siguiente:
- Estar seguros de que existe una tabla de datos a Perforar
  - Ajustar los tiempos de bajada del taladro y encender el motor del mismo
  - Colocar correctamente la Placa a Perforar en la mordaza
  - En el kit, llamar al Programa SALE
  - Vigilar de cerca el desarrollo de la Perforación
  - Al terminar el kit vuelve al Programa Principal

- Cargado en cinta. El cargado en cinta puede hacerse únicamente después de haber realizado la función de cargado de datos. El Procedimiento de esta función es el siguiente:

- Haber realizado el cargado de datos
- Que el kit esté en el Programa Principal
- Seleccionar la opción SACR tecla "1"
- Escribir el nombre del Programa.
- Conectar la salida MIC (micrófono) del kit a la entrada MIC (micrófono) de una grabadora
- Tomar nota del contador de cinta, para poder después facilitar la localización del Programa.
- Presionar las teclas PLAY & RECORD en la grabadora
- Presionar la tecla "GO" en el kit
- Al finalizar el sonido, detener la cinta.
- El kit asigna directamente las direcciones iniciales y finales del Programa, y al finalizar regresa al Programa Principal

- Cargado de cinta. Esta función se lleva a cabo solo después de conocer el nombre del Programa que se desea Perforar. El Procedimiento es el siguiente:

- Posicionar la cinta utilizando el contador de la misma.
- El kit deberá estar en el Programa Principal
- Seleccionar la opción METE presionando la tecla "0"
- Escribir el nombre del Programa que se desea cargar



- Conectar la entrada EAR (earPhone) del kit con la salida EAR (earPhone) de la grabadora
- Presionar la tecla "GO" del kit
- Presionar la tecla PLAY de la grabadora
- Al terminar Presionar la tecla STOP en la grabadora
- Al terminar el kit regresa al Programa Principal

Una vez cargado el Programa solamente se Puede continuar con la función Perforación.

La figura F4, nos muestra un diagrama a bloques de las secuencias de funciones Permitidas Para esta máquina de control numérico, el operario deberá respetar estas secuencias ya que éstas no están dadas directamente Por software.

- Opción de corte. Esta Puede ser considerada en la función de cargado de datos DATS. El Problema aquí es mecánico y relativo al tipo de broca o fresa a utilizar, Pero más aún, relativo al tipo de taladro que se utiliza (Para esta función se requiere de un taladro resistente). Sin embargo el software Para esta opción ha sido Probado y funciona Perfectamente.

- Opción de circuitos integrados. Esta Puede ser considerada en la función de cargado de datos DATS, y Presenta un gran ahorro de memoria cuando se desea cargar los Puntos de Perforación de algun tipo de circuito

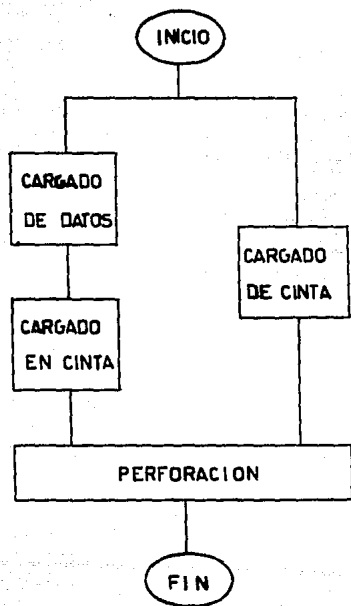


Fig.F4. Diagrama a bloques de las secuencias de funciones permitidas para esta máquina perforadora.

integrado. Sólo se deberá dar el punto inicial, el número de patas del C.I. y especificar si se desea horizontal o vertical. Esto obliga a que en el diseño del layout los circuitos integrados estén alineados ya sea horizontal o verticalmente, esta alineación se hace con respecto a las líneas guías, las cuales se incluirán en todos los circuitos impresos y servirán como guías para colocar las placas en la mordaza de la máquina perforadora.

El uso de las líneas guías deberá existir hasta que se diseñe e implemente un sistema más práctico de sujeción de la placa de C.I. en la mordaza de la máquina.

Para mayor facilidad en el manejo de esta máquina, se sugiere consultar el manual de referencia rápida, este manual está localizado en la máquina a un lado del tablero de control.

## CONCLUSIONES

Durante el desarrollo de este trabajo cabe destacar lo siguiente:

- El uso de la Programación estructurada que facilitó el desarrollo del software (Programación) de este diseño, haciendolo mas claro e incluso Permitió la unión de dos Programas independientes.

- La ayuda , en el uso, de la computadora Cromenco del laboratorio de microProcesadores de esta facultad, en la elaboración de los listados de los Programas y Pruebas de algunas subrutinas de nuestro diseño antes de ser ejecutadas por el kit Micro-Professor I.

- A lo largo del desarrollo de la Programación se presentaron algunas dudas sobre el correcto manejo y aplicación de varias instrucciones del lenguaje ensamblador Z-80, lo que nos llevó a un análisis más Profundo de las mismas.

- El diseño de las interfases del micro kit con el exterior, Para sensar los orígenes X e Y, bajar taladro y enviar Pulsos a los motores de Pasos.

- La adquisición de los motores de Pasos en diferentes deshuesaderos.

- El uso de transistores como elementos de conmutación en la fuente de voltaje de los motores de Pasos, Para obtener el mayor Par de los mismos. El diseño de esta fuente es el resultado del análisis de funcionamiento de los motores de Pasos. Dicha fuente, cuenta con sistema de alarma "Fusible abierto" y sistema de visualización de los Pulsos enviados a los motores.

Concluida la realización física del diseño damos a conocer los resultados obtenidos:

- El éxito de la Programación, especialmente el algoritmo de movimiento simultáneo en los ejes X e Y.

- La facilidad de cargado de datos como resultado de la Programación) subrutinas circuito integrado, línea y corte.

- Funcionalidad en el aspecto mecánico (con opción al perfeccionamiento en la bajada del taladro).

De manera independiente, mencionamos que la realización de este trabajo nos muestra de forma más clara la necesidad de una mayor relación de nuestra facultad con la industria, Para conocer sus necesidades y tratar de resolverlas Por medio de la investigación y realización de trabajos de tesis.

## BIBLIOGRAFIA

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

- COOMBS, CLYDE F. PRINTED CIRCUITS HANDBOOK.  
MC GRAW HILL.
- GAULT & PIMMEL. SISTEMAS DIGITALES BASADOS EN  
MICROPROCESADOR. MC GRAW HILL.
- HERBET, TAUB. PRINCIPIOS DIGITALES Y  
MICROPROCESADORES. MC GRAW HILL.
- MICRO-PROFESSOR I, MONITOR PROGRAM SOURCE LIST.
- MICRO-PROFESSOR I, USER'S MANUAL.
- MOSTEK. MOSTEK MK3880. CENTRAL PROCCESSING  
UNIT HANDBOOK.
- STEVE, CIARCIA. CONSTRUYA SU MICROCOMPUTADORA  
CON UN Z-80. BYTE BOOKS. MC GRAW HILL.