

3 21



# Universidad Nacional Autónoma de México

FACULTAD DE ESTUDIOS SUPERIORES  
CUAUTITLAN

## SISTEMA CAD/CAM PARA DESARROLLO Y FABRICACION DE CIRCUITOS IMPRESOS

TESIS CON  
FALSA DE ORIGEN

T E S I S

QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA  
P R E S E N T A N

GERARDO ANTONIO ARANA GARCIA  
OMAR MORALES GONZALEZ  
RAUL GERARDO OAXACA VAZQUEZ

DIRECTOR DE TESIS  
M. EN C. GENARO DE JESUS ROJAS GONZALEZ

CUAUTITLAN IZCALLI, EDO. DE MEXICO 1990



V N A M



## **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 GENERAL

1	INTRODUCCION .....	1
2	SISTEMAS DE POSICIONAMIENTO LINEAL	
2.1	Sistemas de posicionamiento lineal .....	4
2.2	Control de la trayectoria .....	5
2.3	Los motores de pasos como elementos motrices ....	6
2.3.1	Tipos de motores de pasos .....	7
2.3.2	Principio de funcionamiento .....	8
2.3.3	Nodos de funcionamiento .....	10
2.3.4	Características de los motores de pasos ....	12
2.4	Descripción de los movimientos de la máquina ....	13
2.4.1	Movimiento del eje X .....	15
2.4.2	Movimiento del eje Y .....	16
2.4.3	Movimiento del eje Z .....	19
3	INTERFACES PARA MOTORES DE PASOS	
3.1	Generalidades .....	21
3.2	Estructura de la interfaz HARB-C88 .....	21
3.3	Funcionamiento de la Interfaz HARB-C88 .....	23
3.3.1	Generación de la secuencia A/R .....	23
3.3.2	Circuito RESET .....	24
3.3.3	Funcionamiento de la puerta EX-OR como un inversor programable .....	25
3.3.4	Circuito para la secuencia de A/R .....	26
3.3.5	Circuito de potencia para alimentar los devanados de un motor de pasos .....	27
3.3.6	Recortador simétrico .....	30
3.3.7	Circuito limitador de corriente .....	30
3.4	Estructura de la Interfaz HARB-C89 .....	33
3.5	Funcionamiento de la Interfaz HARB-C89 .....	33
3.5.1	Sensores de posición .....	33
3.5.2	Circuito de alimentación para el motor propulsor .....	35
3.5.3	Control de giro para el motor propulsor ...	36
3.5.4	Circuito de potencia para el motor propulsor .....	37

PROGRAMACION DE CONTROL PARA MOTORES DE PASOS  
Y SISTEMA BARRENADOR

4.1	Sección Preliminar .....	40
4.1.1	Del Papel de la Computadora .....	40
4.1.2	Del Lenguaje .....	42
4.1.3	Del Compilador .....	43
4.1.4	De la Palabra de Control .....	43
4.1.5	De la Palabra de Lectura .....	44
4.1.6	De la Programación .....	45
4.2	Catálogo de Funciones .....	46
4.2.1	El Programa Barrenador .....	46
4.2.2	Preparación de Lectura de Posición de Ejes .....	47
4.2.3	Lectura de Posición de Ejes .....	48
4.2.4	Control de Posicionamiento y Barrenado ..	49
4.3	El Archivo Fuente .....	52
4.3.1	Descripción .....	52
4.3.2	El Programa Barrenador MORGAN.C .....	54

COMUNICACION DE DATOS ENTRE EL SISTEMA  
BARRENADOR Y LA MAQUINA CAD

5.1	El Proceso del Diseño .....	60
5.2	El CAD .....	61
5.2.1	AutoCAD .....	62
5.3	El CAM .....	63
5.4	Comunicación de Datos del Sistema CAD/CAM .....	65
5.4.1	Programa Intermedio de Comunicación .....	65
5.4.2	Programa Generador .....	65
5.4.3	Protocolo de Comunicación .....	67
5.4.4	Configuración de los Puertos .....	68
5.4.5	Procedimiento General de Trabajo .....	70

CONCLUSIONES .....

72

APENDICE .....

73

BIBLIOGRAFIA .....

75

## INTRODUCCION

El gran desarrollo de la electrónica y de técnicas de cómputo han permitido la introducción de las computadoras en los talleres académicos e industriales para controlar procesos de producción.

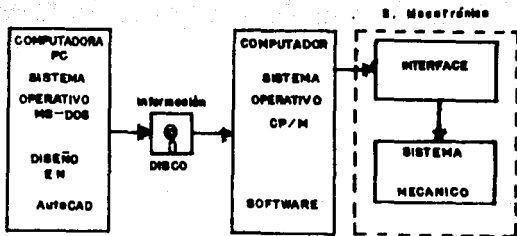
Actualmente existen máquinas automáticas, controladas por computadora, que desarrollan actividades complejas; saben soldar y puntear, ensamblar relojes, montar circuitos, elaborar piezas de precisión en máquinas-herramientas, etc.

Las computadoras han dotado a las máquinas de una especie de cerebro y órganos de percepción, mejorando la eficiencia de las máquinas al trabajar con mayor precisión, alta calidad y rapidez. Además han sustituido al hombre en actividades no deseadas y rutinarias.

El presente trabajo surge de la necesidad de construir una máquina automática, capaz de perforar los barrenos necesarios, en una tarjeta de circuito impreso donde se han de colocar los elementos de un circuito electrónico dado.

La máquina barrenadora automática, es controlada por una computadora con sistema operativo CP/M, ocupando dos sistemas de desplazamiento para localizar las coordenadas de los barrenos sobre el plano X-Y. Un tercer movimiento, permite que el taladro baje y haga la perforación.

Un diagrama a bloques de la estructura de la máquina barrenadora se muestra a continuación:



El primer bloque involucra la utilización de una máquina PC con sistema operativo MS-DOS, debido a que existe una gran variedad de programas de diseño para dichos sistemas de computación.

En ella se diseña el circuito impreso con auxilio de un programa CADD (Computer Aided Design and Drafting) denominado AutoCAD, del cual se obtendrá un archivo con las posiciones de los barrenos. También, se formateará dicho archivo para que sea reconocido por la computadora CP/M.

En el segundo bloque se contempla el diseño del Software que procesa la información.

El tercer bloque está constituido por una parte mecánica y una parte electrónica (sistema mecatrónico).

La parte mecánica se encarga de llevar hasta una posición determinada a la tableta y perforarla. La parte electrónica comunica la parte mecánica con la computadora CP/M.

### Descripción De La Máquina

La máquina utiliza dos sistemas de control:

El sistema de posicionamiento que consiste en dos posicionadores lineales (de un grado de libertad c/u) uno para el eje X, donde va colocada la tarjeta de circuito impreso a ser perforada, y otro para el eje Y, donde viaja la herramienta hasta la posición deseada.

Este es un sistema de coordenadas cartesianas con motores de pasos como elementos motrices. Este tipo de motores permiten obtener una solución de posicionado en bucle abierto o malia abierta suficientemente precisa a condición de manejar cargas débiles y trabajar a velocidades relativamente bajas en comparación con un servomotor por ejemplo.

Los motores son controlados por la interfaz HARB-C88 (diseñada y construida como parte inicial del presente trabajo) conforme al flujo de control enviado por la computadora apoyada en la programación.

El sistema controlador de la herramienta consiste en un mecanismo desplazador en el eje Z (perpendicular a los ejes X-Y), el cual conduce a la herramienta (en este caso una broca) hacia la tarjeta y traerla de regreso hasta su posición original, esperando la nueva orden para realizar el ciclo nuevamente. Este sistema "sube-baja" es controlado por la interfaz HARB-C89 conforme a la señal enviada por la computadora para cada barreno.

Tanto la totalidad de la circuitería como la de la programación han sido seccionadas en diferentes módulos en su diseño y en su implementación con el fin de facilitar tanto la construcción como el mantenimiento de todo el sistema en funcionamiento.

## Organización Del Presente Trabajo

En el capítulo 2 se exponen los dos sistemas de posicionamiento lineal que emplea la máquina y la construcción efectuada.

El capítulo 3 aborda el desarrollo de las interfaces para el control de los sistemas de posicionamiento lineal y para el sistema director de la herramienta.

La programación de control para todos los movimientos están contenidos en el capítulo 4.

Por último en el capítulo 5 se muestra el método empleado para que la comunicación de datos entre el barronador y la máquina CAD fuera el adecuado.

También se encuentra al final una serie de apartados destinados a conclusiones, apéndice y bibliografía consultada con el fin de ampliar ciertos puntos relacionados con el presente trabajo y referenciar fuentes de consulta para emprender posibles mejoras de acuerdo a las bases fundamentales de este tipo de sistemas.

## CAPITULO 2

### SISTEMAS DE POSICIONAMIENTO LINEAL

#### 2.1 POSICION Y DESPLAZAMIENTO \*

Un sistema de posicionamiento lineal está compuesto de un mecanismo que mueve a una partícula en una trayectoria rectilínea.

El movimiento de una partícula en el dominio del tiempo está definido paramétricamente de la siguiente forma:

$$x=x(t) \quad y=y(t) \quad z=z(t)$$

en donde  $x$ ,  $y$ ,  $z$ , son las coordenadas cartesianas de la posición que ocupa el punto en el instante  $t$ .

Si para determinar la trayectoria de un punto son necesarias 3 coordenadas, el punto se mueve en el espacio tridimensional.

Si la trayectoria se puede determinar por 2 coordenadas, esto es, si una de las 3 coordenadas es constante, entonces el punto tiene un movimiento en un plano.

Hay veces en que la trayectoria puede determinarse solamente por una coordenada y en tal caso son constantes 2 coordenadas de posición, en estas circunstancias el punto posee movimiento rectilíneo. Es más claro si se observa la fig. 2.1.

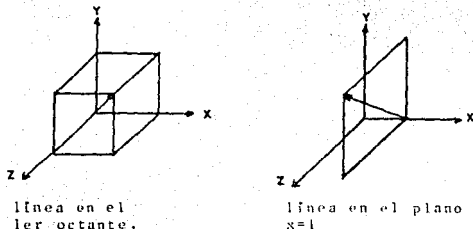


Fig. 2.1

\* Ver ref. 3



## 2.2 CONTROL DE LA TRAYECTORIA \*

Para llevar una partícula de un punto  $RO$  a un punto  $RI$  existe una infinidad de trayectorias, aparte de la rectilínea como se observa en la fig. 2.2a

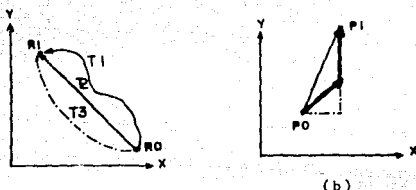
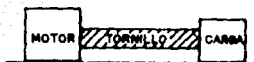


Fig. 2.2

En nuestro caso solo generaremos trayectorias poligonales, (fig. 2.2b) esto lo lograremos mediante alguno de los siguientes sistemas mecánicos.

Se puede controlar el movimiento de una carga para que se desplace a lo largo de un eje mediante un motor y un tornillo como lo muestra la fig. 2.3

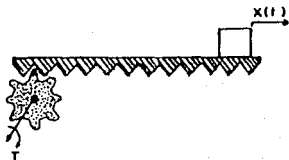


Movimiento de una carga mediante un motor y un tornillo

Fig. 2.3

También mediante una cremallera y un piñón como se indica en la fig. 2.4

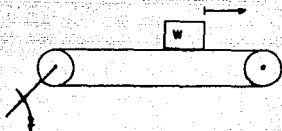
\* Ver ref 2 y 3.



Movimiento de una carga mediante una cremallera y un piñón

Fig. 2.4

Otro es el control de la masa através de una polea mediante un motor, y banda como se representa en la fig. 2.5



Movimiento de una masa mediante el sistema polea-banda

Fig. 2.5

Los elementos motrices para los sistemas de posicionamiento lineal usados, son motores de pasos por las características deseables que tienen.

### 2.3 LOS MOTORES DE PASOS COMO ELEMENTOS MOTRICES \*

El motor de pasos es un dispositivo capaz de transformar pulsos eléctricos en un movimiento rotacional discreto de su eje, éste, gira un determinado ángulo por cada pulso de entrada. El resultado de este movimiento repetible es un posicionamiento preciso y confiable.

En los sistemas de control moderno se presentan a menudo movimientos de tipo incremental, es por esto que los motores de pasos se han convertido en elementos de acción importantes últimamente.

\* Ver ref. 1,2 y 3.

Podemos encontrar estos motores en unidades de disco flexible, unidades de disco duro, impresoras y en gran variedad de máquinas herramientas.

### 2.3.1 Tipos de Motores de Pasos

Existen 3 tipos de motores de pasos.

- 1.- Motores de imán permanente.
- 2.- Motores de reluctancia variable.
- 3.- Motores híbridos o mixtos.

#### 1.- Motores de imán permanente.

Los motores de imán permanente están formados por un estator de forma cilíndrica, con un cierto número de bobinas alimentadas en secuencia, que crean un campo magnético giratorio de manera discreta.

El rotor coaxial con el estator y situado sobre el eje, contiene un imán permanente de N polos, que en cada instante tiende a alinearse con el campo magnético creado por el estator.

#### 2.- Motores de reluctancia variable.

En los motores de reluctancia variable, el estator presenta la forma habitual (de los motores a pasos), con un número determinado de polos electromagnéticos.

Sin embargo el rotor no es de imán permanente, sino que está formado por un núcleo de hierro dulce de estructura cilíndrica, pero con un cierto número de dientes tallados longitudinalmente a lo largo de su superficie lateral.

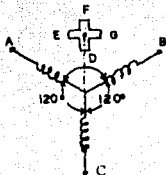


Fig. 2.6 Motor de reluctancia variable de 3 fases

La fig. 2.6 muestra un esquema de este tipo de motor. Se observa que el rotor incorpora los dientes D, E, F, G, y que el estator se ha considerado con 3 bobinas A, B, C, (3 fases son las mínimas para que se tenga control direccional).

Si la bobina C se alimenta con corriente continua, el rotor se alinea con el polo D. Cuando la bobina B es la alimentada, el polo más cercano (G) es el que se alinea con B desplazándose unos 30 grados.

Si ahora la bobina A es excitada, es el polo F más cercano el que se orienta hacia esta bobina girando otros 30 grados. Por lo tanto se obtiene un desplazamiento de 30 grados para cada pulso alimentado, a una de las tres fases



fig. 2.7

En la fig. 2.7 se muestra una curva de velocidad contra torque típica para los motores de reluctancia variable. Se observa que el torque presenta irregularidades a bajas frecuencias.

### 3.- Motores híbridos.

Estos combinan, las características de los 2 tipos descritos anteriormente, obteniéndose importantes torques de accionamiento, un gran número de pasos por vuelta y frecuencia de trabajo elevada.

#### 2.3.2 Principio de Funcionamiento \*

El principio de funcionamiento de los motores de pasos de dos fases es muy simple de entender, se basa en las fuerzas de atracción y repulsión entre polos magnéticos.

\* Ver ref 1 y 3.

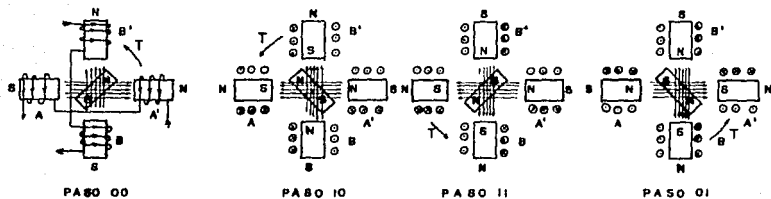
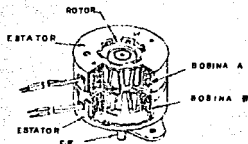


Fig. 2.8 Motor de imán permanente de 2 fases 4 polos.

En base a la fig. 2.8, se hace la explicación del funcionamiento de un motor de pasos de imán permanente.

Se observa que el motor, cuenta en su estator con 4 polos formados por electroimanes y el rotor consiste de un imán permanente, al cambiarse la polaridad en los polos del estator, el rotor girará en sentido contrario al de las manecillas del reloj.

Se nota que el motor gira 90 grados por cambio de polaridad, lo que no representa un dispositivo práctico, generalmente los motores de pasos constan de muchos polos en su estator.



La fig. 2.9 muestra como está construido un motor de n pasos y 2 fases de imán permanente.

El estator está formado por 2 tazas (taza A y taza B), alrededor de las cuales están alojadas las bobinas (bobina A y bobina B), con pares de polos mecánicamente desplazados en medio paso polar.

El imán permanente del rotor está magnetizado con un mismo número de pares de polos de una de las bobinas del estator.

La interacción entre el rotor y el estator (polos opuestos se atraen e iguales se repelen) cuando se energizan alternativamente

polos magnéticos Sur y Norte, fig. 2.10 causa el movimiento del rotor en  $1/4$  de paso polar por cada cambio de polaridad del estator.

Entre las dos bobinas del estator existe un desplazamiento de  $1/4$  de paso polar.

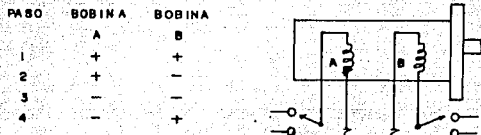
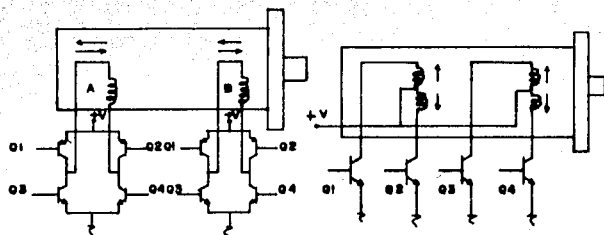


Fig 2.10 Esquema para la secuencia normal de los 4 pasos.

### 2.3.3. Modos de Funcionamiento \*

Existen 2 modos de funcionamiento para los motores de imán permanente :

- Paso Completo (full-step mode).
- Medio paso (Half-step mode).



#### BIPOLAR

#### UNIPOLAR

Step	Q1-Q4	Q2-Q3	Q5-Q8	Q6-Q7	Paso	Q1	Q2	Q3	Q4
1	ON	OFF	ON	OFF	1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON	2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON	3	OFF	ON	OFF	ON
4	ON	OFF	ON	OFF	4	OFF	ON	ON	OFF

#### SECUENCIA NORMAL DE LOS 4 PASOS

\* Ver ref. 1

1	ON	OFF	ON	OFF	1	ON	OFF	ON	OFF
2	ON	OFF	OFF	OFF	2	ON	OFF	OFF	OFF
3	ON	OFF	OFF	ON	3	ON	OFF	OFF	ON
4	OFF	OFF	OFF	ON	4	OFF	OFF	OFF	ON
5	OFF	ON	OFF	ON	5	OFF	ON	OFF	ON
6	OFF	ON	OFF	OFF	6	OFF	ON	OFF	OFF
7	OFF	ON	ON	OFF	7	OFF	ON	ON	OFF
8	OFF	OFF	ON	OFF	8	OFF	OFF	ON	OFF
1	ON	OFF	ON	OFF	1	ON	OFF	ON	OFF

#### SECUENCIA MEDIO PASO; 8 PASOS

1	ON	OFF	OFF	OFF	1	ON	OFF	OFF	OFF
2	OFF	OFF	OFF	ON	2	OFF	OFF	OFF	ON
3	OFF	ON	OFF	OFF	3	OFF	ON	OFF	OFF
4	OFF	OFF	ON	OFF	4	OFF	OFF	ON	OFF
1	ON	OFF	OFF	OFF	1	ON	OFF	OFF	OFF

#### SOLO UNA BOBINA A LA VEZ; 4 PASOS

(Ref. 1)

Fig. 2.11

#### Funcionamiento a Paso Completo

Las bobinas pueden ser energizadas de dos formas; solo una a la vez, o las dos al mismo tiempo con la secuencia normal de los 4 pasos. \*

Alimentando solo una bobina, ver fig. 2.11, el torque de sujeción y el de funcionamiento proporcionado con el voltaje nominal se reduce un 30%. El voltaje nominal puede ser incrementado para aumentar el torque de salida. La ventaja con este tipo de alimentación es que incrementa la eficiencia con la desventaja de que se decremanta la exactitud.

Funcionamiento a medio paso, ver fig. 2.11, la secuencia consta de 8 pasos, así el rotor gira la mitad de su recorrido normal por paso. Las ventajas de trabajar con medio paso consiste en una resolución mayor, con reducción de la resonancia, el torque no es constante sino que en un paso es mayor y en el otro menor, pero se adquiere mayor velocidad (un análisis más detallado de la resonancia en motores de pasos se puede ver en la ref. 2).

\* Ver ref. 1

Existen 2 formas de alimentación ( ver fig. 2.11 )

- 1.- Forma bipolar y
- 2.- Forma unipolar.

Se denomina bipolar porque la corriente en los devandos, circula en ambos sentidos y si solo circula en una dirección se denomina unipolar.

La operación bipolar, es mejor comparada con la unipolar, ya que en esta última para lograr la inversión de flujo, se devanan 2 bobinas en direcciones opuestas, una encima de la otra (devanado bifiliar).

Por lo que, para el mismo número de vueltas que en el bipolar, el diámetro del alambre debe ser menor, aumentando así, la resistencia de la bobina, con el resultado de que, a velocidades de paso bajas, los motores unipolares presentan un torque 30% menor que el de los bipolares ( ver ref.2 ).

#### 2.3.4 Características de los Motores de Pasos \*

##### Exactitud

Sin carga o a carga constante, la exactitud entre cada paso es + - 6.5%, y el error es no acumulativo.

##### Torque

El torque producido por un motor de pasos depende de los siguientes factores:

- a) Velocidad de avance.
- b) Corriente de devanado.
- c) Diseño del circuito de potencia. \*\*

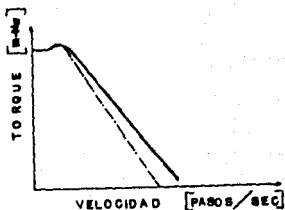
Al inicio, cuando el motor está parado (cero pasos/sec. y energizado), el torque necesario para que el motor gire un paso completo se denomina "Torque de sujeción", este es mayor que el torque dinámico y se usa para frenar la carga.

Estando el motor parado y sin alimentación, existe también un torque llamado "torque residual", que es debido a la fricción en los baleros y al flujo permanente en el caso de los motores de imán permanente, este torque es aproximadamente 1/10 de torque de sujeción.

\* Ver ref. 1 y 3.

\*\* Ver ref. 1.





----- Operación      ..... Al inicio.

Fig. 2.12 Curva característica velocidad vs. torque

la fig. 2.12 muestra una curva característica de torque contra velocidad de los motores Airpax de imán permanente.

Ventajas y desventajas de usar motores de pasos respecto a los de CD y CA:

- Ventajas**
- Software más sencillo.
  - No necesita en Gral. retroalimentación.
  - Hardware más sencillo.

- Desventajas**
- Potencias bajas en la mayoría de ellos.
  - Resolución discreta.

## 2.4 DESCRIPCION DE LOS MOVIMIENTOS DE LA MAQUINA

La máquina barrenadora consta de 3 dispositivos de desplazamiento, que actúan independientemente. Los desplazamientos X, Y, Z, dotan a la máquina de 3 grados de libertad que son descritos a continuación.

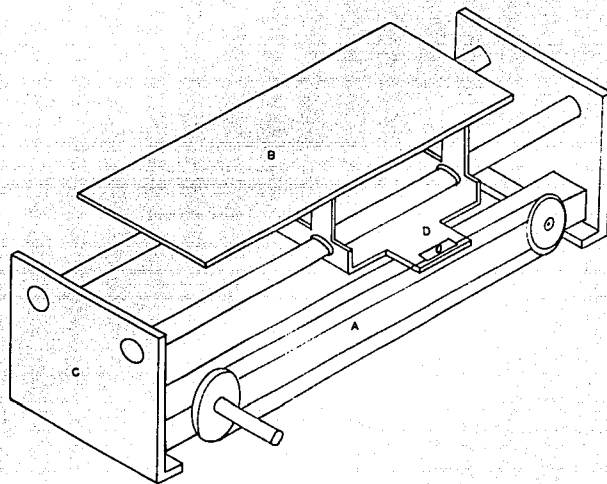


Fig. 2.13

**SISTEMA DEL MOVIMIENTO "X"**

- A) SISTEMA DE TRASLACION POLEA Y BANDA
- B) MESA DE TRABAJO
- C) BARRAS GUIAS
- D) CORREDERA

### 2.4.1 Movimiento del EJE X

En la fig. 2.13, se observa que se convierte el movimiento de rotación del motor de pasos, en un movimiento de traslación a través de una polea y una banda inextensible.

En este trabajo se produce el movimiento de rotación con un motor BERGER de 24 pasos por revolución, de 16 Vcd, 340mA, acoplado a un tren de engranes (Fig. 2.14), para ganar resolución y potencia.

La reducción de velocidad se efectúa de la siguiente forma. Si  $e_1$  y  $e_2$  tuviesen igual número de dientes, la relación de vueltas de  $e_2$  respecto a  $e_1$  sería 1.

Si el número de dientes de  $e_2$  es mayor que el de  $e_1$ , las revoluciones que dará  $e_2$ , serán una fracción de las revoluciones de  $e_1$ . Es decir,  $e_2$  dará  $N_1 / N_2$  revoluciones de  $e_1$ . De forma análoga ocurrirá con  $e_2$  y  $e_3$ ,  $e_3$  y  $e_4$ ,  $e_4$  y  $e_5$ .

$N_1$  ..... 1 vuelta de  $e_1$   
 $N_1/N_2$  ..... Fracción de vuelta de  $e_2$  cuando  $e_1$  dió una vuelta  
 $N_3/N_4$  ..... Fracción de vuelta de  $e_3$  cuando  $e_2$  dió una vuelta  
 $N_4/N_5$  ..... Fracción de vuelta de  $e_4$  cuando  $e_3$  dió una vuelta

Tomado en cuenta la vuelta del  $e_1$ , tenemos:

$N_1/N_2$  ..... Vueltas de  $e_2$   
 $N_3/N_4$  ( $N_1/N_2$ ) ..... Vueltas de  $e_3$   
 $N_4/N_5$  [ $N_3/N_4$  ( $N_1/N_2$ )] ..... Vueltas de  $e_4$

Simplificando, tenemos:

$N_1 N_3/N_2 N_4$  ..... Vueltas de  $e_4$ , cuando  $e_1$  dá una vuelta

Si  $N_1=24$ ,  $N_2=41$ ,  $N_3=11$ ,  $N_4=58$  y  $N_5=75$ , la relación de multiplicación es de :

$N_1 N_3/N_2 N_4 = 0.0858$  vueltas de  $e_4$  cuando  $e_1$  dá una vuelta, en forma inversa si  $e_4$  dá una vuelta, el deberá dar:

$N_2 N_4/N_1 N_3 = 11.647$  vueltas

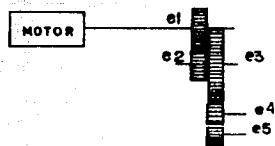


Fig. 2.14

La flecha de e5 mueve la polea que desplaza una corredera por medio de una banda de acero flexible ( fig. 2.13 ). La corredera está montada en 2 barras guías, através de 4 bujes de nylamid.

En este dispositivo de movimiento lineal está colocada la mesa de trabajo, sobre la que está colocada la placa de trabajo a barrenar.

La máxima carrera o avance de este eje está limitada por un par de sensores, que detectan cuando se han excedido los límites y si esto ha sucedido se detiene el movimiento para evitar que alguna parte de la máquina sufra un desperfecto.

Los sensores son del tipo optoelectrónico, Los sensores de límite de carrera se analizará en el siguiente capítulo.

## 2.4.2 Movimiento del EJE Y

### 2.4.2.1 Generalidades

Los términos a usar en esta parte del capítulo son los sigtes:

Paso (p) es la distancia que hay entre 2 hilos adyacentes, medidos paralelamente al eje de la rosca, y es el recíproco de los hilos por pulgada N.

Avance (l) es la distancia que se desplaza una tuerca paralelamente al eje de la rosca de un tornillo, cuando dá una revolución.

En el caso de una rosca simple (o de un solo filete) como el de la fig. 2.15, el avance es igual al paso. Un elemento con rosca múltiple es aquel cuya rosca tiene 2 o más hilos ( imagine 2 o más cordones enrollados juntos alrededor de un eje).

Por lo tanto en un tornillo de rosca múltiple X (donde X es el número de hilos), el avance está dado entonces por la siguiente expresión.

$$l = X * p$$

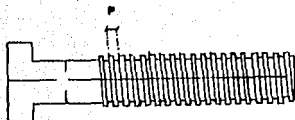


Fig. 2.15

El sistema del movimiento Y (fig.2.16), consta de un motor de accionamiento (a) (motor de pasos), 2 barras guías (b), una barra transportadora (c) (que es un tornillo sinfín de rosca múltiple, 5 filetes), una mesa (d) donde se soporta el movimiento Z, un tren de engranes (e), 4 bujes de nylamid (X).

El tornillo por su forma de uso, es denominado tornillo de fuerza.

Los tornillos de fuerza son elementos que se utilizan en máquinas para convertir un movimiento angular, en un movimiento lineal y transmitir así, fuerza o potencia (así siempre se emplean para husillos o tornillos de avance de tornos, y para tornillos de bancos, prensas y gatos).

El control del desplazamiento está implementado por un motor de pasos, acoplado a un tren de engranes, a su vez está unido al tornillo sinfín en el que viaja una tuerca, que está fija a la mesa y la arrastra.

La mesa está soportada por las 2 barras guías a través de 4 bujes de nylamid.

El tren de engranes tiene una relación de 125:6 revoluciones, o 500 pasos del motor por una revolución en el árbol de la salida.

El tornillo sinfín es de quintuple rosca, con un paso de 0.236 pulg. por lo que el avance es,

$$l = 5 * 0.236 \text{ pulg.} = 1 \frac{5}{32} \text{ pulg.}$$

El motor de pasos consta de 24 pasos por revolución, y el tren tiene una desmultiplicación de 125:6, si se define a la resolución que tiene este sistema como el avance lineal que tiene la mesa cuando se le comanda un paso queda;

$$\begin{aligned} \text{resol} &= (1 \frac{5}{32} \text{ pulg} / (125/6)) / 24 \text{ pasos} \\ \text{resol} &= 0.0024 \text{ pulg. / paso} \end{aligned}$$

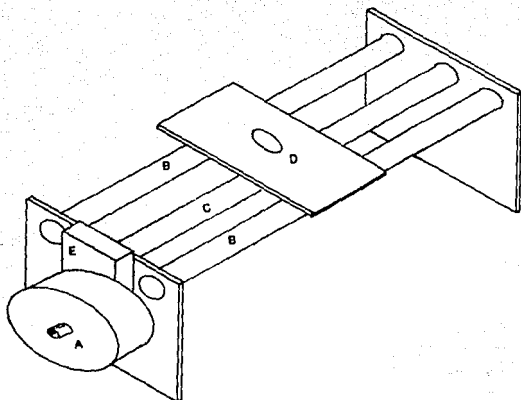


Fig. 2.16

**SISTEMA DEL MOVIMIENTO "Y"**

- A) MOTOR DE ACCIONAMIENTO**
- B) BARRAS GUIAS**
- C) BARRA TRANSPORTE**
- D) MESA SOPORTE**
- E) TREN DE ENGRANES**

### 2.4.3 Movimiento del EJE Z

Movimiento eje Z (Fig.2.18). El sistema que desplaza la herramienta de corte, hasta la tableta de circuito impreso a perforar, está formado por, una base (a), que sirve de soporte para realizar el movimiento del contenedor (b), en el que van alojadas la herramienta de corte (f) y el motor propulsor (d).

Incorporado a la base se encuentra una hembra roscada (e) que proporciona un avance vertical por vuelta del birlo acoplado al eje del motor propulsor.

Existe en la base (a) una escotadura que permite a la herramienta llegar libremente a la tableta a perforar.

La tableta deberá estar colocada sobre la mesa principal (eje X).

El contenedor (b), es la pieza en movimiento en el eje z, que aloja un par de bujes maquinados en teflón, en los cuales se establece el contacto o el desplazamiento con los pernos guía (c), que van fijos a la base. El material de estos pernos es de bronce y su maquinado es preciso para que el contenedor no se trabe.

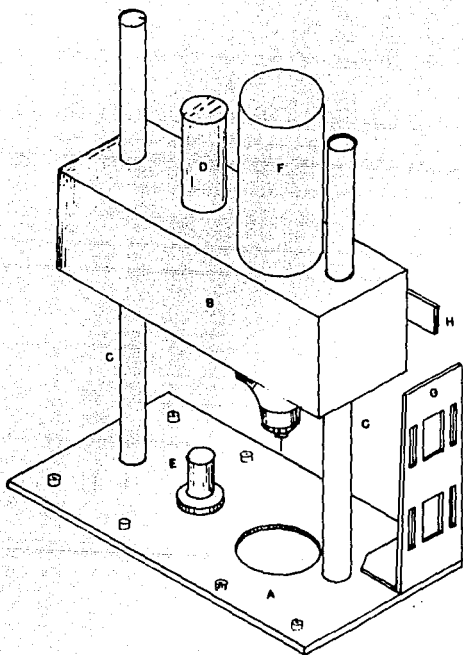
El motor propulsor, es un motor de CD, del tipo en derivación, que posee un pequeño tren de engranes, de éste sobresale su eje el cual se acopló al birlo.

El motor propulsor está fijo en la parte superior del contenedor (b).

La herramienta de corte es un taladro comercial de velocidad variable (de 5000 a 28000 rpm).

Pestaña (g), En ésta se encuentran los sensores optoelectrónicos de límite de carrera, que detectan los extremos de la carrera del contenedor, la detección se hace al hacer pasar por los sensores de límite de carrera, una laminilla (n), que va fija al contenedor.

Los sensores de límite de carrera pueden variar su posición para aumentar o disminuir la carrera del contenedor según convenga.



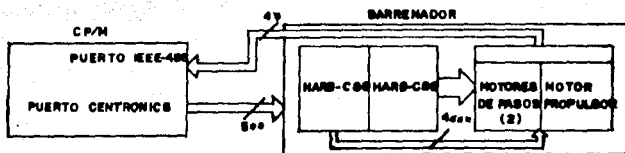


## CAPITULO 3

### INTERFACES PARA MOTORES DE PASOS

#### 3.1 GENERALIDADES

Los sistema que conectan los motores de pasos y el motor propulsor con la computadora CP/M son las interfaces HARB-C88 y HARB-C89, diseñadas en este trabajo, como parte de la máquina barrenadora. Según se muestra en la fig.3.1



\*4 Señales (2 límites del eje X , 2 límites para el eje Y)

\*\*5 Señales (2 señales de reloj, 2 señales control A/R y 1 pulso de activación para el motor propulsor )

Fig. 3.1 Diagrama a bloques

Las señales que hacen funcionar las interfaces HARB-C88 y HARB-C89 son enviadas por la computadora CP/M a través del puerto paralelo CENTRONICS por medio de la programación de control escrita en lenguaje C; encargada de tener el control de la máquina barrenadora en todo momento.

#### 3.2 ESTRUCTURA DE LA INTERFAZ HARB-C88

La interfaz HARB-C88 está compuesta por dos canales, uno para cada motor de pasos (ver fig. 3.17c) a continuación se muestra la estructura con diagramas a bloques de un canal, el otro es igual.

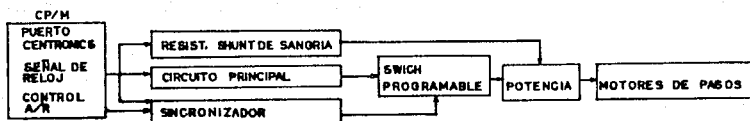


Fig. 3.2 Diagrama a bloques de la estructura de un canal de la interfaz HARB-C88.

En el circuito principal se generan los dos trenes de pulsos defasados entre sí  $90^\circ$ , que sirven para dar las formas de onda para los devanados de los motores de pasos, iniciando los trenes de pulsos en un nivel lógico alto, con un circuito RESET, ver fig. 3.5.

La salida del circuito principal se envía a un switch programable, que decide si la secuencia es de avance o retroceso, con unas puertas X-OR.

La decisión de avance ó retroceso se da con una señal de control a través del sincronizador, para asegurar que la señal esté disponible antes de entrar el siguiente dato.

Las señales del switch programable, son enviadas a un circuito de potencia, que suministre la corriente necesaria, para alimentar los devanados de los motores de pasos.

Cuando los motores de pasos están en reposo y trabados, el torque de sujeción que necesita desarrollar, es mayor que el desarrollado cuando está en movimiento. El torque desarrollado por el motor trabado demanda más corriente, que cuando está en movimiento. Para alargar la vida útil del motor de pasos, protegiendo sus devanados una resistencia shunt sangra la corriente que suministra la etapa de potencia, por medio de un detector de pulsos. La fig. 3.3 muestra con diagramas a bloques el funcionamiento de la resistencia shunt de sangría.

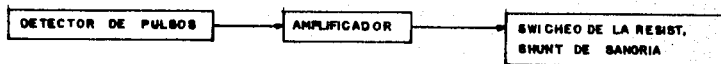
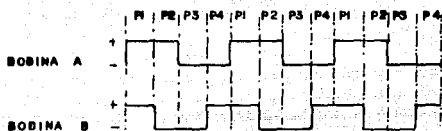


Fig. 3.3 Diagrama a bloques del circuito que desvía la corriente

### 3.3 FUNCIONAMIENTO DE LA INTERFAZ HARB-C88

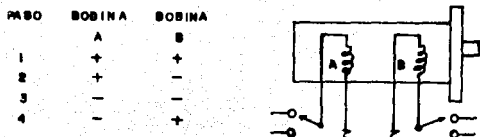
#### 3.3.1 Generación de la Secuencia A/R

La interfaz HARB-C88 que acciona los motores de pasos, para mover los sistemas de posicionamiento lineal, está diseñada para dar la secuencia normal de 4 pasos, ver fig. 3.4 y sección 2.3.3



\* P1-P4 son Paso 1, Paso 2, Paso 3 y Paso 4.

(a) Forma de voltaje para los devanados, secuencia normal 4 pasos



(b) Forma de alimentar a los devanados del motor de pasos

Fig. 3.4

De la fig. 3.2 el bloque de la interfaz HARB-C88 que genera los trenes de pulsos defasados 90°, es el circuito principal, compuesto por un arreglo de FLIP-FLOP'S tipo D, como se indica en la fig.3.7

En la fig. 3.4 se establece que la secuencia de pulsos debe empezar en un nivel alto para ambos trenes de pulsos, el circuito que forza a inicializar a los FLIP-FLOP'S a la condición inicial al encendido en un nivel lógico alto es el circuito RESET, fig.3.5; que envía a los PRESET'S del arreglo de FLIP-FLOP'S un nivel lógico bajo.

### 3.3.2 Circuito RESET

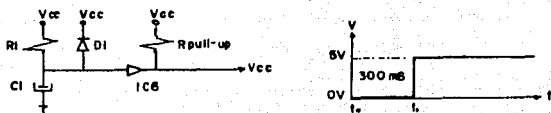


Fig 3.5 Circuito del RESET, para la condición inicial

El circuito RESET envía un pulso bajo que dura 300 ns hasta después de iniciarse la alimentación, tiempo en el que el capacitor  $C_1$  alcanza el voltaje de alimentación ( $V_{cc}$ ), para pasar a un nivel lógico alto, en el que permanece hasta apagar el sistema.

El diodo es empleado para descargar rápidamente al capacitor en ausencia de alimentación, asegurando condiciones iniciales nulas por si existe un apagón repentino.

Para mejorar la señal de transición, relativamente lenta del RESET y que opere confiablemente, es usado un BUFFER, IC6, circuito amplificador que además incrementa la capacidad del manejo de corriente en la línea del RESET, asegurando que se puedan conectar varios circuitos a la salida del RESET (FAN-OUT).

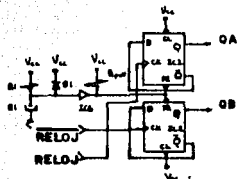
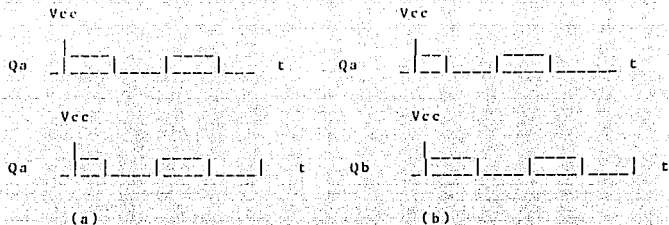


Fig.3.7

El circuito de la figura 3.7, genera la secuencia de avance mostrado en la fig. 3.4 (a), comenzando la secuencia por el paso uno, siempre que se reinicializa la alimentación.

La secuencia de retroceso partiendo de cualquiera de los 4 pasos, se genera al negar alternativamente la secuencia de pasos de avance, inmediatamente después del paso donde se quiere retroceder, como se muestra en la fig. 3.8



(a) Secuencia de avance 11,10;00,01,11; el 5º paso es el último de avance, en 3.8b se ilustra la secuencia de retroceso a partir de este paso.

(b) Secuencia de retroceso 11,01;00,10,11; formas de onda para el voltaje aplicado a los devanados para retroceder el motor de pasos.

Fig. 3.8

### 3.3.3 Funcionamiento de la Puerta X-OR como un Inversor Programable

El switch programable genera las formas de onda de la secuencia de retroceso, a partir de la secuencia de avance, usando inversores programables, con puertas X-OR.

La Fig.3.9a muestra una puerta X-OR usada como inversor programable, cuando la entrada P es puesta en un nivel lógico alto (Fig. 3.9b), la salida es el inverso de la entrada S. Cuando la entrada P es puesta a un nivel lógico bajo (Fig. 3.9c), la salida es idéntica a la entrada S.

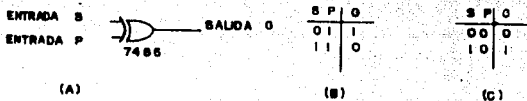


Fig. 3.9 Funcionamiento de la puerta X-OR como inversor programable

### 3.3.4 Circuito para la Secuencia de A/R

El circuito para la secuencia de Avance/Retroceso es mostrado en la Fig 3.10. Si la secuencia es de avance, la señal P debe estar a un 0 lógico. Si la secuencia es de retroceso la señal P debe ser un tren de pulsos, Fig. 3.10 (b)

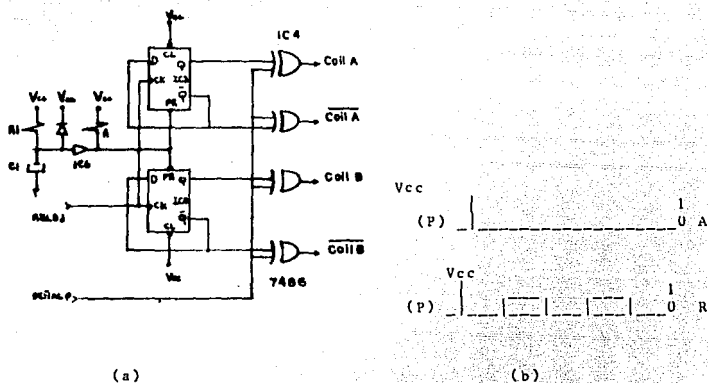


Fig. 3.10 Circuito para la secuencia de Avance/Retroceso

La señal P debe estar disponible antes de que entre el siguiente pulso de reloj en el circuito principal, siendo sincronizada con un FLIP-FLOP y una puerta AND, como se indica en la fig. 3.11

Através del FLIP-FLOP IC1, que funciona al doble de la frecuencia, con que es generada la secuencia de avance, se da la señal de control de cambio de avance del motor de pasos, con el fin de estar disponible el cambio de avance, al activarse la señal de control de Avance/Retroceso.

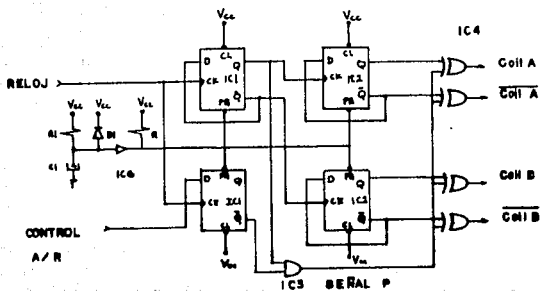


Fig. 3.11 Circuito para la secuencia de Avance/Retroceso, con la señal de control de cambio de A/R en sincronía, con la señal de reloj que alimenta a toda la Interfaz.

Cuando el control de Avance/Retroceso está en un 0 lógico, la puerta AND de la fig. 3.11, bloquea al reloj y aplica un 0 en el control del switch programable, por tanto pasará la secuencia de avance. Si el control de A/R está en 1 lógico, la puerta se abre y pasa el reloj al control de los switches programables, obteniéndose así una secuencia de retroceso.

La lógica TTL maneja poca corriente ( $I_{OL} = 16mA, I_{OEX} = 250mA$ ), que no es suficiente para darle potencia a los motores de pasos. Para controlar con lógica TTL a los motores de pasos se usa la etapa de potencia, que como indica la fig. 3.2 es la siguiente después del circuito de switches programables, que es usada para proporcionar la corriente de los motores de pasos.

### 3.3.5 Circuito de Potencia para Alimentar los Devanados de un Motor de Pasos

El circuito de potencia utiliza 4 transistores y una fuente de alimentación unipolar, como se indica en la fig. 3.13

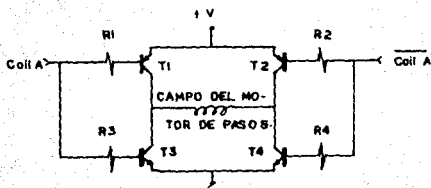
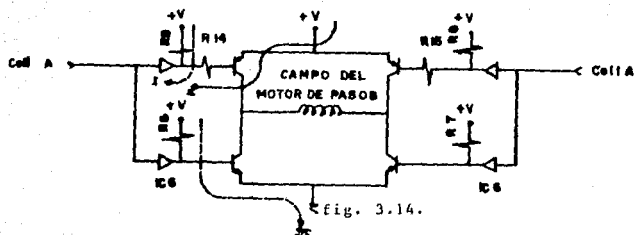


Fig. 3.13 Circuito de potencia para cada devanado de un motor de pasos, T1-T2 son Tip 32A y T3-T4 son Tip31A.

Mótese que esta configuración requiere un circuito PUSH-PULL con los transistores de potencia T1-T4. El funcionamiento de los transistores T1-T4 es siempre en saturación o en corte por parejas, Así, si T1 y T4 conducen T3 y T2 están bloqueados o viceversa, controlando la frecuencia de la señal de excitación se varía la velocidad de los motores.



La fig. 3.14 muestra la forma de polarizar los transistores de la etapa de potencia, para uno de los devanados de los motores de pasos. Cada motor necesita 2 circuitos, como el indicado en la fig. 3.14, uno para devanado del motor bifásico.

#### - Cálculo de las resistencias de polarización.

Para determinar el valor de la resistencia R5, debe tomarse en cuenta, que cuando la salida del buffer de colector abierto IC6, es baja, la corriente de sumidero a través de la salida baja, no exceda el límite  $I_{OL}$  (corriente de salida de nivel bajo aprox. 16 mA.), de IC6.



De la fig. 3.14

- Malla 1  
 $16 \text{ V} - R_5 I = 0 \text{ V}$  si  $I = I_{ol}/2$ ,  $I_{ol} = 16 \text{ mA}$   
 $R_5 = 16 \text{ V} / 8 \text{ mA}$   $R_5 = 2000 \text{ Ohms}$   
Se usó una resistencia de  $2.2 \text{ k Ohms}$ .

Cálculo de la potencia a disipar por la resistencia  $R_5$   
 $I_{trab} = 16 \text{ V} / 2.2 \text{ k Ohms}$   $I_{trab} = 7.273 \text{ mA}$   $P = V \cdot I_{trab}$   
 $P = 0.11 \text{ W}$ .  $\Rightarrow$  la potencia será de  $1/4 \text{ W}$

- Malla 2  
 $16 \text{ V} = V_{eb} + R_{14} I_b$  Si  $V_{eb} = 0.7 \text{ V}$   $I_b = I_e / \text{Beta}$

\*La corriente de emisor es la que demanda cada devanado del motor (I es igual a  $340 \text{ mA}$ )  
\*La beta de trabajo del transistor es de 45

$16 \text{ V} = 0.7 \text{ V} + R_{14} \cdot (340 \text{ mA} / 45)$   $R_{14} = 1800 \text{ Ohms}$

Cálculo de la potencia que disipará la resistencia  $R_{14}$   
 $P = R_{14} \cdot I_b^2$   $P = 0.127 \text{ W}$  la resistencia  $R_{14}$  que se usó es de  $1/4 \text{ W}$ .

El valor de  $R_7$  y  $R_8$ , se calcula cerrando la malla 3. Dando un valor igual al de  $R_{14}$ .

Cuando la corriente que circula por los devanados cambia de dirección se genera una fuerza contraelectromotriz, comúnmente llamada "latigazo", esta fuerza contraelectromotriz de corta duración y de gran magnitud acorta la vida útil de los transistores de potencia.

Para evitar los latigazos de voltaje, se usan dos diodos zener conectados en oposición que recortan los picos de voltaje generados en los campos de los motores de pasos, a  $(V_z + 0.6 \text{ V})$  como se muestra en la fig. 3.15

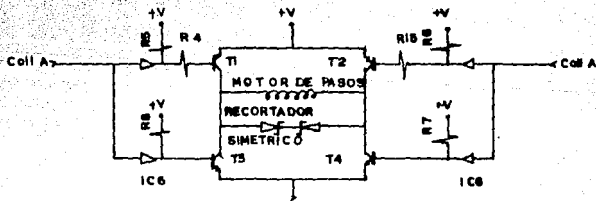
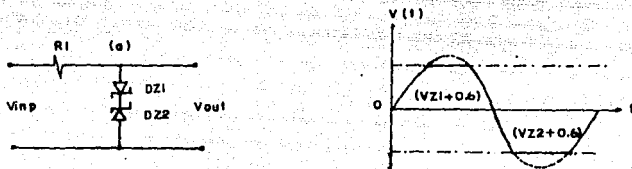


Fig. 3.15

### 3.3.6 Recortador Simétrico

El funcionamiento del recortador simétrico es el siguiente :

Si una onda senoidal alimenta a los diodos zener 1 y 2 , como se muestra en la fig. 3.16 , conectados en serie encontrados, mediante la resistencia  $R_1$ , mientras la amplitud de la onda senoidal es menor que el voltaje de ruptura del diodo zener ( $V_z$ ) más 0.6 V de la tensión directa del otro diodo, no tiene lugar el recorte, entradas mayores que este nivel, la salida queda limitada a ( $V_z + 0.6$  V) como se muestra en la fig. 3.16 (b)



(a) Recortador simétrico

(b)  $V_{out}$  del recortador simétrico

Fig. 3.16

### 3.3.7 Circuito Limitador de Corriente

En la sección 3.2 se estableció que si los motores de pasos están parados y existe alimentación, para desarrollar el torque de sujeción necesitan mayor corriente, que la demandada cuando están en movimiento. Se diseñó un circuito limitador de corriente, que switchea una resistencia shunt de sangría, cuando no existen pulsos en la interfaz HARB-C88.

Un detector de pulsos activa un transistor switchéandolo, mostrado en la fig. 3.17, es alimentado con la señal de reloj, el capacitor  $C_2$  rectifica la señal cuadrada, descargándose el capacitor durante los niveles bajos de la señal de reloj, obteniéndose una señal continua, cuando existe reloj.

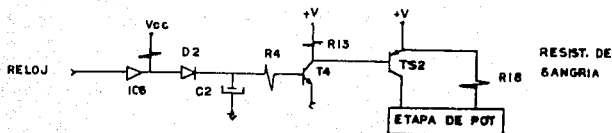


Fig. 3.17 Circuito limitador de corriente

La señal continua se envía a un transistor para amplificarla y ésta switchea un transistor de potencia, conectando o desconectando una resistencia shunt de sangría, según sea el caso.

Cálculo de las resistencias de polarización.

Se diseñó el circuito limitador de corriente, con un tiempo de carga del capacitor C2 de 0.05 S. Para garantizar que funcione C2 rápidamente en presencia de pulsos de reloj, enviando un nivel lógico alto a la base de TS1, saturando a TS2, bloqueando a la resistencia shunt de sangría.

Mientras que el tiempo de descarga del capacitor es muy grande (2 S.) para asegurar que se excite la base de TS1 mientras dure el pulso bajo, como se muestra en la fig. 3.17 b.

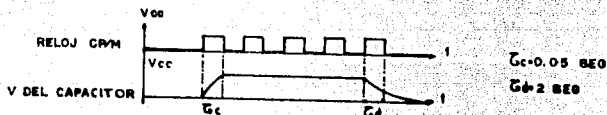


Fig. 3.17b voltaje de carga y descarga del capacitor C2.

Para la carga de C2

Si carga de C2 = 0.05S y C2 = 22  $\mu$ F => R3= 0.05S/22  $\mu$ F  
R3=2200 Ohms a 1/4 W.

Para la descarga de C2

Si descarga de C2 = 2S y C2 = 22  $\mu$ F => R4=2S/22  $\mu$ F  
R4 = 82 000 Ohms a 1/4 W

Cálculo de R13

La resistencia R13 polariza a la base de TS2, si Ie de TS2 es de 340 mA => Ib de TS2 = 340 mA/45, Ib Ts2=7.5 mA = Ic T4

R13 = ( 16 - 0.2 ) V / 7.5 mA R13 = 2107 Ohms

Potencia de R13=2107\*(7.5 mA)<sup>2</sup> Potencia de R13 = 0.118

Se usó una resistencia de 2200 Ohms a 1/4 de Watt.

La fig. 3.17c muestra la estructura final de Interfaz HARB-C88

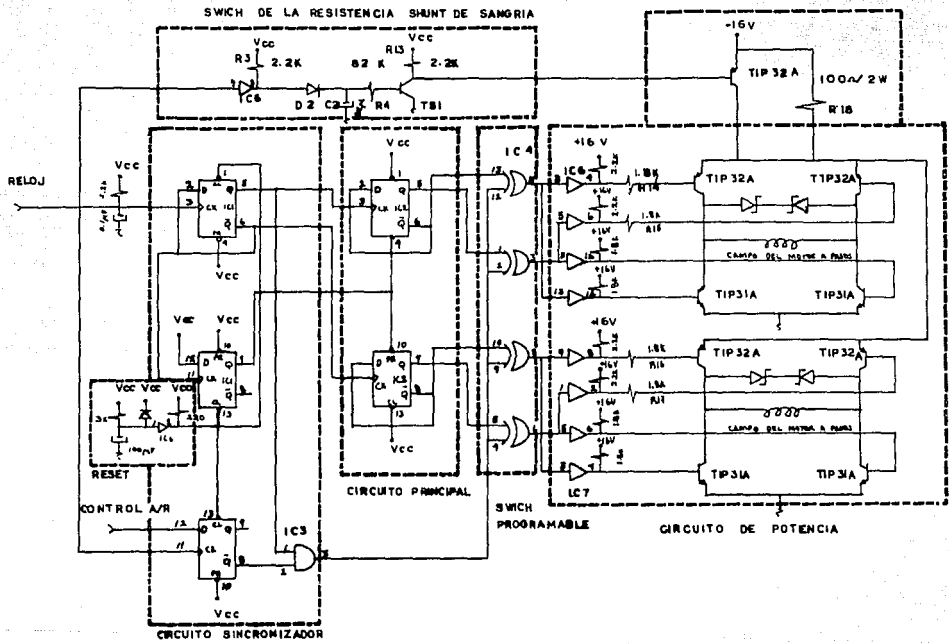


Fig. 3-17C  
 ESTRUCTURA DE LA INTERFAZ HARD-C88

### 3.4 ESTRUCTURA DE LA INTERFAZ HARB-C89

La interfaz HARB-C89 (ver Fig. 3.25) controla un motor de CD del tipo en derivación, que baja la herramienta de corte (Dremel, moto-tool velocidad variable, modelo 370-5) para perforar, y la regresa a lugar de reposo. A continuación se muestra la estructura a bloques de la Interfaz HARB-C89 fig. 3.18

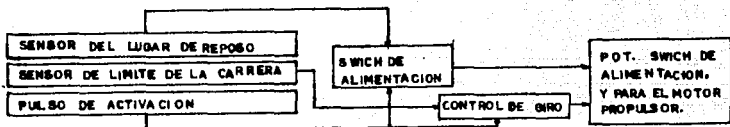


Fig. 3.18 Diagrama a bloques de la estructura de la interfaz HARB-C89

El bloque del switch de alimentación, reconoce la posición de la herramienta de corte, por medio del sensor de reposo y del de límite de carrera, decidiendo si conecta o desconecta la alimentación

El bloque sensor de límite de carrera, junto con el pulso de activación que envía la computadora CP/M, hacen funcionar el circuito de control de giro del motor de CD, que sube el motor propulsor para estacionarlo en su lugar de reposo, o lo saca de ahí bajándolo para que perforé y después lo regresa al lugar de reposo, según sea el caso.

Para controlar al motor propulsor con lógica TTL, se usa un circuito de potencia que proporciona la corriente demandada por el motor propulsor. La fig. 3.18 muestra el bloque del switch de alimentación y el de control de giro, que envían señales a los circuitos de potencia.

### 3.5 FUNCIONAMIENTO DE LA INTERFAZ HARB-C89

#### 3.5.1 Sensores de Posición

El sensor del lugar de reposo y el del límite de la carrera, están diseñados en base a sensores optoelectrónicos.

A continuación se muestra la estructura de uno de ellos, el otro es igual.

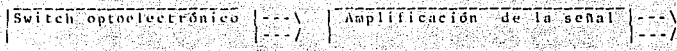
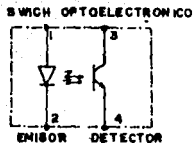
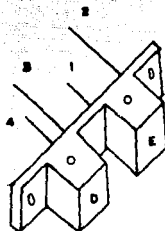


Fig. Estructura del sensor del lugar de reposo y el del límite de carrera

El switch optoelectrónico es un dispositivo que tiene LED-emisor de infrarrojos y un detector de luz infrarroja acoplados como se indica en la fig. 3.20



(a) LED-emisor y Detector



(b) Apariencia física

Fig. 3.20 Switch optoelectrónico

De la Fig. 3.21 el LED-emisor es alimentado con  $R_{10}$ , generalmente para un LED demanda una corriente de aproximadamente 20 mA, entonces  $R_{10} = 220 \text{ Ohms a } 1/4 \text{ W}$

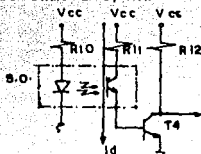


Fig. 3.21 polarización para el switch optoelectrónico

R11 en serie con el detector de infrarrojo, limita la corriente de base del transistor T4, que conduce al recibir la radiación infrarroja del LED hasta que no sea bloqueado el paso de la luz, fig. 3.20 (b); El cálculo de R11 toma en cuenta que la corriente máxima que puede conducir el detector es de 4 mA.

De la malla 1 en la fig. 3.21, tenemos:

$$V_{cc} = R11 \cdot I_d + V_{ce} + V_{be4} \quad \text{Si } V_{ce} = 0.2 \text{ V}, V_{be4} = 0.7 \text{ V}, I_d = 4 \text{ mA}$$

$$V_{cc} = R11 \cdot I_d + 0.9 \text{ V} \quad \Rightarrow R11 = 1025 \text{ Ohms.}$$

En la práctica se usó una resistencia de 1K Ohms a 1/4 W, el transistor es usado para amplificar la señal del switch, que es muy pequeña.

### 3.5.2 Circuito de Alimentación para el Motor Propulsor

Como se indica en la fig. 3.18 el sensor de lugar de reposo y el de límite de la carrera, envían señales al switch de la alimentación que está compuesto por un arreglo de puertas (IC2, IC4, IC5, fig. 3.22) que activan a las señales prioritarias PRESET y CLEAR del IC1, conectando la alimentación o desconectándola, según sea el caso.

Las puertas (IC2, IC4, IC5) envían señales al FF IC1 (A) bajo las siguientes condiciones que indican las posiciones posibles del motor propulsor, cuando ha transcurrido tiempo, de que ha sido conectada la alimentación al sistema.

- Motor propulsor en el lugar de reposo

El switch de lugar de reposo se encuentra desactivado, enviando un pulso alto a través del IC6, que pasa por la puerta AND-IC4 y por la puerta OR-IC2, mandando un pulso alto al PRESET del IC1 (A).

También el pulso alto se desvía hacia IC5, pasando por el IC2, desactivando el CLEAR del IC1 (A).

El CLOCK del IC1 (A) también es alimentado con un pulso ascendente, la alimentación no es conectada.

- Motor de propulsión a la mitad de su carrera

El sensor de lugar de reposo, envía un pulso bajo al IC6, pasando por IC4 y por IC2, que está abierta, el pulso bajo alimenta al PRESET del IC6, conectándose la alimentación, mientras que el CLEAR continúa desactivado

- Motor de propulsión al final de su carrera  
 El sensor de lugar de reposo sigue enviando un pulso bajo, manteniendo la alimentación conectada, como en el caso anterior.

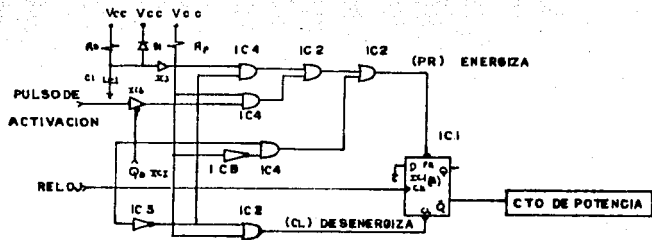


fig. 3.22 Cto que cumple con las condiciones anteriores

El circuito RESET es usado para que al inicializar la alimentación del sistema, se conecte la alimentación del motor propulsor, el circuito RESET se discutió en la sección 3.3

### 3.5.3 Control de Giro para el Motor Propulsor

El control de giro del motor propulsor permite bajar/subir la herramienta de corte. El control de giro del motor está dado por el FF (IC1 B), que recibe dos señales, un pulso enviado por la computadora CP/M y un nivel lógico, que manda el sensor de límite de la carrera por medio de un inversor (IC5) y de una puerta AND (IC4) como se muestra en la fig. 3.23

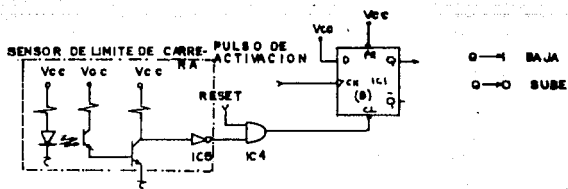


Fig. 3.23 Control de giro del motor propulsor



De la fig. 3.23, cuando se inicializa la alimentación el RESET bloquea la puerta IC4 activando el clear del FF IC1 (B), la salida Q baja hace subir al motor, el switch de alimentación para al motor propulsor al llegar al lugar de reposo.

El pulso de activación es enviada por la computadora CP/H cuando el switch de lugar de reposo está desactivado, y el de límite carrera está activado, situación en que el CLEAR del FF IC1 (B) está desactivado, permitiendo la entrada (conectado permanentemente a Vcc, fig. 3.23). La salida Q del IC1 es puesta a un nivel lógico alto, energizando al motor de forma que éste baje.

Al llegar a la parte más baja el motor propulsor, bloquea al sensor de límite de carrera, y si el detector no recibe luz de emisor, se desactiva mandando un 0 lógico al transistor T5, que pasa por la línea del inversor (IC5) y la puerta AND (IC4), activando el cambio de circulación de la corriente que pasa por los devanados del motor propulsor, que sube.

### 3.5.4 Circuito de Potencia para el Motor Propulsor

Generadas las señales que conmutan la alimentación y las que cambian el sentido de la corriente que circula por los devanados del motor propulsor, son enviadas a una etapa de potencia para que suministren la corriente necesaria para el motor propulsor, como se muestra en la fig. 3.24

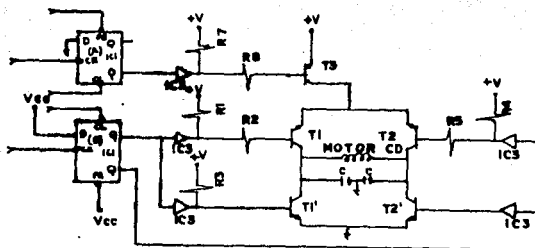


Fig. 3.24 Circuito de potencia para el motor propulsor (alimentación y control de giro).

Cálculo de las resistencias pull-up (R1, R4, R7) de los buffer's IC3 fig. 3.24.

- De la malla 1

$$16V = R7 \cdot I1 \quad \text{si } I1 = I_{ol}/2 \quad I_{ol} = 16 \text{ mA} \Rightarrow R7 = 16V/8 \text{ mA}$$
$$R7 = 2000 \text{ Ohms. Se usaron resistencias de } 2.3 \text{ k Ohms a } 1/4 \text{ de Watt}$$

Para el cálculo de las resistencias de polarización (R2, R3, R5, R6 y R8) se tomó en cuenta que el motor en el momento de subida, necesita mayor torque, que en el momento de estar bajando, en el último caso tiene la fuerza de gravedad en favor, mientras que en el otro caso la tiene a contra, por lo tanto se diseñó la etapa de potencia de forma que suministre la corriente necesaria para desarrollar el torque para lograr subir (mayor torque).

Si la corriente máxima disponible para el torque de subida es con 1 Amp de la fig. 3.24, malla 2, tenemos:

$$16V = V_{eb} + R8 \cdot I_b \quad \text{como } I_b = I_e/\text{Beta} \quad I_{\text{motor}} = 1 \text{ Amp, Beta} = 45$$

$$I_b = 1 \text{ Amp}/45 \quad I_b = 22.222 \text{ mA} \quad R_b = 15.3 \text{ V}/22.22 \text{ mA} \quad R8 = 680 \text{ Ohms}$$

Cálculo de la potencia disipada por la resistencia de 680 Ohms, si la corriente que demanda es de 70 mA

$$P = V \cdot I \quad V = 15.3 \text{ V} \quad I = 70 \text{ mA} \quad P = 15.3 \text{ V} \cdot 70 \text{ mA} \quad P = 1.07 \text{ W}$$

se usaron resistencias de 680 Ohms a 2 Watts

La fig. 3.25 muestra la estructura final de la interfaz HARB-C89 diseñada.

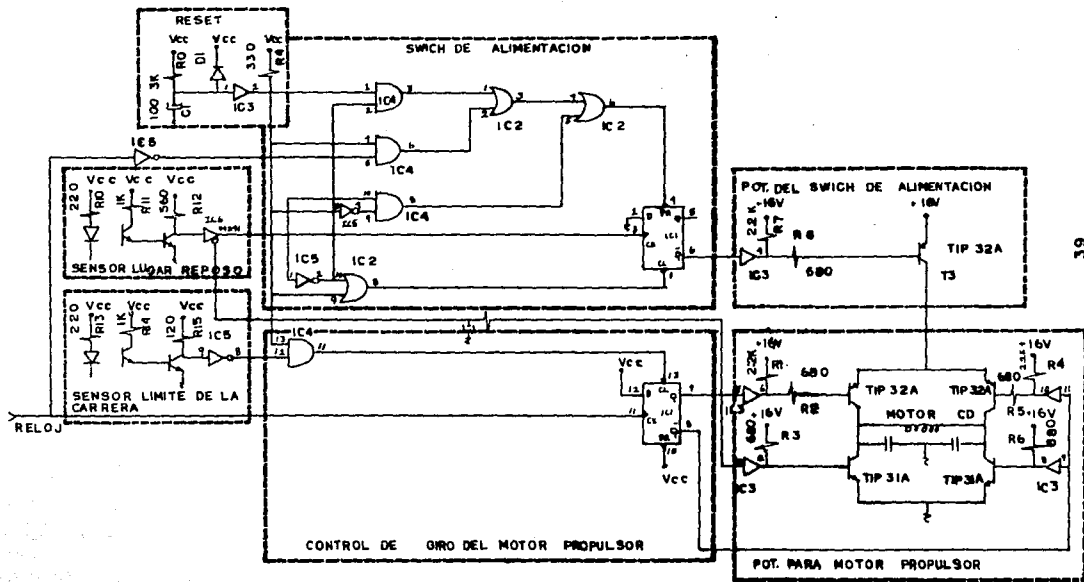


Fig. 3.28

ESTRUCTURA DE LA INTERFAZ HARD-C00

## CAPITULO 4

### PROGRAMACION DE CONTROL PARA MOTORES DE PASOS Y SISTEMA BARRENADOR

#### 4.1 SECCION PRELIMINAR

Para que nuestro sistema realice su función como barrenador, requiere además del conjunto mecatrónico, de un elemento vital : la programación de control. En el presente capítulo se analiza la programación desarrollada para este fin, con la cual una computadora con sistema operativo CP/M procesa la información suministrada por el usuario, y proporciona los resultados de dicho proceso a la circuitería, quien finalmente procesa las órdenes de posicionamiento y perforado haciéndolas llegar a los sistemas de posicionamiento lineal.

Cabe señalar que el hecho de diseñar por módulos un sistema, contribuye a minimizar tiempos muertos, para mantenimiento y reparación de averías. El conjunto mecatrónico diseñado se apegó a esta técnica, y la programación de control no ha sido la excepción.

##### 4.1.1 Del Papel de la Computadora

La computadora utilizada por nuestro sistema es la computadora "ZORBA" desarrollada por otro equipo de trabajo, tomando como base una tarjeta madre de Telcon Industries, Inc. . Cuenta con 64 kbytes de memoria, procesador Z-80 y sistema operativo CP/M versión 2.2 . CP/M (Control Program Monitor) es un sistema operativo en disco para microcomputadoras, producido por la compañía Digital Research (ver refs 11, 12, 14, 15 y apéndice).

Cuenta también con un monitor de 9 pulgadas con capacidad para 25 líneas de 80 caracteres. Cada carácter es elaborado con una matriz de 5 (ancho) por 7 (alto) puntos .

Posee además, dos unidades para disco flexible (5" 1/4) de 390 kbytes c/u y un teclado dividido en 3 áreas diferentes que son : teclado principal ; teclado de función y teclado de control (ver refs. 11 y 12) .

El papel de la computadora, en nuestro sistema y en general en cualquier sistema automatizado, interviene fundamentalmente en dos aspectos:

- Adaptación de las entradas (sensores y comandos) y de las salidas (actuadores o motores).
- Control general tanto de los movimientos, como de los cálculos y otros parámetros de trabajo, como retardos, mensajes al usuario, decisiones dentro del programa, toma de lecturas, etc.

En un sistema CAD/CAM la computadora se encarga del proceso -- miento de la información proveniente de sensores que proporcionan el estado del mundo exterior e incluso puede intervenir en coordinación con varias máquinas. En nuestro sistema, la computadora se ocupa de un solo tipo de sensores: los sensores de límite de carrera (ver cap. 3), sin la intervención directa de alguna otra computadora durante el proceso.

Los puertos E/S permiten al microprocesador tener contacto con el "mundo exterior". Cuando se trata de una comunicación asíncrona con dispositivos de entrada/salida deberá asegurarse que la comunicación tenga lugar cuando ambos comunicandos (computador y elemento de E/S) estén ya dispuestos. Esto puede realizarse ya sea mediante la técnica de Handshaking (la cual ha sido empleada en el presente trabajo) o bien a través de la técnica de interrupciones.

El proceso de Handshaking al efectuar la entrada es el siguiente:

- a) El microprocesador pregunta al componente E/S si este tiene un caracter válido.
- b) El componente E/S responde al microprocesador SI/NO

El microprocesador podrá leer un caracter válido del componente de entrada/salida únicamente cuando éste haya dado el SI correspondiente.

El proceso de Handshaking para efectuar la salida es:

- a) El microprocesador pregunta al componente E/S si está dispuesto a leer.
- b) El componente E/S responde SI/NO.

En el campo de la transmisión de datos en paralelo, las interfaces Centronics y MC68B21 son las más importates a ser mencionadas. La interface Centronics se utiliza frecuentemente en impresoras y se caracteriza por su sencilla manipulación. La transmisión se realiza generalmente bajo código ASCII de 8 bits.

La interfaz para el puerto IEEE (MC68B21 Peripheral Interface Adapter PIA) proporciona un método flexible para la conexión de periféricos al CPU. El PIA aunque relativamente complejo en su estructura permite al CPU manejar una amplia variedad de equipos con una lógica adicional mínima y una programación sencilla.

Para una descripción exacta tanto de la técnica de transmisión de datos como de las interfaces para la comunicación en paralelo empleados en este sistema véase el capítulo 5.

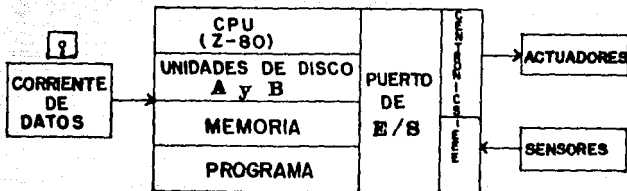


fig 4.1 -- El papel de la computadora CP/M en nuestro sistema.

#### 4.1.2 Del Lenguaje

Ya que el lenguaje es el medio que se utiliza para comunicarse con la computadora; una gran parte del rendimiento y productividad de las máquinas controladas por sistemas numéricos, radica en su buena programabilidad, fundamentada en el empleo de un lenguaje adecuado. La elaboración de la programación representa una parte de su explotación.

El lenguaje empleado para controlar las interfaces HARB-C88 y HARB-C89 con la computadora CP/M ha sido el lenguaje C.

C es un lenguaje de nivel mediano, ofreciendo posibilidades que resultan muy poderosas para los programadores profesionales. Es un lenguaje estructurado que permite modularizar en gran medida nuestro trabajo. Sus grandes virtudes lo han hecho una herramienta muy utilizada en la escritura de sistemas operativos y ha sido utilizado con el mismo éxito para escribir programas de control numérico, programas de procesamiento de textos y bases de datos.

Usualmente C es compilado y está basado en estructuras llamadas funciones. La función principal (o función raíz) es la función Main() que es la que siempre debe existir para que se pueda ejecutar un programa (ver refs. 16 , 17 , 18 y 20).

#### 4.1.3 Del Compilador.

El compilador utilizado es el MIX C versión 1.0 producido por MIX Software Inc. apegado a las normas de Kernighan y Ritchie, cuyo funcionamiento básico consiste en leer nuestro programa fuente (que es un archivo ASCII) línea por línea y generar para cada línea un código intermedio (todavía no binario) y guardarlo en otro archivo que todavía no es ejecutable.

El código intermedio es aceptado por un programa más; el LINKER, que nos proporciona un programa ejecutable (en binario) que puede ejecutarse muy rápidamente, contrario a lo que realiza un intérprete, ya que este último convierte las instrucciones del usuario a código máquina conforme va transcurriendo el programa, consumiendo un tiempo considerablemente mayor durante su ejecución. Por esta razón existen muy pocos intérpretes para C (ver refs. 18 , 19 y 20).

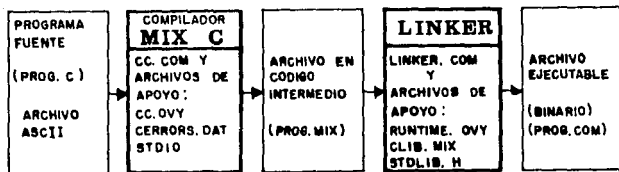


Fig.4.2 .- Funcionamiento del compilador MIX C .

#### 4.1.4 De la Palabra de Control

La organización de los bits en la palabra de control que se envía por el puerto CENTRONICS de la computadora CP/M es la siguiente :

nu nu nu s s s s s

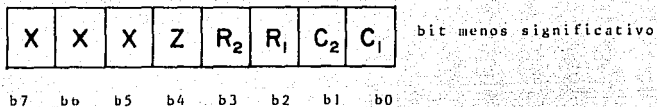


fig 4.3 .- Codificación de la palabra de control.  
s = salida, nu = no usado.

Como se observa en la figura anterior el bit b0 resulta ser una salida de la computadora y representa la señal de dirección del motor en el eje X (con un valor de 0 lógico avanza y con un 1 lógico retrocede).

El bit b1 representa la señal de dirección del motor en el eje Y, y es también una señal de salida.

El bit b2 es el bit de señal de reloj para el motor X, y el bit b3 lo es, para el motor Y; y ambas son salidas.

El bit b4 es el bit para la señal de activación del eje Z (sistema sube/baja), el cual se activa con el borde de subida del pulso que manda este bit.

Los tres últimos bits de la palabra de control no son utilizados, quedando disponibles para alguna otra posible aplicación en que pudieran ser ocupados.

Tanto el protocolo de comunicación utilizado por el puerto CENTRONICS y la configuración del conector correspondiente se ilustran en el siguiente capítulo.

#### 4.1.5 De la Palabra de Lectura

La organización de los bits en la palabra para la lectura de sensores de posición de los ejes realizada mediante el puerto de entrada/salida IEEE de la computadora CP/M es la que a continuación se muestra:



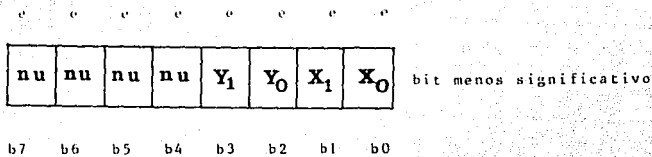


fig. 4.4 .- Codificación de la palabra de lectura .  
e=entrada   nu = no usado .

Donde los bits b0 y b1 indican que se ha alcanzado la posición mínima y máxima respectivamente sobre el eje X, al ser activados con un "1" lógico. Mientras que los bits b2 y b3 lo hacen para el eje Y.  
El puerto de entrada / salida utilizado es el IEEE configurado para entrada .

La configuración tanto del puerto IEEE como del conector correspondiente se muestran en el siguiente capítulo.

#### 4.1.6 De la Programación

La programación incorporada en este sistema ha sido categorizada en dos diferentes tipos de acuerdo a su función :

- De Barrenado o de Control (programa Barrenador)
- De Generación y Almacenamiento de datos para el barrenado que forma parte de la programación de comunicación (programa Generador).

En la sección siguiente se muestra únicamente el catálogo de funciones del programa Barrenador o de control . El programa Generador se expone en el siguiente capítulo.

## 4.2 CATALOGO DE FUNCIONES

### 4.2.1 El Programa Barrenador

En este programa se establece el tráfico de la información proveniente de los sensores de límite (através del puerto IEEE) y de la base de datos realizada por el programa Generador. Está constituido por un programa principal el cual ha sido modularizado u organizado en Funciones con tareas específicas.

El programa utiliza como puertos de salida de la computadora el puerto CENTRONICS y el puerto IEEE que son los dos puertos para la comunicación en paralelo (ver refs. 11,12,13 y cap. 5) con las dos interfaces HARB-C88 y HARB-C89 (ver cap. 3).

Las tareas a realizar por parte del programa principal son las siguientes :

- Reconocimiento del origen para preparar la actual sesión con la coordenada de inicio adecuada.
- Realizar el desplazamiento hacia el punto "Casa" que es el punto en que las variables de posicionamiento se inicializan con un valor de cero.

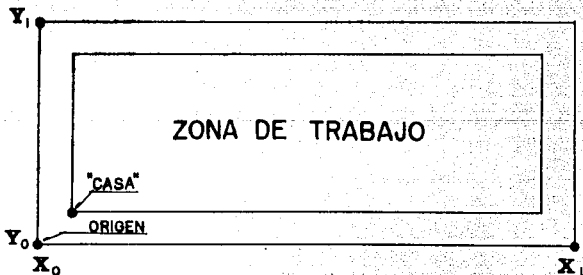


fig 4.5 .- El punto "Casa" como punto de referencia cero o punto de inicio.

- Una vez que es localizada la coordenada inicial (que como ya sabemos es el punto de partida), leer las coordenadas de los barrenos contenidas en la base de datos, para poder dar inicio a los movimientos de los ejes.
- Decidir el momento en que la herramienta deba bajar para hacer la perforación, una vez que los dos ejes X y Y han quedado en reposo, y cuándo estos ejes ya puedan continuar el movimiento hacia una nueva coordenada.
- Continuar con los demás barrenos hasta detectar el último y dar por terminada la sesión.

A continuación se presenta una explicación de las tareas realizadas por cada una de las funciones contenidas en el programa principal.

#### 4.2.2 Preparación de Lectura de Posición de Ejes

La función INI68B21 () inicializa el puerto IEEE para que pueda leer el estado de los sensores de límite colocados en cada uno de los ejes X y Y.

**FUNCION: INI68B21 ()**

**CATEGORIA:** De Barrenado

**TAREA:** Prepara a la computadora para leer el estado de los sensores, inicializando el chip HC 68B21 (ver ref, 24).

**FUNCIONES LLAMADAS:** OUTP () e INP () que son funciones que no forman parte del conjunto standard de C. Son proporcionadas por el compilador MIX C (ver ref. 18).

**DATOS ENTRADA :** No externos (0)

**DATOS SALIDA :** No externos (0)

**PARAMETROS DE ENTRADA :** Ninguno

**VALORES DE RETORNO :** Ninguno

#### 4.2.3 Lectura de Posición de Ejes

La función POS\_INIC () lee por el puerto IEEE el estado de los sensores quienes son obturados por sendas pestañas incorporadas en cada eje. Posteriormente evalúa la lectura y decide el movimiento de cada eje hacia el origen según convenga para finalmente realizar el reconocimiento de dicho punto.

##### FUNCION: POS\_INIC ()

CATEGORIA: De Barrenado

TAREA: Posiciona ambos ejes en el origen analizando la lectura de sensores y reconoce repetidamente el origen.

FUNCIONES LLAMADAS: HOME (lec), INP () (func. del compilador MIX C).

DATOS ENTRADA: Variable lec (representa la lectura en hexadecimal tomada directamente de los sensores).

DATOS SALIDA : Flujo de control a HARB-C88

PARAMETROS DE ENTRADA: Ninguno

VALORES DE RETORNO : Ninguno

##### FUNCION: HOME (lec)

CATEGORIA: De Barrenado

TAREA: Movimiento de ejes hacia el origen habiendo evaluado la lectura.

FUNCIONES LLAMADAS: AVANZAX (npas,dir1),  
AVANZAY (npas,dir2),  
AVANZAXY (npas,dir1,dir2)

DATOS ENTRADA : Ninguno

DATOS SALIDA : Flujo de control a HARB-C88

PARAMETROS ENTRADA: lec (lectura hexadecimal)

VALORES DE RETORNO: Ninguno

#### 4.2.4 Control del Posicionamiento y Barrenado

La función COMAND () comanda y controla el número de pasos y la dirección de los motores de los ejes X y Y, mandando la orden de activación de la herramienta una vez obtenida la coordenada deseada. Inicia su tarea accediendo a datos por pares coordinados provenientes de la Base generada con anterioridad a la cual se ha direccionado el programa Barrenador, aceptando solamente valores positivos de dichos datos.

Cuando los motores necesitan idéntico número de pasos para llegar a una coordenada específica se dice que ningún eje demanda compensación. Si se requiere compensar alguno de los ejes, esta función compensa al eje en cuestión con el número de pasos necesarios, reteniendo al otro eje mientras termina la compensación.

COMAND () también pide el estado de los sensores de límite para decidir si continúa enviando órdenes a las interfaces o si se ha detectado algún problema en los límites y se tenga que reiniciar la sesión, mandando un mensaje de error al usuario.

#### **FUNCION: COMAND ()**

**CATEGORIA:** De Barrenado

**TAREA:** Comanda los movimientos de los ejes de acuerdo a los datos de entrada.

**FUNCIONES LLAMADAS:** BDOS (5,arg) (llamada al sist. op., ver Apéndice), INP () (función del compilador, ver ref. 18), y DEC\_AVAN(dir1,dir2,x2,y2)

**DATOS ENTRADA:** xl,y1,lec (coordenadas absolutas obtenidas de la base de datos, y estado de los sensores respectivamente).

**DATOS SALIDA:** Flujo de control hacia HARB-C89

**PARAMETROS DE ENTRADA :** Ninguno

**VALORES DE RETORNO :** Ninguno

FUNCION: DEC\_AVAN (dir1,dir2,x2,y2)

CATEGORIA: De Barrenado

TAREA: Decide que eje necesita compensación en el avance y compensa adecuadamente.

FUNCIONES LLAMADAS: AVAZAX (npas,dir1,dir2),  
AVANZAX (npas,dir1) ,  
AVANZAY (npas,dir2) .

DATOS ENTRADA : Ninguno

DATOS SALIDA : Flujo de control a HARB-C88

PARAMETROS DE ENTRADA: Sentidos de movimiento (dir1,dir2), x2, y2 (que son las diferencias entre coordenadas nuevas y coordenadas anteriores).

VALORES DE RETORNO : Ninguno

FUNCION: AVANZAX (npas,dir1)

CATEGORIA: De Barrenado

TAREA: Avanza el eje X con la dirección y número de pasos necesarios.

FUNCION LLAMADA : SALE (byte,sube,baja,npas)

DATOS ENTRADA : Ninguno

DATOS SALIDA : Flujo de control a HARB-C88

PARAMETROS DE ENTRADA: npas (distancia por avanzar), dir1 (sentido sobre el eje X).

VALORES DE RETORNO : Ninguno

**FUNCION: AVANZAY (npas,dir2)**

**CATEGORIA: De Barrenado**

**TAREA: Hace avanzar al eje Y con la dirección y número de pasos necesarios .**

**FUNCION LLAMADA : SALE (byte,sube,baja,npas)**

**DATOS ENTRADA: Ninguno**

**DATOS SALIDA : Flujo de control a HARB-C88**

**PARAMETROS DE ENTRADA: npas ( distancia a recorrer), y dir2 (sentido de avance en Y) .**

**VALORES DE RETORNO : Ninguno**

**FUNCION: AVANZAXY (npas,dir1,dir2)**

**CATEGORIA: De Barrenado**

**TAREA: Hace avanzar a los dos ejes X y Y con la dirección y número de pasos necesarios .**

**FUNCION LLAMADA : SALE (byte,sube,baja,npas)**

**DATOS ENTRADA : Ninguno**

**DATOS SALIDA : Flujo de control a HARB-C88**

**PARAMETROS DE ENTRADA: npas (distancia a recorrer en ambos ejes), dir1 sobre dir2 (sentidos sobre los ejes X , y Y) .**

**VALORES DE RETORNO : Ninguno**

**FUNCION:** SALE (byte, sube, baja, npas)

**CATEGORIA:** De Barrenado

**TAREA:** Manda señales de reloj y dirección para avance de motores através de la palabra de control (por el puerto CENTRONICS de CP/M) . Tambien pide el estado de los sensores de límite de los ejes para continuar .

**FUNCIONES LLAMADAS:** BDOS (5, arg) (LLamada al sis. op., ver Apéndice) e INP () (func. que no forma parte del conjunto standard de C. Es proporcionada por el compilador MIX C) .

**DATOS ENTRADA:** c (estado de los sensores).

**DATOS SALIDA :** La variable byte (palabra de control enviada hacia el puerto CENTRONICS), como flujo de control .

**PARAMETROS DE ENTRADA:** sube, baja (variables para enmascaramiento de byte), byte, y npas

**VALORES DE RETORNO :** Ninguno

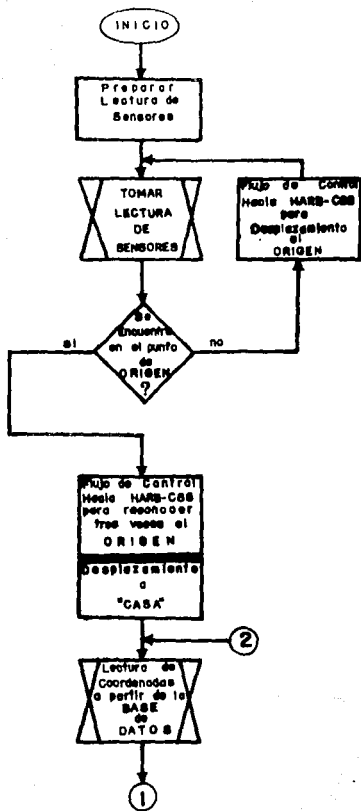
### 4.3 EL ARCHIVO FUENTE

En la presente sección se muestra el archivo fuente del Software de control implementado : El programa Barrenador (Morgan.c) y su correspondiente descripción mediante un sencillo diagrama de bloques.

#### 4.3.1 Descripción

El siguiente diagrama ilustra el conjunto de actividades realizadas por la programación de control. La secuencia de trabajo es la indicada siguiendo las sagítas.





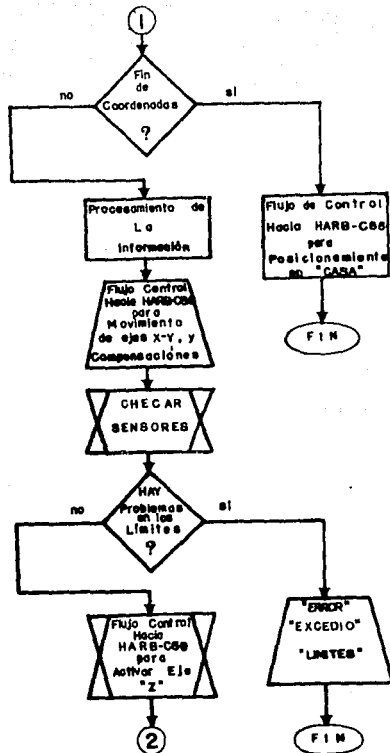


fig. 4.6 .- Secuencia de actividades de la programación de control.

#### 4.3.2 EL Programa Barrenador Morgan.c

El grupo total de funciones que definen y calculan en el programa las acciones de control ha sido confeccionado de tal manera que cada función se basa en una descripción de la tarea específica a realizar por nuestro sistema y mediante la cual se toman las decisiones correspondientes .

El archivo fuente del programa Barrenador es el siguiente:  
MORGAN.C

```
#include <stdio>
#define DRA 0x60 /* PIA Data Register A */
#define CRA 0x61 /* PIA Comand Register A */

int c, cont, x, x2, y, y2;

Main ()
{
    int byte, sube, baja, npas;
    byte=0x0c; sube=0x0c; baja=0x03; npas=20;

    IN168B21 ();
    POS_INIC ();
    SALE (byte, sube, baja, npas);
    COMAND ();
}

IN168B21 () /* Función que prepara a la compu */
{ /* tadora para leer desde el puer */
    c=0; /* to IEEE . */
    OUTP (CRA, c); /* DRA seleccionado */
    c=0;
    OUTP (DRA, c); /* Puerto A como entradas */
    c=0x24; /* ORA seleccionado */
    OUTP (CRA, c);
    c=INP (DRA);
    c=INP (DRA);
}

POS_INIC () /* Posicionamiento y reconocimiento */
{ /* del origen */
    int lec;
    cont=0; lec=0x00;

    while (cont < 3)
    {
        lec=INP (DRA);
        if (lec == 0xf3)
        {
            HOME (lec);
            cont ++ ;
        }
        else
        { HOME (lec) ; }
    }
}
```

```

)
HOME (lec) /* Función que mueve los ejes */
           /* hacia el ORIGEN habiendo eva - */
           /* ludo la lectura de los senso- */
           /* res de limite de carrera. */

int lec;

{
int npas, dir1, dir2;

switch (lec)
(
case 0xf0 :
    npas=4;   dir1=1;   dir2=1;
    AVANZAXY (npas,dir1,dir2);
    break;
case 0xf1 :
    npas=4;   dir2=1;
    AVANZAY (npas,dir2);
    break;
case 0xf2 :
    npas=4;   dir1=1;
    AVANZAX (npas,dir1);
    break;
case 0xf3 :
    npas=20;  dir1=0;   dir2=0;
    AVANZAXY (npas,dir1,dir2);
    break;
case 0xf4 :
    npas=20;  dir1=1;
    AVANZAX (npas,dir1);
    break;
case 0xf6 :
    npas=20;  dir1=1;
    AVANZAX (npas,dir1);
    break;
case 0xf8 :
    npas=20;  dir2=1;
    AVANZAY (npas,dir2);
    break;
case 0xf9 :
    npas=20;  dir2=1;
    AVANZAY (npas,dir2);
    break;
case 0xfc :
    npas=20;  dir1=1;   dir2=1;
    AVANZAXY (npas,dir1,dir2);
    break;

default :
    printf (" error \n ");
    break;

```

```

    }
    COMAND ()

```

```

/* Esta función comanda los*/
/* movimientos de los ejes */
/* según datos de entrada */
/* Bits en uso: */
/*      B0 Sentido x */
/*      B1 Sentido y */
/*      B2 Reloj x */
/*      B3 Reloj y */
/*      B4 Taladro */

while (x1 >= 0)
(
    scanf ("%d %d" , &x1, &y1);
    if (x1 >= 0 | y1 >= 0)
    (
        if (x1 >= x)
        (
            if (x1 != x)
            (
                x2 = (x1-x) * 15.39 ; dir1=0;
            )
            else (x2=0; dir1=0;)
        )
        else { x2 = (x-x1) * 15.39 ; dir1=1; }
        if (y1 >= y)
        (
            if (y1 != y)
            (
                y2 = (y1-y) * 22.22 ; dir2=0 )
            else { y2=0; dir2=0; }
        )
        else { y2 = (y-y1) * 22.22 ; dir2=1; }
        DEC_AVAN (dir1,dir2,x2,y2);
        if (lec=INP (DRA) == 0xf0)
        (
            for (j=0; j<50; j ++);
            BDOS (5,0x10);
            for (j=0; j<2; j ++);
            500/300; 200/333;
            BDOS (5,0x00);
            for (j=0; j<5000; j ++);
        )
        else (goto error;)
        x=x1; y=y1;
    )
)
error : printf("error, excedió límites");

```

```

    }
    DEC_AVAN (dir1,dir2,x2,y2) /* Decide que eje necesita compensa */
    int dir1, dir2; /* ción en el avance y lo compensa */
    {
        int npas;
        if (x2 <= y2)
        {
            npas=x2;
            AVANZAXY (npas,dir1,dir2);
            npas=y2-x2;
            AVANZAY (npas,dir2);
        }
        else
        {
            npas=y2;
            AVANZAXY (npas,dir1,dir2);
            npas = x2-y2 ;
            AVANZAX (npas,dir1);
        }
    }
}

```

```

AVANZAXY (npas,dir1,dir2) /* Ejecuta el avance de los ejes */
int npas,dir1,dir2 ; /* X y Y */
{
    int byte,sube,baja ;
    baja=0x03; sube=0x0c;

    if (dir1 == 0)
    {
        if (dir2 == 0)
            ( byte=0x0c; )
        else ( byte=0x0e; )
    }
    else
    {
        if (dir2 == 0)
            byte=0x0d;
        else byte=0x0f;
    }
    SALE (byte,sube,baja,npas);
}

```

```

AVANZAX (npas,dir1) /* Ejecuta el avance del eje X */
int npas,dir1; /* con la dirección y cantidad de */
/* pasos necesarios para el motor */
{
    int byte,sube,baja;
    sube=0x04; baja=0x03;

    if (dir1 == 0)
        byte=0x04;
    else byte=0x05;
    SALE (byte,sube,baja,npas);
}

```

```
    }  
    AVANZAY (npas,dir2)
```

```
    int npas,dir2; /* Ejecuta el avance del eje Y */  
                  /* con la dirección y cantidad de */  
                  /* pasos necesarios para el motor */
```

```
    {  
        int byte,sube,baja;  
        sube=0x08; baja=0x07;  
        if (dir2 == 0)  
            byte=0x08;  
        else byte=0x0a;  
        SALE (byte,sube,baja,npas);  
    }
```

```
    SALE (byte,sube,baja,npas)
```

```
    int byte,sube,baja,npas; /* Función que manda señales de */  
                              /* reloj y dirección para avance */  
                              /* de los motores auxiliada por la */  
                              /* variable byte,através del puer */  
                              /* to CENTRONICS. También pide el */  
                              /* estado de los sensores */  
    {  
        int i;  
        for (i=1; i<=npas; i++)
```

```
        {  
            BDOS (5,byte);  
            byte = byte & baja;  
            printf (" "); printf (" ");  
            BDOS (5,byte);  
            byte = byte | sube;  
            printf (" ");  
            c=INP (DRA);
```

```
            if (c != 0xf0 & cont>2)  
                {  
                    x2=y2=0;  
                    break;  
                }  
        }
```

```
    }
```

COMUNICACION DE DATOS ENTRE EL SISTEMA BARRENADOR  
Y LA MAQUINA CAD

5.1 EL PROCESO GENERAL DEL DISEÑO

Para llevar a cabo cualquier diseño, existen 6 pasos a seguir:

- 1.- Reconocer la existencia de una necesidad en un proceso, producto o máquina, por la cual deba aplicarse una operación correctiva.
- 2.- Definición del problema. Especificación completa de los componentes a diseñar, como son características físicas y de funcionalidad, costo, calidad y especificaciones de operación.
- 3.- Síntesis, Análisis y optimización. Estos pasos se encuentran íntimamente relacionadas, para la obtención de un sistema óptimo, de acuerdo con un criterio determinado. Es necesario la aplicación repetitiva de algún método de análisis. Esta forma de diseño consiste en crear un sistema de partida y analizarlo, en función de los resultados obtenidos y de los buscados, hasta que se obtenga una solución satisfactoria.
- 5.- Evaluación. Aquí se compara el diseño con las especificaciones establecidas.
- 6.- Presentación del diseño. Incluye la documentación, con dibujos, esquemas y especificaciones de los materiales ensamblado, etc.



fig. 5.1 El proceso general del diseño



## 5.2 El CAD

CAD (Computer Aided Design) es una técnica por medio de la cual las descripciones geométricas de componentes físicos pueden ser creadas y almacenadas en forma de modelos matemáticos, dentro de bases de datos. Puede definirse como el uso de sistemas de cómputo en apoyo a la creación, análisis, modificación y optimización de un diseño. El software de apoyo para sistemas CAD consiste en programas para implementar gráficos y programas que facilitan las funciones de ingeniería (ver ref. 25).

En otras palabras la combinación de hardware y software de computadora facilita la construcción de modelos y su análisis, permitiendo además una amplia variedad de representaciones visuales.

El CAD tiene 4 áreas que se relacionan con el proceso del diseño:

- a.- Modelo geométrico. Paso de síntesis, en el cual el diseño se toma desde el sistema de gráficos de la computadora para crear y transformar datos en forma de dibujos y símbolos.
- b.- Análisis de ingeniería. En esta área los diseños son analizados por métodos de optimización.
- c.- Revisión del diseño y evaluación.
- d.- Procedimiento de conversión de los datos de la imagen residentes en la memoria de la computadora en un documento físico.



- III Modelaje geométrico
- IV Analisis Ingenieril
- V Revisión del diseño y evaluación
- VI Dibujo automatizado.

fig. 5.2 Papel del CAD en el proceso del Diseño

Otra alternativa en el uso de sistemas CAD es la posibilidad de poder desarrollar una base de datos para la manufactura de un producto.

Entre los beneficios del CAD se encuentran:

- Incremento de la productividad en el área de diseño.
- Análisis de diseño, con lo cual los diseños se acercan más a su nivel óptimo. Un sistema CAD cuenta con rutinas de análisis de diseño.
- Errores de diseño menos frecuentes.
- Mayor precisión en los cálculos de diseño.
- Los dibujos pueden ser más comprensibles.

### 5.2.1 AutoCAD

Los paquetes para sistemas CAD (paquetes CADD) emplean todos los elementos básicos para el diseño (puntos, líneas rectas, líneas paralelas, líneas curvas, multicurvas, figuras geométricas, círculos, medios arcos, parábolas, elipses, etc.) para que el usuario pueda elaborar sus diseños en la pantalla de la computadora.

Estos programas le permiten al diseñador especificar las coordenadas exactas de cada punto del objeto que está desarrollando, ya sea con la ayuda del cursor, con el ratón, o con el digitalizador. Algunos también permiten especificar otro conjunto de coordenadas relativas a otra sección de la pantalla. Así mismo es posible usar coordenadas polares, en las cuales es necesario especificar la longitud de una línea y su ángulo.

En cada diseño se establece la escala del dibujo, para posteriormente transformar el modelo diseñado, en algo que tenga medidas reales.

AutoCAD es uno de los programas desarrollados para el sistema operativo MS-DOS y es el paquete escogido para obtener una base de datos primaria, que posteriormente será filtrada por el programa de Generación (Mask.C) desarrollado en el presente trabajo. AutoCAD permite el dibujo y diseño por computadora, para diversas áreas (que en nuestro caso, es el diseño de circuitos impresos), ofreciendo ciertas ventajas de operación.

Como AutoCAD almacena la información sobreponiendo "subpartes" llamadas Layers, para analizar cada una de ellas por separado. La base de datos primaria solamente contendrá el Layer de puntos (referente a las coordenadas de los barrenos) a la escala requerida. Es decir, el archivo creado en AutoCAD puede contener sin representar un estorbo, toda la información de un circuito electrónico completo, y a partir de él, utilizar solamente la información que necesita la programación de control.

La información que contiene el layer de puntos deberá estar en escala 1:1. Este layer manda crear un archivo XXX.PLT (información para el plotter), que contiene la posición de inicio, las posiciones de todas las coordenadas, cuándo debe subir y bajar la plumilla del plotter y el fin del archivo para que el plotter se desactive (ver ref. 23). La versión de AutoCAD empleada para este sistema fue la versión 2.3.

### 5.3 CAD/CAM

CAM (Computer Aided Manufacturing) se define como el uso de sistemas de cómputo para planear, dirigir y controlar las operaciones de manufactura a través de una interfaz conectada a tal proceso (ver ref. 25).

Existen 2 formas para implementar el CAM :

1.- CAM mediante monitoreo :

Relaciona la interfaz de la computadora con el proceso de manufactura para observar el proceso, el equipo asociado y la recopilación de datos del proceso.

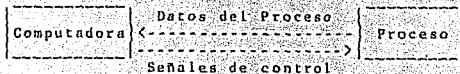
El control queda en manos del ser humano y puede guiarse mediante la información obtenida a través de la computadora.

2.- CAM mediante la vía del control

Este va más allá de un simple monitoreo, hasta el grado de controlar la totalidad del proceso basándose en las observaciones.



MONITOREO



CONTROL

fig. 5.3 Aspectos fundamentales del CAM

Algunas aplicaciones del CAM para soporte de manufactura son:

- Programación de control numérico para máquinas-herramientas automáticas.
- Planeación del proceso automático, en el que la computadora prepara una secuencia para procesar un producto particular o componente.
- Determinación del tiempo standard para una operación de producción particular.
- En control de calidad los sistemas CAM son usados también en la inspección, y pruebas de componentes.

Entre los beneficios del CAM se encuentran :

- Diseño y optimización de herramientas para la manufactura
- Programación modular del control numérico
- Planeación de procesos
- Ensamblado de partes
- Inspección por computadora

La tecnología CAD/CAM se refiere al uso de computadoras digitales que presentan diversas funciones dentro del diseño y la producción. En un sistema CAD/CAM se establece una relación directa entre el diseño y la construcción.

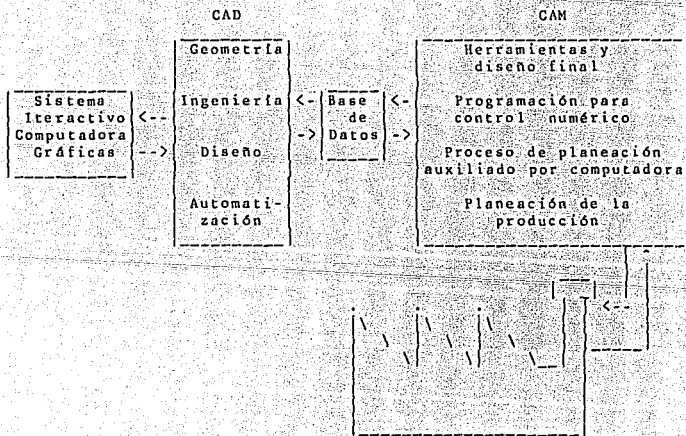


Fig. 5.4 Relación Diseño-Construcción en sistemas CAD/CAM

## 5.4 COMUNICACION DE DATOS DEL SISTEMA CAD/CAM

### 5.4.1 Programa intermedio de comunicación

Media Master (MM) es un programa comercial de Intersecting Concepts, Inc., que permite copiar, borrar, leer o imprimir el contenido de discos flexibles formateados en sis. op. MS-DOS, desde o hacia discos flexibles formateados en sis. op. CP/M. La función principal de MM en nuestro sistema, es la de obtener una copia del archivo que manda crear AutoCAD, con la información que utiliza el plotter (archivo XXX.PLT).

Esta copia se realiza desde un disco formateado en MS-DOS hacia otro disco formateado para CP/M, cuya información referente al Layer de "puntos" únicamente, contiene las coordenadas de los barrenos. Sin embargo, un archivo así (archivo Madre), todavía no es utilizable directamente por el programa del Barrenador (Morgan.C), es necesario procesarlo en la computadora ZORBA (descrita anteriormente), mediante el programa de Generación con el fin de obtener una base de datos útil para la programación de Control (Programa Morgan.C).

Dicho proceso parte en la detección de los movimientos absolutos que realizaría el plotter. El almacenamiento se realiza mediante direccionamiento, en el momento de ejecutar el programa Generador :

```
MASK < XXX.PLT > XXX  
MORGAN < XXX
```

### 5.4.2 Programa Generador

El programa Generador (Mask.C) realiza la base de datos que requiere el programa Barrenador (Morgan.C). La base contiene las coordenadas X - Y de todos y cada uno de los barrenos y es directamente accesable por Morgan.C. La generación se basa en el filtrado o enmascaramiento del archivo Madre (cuya extensión es .PLT) para que el programa Barrenador realice sus funciones adecuadamente (ver ref.23).

Cualquier información "basura", contenida en el archivo Madre es ignorada, dejando pasar solamente las coordenadas por pares. El resultado de dicho enmascaramiento se almacena finalmente en una base de datos (sin extensión), que es a la cual se direcciona al programa Barrenador en el momento de su ejecución.

FUNCIÓN : Main ( )

CATEGORÍA: De Generación

TAREA : Reconocer todos los movimientos absolutos del plotter hacia las coordenadas X-Y para guardarlas en una base de datos .

FUNCIONES LLAMADAS : Ninguna

DATOS DE ENTRADA : Información proveniente del archivo madre.

DATOS DE SALIDA : Coordenadas por pares (separadas por espacios en blanco conteniendo como fin de archivo los valores : -1 -1 ) almacenadas en una base de datos.

PARAMETROS DE ENTRADA : Ninguno

VALORES DE RETORNO : Ninguno

El archivo fuente para el programa Generador es el siguiente:

MASK.C

```
#include <stdio>
main ()
{ int i, c;
  while ( (c = getchar ()) != 'S' )
  { if (c == 'M')
    { for (i=1; i=1; i++)
      { c = getchar ();
        if (c >= '0' & c <= '9' )
          { putchar (c); }
        else if (c == ';' )
          { break; }
        else if (c == ',' )
          { printf (" "); }
        else
          { printf (" "); }
      }
    }
  }
}
```

### 5.4.3 Protocolo de Comunicación

Para enviar las señales necesarias a la interfaz HARB-C88 se utilizó el puerto CENTRONICS cuyo protocolo de comunicación empleado es el denominado de "Handshaking", el cual hace uso de tres señales:

\*STROBE  
\*ACKNOWLEDGE  
BUSY

Las señales marcadas con asterisco son activadas con bordes de bajada.

El "Handshaking" para un puerto de salida se lleva a cabo de la siguiente forma:

La computadora pone una palabra dato en el bus de datos, un poco después activa la señal STROBE, indicando al periférico que existe un dato válido en el bus de datos. Posteriormente la señal BUSY es activada por parte del dispositivo periférico, lo que indica a la computadora que se está procesando dicho dato, y que no debe enviar otro dato a procesar.

Cuando se ha procesado completamente el dato el dispositivo periférico desactiva la señal BUSY e inmediatamente activa durante un pequeño lapso de tiempo la señal ACKNOWLEDGE (ACK) con lo que indica a la computadora que puede procesar otro dato.

La siguiente fig. muestra las relaciones entre las señales antes mencionadas.



Fig. 5.2.4 Señales activadas durante el "Handshaking".

En el caso de un puerto de entrada, el periférico debe informar con una línea cuándo la información transportada es

disponible para la computadora. Esta línea es leída por la computadora en un puerto de entrada separado, conocido como puerto de estatus.

En un puerto de salida la línea de STROBE, es enviada automáticamente por el hardware de la computadora al puerto cada vez que escribe un dato en él (ver refs. 9, y 13) .

#### 5.4.4 Configuración de Puertos.

##### Interfaz de salida paralelo

La computadora envía 8 bits a la interaz HARB-C88 de los cuales sólo se ocupan 5 bits:

B0 control 1  
B1 control 2  
B2 reloj 1  
B3 reloj 2  
B4 activa taladro

El conector de la computadora para el puerto CENTRONICS es un conector DB25 hembra con la siguiente configuración:

Número de contacto	Función
1	GROUND
2	Transmit data output to printer
3	Receive data input from printer
4	Request to send output to printer
5	Clear to send input from printer
6	Not used
7	Ground
8	Not used
9-12	Ground
13	Vcc
14-15	Ground
16	CB2
17	CB1
18	PB7
19	PB6
20	PB5
21	PB4
22	PB3
23	PB2
24	PB1
25	PB0



### Interfaz de entrada paralelo

Para leer los sensores de posición extrema se utilizó un puerto IEEE (Peripheral Interface Adapter) de Motorola.

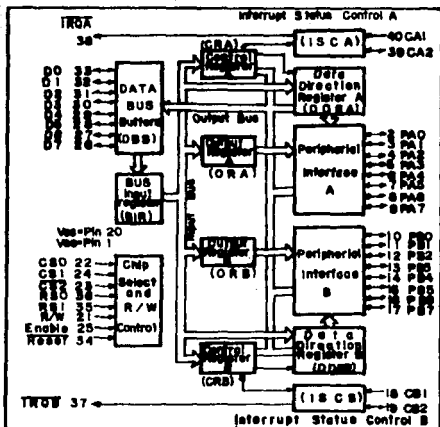
Este puerto solo hace uso de 4 bits.

B0	Posición mínima en el eje X
B1	Posición máxima en el eje X
B2	Posición mínima en el eje Y
B3	Posición máxima en el eje Y

Internamente, el PIA es dividido en dos configuraciones de registro independientes pero simétricas. Cada parte tiene tres características principales: un registro de salida, un registro de control, y un registro de dirección de datos. El CPU trata estos registros como localidades de memoria y puede escribir o leer tanto en un como en el otro.

El registro de dirección de datos (DDR) y el registro de salida (ORA u ORB) de cada configuración representan una localidad de memoria para el CPU. La selección de uno de ellos se hace internamente en el PIA mediante un bit en el Registro de Control (CRA o CRB).

Los DDR son utilizados para establecer el funcionamiento como entrada o como salida de cada configuración de registro. Esto es llevado a cabo mediante la escritura de unos o ceros por el CPU en los 8 bits del DDR, los 8 bits puestos en ceros causan que la correspondiente configuración de registro funcione como entrada, y puestos en "unos", que funcione como salida (ver ref. 24).



El conector del puerto IEEE de la computadora tiene la siguiente configuración :

Número de contacto	Función
1	Service request
2	Attention
3	Interface clear
4	Not ready for data
5	Data I/O D7
6	D6
7	D5
8	D4
9	D3
10	D2
11	D1
12	D0
13	Not used
14	Data valid
15	Remote enable
16	Not data accepted
17	End or identify
18-25	Ground

La función en lenguaje C, llamada INI\_68B21 que forma parte de la programación de control se encarga de inicializar al puerto IEEE (ver cap. 4) .

#### 5.4.5 Procedimiento General de Trabajo

- Diseñar en AutoCAD (sis.op. MS-DOS) ya sea un circuito electrónico completo en uno de cuyos Layers se proporcionen las coordenadas exactas de los barrenos (Layer de puntos o base de datos primaria), o un único Layer con las coordenadas necesarias, y salvarlo. La información que contenga el Layer de puntos deberá estar en escala 1:1 .
- Con ayuda del programa intermedio de comunicación M M (Media Master, sis. op. MS-DOS) formatear un disco para sistema operativo CP/M ZORBA 40 Tracks y en ese disco hacer una copia del archivo que contiene la información para el plotter (archivo con extensión PLT ó archivo Madre). El archivo Madre es creado automáticamente por el Layer de puntos .
- Poner en marcha el programa Generador MASK.C para filtrar la información y crear la base de datos con la información que ya podrá ser entendida por la programación de Control .

Teclear :

```
(MASK) (<) (nombre.PLT) (>) (nombre)
      programa      archivo Madre      Base de
      Generador      Datos
```

- Colocar la tarjeta de circuito impreso a barrenar y encender el sistema barrenador .
- Ejecutar el programa de Control MORGAN.C (sis. op.CP/M) direccionándole la Base de Datos obtenida en el punto anterior .

Teclear :

```
(MORGAN) (<) (nombre)
      programa      Base de
      de Control    Datos
```

## CONCLUSIONES

La experiencia y los conocimientos adquiridos en la realización de este trabajo, nos permiten visualizar algunas modificaciones.

EL proyecto aquí realizado es un Sistema CAD/CAM básico, el cual está sujeto a mejoras entre las cuales podemos citar:

Un sistema más flexible;

- En cuanto a la selección de diferentes herramientas (brocas, seguetas, fresas o cortadores, electrodos).
- En la selección de diferentes bases de datos.
- En la selección de la trayectoria a seguir para un trabajo específico.
- En cuanto al uso de un sistema de cómputo MS-DOS
- En cuanto a la posibilidad de variar la velocidad de trabajo del barrenador.

Estamos seguros que el proyecto realizado será de gran utilidad para las personas interesadas en la fabricación de circuitos impresos.

Con algunas modificaciones el presente proyecto, podrá ser utilizado como un plotter, una fresa para contornos, una punteadora, por mencionar algunas aplicaciones.

## APENDICE

### SISTENA OPERATIVO CP/M

CP/M (Control Program Monitor) es un sistema operativo almacenado en disco para microcomputadoras producido por Digital Research.

Una parte de CP/M permanece continuamente en monitoreo de actividades mientras el sistema esté cargado, y es llamado MONITOR. El MONITOR consiste en tres programas; dos de ellos son: BDOS (Basic Disk Operating System, para manejo de archivos y programas), y BIOS (Basic I/O System, para la transmisión de datos a puertos), que están al servicio de otros programas.

El tercero es el CCP (Console Command Processor), que está al servicio del usuario através de las órdenes y se encarga del acceso a discos. CCP es el módulo con el que el usuario tiene su primer contacto con CP/M y es el responsable de leer e interpretar los comandos que el usuario teclea.

#### Función BDOS (5, argumento) de CP/M

Se pide a través de BDOS, el conjunto de llamadas para comunicarnos con el Hardware de la computadora desde el programa de aplicación, escogiendo de entre ellas a la llamada 5. La llamada 5 del sistema operativo envía lo que se tiene en el argumento de la función al puerto paralelo (en este caso CENTRONICS) de la impresora (b7 - b0).

Esta forma de trabajar por parte del programa de aplicación tiene la ventaja de que como se recurre directamente con el hardware los procesos requeridos se vuelven muy rápidos. (Para mayor detalle ver ref. 15).

Todas las versiones de CP/M tienen dos tipos de órdenes :

- órdenes permanentes
- órdenes transitorias

Los programas cortos que ejecutan las órdenes permanentes siempre están en memoria. Las órdenes permanentes son ejecutadas inmediatamente por CP/M sin consultar ninguna instrucción del disco. Los programas que ejecutan órdenes o comandos transitorios no son cargados automáticamente en memoria cuando CP/M es arrancado. Una orden transitoria provoca que CP/M acceda a un programa del disco ; lo cargue en la memoria de la computadora y lo ejecute. Es decir, requieren que un conjunto de instrucciones almacenadas en el disco sean cargadas en la memoria para poder ser ejecutados.

#### Resumen De Ordenes Permanentes

- DIR           mostrar directorio del disco
- TYPE         visualizar archivo en ASCII
- ERA          borrar archivo
- REN          renombrar archivo
- SAVE         guardar contenido de memoria en un archivo en disco
- D:           cambio de unidad de disco implícita
- USER        cambio de usuario actualmente registrado

#### Ordenes De Edición De Línea Interpretadas Inmediatamente :

- ^C           reinicializar CP/M
- ^E           continuación del mecanografiado en la siguiente línea
- ^H           suprimir el último caracter (back space)
- ^J           line feed
- ^M           carriage return
- ^P           asignación de salida para impresora
- ^R           repetir línea de orden actual
- ^S           pausa de la pantalla
- ^U           cancelar orden actual
- ^Y           cancelar orden actual

#### Algunas Ordenes Transitorias en CP/M

- PIP          transporte de datos de un dispositivo a otro
- DDT          inicialización para operación de debug
- DUMP         visualizar en pantalla un archivo en formato hexadecimal
- STAT         proporcionar información y control de discos y/o archivos

## BIBLIOGRAFIA

- 1 Stepper Motors Handbook  
North American Philips      Airpax Corp.      1982
  
- 2 Electromagnetic Resonance in Permanent Magnet Stepper Motors  
Proc. 2nd. Int. Conf. on Stepping Motors and Systems Leeds 1976
  
- 3 Curso de Robótica  
José María Angulo Usategui  
Rafael Avilés González      Paraninfo      1985
  
- 4 Análisis Cinemático de Mecanismos  
Shigley      McGraw-Hill      1981
  
- 5 Diseño de Mecanismos  
Shigley      McGraw-Hill      1987
  
- 6 "TTL Logic Data Book"  
National      National Semiconductor Corp.      1984
  
- 7 "Discrete Data Book"  
National      National Semiconductor Corp.      1984
  
- 8 Diseño con Circuitos Integrados TTL  
Robert L. Morris  
Jhon R. Miller      Texas Instruments      1984
  
- 9 Z80-CPU  
Technical Manual      Zilog Inc.      1987
  
- 10 Z80-CTC  
Counter/Timer      Zilog Inc.      1982
  
- 11 Users Reference Guide  
For: Gemini's "Zorba"  
Portable Computers      Gemini Electronics, Inc.      1981
  
- 12 Users Reference Guide  
For: Telcon  
Portable Computers      Telcon Industries, Inc.      1981

- |    |  |                                |      |
|----|--|--------------------------------|------|
| 13 | Interfacing to S-100/IEEE 696 Microcomputers<br>Sol Libes<br>Mark Garetz                     | McGraw-Hill                    | 1986 |
| 14 | Sistema Operativo CP/M Guía del Usuario<br>Thom Hogan  | McGraw-Hill                    | 1985 |
| 15 | Inside CP/M - A Guide for Users and Programers with CP/M and MP/M2<br>David E. Cortesi       | CBS College Publishing         | 1982 |
| 16 | El Lenguaje de Programación C<br>Brian W. Kernighan<br>Dennis M. Ritchie                     | Prentice Hall                  | 1986 |
| 17 | Lenguaje C Biblioteca de Funciones<br>Kris Janse   | Osborne/McGraw-Hill            | 1987 |
| 18 | Mix C Compiler Manual<br>Full K & R Standard C   | Mix Software Inc.              | 1985 |
| 19 | Run /C -- The Interpreter<br>Version 1.1 of Run/C  | Lifeboat Associates            | 1985 |
| 20 | C The Complete Reference<br>Herbert Schildt  | Osborne/McGraw-Hill            | 1987 |
| 21 | Microsoft MS-DOS<br>Users Guide & Users Reference  | Microsoft Corp.                | 1987 |
| 22 | Sistema Operativo MS-DOS Guía del Usuario<br>Paul Hoffman<br>Tamara Nicoloff                 | Osborne/McGraw-Hill            | 1985 |
| 23 | The AutoCAD Drafting Package<br>User Guide   | Auto Desk Inc.                 | 1982 |
| 24 | Microprocessor Application Manual<br>Motorola  | Motorola Semiconductor P. Inc. | 1978 |
| 25 | CAD/CAM Computer-Aided Design and Manufacturing<br>Nikell P. Groover<br>Emory W. Zimmers Jr. | Prentice Hall                  | 1984 |