



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

37

Zejun

FACULTAD DE INGENIERIA

DISEÑO E IMPLEMENTACION DE UN  
CONTROLADOR ELECTRONICO DIGITAL PARA  
CONVERTIR UNA MAQUINA CONVENCIONAL DE  
CORTE DE MATERIALES EN UN TALADRO CON  
CONTROL NUMERICO

**T E S I S**  
QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA EN  
EL AREA DE ELECTRICA Y ELECTRONICA  
**P R E S E N T A :**  
JOSE CARLOS BENITEZ ALVAREZ

DIRECTOR DE TESIS:

M. I. VICTOR J. GONZALEZ VILLELA



MEXICO, D. F.

1995

FALLA DE ORIGEN

TESIS CON  
FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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 esta 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.

**Diseño e implementación de un  
controlador electrónico digital para  
convertir una máquina  
convencional de corte de materiales  
en un taladro con control numérico**

**A mi madre:**

**Por todo su cariño, por que me ha enseñado la vida tal y como es, por que me ha apoyado siempre para que sea y siga siendo una persona útil.**

**A mi hermano:**

**Por su ejemplo y su cariño.**

**A mi hermana:**

**Por su cariño.**

**A mi abuelita Luz:**

**Porque siempre se ha preocupado por mí.**

**A mi abuelito José:**

**Que siempre estuvo orgulloso de mí.**

**A mi padre ...**

**A mis abuelitos Toño y Toya**

**A Erika, mi neta:**

**Por su amor y por su apoyo en tantos momentos difíciles.**

**A mis tíos Arturo y Héctor:**

**Por su apoyo y cariño.**

**A todos mis tíos que me han apoyado siempre.**

**A Alejandra, mi gran amiga ...**

**A mis amigos que han compartido conmigo tantos momentos**

**A la Universidad Nacional Autónoma de México y la Facultad de Ingeniería.**

**A Chiapas**

**A México**

## **Agradecimientos**

Este trabajo no hubiera sido posible sin la ayuda de mi director de tesis el M.I. Victor Manuel González Villela y de la Ing. Ma. Jacqueline Guadarrama Li-Ho que me brindaron su confianza y apoyo durante el desarrollo de este trabajo y por invertir parte de su tiempo en darme opiniones y consejos.

Quiero agradecer además a las siguientes personas. A Marco Ríos y su familia por su amistad y sencillez, a los tíos Arturo y Héctor por las comidas en Sanborns, a los tíos Paco y Edith (y familia) por su cariño, al tío Pepe por su cariño y sus locuras, a mis tíos de Tuxtla que siempre se han preocupado por mi hermano y por mí, a mis amigos de la Facultad, a José Jaime Villegas Blanco por sus valiosos consejos en la parte mecánica y al Ing. José Alberto Vargas por su ayuda en el editor.

Agradezco también a todos mis maestros de la Academia Militarizada "México", al Profr. Carlos Alemán Vásquez, al Profr. David Baltazar Miranda, al Profr. Carlos Baltazar, al Profr. Filiberto Torres, al Profr. Héctor Omar Peña Avila, a todos mis maestros de la Facultad de Ingeniería, a José Luna Gutiérrez y a todos aquellas personas que han contribuido a mi desarrollo personal.

José Carlos Benítez Alvarez

# Índice

<b>Introducción</b> .....	1
<b>1.-Generalidades sobre las máquinas CNC</b> .....	2
1.1.- Aparición del control numérico .....	2
1.2.- Secuencia operativa en la máquina-herramienta .....	4
1.3.- Historia del control numérico .....	4
1.4.- Evolución del control numérico .....	7
1.5.- Ámbito de operación del control numérico .....	9
1.6.- Ventajas del control numérico .....	9
1.7.- Clasificación de las máquinas con control numérico .....	10
1.7.1.- Control numérico punto a punto .....	11
1.7.2.- Control numérico paraaxial .....	11
1.7.3.- Control numérico continuo .....	12
1.8.- Cálculo de la trayectoria .....	15
1.8.1 - Interpolación lineal .....	15
1.8.2.- Interpolador circular .....	16
1.9.- Función del equipo de control numérico .....	16
1.9.1 - Control de los desplazamientos .....	18
1.9.2 - Interfaz con el hardware .....	18
1.9.3.- Soporte de la información .....	18
1.9.4.- Cálculo de las posiciones y de los desplazamientos .....	18
1.9.5 - Modos de funcionamiento de la máquina-herramienta con CNC .....	18
1.9.6.- Pupitre de control .....	19
1.9.7.- Función de la interfaz-máquina .....	19
1.10.- Control de posicionamiento .....	19
1.10.1 - Sistema de bucle abierto .....	19
1.10.2.- Sistema de bucle cerrado .....	20
1.11.- Movimiento de los ejes en una máquina CNC .....	21
1.12.- Modos de posicionamiento de la herramienta .....	22
1.12.1 - Posicionamiento incremental .....	23
1.12.2 - Posicionamiento absoluto .....	23
1.12.3.- Unidades empleadas en las coordenadas de posicionamiento .....	25

1.13.- Procedimiento de maquinado en una máquina CNC .....	25
1.13.1.- Diseño de la pieza .....	25
1.13.2.- Metodología de operaciones .....	25
1.13.3.- Elección de la máquina CNC .....	26
1.13.4.- Métodos para sujetar la pieza durante el maquinado .....	26
1.13.5.- Definición del maquinado .....	26
1.13.6.- Programación del maquinado .....	26
1.13.7.- Documentación y configuración de la máquina .....	27
1.14.- Conceptos de programación CNC .....	28
1.14.1.- Terminología del lenguaje de programación .....	30
1.14.1.1.- Carácter de programación .....	30
1.14.1.2.- Comandos .....	30
1.14.1.3.- Palabras .....	31
1.14.1.4.- Bloques .....	31
1.14.1.5.- Programa .....	31
1.15.- Ordenamiento de comandos en un bloque .....	31
1.16.- Números de programa y de secuencia: Códigos O, N .....	32
1.17.- Funciones preparatorias: Códigos G .....	32
1.18.- Palabras de dimensión: Códigos X, Y, Z, A, B, C, U, V, W, I, J, K, R, Q .....	34
1.19.- Velocidad de avance (feed rate): Código F .....	34
1.20.- Velocidad del Husillo: Código S .....	35
1.21.- Funciones misceláneas de la máquina: Códigos M .....	35
<b>2.- Planteamiento, Análisis y Solución del problema .....</b>	<b>37</b>
2.1.- Planteamiento del problema .....	37
2.2.- Análisis del problema .....	38
2.3.- Solución del problema .....	41

<b>3.- Diseño e implementación del sistema .....</b>	<b>43</b>
3.1.- Diseño del Hardware y el software del controlador .....	43
3.1.1 - Diseño del hardware .....	43
3.1.2.- Diseño del software del controlador .....	46
3.2.- Diseño del software .....	51
3.3.- Etapa de potencia .....	54
3.4.- Etapa y acoplamiento mecánica .....	55
<b>4.- Conclusiones .....</b>	<b>59</b>
<b>Apéndice #1, Diagramas y programas.....</b>	<b>61</b>
<b>Apéndice #2, SISTEMA UNIMAT, Manual del usuario .....</b>	<b>92</b>
<b>Bibliografía</b>	



## Introducción

Desde que el hombre fue capaz de construir máquinas que le permitieran mejorar su nivel de vida, éstas han pasado de ser simples herramientas para la producción a verdaderos pilares de la economía y de la industria.

México es un país que necesita modernizarse en muchos aspectos, ya que al formar parte de un sistema de economía globalizada como lo es el TLC, debe ser capaz de enfrentar los retos de productividad y competitividad. Sin embargo esto no se logrará si en México no se presentan los cambios necesarios para lograr que la planta productiva nacional se fortalezca.

El gran abismo que se forma entre los países desarrollados y los subdesarrollados está marcado básicamente por el nivel tecnológico que han alcanzado, esto aunado a un sistema político-económico que propicia el desarrollo y el bienestar de las sociedades.

Actualmente el uso de sistemas automatizados permite que los procesos de manufactura sean más eficientes logrando productos con mayor calidad lo cual ha fomentado el desarrollo de sistemas de manufactura computarizados, que han alcanzado niveles de perfeccionamiento verdaderamente impresionantes, al punto de que es posible encontrarse con empresas "fantasma" ya que únicamente son las máquinas y robots los que intervienen en los procesos de producción.

Uno de los sistemas de manufactura computarizada que más aceptación ha tenido es el uso de máquinas controladas con control numérico debido a las ventajas y a la flexibilidad que presentan.

La idea de realizar un controlador CNC surge del hecho de obtener los conocimientos necesarios para la realización de máquinas-herramienta con control numérico a partir de máquinas convencionales, para ello decidimos iniciar con el taladrado ya que es la operación más sencilla de un controlador CNC, tratando de aprovechar los recursos de maquinado existentes en la industria nacional, utilizando tecnologías propias y sin la necesidad de hacer grandes inversiones para lograrlo.

Todo ello nos condujo a realizar el sistema UNIMAT, nombre que adoptamos de la máquina-herramienta utilizada y que no tiene un significado específico, y que será utilizado de ahora en adelante para identificar en su totalidad al sistema. El sistema UNIMAT consta de tres partes diferentes que interactúan entre sí. Consiste de un programa que nos permite programar los movimientos de la máquina, de un sistema de control por microcontrolador que administra las señales de control de los diferentes dispositivos de la máquina y por último, la máquina-herramienta que ejecuta los movimientos programados.

UNIMAT es el resultado de un estudio detallado para la obtención de una máquina taladradora controlada con control numérico por computadora a partir de una máquina convencional.

# Capítulo 1

## Generalidades sobre las máquinas CNC.

Las máquinas-herramienta han desempeñado un papel fundamental en el desarrollo tecnológico mundial hasta el punto que no es una exageración decir que la tasa de desarrollo de máquinas-herramienta gobierna directamente la tasa de desarrollo industrial. Gracias a la utilización de dichas máquinas se ha podido realizar, de forma práctica, maquinaria de todo tipo que, aunque concebida y realizada, no podía ser comercializada, ya que no existían medios adecuados para su construcción industrial. El ejemplo más característico aparece con la máquina de vapor. Esta máquina, aunque inventada por James Watt en 1766, no tuvo desarrollo conveniente hasta 1776, en que John Wilkinson construyó la primera mandrinadora, gracias a la cual fue posible fabricar máquinas de vapor a gran escala.

Este hecho prácticamente marcó el inicio de la revolución industrial. Las elevaciones de las potencias disponibles, la demanda de productos manufacturados y el desarrollo de las comunicaciones, debido fundamentalmente a la aparición de las locomotoras, hicieron posible el desarrollo paralelo de las herramientas de corte de las máquinas-herramienta utilizadas en la manufactura de dichos productos. También los materiales se fueron desarrollando obteniéndose el llamado acero de alta velocidad, que contenía ciertas proporciones de cromo y tungsteno. Este nuevo material tuvo una influencia decisiva en el rápido crecimiento de la máquina-herramienta. Posteriormente, en 1930, se introdujo el carburo de tungsteno que, una vez perfeccionado, es el más usado en la actualidad por la industria.

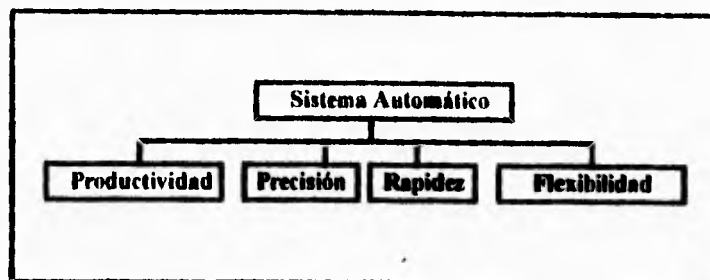
### 1.1 - Aparición del control numérico

En los sistemas de producción convencionales, usados en la mayor parte de la industria, las máquinas-herramienta del mismo tipo están agrupadas, siendo cada máquina manejada independientemente. Las operaciones necesarias para la fabricación de una pieza se dividen en un número de operaciones independientes, cada una de las cuales se pueden realizar de manera más eficiente en una máquina en particular. Actualmente la fabricación de piezas mecánicas exige niveles de precisión, cantidad de piezas manufacturadas y diseños muy complejos que la fabricación de piezas no puede basarse únicamente en la utilización de las máquinas-herramienta por el hombre y a las limitaciones que ello implica. Debido a esta situación, se introdujeron los sistemas automáticos en los procesos de fabricación, que fue marcada básicamente por las siguientes razones:

- a) Necesidad de fabricar productos que no se podían conseguir en cantidad y calidad suficiente sin recurrir a la automatización del proceso de fabricación.
- b) Necesidad de obtener productos hasta entonces imposibles o de muy difícil fabricación, por ser excesivamente complejos para ser controlados por un operador humano.
- c) Necesidad de fabricar productos a precios suficientemente bajos.

d) Aumento de la productividad.

Actualmente, todos los dispositivos automáticos ideados tienden a optimizar la función de las siguientes cuatro variables:



Los primeros automatismos no optimizaban esta función, dado que eran fundamentalmente dispositivos de propósito particular y, por tanto, de una rigidez casi absoluta.

Hacia 1942 surgió lo que podríamos llamar el primer control numérico verdadero, y que apareció como una necesidad impuesta por la industria aeronáutica.

**El Control Numérico (CN) es un sistema, que aplicado a máquinas-herramienta, automatiza y controla todas las acciones de la máquina.**

En general, con un CN pueden controlarse:

- Los movimientos de los carros o del cabezal;
- El valor y el sentido de las velocidades de avance y de corte;
- Los cambios de herramientas, así como de piezas;
- Las condiciones de funcionamiento de la máquina, en cuanto a su modo de trabajar (con/sin refrigerante, frenos, etc.), o en cuanto a su estado de funcionamiento (deficiencias, averías, etc.)

Paralelamente, el Control Numérico se encarga de coordinar otras funciones que le son propias.

Por ejemplo

- Control de flujos de información;
- Control de la sintaxis de programación;

- Diagnóstico de su funcionamiento, etc.

Toda la información necesaria para la ejecución de una pieza constituye el PROGRAMA, que es escrito en un lenguaje especial (código) por medio de caracteres alfanuméricos sobre un soporte físico (cinta perforada, cinta magnética).

La pauta a seguir en el lanzamiento de un pieza puede ser la siguiente:

- 1.- Se escribe el programa de la pieza a partir de la información que se dispone de ella (plano), del proceso de maquinado trazado y de las herramientas elegidas para las operaciones necesarias. Esta fase de trabajo, al hacerse externa e independiente de la máquina, permite que ésta permanezca en funcionamiento.
- 2.- Previamente a la terminación de la fabricación de la serie anterior, se realiza el acopio de piezas, herramientas y del utillaje necesario, preparándolos alrededor de la máquina.
- 3.- Después se colocan las herramientas, el utillaje y la pieza sobre la máquina, así como la cinta perforada, en la unidad de control.
- 4.- Se pone en marcha la máquina y se comprueba si la pieza es correctamente elaborada. Si no lo es y el error se debe a una mala programación, se corrige de inmediato a pie de máquina y, si el error es debido al ajuste de las herramientas, se modifican los reglajes en la unidad de control.

### 1.2.- Secuencia operativa en la máquina-herramienta

Cada operación que un operario realiza en una máquina-herramienta tradicional constituye una secuencia de posicionamientos relativos entre la herramienta y la pieza que trabaja. Véase la figura 1. En el esquema de la figura 2, confiamos los posicionamientos relativos entre la herramienta-pieza a un equipo de control. La combinación de la máquina-herramienta y dicho equipo recibe el nombre de máquina-herramienta con control numérico.

### 1.3.- Historia del control numérico

En principio, contrariamente a lo que se pudiera pensar, el Control Numérico de máquinas herramienta no fue concebido para mejorar los procesos de fabricación, sino para dar solución a problemas tecnológicos surgidos a consecuencia del diseño de piezas cada vez más difíciles de maquinar.

En 1942, la Bendix Corporation tuvo problemas con la fabricación de una leva tridimensional para el regulador de una bomba de inyección para motores de avión. El perfil tan especial de dicha leva es prácticamente imposible de realizar con máquinas comandadas manualmente.

La dificultad provenía de combinar los movimientos del útil simultáneamente según varios ejes de coordenadas, hallando el perfil deseado. Se acordó entonces confiar los cálculos a una máquina

automática que definiera un gran número de puntos de la trayectoria, siendo el útil conducido sucesivamente de uno a otro.

En 1947, John Parsons, constructor americano de hélices de helicóptero, concibe un mando automático con entrada de información numérica.

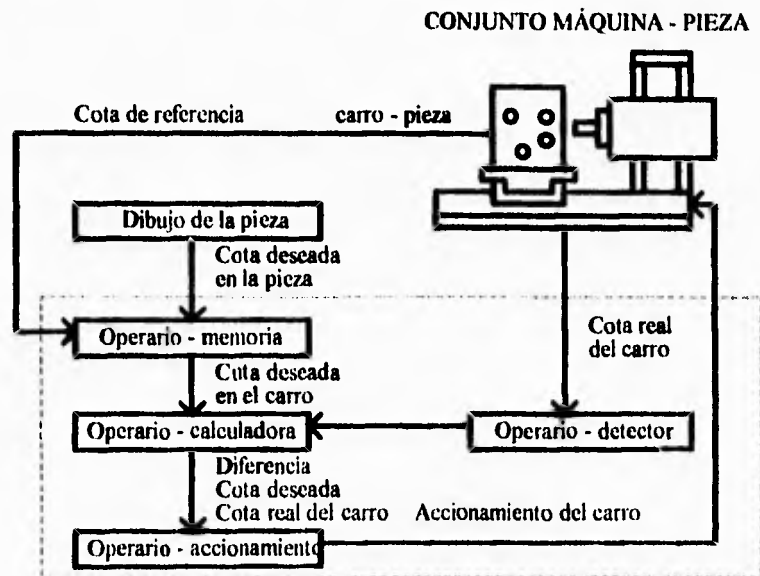


Figura 1.- Función realizada por el operario

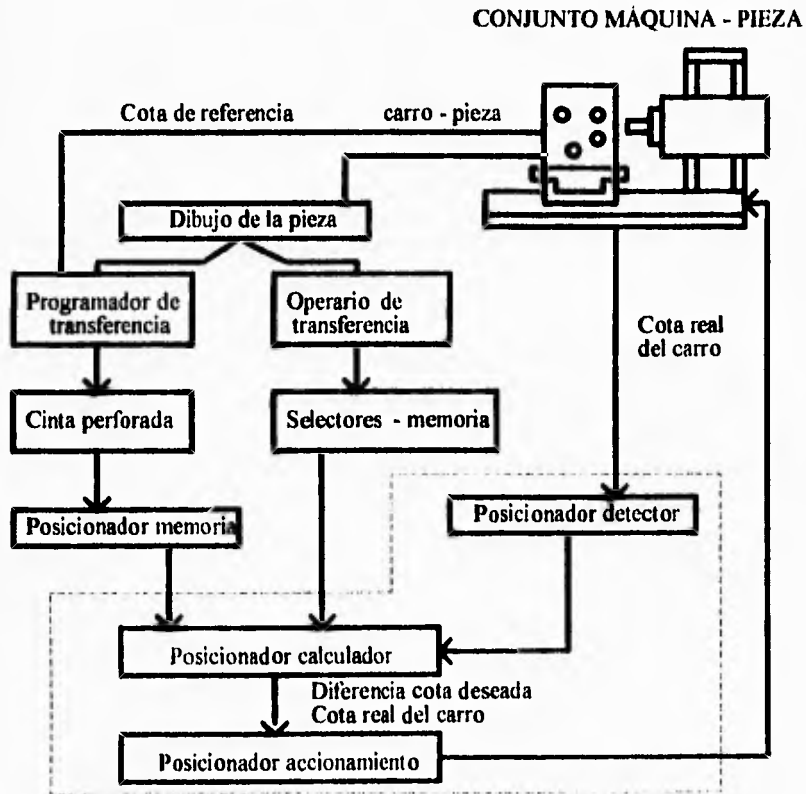


Figura 2.- Función realizada por el control numérico

Antes, en su afán por controlar la forma de las hélices, así como su paso, Parsons debía utilizar un gran número de plantillas y su realización estaba lejos de ser rápida y económica.

La idea de utilizar cartas perforadas (transportando las coordenadas de los ejes de los agujeros) en un lector que permitiera traducir las señales de mando a los ejes, permite a Parsons desarrollar su sistema Digitón.

En esta época, la U.S. Air Force estaba preocupada con la fabricación de estructuras difíciles de trabajar por copiado y susceptibles de ser modificadas rápidamente. Gracias a su sistema, Parsons obtiene un contrato y el apoyo del Laboratorio de Servomecanismos del Massachusetts Institute of Technology M.I.T.

El Gobierno americano apoya la iniciativa para el desarrollo de una fresadora de ejes en contorno comandado por control digital.

En 1953, después de cinco años de puesta a punto, el M.I.T. utiliza por primera vez el nombre de Numerical Control.

En 1956, la U.S.A.F. hace un pedido de 170 máquinas de Control Numérico a tres grandes constructores americanos:

- Cincinnati Milling Machine Company,
- Giddin & Lewis,
- Kearney & Trecker.

Paralelamente a esta evolución, ciertos constructores se interesan por el desarrollo de máquinas más simples para trabajos tales como taladrado, mandrilado y punteado, que no requieren ningún movimiento continuo, pero sí un posicionamiento preciso. Es por ello que el Control Numérico Punto a Punto hizo su aparición más tarde que el Control Numérico en Contorno. Después aparecería el Control Numérico Paraaxial.

En 1960, también en el M.I.T. se realizaron las primeras demostraciones de Control Adaptable (un perfeccionamiento del Control Numérico que permite, además, la autorregulación de las condiciones de trabajo de las máquinas).

A finales de 1968 tuvieron lugar los primeros ensayos de Control Numérico Directo (CND).

#### **1.4.- Evolución del control numérico**

Aunque el control numérico es una técnica relativamente joven, su evolución ha conocido ya varias fases. El origen de esta evolución hay que buscarlo en los desarrollos acelerados de la electrónica y, en particular, de la microelectrónica.

En la historia del control numérico, se pueden distinguir cuatro generaciones:

- En la primera, la lógica de control estaba basada en tubos electrónicos y en relés. esta era la tecnología de las primeras máquinas con control numérico.
- A partir de 1965, lógica del control fue elaborada a partir de semiconductores: transistores, diodos y tiristores. se pudo de esta forma construir equipos de control más compactos.

- Inmediatamente después, y disponiendo siempre de los semiconductores, ciertas combinaciones de elementos lógicos fueron englobados en un mismo soporte, lo que le valió el nombre de circuitos integrados. Son las máquinas de tercera generación.

- En la cuarta generación que está situada hacia los años de 1974 - 1975, la lógica del tratamiento fue confiada a una computadora: en principio, una minicomputadora, actualmente una microcomputadora. Se ha llegado así a la lógica por software o lógica programada. En efecto, la lógica del equipo no está realizada por montaje de elementos digitales, sino por programación de una computadora. Se trata de esta forma del CNC (Computer Numerical Control) o del control numérico con lógica programada.

Los sistemas de Control Numérico (NC) ofrecen las siguientes ventajas, entre otras:

- 1.- Mejor control de la herramienta de corte por medio de condiciones óptimas de maquinado.
- 2.- Aumento de la calidad y repetitividad de la pieza.
- 3.- Reducción de los costos de maquinado, desgaste de la herramienta.
- 4.- Reducción del tiempo de manufactura de las piezas.
- 5.- Reducción de desperdicios.
- 6.- Planeación de la producción y permite que el trabajo de maquinado quede en manos de los ingenieros.

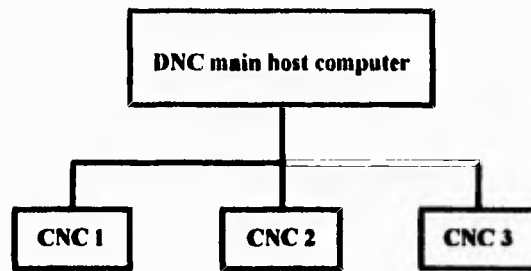
Sin embargo en los sistemas de NC la mayoría de las funciones de control se encuentran implementadas electrónicamente, esto significa que la mayoría de las funciones de la máquina son controladas por los componentes electrónicos físicamente localizados dentro del controlador. Es por ello que se optó por el control numérico por computadora (CNC), en donde la mayoría de las funciones de la máquina son controladas por programa y no son borradas cuando la máquina se apague. La memoria de computadora que permite mantener tal información se conoce como ROM (Read Only ). La entrada de información generalmente se realiza por medio de un teclado alfanumérico para permitir la entrada directa de datos en forma manual (MDI, Manual Data Input) para programar la pieza. Estos programas son almacenados en la memoria RAM (Random Access Memory) de la computadora, en donde son editados, listados, procesados y ejecutados por la unidad de control. Estos programas pueden ser almacenados en unidades auxiliares de almacenamiento, tales como, diskettes, cintas magnéticas, cintas perforadas o CD-ROMs.

Los sistemas CNC prometen nuevas alternativas y ventajas sobre los sistemas antiguos de NC. Algunas son las siguientes:

- 1.- Reducción del hardware necesario para agregar nuevas funciones a la máquina, estas pueden ser añadidas por software.
- 2.- El programa CNC puede ser escrito, almacenado y ejecutado directamente en la máquina CNC.
- 3.- Cualquier porción del programa CNC proporcionado puede ser ejecutado y editado como sea necesario. Los movimientos de la máquina pueden ser mostrados mientras se ejecutan.
- 4.- Diferentes programas CNC pueden ser almacenados en el controlador.



5.- Varias máquinas CNC pueden ser conectadas a una computadora central. Los programas pueden ser almacenados en cada una de las máquinas por medio de esa computadora central. Esto se conoce como control numérico directo:



Control numérico directo DNC

#### 1.5.- Ámbito de aplicación del control numérico

Los sistemas CNC pueden utilizarse de acuerdo a las cuatro variables ya mencionadas (productividad, precisión, rapidez y flexibilidad), a fin de elegir el más conveniente, de acuerdo con el número de piezas que se han de fabricar, es decir, de acuerdo a la serie de fabricación, que puede ser una de las siguientes:

- 1.- Grandes series (> 10,000 piezas);
- 2.- Series medias (entre 50 y 10,000 piezas);
- 3.- Series pequeñas (> 5 piezas).

#### 1.6.- Ventajas del control numérico

Con la utilización del control numérico se obtienen grandes ventajas en la fabricación de piezas y en muchos casos resulta ser la solución ideal. Entre las ventajas más importantes podemos citar las siguientes:

- Posibilidad de fabricación de piezas casi imposibles o muy complicadas.
- Precisión, la cual se debe principalmente a la mayor precisión que presentan las máquinas CNC frente a las clásicas. La precisión de las máquinas CNC actuales se encuentra entre 1 a 10 micras.
- Reducción de los tiempos de ciclos operacionales. Lo cual incluye los siguientes aspectos:
  - \* Trayectorias y velocidades más ajustadas que en las máquinas convencionales;
  - \* Menor revisión constante de los planos y hojas de instrucciones;

\* Menor verificación de medidas entre operaciones.

- Ahorro de herramientas y utillaje. Esto se obtiene como consecuencia de la utilización de herramientas más universales.
- Mayor precisión e intercambiabilidad de piezas.
- Reducción del porcentaje de piezas defectuosas.
- Reducción del tiempo de cambio de pieza.
- Reducción del tamaño del lote.
- Reducción del tiempo de inspección.
- Un nivel de calidad más alto.

Actualmente el CNC se aplica a diversos tipos de máquinas como son:

- Taladradoras
- Mandriladoras
- Tornos
- Centros de maquinado
- Rectificadoras
- Punzonadoras
- Máquinas de electroerosión
- Máquinas de soldar
- Máquinas de oxicorte
- Dobladoras
- Plegadoras
- Máquinas de dibujar
- Máquinas de trazar
- Bobinadoras
- Máquinas de medir por coordenadas
- Manipuladores
- Robots
- etc.

### **1.7.- Clasificación de las máquinas con control numérico**

Como consecuencia de las diferencias entre las máquinas por automatizar, de las dificultades técnicas en el diseño de las unidades de control y de condicionamientos económicos, se han originado diversos tipos de control numérico, que pueden ser clasificados en tres tipos:

- 1.- Control numérico punto a punto.
- 2.- Control numérico paraaxial.
- 3.- Control numérico Continuo.

### 1.7.1.- Control numérico punto a punto

Este sistema controla el posicionamiento de la herramienta en los sucesivos puntos donde deben efectuarse una o varias operaciones de maquinado. La trayectoria seguida para pasar de un punto al siguiente no tiene importancia, pues las funciones de posicionamiento y de maquinado son distintas Fig. 3.

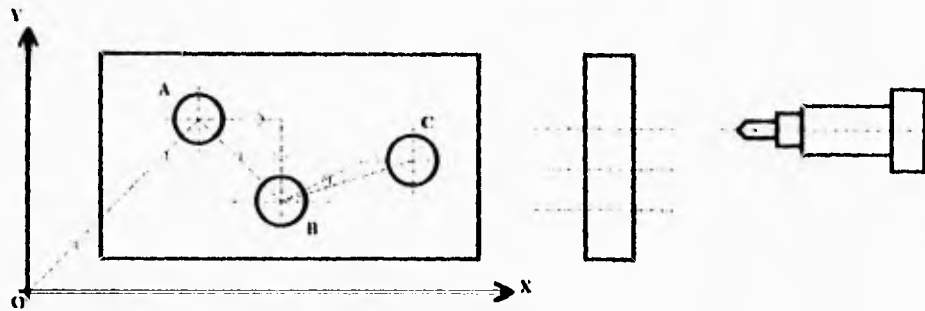


Fig. 3.- Control numérico punto a punto

En este caso, para desplazarse a los puntos A, B, y C, el útil puede seguir varios tipos de trayectoria:

- 1) Desplazamientos efectuados según direcciones paralelas, es decir OX y OY.
- 2) Desplazamiento dando una orden simultánea a los dos ejes, pero sin existir coordinación alguna en los sistemas de mando de cada uno de los motores. La trayectoria se aproxima a  $45^\circ$ .

Como aplicaciones principales tenemos: punteadoras, taladros, punzonadoras y mandriladoras.

### 1.7.2.- Control numérico paraxial

Con este sistema es posible controlar, además de la posición del órgano móvil, la trayectoria seguida por el mismo según la dirección de algunos de los ejes coordenados.

Esta posibilidad se ofrece, generalmente, como una opción adaptable a un sistema de posicionado punto a punto. Fig. 4

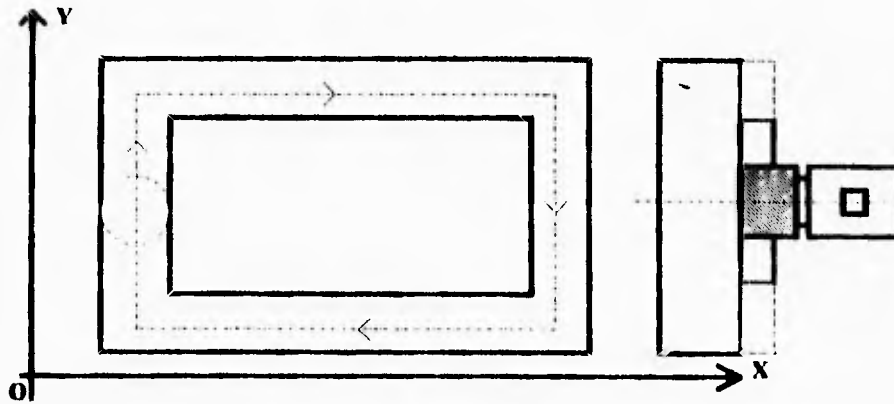


Fig.4.- Control numérico paraxial

Una aplicación corriente de este tipo de sistema, dentro de la máquina herramienta, se halla en la **taladora-fresadora**. En cualquier caso, el fresado sólo puede realizarse según trayectorias rectilíneas paralelas a alguno de los ejes coordenados.

### 1.7.3.- Control numérico continuo

En los sistemas de posicionado continuo, los desplazamientos del órgano móvil son controlados en todo momento, de manera que las posiciones sucesivas del mismo deben corresponder siempre a la trayectoria preestablecida.

Se consigue este resultado relacionando entre sí los movimientos elementales realizados según los distintos ejes coordenados, de modo que estos movimientos elementales sigan cada uno una ley prescrita por una trayectoria común.

El control numérico continuo generalmente se utiliza para mantener la herramienta de corte en contacto constante con la pieza de trabajo mientras éste la corta.

Una aplicación de estos sistemas se encuentra en fresadoras, tornos, centros de maquinado, máquinas a electroerosión, oxicorte, mesas de dibujo automáticas. Fig. 5.

El método utilizado es el siguiente. Las órdenes de desplazamiento son comunicadas al órgano móvil bajo forma de componentes de un vector que permite alcanzar un punto próximo. La curva elemental descrita por este órgano será entonces un segmento de recta. Cuando se quiera seguir un contorneado cualquiera, se descompondrá en segmentos elementales llamadas incrementos. Su número dependerá de la precisión con que se quiera efectuar el trabajo. Si es elevada, el número de incrementos será grande.

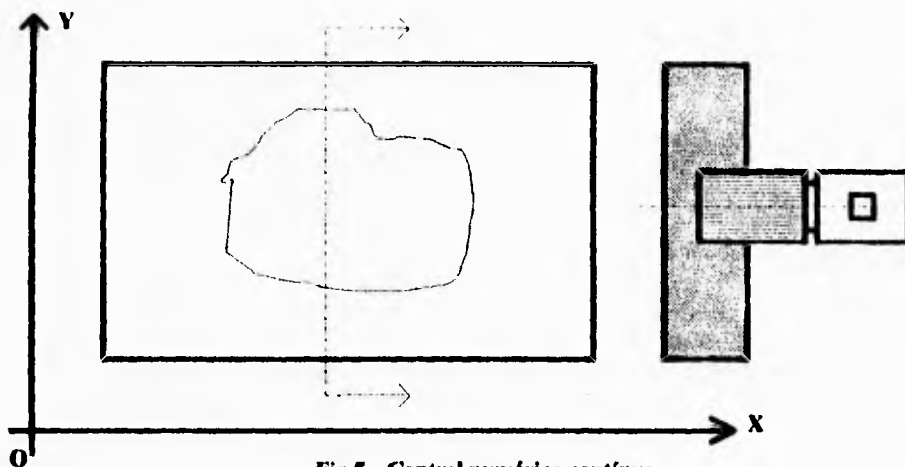


Fig.5.- Control numérico continuo

Esta descomposición de la curva en segmentos se llama *interpolación*. Es necesario sincronizar las velocidades siguiendo las direcciones O-X Y O-Y, a fin de definir correctamente los vectores VA, VB, etc. La curva real es así situada en el interior de una zona delimitada por dos curvas paralelas y donde la anchura está definida por la precisión de maquinado. Fig.6.

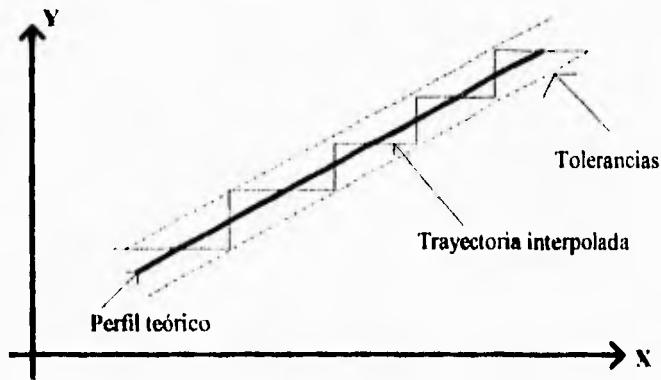


Fig.6.- Interpolación.

En el caso de un desplazamiento continuo en el plano, se sincronizarán dos ejes, denominándose entonces "máquina de dos ejes". Entre las máquinas más características de dos ejes se encuentran los tornos con CN, ya que las piezas de revolución se ejecutan con la sincronización de los ejes X y Z. Para un maquinado en el plano deberá utilizarse una máquina de dos ejes. Fig. 7.

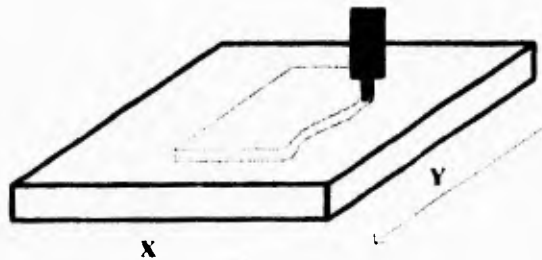


Fig.7.- Maquinado en el plano.

### 1.8.- Cálculo de la trayectoria

Dos procedimientos son empleados para efectuar estos cálculos:

1.- Hacer que una computadora calcule las coordenadas de los puntos necesarios para seguir correctamente la trayectoria en función de la precisión requerida, y dar bajo formas numéricas estas coordenadas al Control Numérico. Este procedimiento se utiliza en programación automática.

2.- El control de la máquina incluye un calculador llamado *interpolador*, que efectúa el cálculo de las coordenadas y las suministra a la cadena de control. Estos interpoladores son de varios tipos, según las curvas que permiten realizar.

Existen básicamente dos tipos de interpoladores a saber:

#### 1.8.1.- Interpolador lineal

Descompone un segmento recto. Es suficiente con indicar las coordenadas de las extremidades del segmento. El equipo, al recibir la instrucción en forma de G01 ..., pasa la información a la tarjeta destinada exclusivamente a este tipo de cálculo y manda desplazarse a los motores X y Y en forma sincronizada hasta alcanzar la posición B siguiendo una línea recta. Fig. 8.

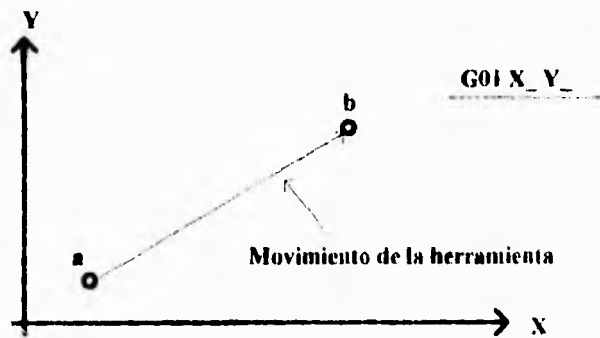


Fig. 8.- Interpolación lineal

### 1.8.2.- Interpolador circular

Descompone un arco de circunferencia conociendo los elementos necesarios para definirlo; con frecuencia se define al arco con el punto inicial, punto final y el centro. En este caso al recibir la información en forma de  $G02$  ..., la tarjeta manda movimientos sincronizados a los motores X y Y teniendo en cuenta un punto (I, J) que es el centro del arco. Fig.9.

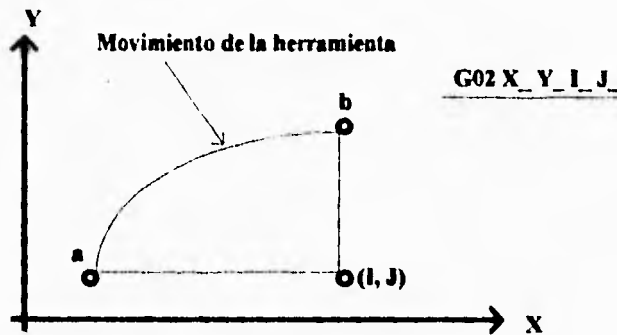


Fig. 9.- Interpolación circular

### 1.9.- Función del equipo de control numérico

En la conducción de máquinas con control numérico, el equipo desempeña dos papeles principales. Fig. 10.

- Asegura el desplazamiento de los ejes de la máquina, de manera individual o combinada, y provoca las trayectorias de las herramientas programadas.
- Alimenta el hardware con instrucciones de control y de funciones auxiliares de la máquina (informaciones tecnológicas).

Desde el punto de vista general, ejerce todavía las siguientes funciones:

- Lee el soporte de informaciones e interpreta el contenido.
- Permite la modificación eventual de estas informaciones (edición de programa).
- Calcula eventualmente la posición de los ejes y los desplazamientos de la máquina.
- Controla los diferentes modos de funcionamiento de la máquina.
- Dirige todas las señales que vienen o van al pupitre del control de la máquina.



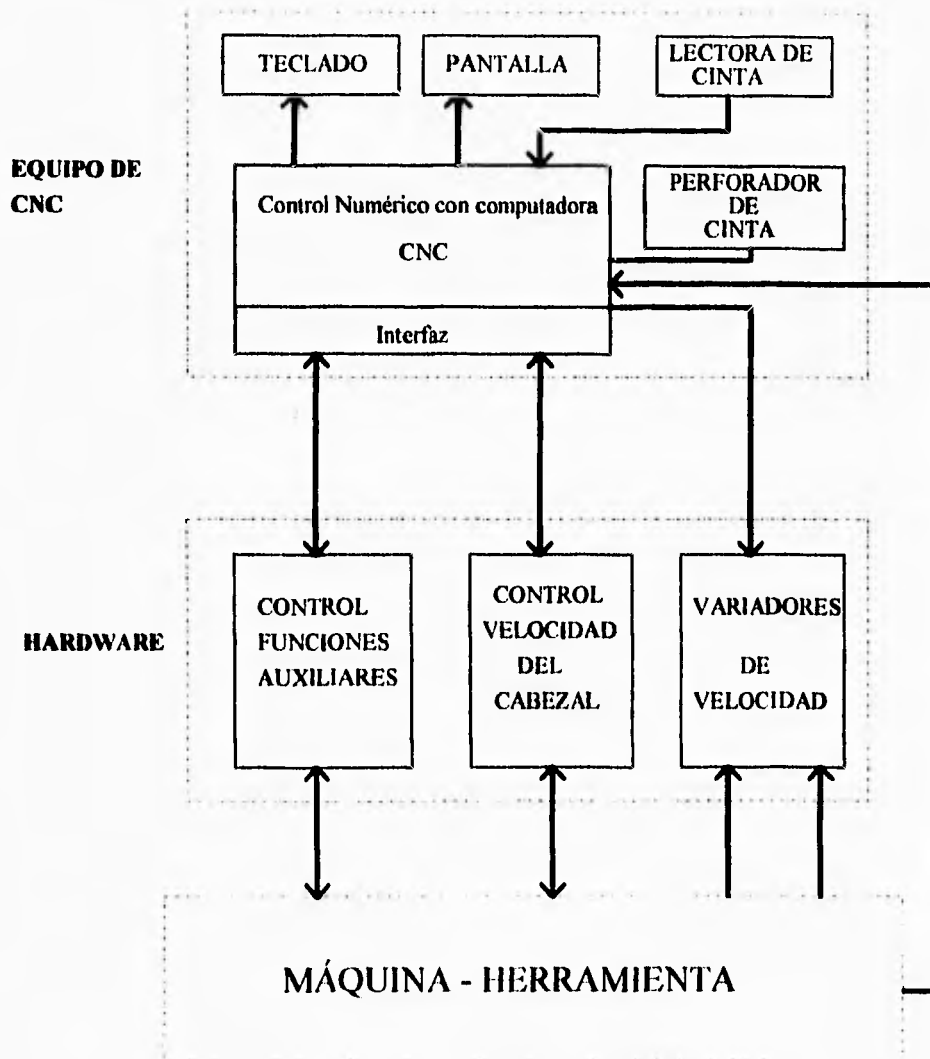


Fig.10.- Estructura general de un Control Numérico con Computadora

### **1.9.1.- Control de los desplazamientos**

El equipo de control numérico es el que está encargado de conducir los ejes de la máquina, cuyo número y tipo varían de una aplicación a otra. El movimiento de los ejes es controlado en desplazamiento y en velocidad.

### **1.9.2.- Interfaz con el hardware**

Todos los controles de las funciones auxiliares de la máquina que son programados o eventualmente creados por el equipo de control, transitan hacia el hardware a través de una interfaz.

Las señales que corresponden a estas funciones auxiliares son generalmente codificadas y éste será precisamente el papel de la interfaz, decodificarlas y cuidar de su buena ejecución. El equipo recibe como respuesta de la interfaz un cierto número de señales destinadas a informarle de los defectos sobrevenidos en la máquina y a autorizar o a prohibir el seguimiento de la ejecución del programa.

La gestión y el control de este diálogo con la interfaz tiene, pues, un lugar más o menos importante en la lógica funcional de un equipo de control numérico, en función de la complejidad de la máquina. Esta interfaz puede ser cableada o programada.

### **1.9.3.- Soporte de la información**

El equipo lee, decodifica e interpreta la información registrada en el soporte. Carga sus diversos registros internos de control.

### **1.9.4.- Cálculo de las posiciones y de los desplazamientos**

Además de la información de los desplazamientos programados en la cinta, todos los equipos de control numérico aceptan cierto número de correcciones (o compensaciones) de trayectorias de herramienta introducidas en el pupitre de control; correcciones sobre las longitudes y los diámetros de las herramientas y sobre los orígenes de programación.

La posición absoluta de los ejes de la máquina debe, pues, calcularse eventualmente por el equipo de control numérico después de estas diversas informaciones. El modo de programación absoluto o incremental, se deja en los equipos de control modernos a elección del programador, mientras que el modo de control de los movimientos está determinado por los dispositivos de accionamiento y de medida.

Estas consideraciones también valen para las velocidades de desplazamiento, programadas por lo general a lo largo de la trayectoria de la herramienta, ellas deben ser transformadas en consignas reales para cada uno de los ejes en movimiento.

### **1.9.5.- Modos de funcionamiento de la máquina-herramienta con CNC**

El equipo de control numérico asegura los diferentes modos de funcionamiento de la máquina-herramienta, a saber:

- Automático: las diversas secuencias del maquinado se encadenan automáticamente.
- Semiautomático o "bloque por bloque": la máquina ejecuta cada secuencia después de que el operario haya dado la orden.
- Entrada manual de datos (MDI: Manual Data Input): las instrucciones se introducen paso a paso al pupitre de control.

#### **1.9.6.- Pupitre de control**

El pupitre de control está provisto de botones, selectores, etc., y envía una cantidad importante de señales al equipo de control numérico, y viceversa, ciertas informaciones pertinentes en el desarrollo del programa de maquinado se envían por el equipo hacia el pupitre del operario y son visualizados sobre éste.

#### **1.9.7.- Función de la interfaz-máquina**

La función principal de la interfaz-máquina es administrar y controlar las diversas funciones de la máquina-herramienta. Con este fin, decodifica las señales enviadas por el equipo de control numérico, transforma éstas en órdenes eléctricas que envía a los diversos dispositivos de la máquina y cuida de la buena ejecución de estos mandos. Vigila también los circuitos de engrase y refrigerante de la máquina. En fin, controla el equipo de control numérico autorizándole o prohibiéndole el seguimiento de la ejecución del programa de maquinado e informándole de los efectos de la máquina que él mismo constata.

#### **1.10.- Control de posicionamiento**

El control numérico de un órgano móvil tiene por objeto conducirlo automáticamente a una posición determinada, sea siguiendo una trayectoria recta o curva. Cuando una orden ha sido emitida, es preciso cerciorarse de que el útil adopta la posición exacta. Para ello se utilizan dos sistemas:

- 1.- Sistema de bucle abierto.
- 2.- Sistema de bucle cerrado.

##### **1.10.1.- Sistema de bucle abierto**

En este sistema se suprime el retorno de la información del captador de posición del bucle cerrado, por la introducción de motores paso a paso. El principio de funcionamiento de estos motores se presta particularmente bien para máquinas que trabajan por conteo de impulsos. (Fig. 11);

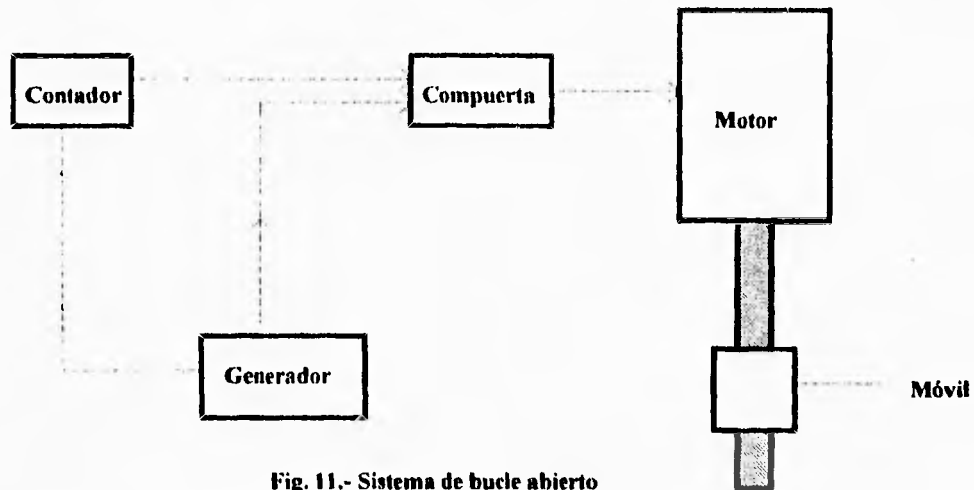


Fig. 11.- Sistema de bucle abierto

El motor paso a paso contiene un rotor que efectúa una rotación de un ángulo determinado cada vez que su bobina de mando recibe un impulso eléctrico.

Entre el generador y el motor se dispone de un circuito de conmutación. Este circuito está controlado por un contador que se encarga de encender y a pagar las bobinas correspondientes del motor de pasos a medida que se agota la cuenta de pasos a realizar.

El motor de pasos permite controlar desplazamientos y velocidades de manera muy sencilla. Está alimentado por un tren de impulsos, en el que el número fija el número y la cadencia, la velocidad a la que debe efectuarse. A cada impulso, el rotor gira un paso; por ejemplo  $1.8^{\circ}$ . Es suficiente pues calcular el número de pasos para deducir el desplazamiento angular; además, la cadencia de los impulsos determina la velocidad de rotación media. También es fácil invertir el sentido de rotación.

El funcionamiento es el siguiente: el desplazamiento deseado es comunicado a un control bajo la forma de impulsos. Este traduce los impulsos en señales que excitan sucesivamente las diversas fases del estator del motor paso a paso accionando el husillo. Si se desea cambiar el sentido de rotación, es suficiente cambiar la secuencia de excitación.

Si el paso del motor fuese demasiado grande, es posible disminuir su tamaño a partir de la excitación de varias fases del estator, es decir, operar el motor en medio-paso.

#### 1.10.2.- Sistema de bucle cerrado

El sistema de posicionamiento en bucle cerrado constituye el principio de funcionamiento básico de un servomecanismo de posición, y consiste en comparar en todo momento la posición del móvil con la orden dada. La señal que se emite al momento es función de la relación entre la posición y la orden. Fig. 12

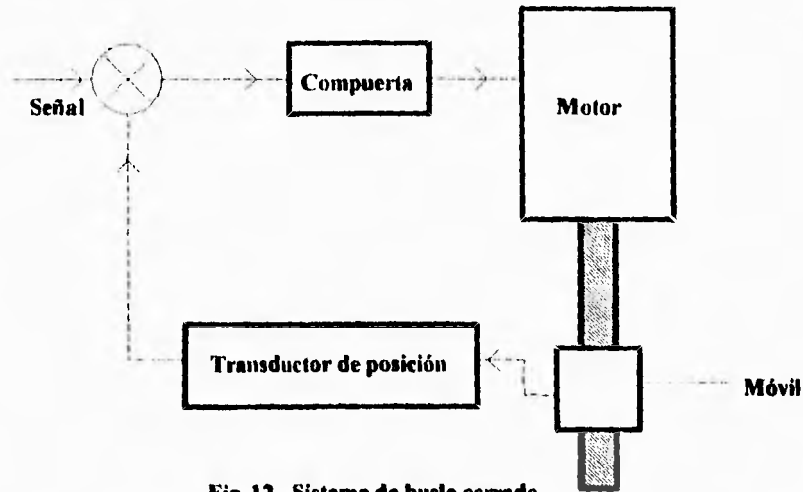


Fig. 12.- Sistema de bucle cerrado

Las máquinas que utilizan este sistema normalmente tienen dos bucles de retorno de información, uno para el control de la posición y otro para el control de la velocidad de desplazamiento del móvil, ya que momentos antes de alcanzar la cota deseada disminuye la velocidad de éste para conseguir un posicionado correcto.

Existen en el mercado un gran número de transductores de posición y de velocidad, y se pueden clasificar en función de los siguientes conceptos:

- Por la naturaleza de la información entregada: analógica o digital;
- por la relación entre la magnitud mecánica y la magnitud eléctrica: absoluta o incremental;
- por el emplazamiento del captador en la cadena de control: medida directa o indirecta;
- por la forma física del captador: lineal o rotativo.

### 1.11.- Movimiento de los ejes en una máquina CNC

Los equipos de CNC ejecutan las operaciones de maquinado realizando movimientos en forma de deslizamiento lineal y movimientos giratorios. El método real de movimiento está diseñado por el fabricante de la máquina y varía de una máquina a otra. Por ejemplo, la mesa puede moverse en el plano horizontal (movimientos en los ejes X-Y), y el husillo en un plano vertical (movimiento en el eje Z).

El sistema responde a una orden para mover la herramienta de corte sobre el eje X ó Y efectuando movimientos en la mesa en dirección opuesta, -X ó -Y. Debido a que la máquina sabe automáticamente como mover los ejes en respuesta a una orden, el programador no necesita saber que es lo que se mueve, si la herramienta o la mesa. Lo importante de esto, es que, a fin de cuentas, la herramienta llega al punto programado. Entonces, los ejes de la máquina están definidos en términos del movimiento de la herramienta de corte.

*Nota: Desde el punto de vista del programador, los movimientos de la mesa de la máquina CNC deben ser considerados como movimientos de la herramienta sobre los ejes X, Y y Z.*

Actualmente existen normas para establecer los ejes de una máquina CNC y se basan en las Normas EIA RS 267A, las cuales incluyen:

- 1.- Los movimientos de los ejes primarios se establecen de acuerdo a la regla de la mano derecha.
- 2.- Los movimientos del husillo se realizan primariamente sobre el eje Z. Los movimientos hacia la pieza sobre el eje - Z y los movimientos desde la pieza en el eje +Z.
- 3.- En la mayoría de las fresadoras, los movimientos sobre el eje X son aquellos que se realizan en el eje con mayor longitud y que es perpendicular al eje Z. Los movimientos en -X, son directamente opuestos a los indicados en +X. El eje X es paralelo al sujetador de la pieza y se encuentra en el plano horizontal. Los movimientos del eje X sobre un plano se realizan hacia la derecha , tal como lo ve el operador.
- 4.- Con respecto a la fresadoras, el movimiento sobre el eje Y es el más corto y es perpendicular al eje Z. Los movimientos indicados en dirección -Y son directamente opuestos a los indicados en +Y. El eje Y está en el mismo plano que el eje X. Observando el plano, el operador notará que el eje Y es perpendicular al eje X.

Muchas máquinas CNC pueden realizar movimientos simultáneos sobre el eje X, Y, y Z, y son conocidas como máquinas de tres ejes. Máquinas CNC más complejas pueden ejecutar también movimientos giratorios. Fig. 13.

#### **1.12.- Modos de posicionamiento de la herramienta**

Dentro del sistema de coordenadas de una máquina CNC, el posicionamiento de la herramienta puede programarse de dos formas: incremental, absoluto, o incremental y absoluto.

### 1.12.1.- Posicionamiento incremental

Si la máquina opera en este modo, el posicionamiento de la herramienta se realiza tomando como referencia el último punto, es decir, la máquina se desplaza las unidades indicadas a partir del punto en que se encuentra.

Este modo de posicionamiento tiene algunos inconvenientes, el más notable es que si existe un movimiento incremental con error, todos los demás resultan en un error también. Fig. 14.

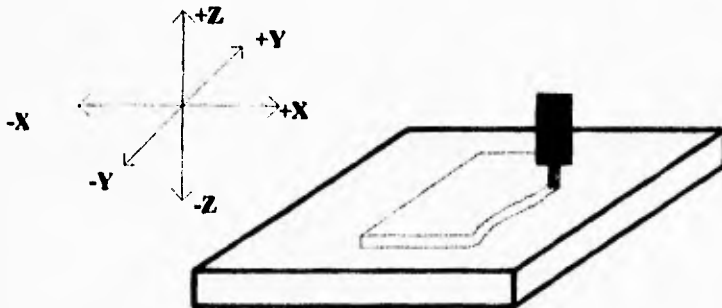


Fig. 13.- Sistema de coordenadas según la Norma EIA RS-267A.

### 1.12.2.- Posicionamiento absoluto

Cuando la máquina opera en este modo, la máquina determina cada nueva posición de la herramienta a partir de un punto fijo u origen específico (0,0). Fig. 14. Muchos controladores modernos, tienen la capacidad de operar en ambos modos de posicionamiento. El operador puede cambiar de un modo a otro simplemente escribiendo un solo código.

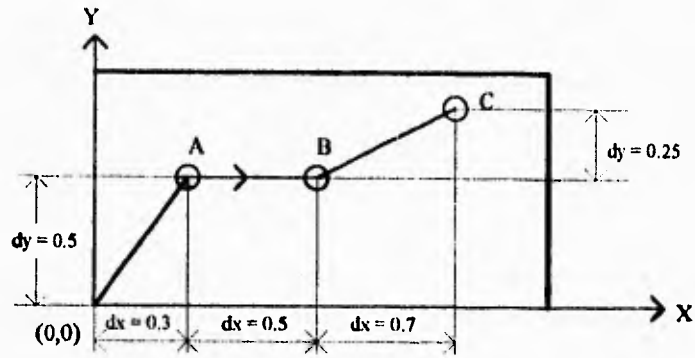


Figura 14.- Dimensionamiento para posicionamiento incremental

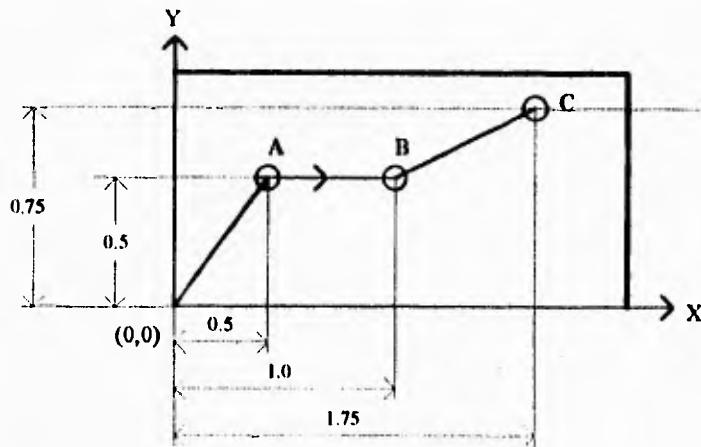


Figura 15.- Dimensionamiento para posicionamiento absoluto



### 1.12.3.- Unidades empleadas en las coordenadas de posicionamiento

En Estados Unidos, las coordenadas de posicionamiento están especificadas ya sea en el sistema Inglés (e.g., Pulgadas) o en el sistema métrico (e.g., milímetros). Fuera de Estados Unidos, el sistema mayormente empleado es el sistema métrico. Las máquinas CNC más actualizadas son capaces de procesar información en cualquiera de los sistemas. Para utilizar el sistema métrico, las piezas deben ser dimensionadas en metros. La conversión es sencilla si también se utilizan herramientas dimensionadas en metros. Si se utilizan herramientas dimensionadas en pulgadas, las dimensiones deben ser convertidas primero al sistema métrico para asegurar que las coordenadas apropiadas son proporcionadas para al maquinado. Las pulgadas y los milímetros se relacionan de la siguiente manera:

$$\begin{aligned} 1 \text{ mm} &= 0.0394 \text{ in} \\ 1 \text{ in} &= 25.4 \text{ mm} \end{aligned}$$

### 1.13.- Procedimiento de maquinado en una máquina CNC

Las actividades que se llevan a cabo durante un proceso de maquinado con una máquina CNC son realizadas en forma departamental. Tales departamentos se mencionan a continuación.

#### 1.13.1.- Diseño de la pieza

El primer paso antes de preparar cualquier parte del programa es un estudio detallado de los planos de diseño de la pieza. Los planos indican la forma de la pieza, el material, las tolerancias, acabados, tratamientos de los materiales (si existe alguno), y cualquier otro requerimiento. El programador determina a partir del plano si la pieza es factible de maquinarse por medio de una máquina CNC.

Otros factores importantes en la decisión de utilizar CNC incluyen la cantidad de partes a ser maquinadas, los requerimientos de calidad de la pieza, los costos del maquinado, la manufactura y los costos de emplear CNC. Un plan de trabajo se realiza después de tomar la decisión así como la forma en que se realizará la manufactura de la pieza.

#### 1.13.2.- Metodología de operaciones

La metodología involucra la creación de un plan indicando la secuencia y métodos que se seguirán para producir una pieza. En manufactureras pequeñas el programador toma la responsabilidad de realizar la manufactura de la pieza. En una fábrica más grande, ingenieros especializados se dedican a ésta actividad. La hoja de trabajo contiene información referente a las condiciones del material de la pieza, información necesaria para el programador antes de realizar el maquinado.

### **1.13.3.- Elección de la máquina CNC**

El programador selecciona una máquina CNC de acuerdo a su habilidad de optimizar las operaciones de maquinado necesarias para la manufactura de la pieza. La disponibilidad y la capacidad de la máquina CNC es otro de los criterios que deben tomarse en cuenta.

La capacidad de la máquina se mide en función de su carrera, potencia, tamaño del portaherramientas, precisión, y número de ejes de movimiento.

### **1.13.4.- Métodos para sujetar la pieza durante el maquinado**

Es una tarea del programador decidir cómo será sujeta la pieza durante el maquinado. Las personas encargadas de la configuración de la máquina sujetarán la pieza a la máquina según las indicaciones del programador. Si la superficie es muy compleja, el programador diseña un sujetador. Todos los puntos de sujeción, topes, prisioneros y sujetadores deben ser claramente documentados. El programador deberá evitar estos obstáculos cuando se programan los movimientos de la herramienta.

### **1.13.5.- Definición del maquinado**

Es importante definir la secuencia de maquinado dependiendo de la pieza que se fabrica. En general, el maquinado debe seguir el siguiente orden:

- Maquinado pesado
- Fresado pesado
- Perforado
- Acabado de superficies
- Fresado de acabados

Se documentan la secuencia del maquinado, así como las herramientas necesarias. Si una herramienta especial se utilizara, el departamento de herramientas correspondiente se encargará de ello. Después de documentar lo anterior, el programador prepara una lista de herramientas, mismas que deberán estar listas en la máquina en el momento en que se decida maquinar la pieza.

La velocidad de corte y la velocidad de penetración de la herramienta deben determinarse y ser incluidas.

### **1.13.6.- Programación del maquinado**

Una vez que la planeación del maquinado se ha terminado, la programación de la máquina CNC se realiza. El programador toma ciertas decisiones basadas en los requerimientos de los planos y la configuración de la fijación de la parte. Lo cual incluye:

- Establecer las coordenadas del cero programa u origen de la pieza
- Establecer la posición Zo de la parte

- Determinación del modo de programación (Absoluto o incremental)
- Determinación del sistema de unidades de programación (Inglés o métrico)

#### 1.13.7.- Documentación y configuración de la máquina CNC

Como ya se mencionó, toda la información para fijar, maquinar y la configuración deben indicarse. Una buena documentación asegurará que el procedimiento de configuración se realice rápidamente y con un mínimo de correcciones. Una hoja de configuración típica incluye la siguiente información:

- 1.- Número de la parte
- 2.- Número del proceso
- 3.- Número de operación
- 4.- Nombre del programador
- 5.- Fecha de la programación
- 6.- Número de página
- 7.- Información de la sujeción de la pieza
- 8.- Ubicación del origen de programa
- 9.- Un bosquejo de cómo será preparada la pieza en la máquina CNC
- 10.- Número de herramienta
- 11.- Descripción de las herramientas
- 12.- Operaciones de maquinado realizadas por la herramienta

Con ésta información se procede a realizar la configuración de la máquina. Generalmente el encargado de la configuración comienza por montar las herramientas. Se coloca el porta - pieza sobre la mesa de la máquina. Si es necesario se sujeta y se realiza un maquinado menor. La pieza se monta sobre el porta - pieza como lo indique el programador. Después de cargar el programa en la memoria del controlador CNC, el encargado de la configuración procede a especificar otro parámetro muy importante, que el programador no considera: la ubicación del cero pieza o cero programa  $X_0 Y_0$  con respecto al cero máquina.

La posición de reset o cero máquina se fija por el fabricante de la máquina CNC. La máquina es inicializada al inicio del proceso de configuración de la pieza. Cuando se inicializa la máquina, ésta retrae la herramienta a su altura máxima sobre la mesa ( $Z_0$  máquina) y posiciona la mesa a una posición máquina inicial preestablecida ( $X_0, Y_0$  máquina).

El origen de la pieza (cero pieza, cero programa), es la posición (0,0,0) respecto al sistema de coordenadas XYZ de la pieza. En el modo de programación absoluto, todos los movimientos se realizan con respecto a este origen. Fig.16.

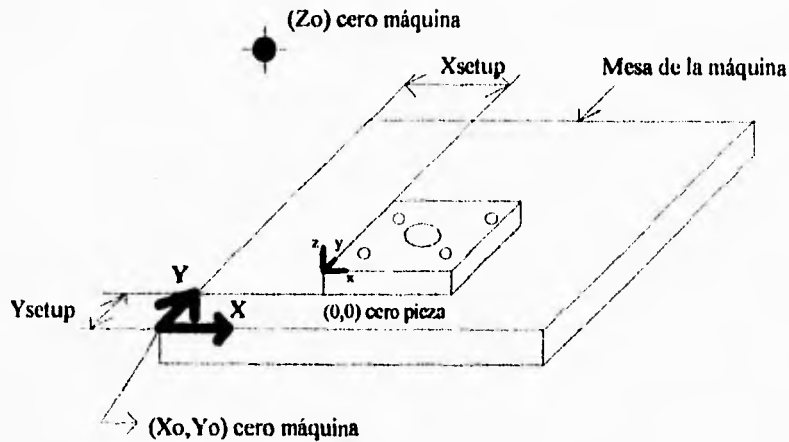


Fig.16.- Cero máquina y cero pieza

Después se miden y se especifican los valores de la longitud de la herramienta. Esta longitud puede variar, por lo cual, el controlador debe ajustar sus movimientos en el eje Z a fin de compensar estas variaciones cuando se desplaza la herramienta sobre el eje Z. Solamente puede hacerlo si el controlador conoce la distancia entre la parte baja de la herramienta y la pieza  $Z_0$ . La herramienta se retrae hasta su posición máxima sobre el eje Z, se mide la distancia mencionada y se ingresa este valor al controlador. Fig.17.

#### 1.14.- Conceptos de programación CNC

Un formato de programa permite la entrada de instrucciones al controlador de modo que las ejecute. Existen diferentes formatos de programación. El formato más utilizado es el conocido como *programación por bloques*. Fue diseñado originalmente para la programación de NC utilizando cintas perforadas y se ha mantenido en la programación CNC. Programar un equipo CNC implica la introducción de códigos a una máquina CNC con el propósito de controlar todos los movimientos de la misma. El lenguaje de programación es similar al inglés y está dividido en las siguientes partes:

Inglés	Lenguaje CNC
Caractéres ingleses	Caractéres de programa
Palabra en inglés	Palabra de programa
Enunciado en inglés	Bloque de programa
Periodo	Fin de bloque

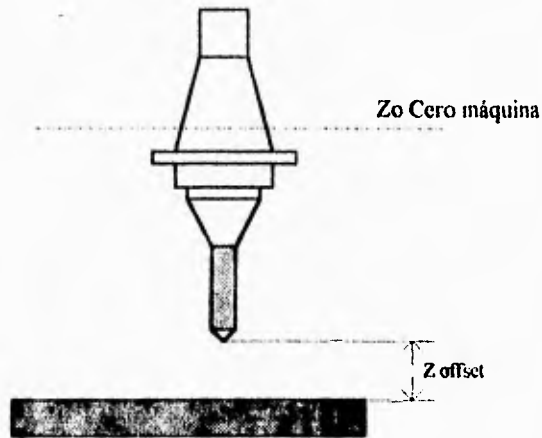


Fig.17.- Longitud de los offsets de la herramienta

Es importante mencionar que existen diferentes "dialectos" de programación. Cada controlador CNC está diseñado para aceptar un dialecto particular. Esto implica, que programa tenga que ser modificado para poder operar en otra máquina diferente. Sería imposible tratar de describir todos los diferentes códigos para cada controlador, por lo que se describirá el formato general de programación CNC según las normas ISO.

En general, la información necesaria para la ejecución de una pieza en la máquina CNC puede ser de los siguientes tipos: geométrica, tecnológica y de movimientos.

La *información geométrica* es la que contiene los datos referentes a la superficie de referencia, origen de los movimientos, etc. Puede incluir: Dimensiones de la pieza, acabado superficial, tolerancias, dimensiones de la herramienta, longitud de las carreras, etc.

La *información tecnológica* describe los datos referentes a las condiciones de maquinado, los materiales, el modo de funcionamiento de la máquina, etc. Puede incluir: La velocidad de avance, velocidad de rotación, sentido de rotación, características del material de la pieza, características de la herramienta, tipo de refrigerante, modo de funcionamiento de la máquina CNC, etc.

La *información de movimientos* indica el orden secuencial de las operaciones y del tipo de función de desplazamiento. Esta información es la que indica cómo se va a mover la máquina.

La preparación de esta información en forma inteligible para el control numérico se llama programación. El documento base del programador es el plano de la pieza: si los departamentos de estudio y diseño, la conciben en la óptica de una producción eventual en la máquina CNC, la acotación de los planos puede hacerse de manera que facilite la tarea del programador (acotación absoluta o indicación más completa de las coordenadas).

El modo de acotación de los planos ha perdido, sin embargo, su importancia frente a la programación, ya que todos los equipos de control numérico modernos permiten, en un mismo programa, utilizar tanto la programación absoluta, como la programación incremental.

Las especificaciones del material, del acabado y de sus tolerancias indicadas en los planos, condicionan la programación, puesto que éstas determinan la elección de las herramientas y las condiciones de trabajo.

#### 1.14.1.- Terminología del lenguaje de programación

La terminología siguiente es importante para la programación en CNC, ya que a partir de ella se genera un programa.

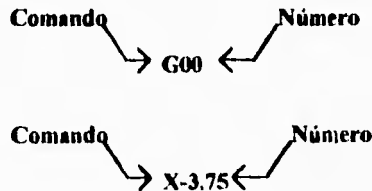
##### 1.14.1.1.- Carácter de programación

Un carácter de programación es un carácter alfanumérico o signo de puntuación. Los siguientes son caracteres de programación:

NG ;

##### 1.14.1.2.- Comandos

Un comando es una letra que describe el significado del valor numérico inmediato a ese comando. Son ejemplos de comandos:



Es importante notar que el signo menos (-) puede ser insertado entre los comandos y el valor numérico. Los valores numéricos positivos no necesitan de un signo más (+).

**1.14.1.3.- Palabras**

Los caracteres se utilizan para formar palabras. Las palabras de programa se componen de dos partes principales: un comando seguido de un número. Las palabras son utilizadas en el programa para describir información importante tal como los movimientos de la máquina y las dimensiones.

**1.14.1.4.- Bloques**

Un bloque es una línea de información completa para la máquina CNC. Se compone de una palabra o un arreglo de palabras. Los bloques pueden variar en tamaño. El programador únicamente incluye en un bloque aquellas palabras necesarias para ejecutar una función particular de la máquina. Un ejemplo de bloque es el siguiente:

```
N0020 G90 G00 X-25 Y-3.75 S200 ;
```

Cada bloque se separa uno del otro por medio de un carácter de fin de bloque, que en la mayoría de los casos lo forman el par CR/LF

**1.14.1.5.- Programa**

Un programa es una secuencia de bloques que describen en detalle la secuencia de movimientos de una máquina CNC para la manufactura de una pieza. El controlador ejecuta un programa bloque a bloque, es decir, LEE-INTERPRETA-EJECUTA cada bloque. El orden en que cada bloque aparece, es el orden en el cual se procesan. Un ejemplo de programa es el siguiente:

```
INICIO:  N0010 G90G00X0Y-1.0S800 ----- BLOQUE 1
          N0020 G43Z.1M03H01 ----- BLOQUE 2
          N0030 G99G81X1.0Z-.3R.1F5.0 ----- BLOQUE 3
          N0040 G98X-1.0 ----- BLOQUE 4
          N0050 G80 ----- BLOQUE 5
          N0060 G91G28X0Y0Z0 ----- BLOQUE 6
FIN:     N0070 M30 ----- BLOQUE 7
```

**1.15.- Ordenamiento de comandos en un bloque**

El orden en el cual los comandos aparecen en un bloque puede variar. Sin embargo la secuencia siguiente es generalmente utilizada:

**SINTAXIS GENERAL**

```
N .. G .. X .. Y.. Z.. J.. J .. K .. U .. (V .. W .. A .. B .. C) .. P .. Q.. R .. F.. S.. T.. M.. H
```

- N** Número de secuencia, indica el número secuencial del bloque  
**G** Función preparatoria, especifica el modo de operación en el cual un comando será ejecutado.  
**X,Y,Z** Palabras de dimensionamiento, designa las cantidades en los movimientos de los ejes  
**I,J,K**  
**U,V,W**  
**A,B,C**  
**P,Q,R**  
**F** Razón de avance, especifica la velocidad relativa de la herramienta de corte con respecto a la pieza.  
**S** Función del husillo, especifica la velocidad del husillo en revoluciones por minuto (rpm).  
**T** Función de herramienta, designa el número de herramienta a utilizarse.  
**M** Funciones misceláneas, designan funciones de la máquina tales como on/off del husillo, on/off del refrigerante, y otras.  
**H,D** Función auxiliar de entrada, especifica la longitud del offset de la herramienta, número de repeticiones de un ciclo fijo, y otras.

Un ejemplo de comandos arreglados en forma de bloque es el siguiente:

```
N100 G01 G42 X5.0 Y10.0 F10.0 S500 M03 D01
```

#### 1.16.- Números de programa y de secuencia: Códigos O,N

Número de programa (O)

Los programas son almacenados en la memoria del controlador CNC por medio de un número de programa. La máquina reconoce los programas de acuerdo a un código numérico. La mayoría de las máquinas pueden almacenar diferentes programas al mismo tiempo. El rango de programas puede ser de O1 a O999.

Número de secuencia (N)

Un número de secuencia es un parámetro que puede ser codificado al inicio de un bloque si es necesario. El controlador CNC ejecutará los bloques de acuerdo en el orden en que aparecen, independientemente del número de secuencia que tengan. Los números de secuencia se incluyen como una forma de localización rápida de bloques cuando se está depurando el programa. Estos números tienen un rango de N1 a N9999.

#### 1.17.- Funciones preparatorias: Códigos G

Un código G es un comando en el programa que especifica el modo en el cual una máquina CNC se mueve a lo largo de los ejes de programación. El modo de movimiento está indicado por el valor numérico inmediato al código G. Un código G es usualmente colocado al principio de un bloque, de tal manera que pueda configurar el control a un modo particular al momento de ejecutar los otros comandos de un bloque.



Los códigos G se dividen en dos principales categorías:

**Modal:** La especificación del código G permanecerá activa para todos los bloques siguientes a menos de que sea remplazado por otro código G modal.

**Nomodal:** La especificación del código G solamente tendrá efecto en el bloque en el cual aparece.

Los siguientes códigos G son muy utilizados en controladores que se basan en las normas ISO. Algunas de las funciones preparatorias previstas por la norma son:

**G00** Posicionado punto a punto.

Se hace uso de ésta función en sistemas de control numérico continuos o paraaxiales, cuando no se desea seguir una trayectoria determinada, sino la propia del control; por ejemplo, en el desplazamiento de aproximación a la pieza.

**G01, G10, G11.** Interpolación lineal

Siempre que se quiera dar a la herramienta un desplazamiento rectilíneo, debe indicarse con ayuda de éstas funciones, según sea este desplazamiento de dimensión media, grande o pequeña.

La dimensión de desplazamiento se dice que es pequeña si es menor de 100 mm, media, si está comprendida entre 100 y 1000 mm, y grande, si lo está entre 1000 y 10,000 mm.

**G02, G03, G20, G21, G30, G31.** Interpolación circular

De igual manera que para las funciones anteriores, éstas se utilizan para obtener desplazamientos circulares.

El tamaño de la dimensión se define de forma análoga a las de interpolación lineal, pero considerando ésta vez el radio. En los sistemas con mayor potencia de cálculo no se hace distinción del tamaño de la dimensión, considerando únicamente G01 para la interpolación lineal y G02, G03 para la circular.

Según la capacidad del CN, sólo es posible programar en un solo bloque recorridos de un cuarto de circunferencia, una semicircunferencia o una circunferencia entera.

**G04, G05.** Parada temporizada y parada suspensiva

Estas funciones permiten detener el funcionamiento de la máquina en el momento en que se va a realizar la operación que se describe en el bloque donde aparecen.

En el primer caso durante un tiempo determinado y, en el segundo, hasta que el operario de la orden de arranque.

La duración de la parada temporizada se programa en algunos CN con la letra F, que corresponde a la función velocidad de avance, pero interpretando el valor de ésta función en milisegundos.

Estas detenciones se disponen a lo largo del programa según las necesidades del maquinado y del operario de la máquina, como por ejemplo: cuando va a realizar alguna de las siguientes operaciones:

- 1.- Cambio de herramienta en aquellas máquinas que no posean cambio automático.
- 2.- Cambio de la posición de la pieza.
- 3.- Medición y control del trabajo realizado.
- 4.- Verificación del estado de la herramienta.
- 5 - Desalajo de viruta, etc.

**G90. Posicionamiento absoluto**

Este código es del tipo modal y se utiliza para especificar la forma en que serán medidos los desplazamientos de los ejes con respecto a un punto de origen el cual es fijo. Todos los movimientos se realizan tomando como referencia un punto fijo.

**G91. Posicionamiento incremental**

Este código es del tipo modal y se utiliza para indicar que los desplazamientos de los ejes se tomarán con respecto al último punto de maquinado, es decir, cada eje se moverá de acuerdo a un incremento indicado en un bloque de instrucciones.

Debido a que los códigos G varían de controlador a controlador, resulta poco práctico listar todos los códigos G existentes, además que cada controlador interpreta un código G de diferente manera aún siendo del mismo valor.

**1.18.- Palabras de dimensión: Códigos X, Y, Z, A, B, C, U, V, W, I, J, K, R, Q**

Las palabras de dimensionamiento se utilizan para especificar los movimientos programados de los ejes. Recordando que la programación de los ejes se basa en el sistema de coordenadas Cartesianas. La dirección positiva o negativa del movimiento sobre un eje está dada por la regla de la mano derecha. Fig. 18.

**1.19.- Velocidad de avance (feed rate): Código F**

La velocidad de avance, es la velocidad con la cual el husillo se mueve a lo largo de un eje programado. Se especifica por el valor numérico siguiente al comando F. En el sistema inglés, la velocidad de avance se expresa en pulgadas por minuto, y en el sistema métrico en milímetros por minuto.

Una especificación F es de tipo modal, y tiene efecto en el programa para todos los movimientos de la herramienta subsecuentes. La velocidad puede ser cambiada ingresando en el programa un nuevo comando F. Cantidades decimales son permitidas con éste comando.

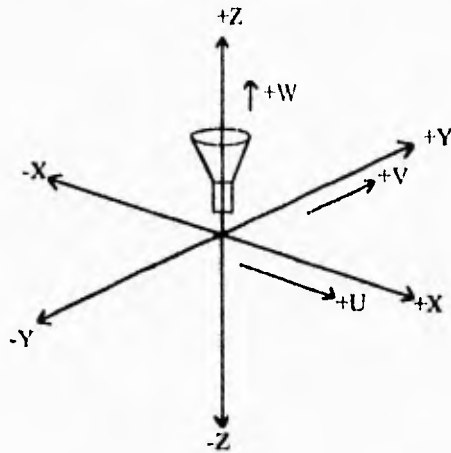


Fig.18.- Regla de la mano derecha para la asignación de los ejes

#### 1.20.- Velocidad del husillo: Código S

Los códigos S controlan la velocidad de rotación del husillo. Un valor numérico de hasta cuatro dígitos se ingresa después del comando S. El comando S es de tipo modal y permanece en efecto para todos los movimientos subsecuentes hasta que un nuevo código S sea ingresado dentro del programa o cancelado por un comando de apagado del husillo (M05). Este comando deberá ser ingresado antes de cualquier comando que implique movimientos de corte. Un ejemplo del uso de los códigos S es S1600, el cual indica que el husillo debe girar a 1600 rpm.

#### 1.21.- Funciones miscelaneas de la máquina: Códigos M

Los códigos M especifican funciones de la máquina CNC no relacionadas a dimensiones o movimientos de los ejes. A diferencia de los códigos G, no preparan al controlador para actuar de algún modo en particular cuando procesa otras palabras en un bloque, sino que ordenan al controlador ejecutar inmediatamente la función de la máquina indicada. Los números siguientes al comando M es una llamada a las funciones miscelaneas, tales como apagado del husillo, encendido del husillo, encendido del refrigerante, detener ejecución, cambio automático de herramientas, fin de programa, etc. Los códigos M se clasifican generalmente en dos grandes grupos:

- A: Aquellos que son ejecutados al inicio de los movimientos de los ejes en un bloque
- B: Aquellos ejecutados en conjunción con los movimientos de los ejes en un bloque.

Algunos de los códigos M más comunes son los siguientes.

**M00. Parada programada**

Esta instrucción detiene el funcionamiento de la máquina-herramienta hasta que el operador de la orden de reanudación.

**M02. Fin de programa**

Indica que el trabajo a concluido, rebobinándose la cinta hasta el principio del programa.

**M03, M04. Rotación del husillo en sentido antihorario u horario respectivamente.**

**M05. Parada del husillo.**

**M06. Cambio de herramienta**

Esta instrucción indica que se va a realizar un cambio de herramienta indicada por la función T.

Para una mayor referencia sobre los códigos CNC disponibles consulte la bibliografía recomendada en la sección **Bibliografía** de este documento.

## Capítulo 2

### Planteamiento, Análisis y Solución del problema

Toda solución surge de un problema. El papel del ingeniero, entre otros, es el de aplicar sus conocimientos técnicos para resolver los problemas que surgan de las necesidades de la sociedad en beneficio de ésta. Pero muchas veces las necesidades de la sociedad no aparecen a simple vista, sino que es necesario que el ingeniero las detecte. Por ello es indispensable observar cada proceso o actividad de la sociedad en general y determinar las fallas que presentan para poder generar soluciones o mejoras. Sin embargo, a medida que la tecnología avanza (y lo hace rápidamente) una solución que fue buena anteriormente podría no serlo en la actualidad, por esta razón es necesario hacer un planteamiento adecuado del problema a resolver para realizar posteriormente un estudio exhaustivo de las posibles soluciones a fin de encontrar la mejor.

#### 2.1.- Planteamiento del problema

Actualmente el uso de sistemas automatizados es muy común, sin embargo la mayoría de estos sistemas son de origen extranjero, lo cual nos pone en desventaja tanto económica como tecnológicamente ya que la única forma de generar bienes de capital es la industria.

En México son pocas las empresas que han implantado sistemas automatizados dentro de sus plantas productivas, lo cual obedece entre otras razones a la falta de recursos económicos que les permitan adquirir la tecnología necesaria. En muchos casos el empresario tiene la idea de automatización como algo muy caro y complejo, que les obligará a hacer grandes inversiones en maquinaria y equipo nuevos. La crisis por la que atraviesa nuestro país se debe en gran parte a la diferencia entre la cantidad de importaciones y la cantidad de exportaciones que hacemos. Dichas importaciones incluyen, entre otras cosas, maquinaria, que aunque tecnológicamente muy avanzadas, muchas veces rebasan las necesidades de la industria nacional y (la gran mayoría) resultan muy costosas, además de que generan dependencia tecnológica hacia el extranjero. Es por ello necesario realizar procesos de automatización tratando de aprovechar los recursos que se tienen a nivel nacional y aplicando la tecnología que comercialmente se encuentra en México y si es posible con tecnologías cien por ciento mexicanas, a fin de lograr sistemas automatizados o semiautomatizados que permitan a las empresas aumentar la calidad de sus productos sin necesidad de hacer grandes inversiones y aprovechando los recursos que se tengan.

Veamos un ejemplo de como transformar una máquina-herramienta convencional en una máquina CNC aplicada al perforado de los circuitos impresos en la industria de la electrónica.

## Capítulo 2

### Planteamiento, Análisis y Solución del problema

Toda solución surge de un problema. El papel del ingeniero, entre otros, es el de aplicar sus conocimientos técnicos para resolver los problemas que surgan de las necesidades de la sociedad en beneficio de ésta. Pero muchas veces las necesidades de la sociedad no aparecen a simple vista, sino que es necesario que el ingeniero las detecte. Por ello es indispensable observar cada proceso o actividad de la sociedad en general y determinar las fallas que presentan para poder generar soluciones o mejoras. Sin embargo, a medida que la tecnología avanza (y lo hace rápidamente) una solución que fue buena anteriormente podría no serlo en la actualidad, por esta razón es necesario hacer un planteamiento adecuado del problema a resolver para realizar posteriormente un estudio exhaustivo de las posibles soluciones a fin de encontrar la mejor.

#### 2.1.- Planteamiento del problema

Actualmente el uso de sistemas automatizados es muy común, sin embargo la mayoría de estos sistemas son de origen extranjero, lo cual nos pone en desventaja tanto económica como tecnológicamente ya que la única forma de generar bienes de capital es la industria.

En México son pocas las empresas que han implantado sistemas automatizados dentro de sus plantas productivas, lo cual obedece entre otras razones a la falta de recursos económicos que les permitan adquirir la tecnología necesaria. En muchos casos el empresario tiene la idea de automatización como algo muy caro y complejo, que les obligará a hacer grandes inversiones en maquinaria y equipo nuevos. La crisis por la que atraviesa nuestro país se debe en gran parte a la diferencia entre la cantidad de importaciones y la cantidad de exportaciones que hacemos. Dichas importaciones incluyen, entre otras cosas, maquinaria, que aunque tecnológicamente muy avanzadas, muchas veces rebasan las necesidades de la industria nacional y (la gran mayoría) resultan muy costosas, además de que generan dependencia tecnológica hacia el extranjero. Es por ello necesario realizar procesos de automatización tratando de aprovechar los recursos que se tienen a nivel nacional y aplicando la tecnología que comercialmente se encuentra en México y si es posible con tecnologías cien por ciento mexicanas, a fin de lograr sistemas automatizados o semiautomatizados que permitan a las empresas aumentar la calidad de sus productos sin necesidad de hacer grandes inversiones y aprovechando los recursos que se tengan.

Veamos un ejemplo de como transformar una máquina-herramienta convencional en una máquina CNC aplicada al perforado de los circuitos impresos en la industria de la electrónica.

Cuando se fabrican sistemas electrónicos, todo el montaje de los componentes se realiza sobre una tarjeta o circuito impreso. El circuito impreso ofrece soporte mecánico a la circuitería así como el conexionado de los componentes según el circuito eléctrico del sistema.

Anteriormente el proceso de diseño de un circuito impreso se realizaba prácticamente de manera artística, era un trabajo tedioso y tardado. Afortunadamente en la actualidad este proceso se ha modernizado gracias al avance de las computadoras y al desarrollo de los programas de CAD. Uno de los programas de diseño electrónico asistido por computadora más conocidos y potentes es el TANGO.PCB PLUS de ACCEL INC. el cual nos permite diseñar circuitos impresos en forma manual o automática. Con TANGO es posible diseñar circuitos impresos de una o dos caras, generar el grabado de los componentes (silk), etcétera. Una vez realizado el diseño, se imprime y se fabrica. Sin embargo, algunas compañías que se dedican a la fabricación de circuitos impresos se enfrentan a otro problema: el perforado de los agujeros que permiten el ensamble de los componentes a la tarjeta impresa. Este, aunque es un problema poco relevante, resulta crítico cuando el circuito impreso presenta miles de agujeros o cuando el número de tarjetas a fabricar es muy grande. Estos agujeros deben realizarse con una precisión de hasta milésimas de pulgada, lo cual resulta en un trabajo sumamente complejo. Por ello es poco práctico realizar este proceso en forma manual. Entonces, ¿Cómo podemos realizar el perforado en forma automatizada con la precisión requerida? La respuesta: utilizando una máquina CNC. Esta solución podría parecer la menos indicada ya que, para utilizar una máquina CNC, es necesario programarla. Sin embargo esa tarea queda resuelta en el momento de utilizar TANGO ¿Por qué? Porque este programa nos permite generar un programa CNC para el perforado de los agujeros del circuito impreso que se diseñó, por lo que el objetivo es aprovechar esta ventaja del programa TANGO para convertir una máquina-herramienta universal en una máquina CNC cuya programación será realizada en forma automática o manual para obtener el perforado de los agujeros de un circuito impreso.

Del análisis anterior se observa que no es necesario adoptar tecnologías nuevas, sino que es posible aprovechar los recursos de los cuales se dispone. Es importante enfatizar que el control numérico no se limita únicamente al taladrado de piezas, pero constituye la función más simple de un controlador CNC.

De todo lo anterior, el problema consiste en convertir una máquina-herramienta convencional de uso manual en una máquina perforadora CNC, cuya principal aplicación está orientada a la fabricación de circuitos impresos de uso experimental.

## 2.2.- Análisis del problema

Para fabricar circuitos impresos se requiere de una máquina capaz de desplazar su herramienta de corte (taladro) en las direcciones X-Y-Z. Los movimientos en los ejes X-Y definen la posición de la isla, punto de soldadura o perforación a realizar. La separación mínima entre las patillas de un circuito integrado es de 0.100 in. es decir, 2.54 mm. El espesor de una tarjeta de baquelita es de aproximadamente de 1/16", es decir, 1.587 mm. La perforación de un agujero se realiza en la dirección del eje Z.

Partiendo de la información anterior, se decidió resolver el problema utilizando una máquina-herramienta de pequeña escala utilizada para realizar trabajos de modelismo y piezas pequeñas. La máquina-herramienta modelo UNIMAT de EMCO Co. posee las siguientes características. Es una máquina de tipo universal, es decir, se pueden realizar operaciones de torneado, fresado y taladrado. Los ejes de movimiento incluyen las direcciones X-Y-Z. Los movimientos en las tres direcciones se pueden realizar en forma independiente y todos a la vez si es necesario. El movimiento en el eje X se logra a partir de un tornillo sin fin cuyo avance es de 1 mm/rev el cual puede posicionarse con una manivela con una resolución de 0.05 mm (0.00196 in); la carrera del tornillo es de 255.6 mm (10.0629 in). El movimiento en el eje Y se logra de la misma manera que para el eje X, sin embargo la carrera es más corta, siendo ésta de 50.0 mm (1.9685 in). El movimiento en el eje Z se logra a partir de un mecanismo Tornillo- cremallera-palanca el cual posee un resorte de apoyo al movimiento en sentido +Z. La carrera de este mecanismo de avance es de 35 mm con el resorte comprimido al 100%. La transmisión de potencia en la herramienta se realiza por medio de un arreglo de poleas acopladas a un motor universal de corriente alterna 125 Vac/1.5 A con dos velocidades. Fig. 19.

Para poder taladrar circuitos impresos, fue necesario realizar algunas modificaciones a la máquina basándose en los tres siguientes puntos:

- 1.- Automatizar los movimientos de la máquina-herramienta y modificarla de una manera mínima a fin de obtener una máquina CNC.
- 2.- Lograr los requerimientos de precisión necesarios.
- 3.- Utilizar los recursos que proporciona TANGO.

Antes de proponer soluciones, es necesario definir el conjunto de entradas y salidas que el sistema requiere. El sistema necesita de un programa CNC el cual deberá controlar los movimientos de los ejes y de la herramienta de corte en forma automática, a partir de este programa, el sistema deberá interpretarlo y ejecutarlo. Para el caso del taladrado, es necesario únicamente programar los movimientos de los ejes X-Y y el eje Z puede mantenerse con una trayectoria fija. Sin embargo utilizando ésta opción se limitaría el uso de la máquina; por ello es necesario poder programar también el eje Z a fin de aumentar la versatilidad de la máquina.

Existen varias soluciones a este problema, de las cuales consideré la siguientes tres:

- 1.- Control por medio de una computadora personal.
- 2.- Control por medio de una microcomputadora.
- 3.- Control por medio de una computadora personal - microcomputadora



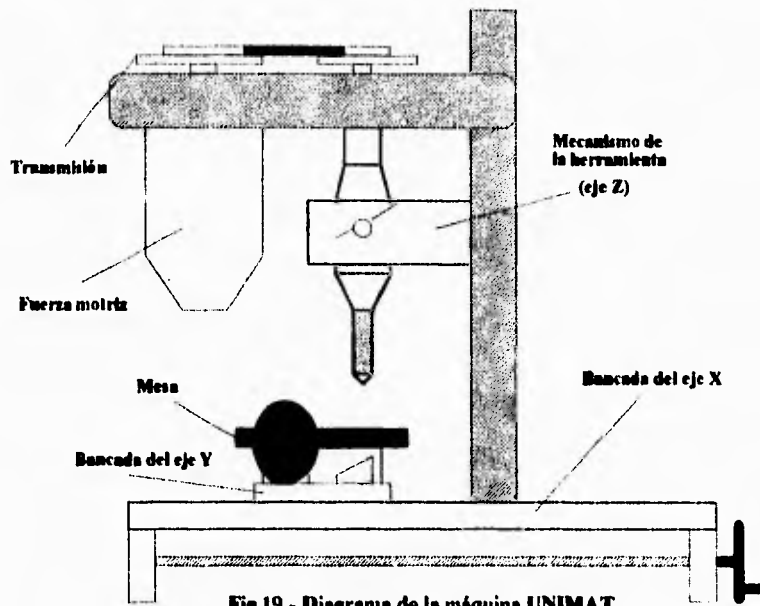


Fig.19.- Diagrama de la máquina UNIMAT

Control por medio de una computadora personal. Esta opción es sin duda la más sencilla en muchos aspectos ya que ofrece la posibilidad de aprovechar los recursos, tanto electrónicos como lógicos, que ofrece una computadora personal. El software podría ser de alto nivel y el control de los movimientos podría realizarse por medio de uno de los puertos de la computadora con ayuda de un hardware adicional mínimo. Sin embargo esta solución presenta un problema: la velocidad de direccionamiento se ve afectada por una interrupción de tiempo real cada 30 ms aproximadamente. Aunado a esto, no resulta óptimo la utilización del potencial tanto de software como de hardware de una computadora para operaciones, que aunque con cierta complejidad, se limitan a control. En cierto modo se desperdiciarían recursos.

Control por medio de una microcomputadora. Esta solución ofrece la ventaja de diseñar un hardware específico y optimizado a las necesidades de la máquina-herramienta, sin embargo presenta mucho mayor complejidad tanto en programación como en hardware, y si añadimos a esto que las posibilidades de interacción entre el usuario y la máquina se ven limitadas. Esto se debe principalmente a que sería necesario añadir más componentes al hardware lo cual resulta en un sistema más complejo al requerir un número mayor de componentes electrónicos y por ello menos confiable.

Control por medio de una computadora personal - microcomputador. Esta solución ofrece la versatilidad de utilizar los recursos de una PC en conjunto con un hardware específico a las necesidades de la máquina herramienta, sin necesidad de esclavizar la computadora, pero sí aprovechar las facilidades que ésta ofrece con la programación de alto nivel, lo que permitiría un programa de interacción usuario-máquina más óptimo y amigable, así como el manejo de medios masivos de almacenamiento de información, lo cual simplificaría el hardware adicional. Esta opción es la que se utilizó en el desarrollo de este proyecto ya que consideramos que es mucho más sencillo utilizar un hardware de uso general tanto de la PC como del sistema adicional, y hacer uso de un recurso más versátil y cómodo como lo es el software, a fin de que el mantenimiento del sistema sea más fácil y cualquier modificación posterior se realice con un mínimo de requerimientos de hardware.

### 2.3.- Solución del problema

El sistema fue desarrollado en base a un diseño modular y descendente a fin de identificar perfectamente cada una de las siguientes partes.

1.- El Software.- El software del sistema lo forman el programa en lenguaje de alto nivel de la PC y el programa en lenguaje ensamblador del microcontrolador utilizado. El programa de alto nivel en la computadora tiene dos objetivos fundamentales: servir de interfaz entre el usuario y la máquina-herramienta así como servir de medio de edición, depuración y almacenamiento de los programas CNC. El programa en lenguaje ensamblador constituye el soporte lógico del hardware respectivo del sistema de control de la máquina y tiene como objetivos administrar y controlar los movimientos de la máquina-herramienta y servir como medio de almacenamiento temporal para la ejecución de los programas CNC.

2.- El hardware.- Lo constituyen el hardware de la PC y la tarjeta del controlador de la máquina. El hardware de la PC se basa en la arquitectura de la IBM-PC y tiene como objetivo servir de interfaz entre el usuario y la máquina herramienta. La tarjeta del controlador de la máquina tiene como objetivos permitir que la máquina funcione de manera autónoma y administrar los recursos electrónicos del sistema de control.

3.- La etapa de control. La constituye el programa en lenguaje ensamblador del microcontrolador y tiene como objetivo ejecutar las instrucciones correspondientes para posicionar los ejes y administrar las señales de los sensores de la máquina. El tipo de control de posición que se realiza es de lazo abierto. Se utilizó este tipo de control debido a su sencillez y a que actualmente, con el auge que han tenido los motores de pasos, resulta mucho más práctico y barato, además de que el nivel de precisión requerido no demanda otro tipo de control de posición.

4.- La etapa de potencia. La constituyen la fuente de alimentación y los circuitos amplificadores de corriente (buffers) para mover los motores. Tiene como objetivo suministrar la energía necesaria a todos los componentes del sistema.

5.- La etapa de acoplamiento mecánico. La forman los engranes y mecanismos necesarios para el movimiento de los ejes así como el soporte mecánico de todas las partes de la máquina.

La figura No. 20 muestra cada una de las partes del sistema.

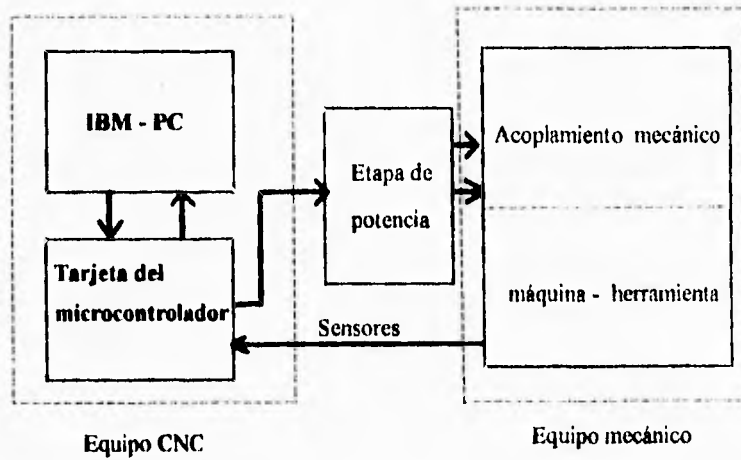


Fig. 20.- Diagrama a bloques del sistema UNIMAT

5.- La etapa de acoplamiento mecánico. La forman los engranes y mecanismos necesarios para el movimiento de los ejes así como el soporte mecánico de todas las partes de la máquina.

La figura No. 20 muestra cada una de las partes del sistema.

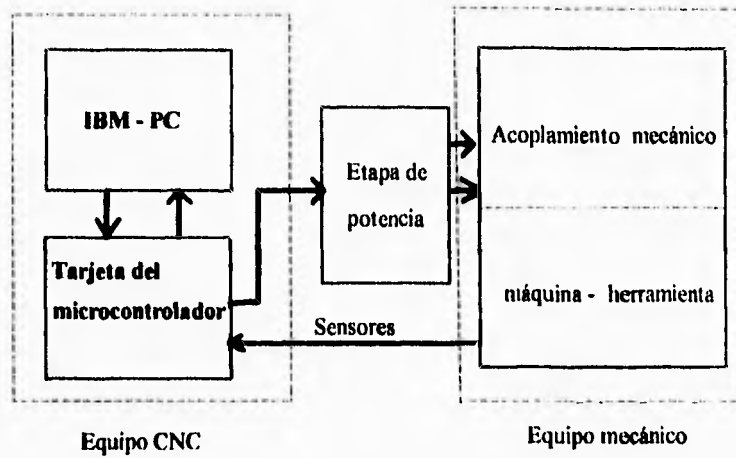


Fig. 20.- Diagrama a bloques del sistema UNIMAT

## Capítulo 3

### Diseño e implementación del sistema

El diseño de un sistema requiere de un análisis detallado de los posibles circuitos electrónicos a utilizarse a fin de lograr un desarrollo óptimo que asegure un buen y confiable funcionamiento. Una vez planteado el problema, es posible elegir los componentes electrónicos que se ajusten a las necesidades requeridas para no utilizar recursos demasiados sobrados o que limiten el funcionamiento del sistema. A partir de este momento el sistema implementado se llamará *UNIMAT : CNC Drilling Machine*.

Este capítulo habla acerca del diseño e implementación electrónicos del sistema UNIMAT. El hardware del sistema UNIMAT está desarrollado en forma modular y descendente, lo cual permite identificar las partes que lo componen.

El software se refiere a la programación de alto nivel de la IBM-PC. El diseño e implementación también incluye las etapa de potencia y mecánica.

#### 3.1.- Diseño del hardware y del software del controlador

##### 3.1.1.- Diseño del hardware

El diseño del hardware del sistema UNIMAT lleva implícito un estudio detallado a fin de encontrar la configuración óptima atendiendo los siguientes aspectos:

- Funcionalidad
- Componentes estándares
- Mercado nacional
- Costo
- Mantenimiento y posibles expansiones posteriores

Cabe mencionar que el uso de un sistema personal IBM-PC proporciona un soporte electrónico muy grande, sin embargo, fue necesario hacer un balance para aprovechar al máximo la PC sin desperdiciar los recursos electrónicos que ofrece.

Los objetivos del hardware del sistema UNIMAT son los siguientes:

- Debe ser un sistema electrónico capaz de realizar las operaciones básicas que ofrece una máquina CNC para el taladrado
- Debe administrar las señales de control para posicionar la herramienta de corte (taladro) por medio de motores de pasos en base a un sistema de control de lazo abierto
- Además debe tener la capacidad de funcionamiento esclavo o autónomo
- Debe ofrecer un sistema de comunicación serial con la IBM-PC de acuerdo a la norma RS232C para el intercambio de información y permitir modificaciones posteriores fácilmente.

El hardware del sistema UNIMAT tiene el siguiente conjunto de entradas:

- Un fotosensor de posición  $X_{home}$  (PTOE0).
- Un fotosensor de posición  $Y_{home}$  (PTOE1).
- Un sensor de posición  $Z_{home}$  (PTOE2).
- Un sensor de posición  $Z_{fin}$  (PTOE5).
- Una línea de recepción serial Rx (PTOD0).

El conjunto de salidas es el siguiente:

- 4 bits de control para la secuencia de pasos del motor en el eje X ( $X_3X_2X_1X_0$ ) (PTOG0 - PTOG3).
- 4 bits de control para la secuencia de pasos del motor en el eje Y ( $Y_3Y_2Y_1Y_0$ ) (PTOG4 - PTOG7).
- 4 bits de control para la secuencia de pasos del motor en el eje Z ( $Z_3Z_2Z_1Z_0$ ) (PTOA0 - PTOA3).
- 1 bit de control del encendido del motor del taladro (PTOD5).
- 1 bit para el control de un LED indicador cada 500 ms (PTOA5).
- 7 bits para señalización de las operaciones que realiza
- Una línea de transmisión serial Tx (PTOD1).

Atendiendo a los objetivos planteados y al conjunto de entradas y de salidas del sistema UNIMAT, se decidió utilizar un sistema basado en un microcontrolador de la familia MC68HC11F1 de Motorola debido a lo siguiente:

- Ofrece un conjunto de instrucciones acorde a nuestros requerimientos
- Es un microcontrolador que se encuentra en el mercado nacional
- Se tienen los soportes técnico y teórico para su implementación
- Es de bajo costo y de velocidad aceptable
- Su capacidad de direccionamiento de memoria

El MC68HC11F1 es un microcontrolador de 8 bits cuyas características son las siguientes. Es de tecnología HCMOS, tiene 1024 bytes de memoria SRAM y 512 bytes de memoria EEPROM. Su capacidad de direccionamiento de memoria es de 64K bytes. Tiene un timer de 16 bits y un contador de eventos de 8 bits. Cuenta además con un sistema de interrupción en tiempo real, un sistema de comunicación serial NRZ, un convertidor A/D de 8 canales de 8 bits c/u, cuatro puertos bidireccionales de 8 bits c/u, dos puertos de salida de 8 bits c/u y un puerto de entrada de 8 bits c/u. Dos acumuladores de 8 bits c/u y dos registros índice de 16 bits c/u. Puede trabajar en cuatro modos:

- Single Chip
- Expandido no-multiplexado
- Bootstrap
- Test

La velocidad de operación es de 8 Mhz (ésta velocidad de operación es la de uso común) y su voltaje de operación es de 5 Vdc, lo cual lo hace compatible con la tecnología TTL.

El modo de operación en el que opera el HC11F1 en este sistema es el modo Expandido no-multiplexado debido a que se requiere el uso de memoria externa adicional tanto para el programa en lenguaje ensamblador en memoria ROM como para el almacenamiento de los códigos CNC en memoria RAM. La figura No. 21 muestra el diagrama a bloques del 68HC11F1.

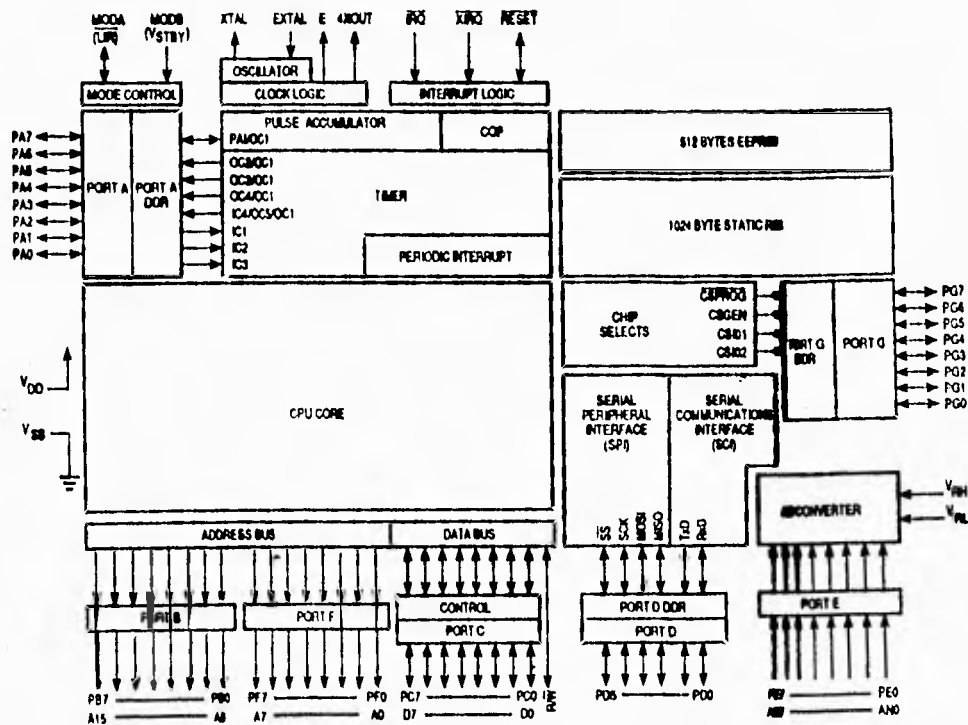


Figura No. 20.- Diagrama a bloques del microcontrolador 68HC11F1

El sistema UNIMAT consta de una memoria de programa ROM externa es de 8K bytes lo cual resulta suficiente para las características del sistema, para posibles modificaciones siguientes y si es necesario aumentar esta cantidad de memoria según se requiera. La memoria RAM externa está destinada para almacenar 8K bytes de información de programa en códigos CNC y nuevamente con la posibilidad de extenderse si resulta necesario.

El sistema no cuenta con despliegues en forma de textos ya que consideramos no es necesario y complicaría el sistema en una etapa que no es de importancia, por lo que únicamente cuenta con señales luminosas por medio de LEDs a saber:

- 1 bit indicador de inicialización (reset) de ejes (PTOD4)
- 1 bit indicador de motor X en operación (PTOD2)
- 1 bit indicador de motor Y en operación (PTOD3)
- 1 bit indicador de motor Z en operación (PTOD5)
- 1 bit indicador de Recepción de programa (PTOA6)
- 1 bit indicador de buffer de programa lleno (PTOA7)
- 1 bit indicador de modo de operación manual (PTOA4).

Las comunicaciones entre el controlador y la IBM-PC se realizan utilizando la norma RS232C para comunicaciones seriales en formato NULL-MODEM, lo cual deriva en un cable trenzado de tres hilos, Tx, Rx y GND. La velocidad de transmisión es de 2400 bauds para 8 bits de información, 1 bit de inicio, 1 bit de parada y no paridad. La velocidad de transmisión puede aumentar hasta 9600 bauds, sin embargo consideramos suficiente una velocidad de 2400 bauds debido a las características del controlador y de la información que se transmite.

El diagrama eléctrico del sistema UNIMAT se muestra en el apéndice #1 bajo el título de "CONTROL.SCH".

### 3.1.2.- Diseño del software del controlador

Habiéndose establecido las características del hardware se procedió al diseño del software del controlador basándose en las siguientes características de programación:

- Programación estructurada: Por medio de esta técnica de programación se implementaron las operaciones lógicas definidas dentro de la programación estructurada tales como IF\_THEN\_ELSE, DO\_WHILE y SWITCH\_CASE, esto con el fin de facilitar la programación desde el punto de vista lógico utilizando las instrucciones del lenguaje ensamblador disponibles en el 68HC11F1.

La implementación del bloque IF\_THEN\_ELSE básicamente tiene la siguiente estructura. Por ejemplo para implementar la estructura lógica

```
if (sensor) then led = on else led =off
```



la implementación en lenguaje ensamblador sería:

```
SENSOR    RMB 1
TRUE      EQU #$01
FALSE     EQU #$00

IFSENSOR: LDAA SENSOR
          CMPA #TRUE
          BNE LED_OFF
LED_ON:   -
          -
          ...
          JMP END_IF
LED_OFF:  -
          -
          ...
END_IF:   -
          ...
```

La implementación del bloque DO\_WHILE básicamente tiene la siguiente estructura. Por ejemplo para implementar el bucle

```
do{ -
-
...
}while (xini < xfinal)
```

la programación correspondiente en lenguaje ensamblador sería:

```
XINI DW 2
XFIN DW 2

DO:   -
      -
      -
      ...
      LDX XINI
      CPX XFIN
WHILE: BLO DO
      -
      ...
```

La programación correspondiente del bloque SWITCH-CASE básicamente tiene la siguiente estructura:

```
switch (opción){
case 1: -
-
break;
case 2: -
-
break;
default: -
-
}
```

La programación correspondiente en lenguaje ensamblador sería:

```
SWITCH:  LDAA OPCION
CASE_1:  CMPA OPCION1
         BNE CASE_2
         -
         ...
CASE_2:  JMP ENDCASE
         CMPA OPCION2
         BNE DEFAULT
         -
         ...
DEFAULT: JMP ENDCASE
         -
         ...
ENDCASE: -
```

- Programación en forma modular, lo cual consiste en la división de un problema en diferentes partes más pequeñas, lo que permite la simplificación, depuración, corrección y programación más eficiente de un programa, a la vez que proporciona un diseño descendente. Además la probabilidad de error

disminuye. Otra ventaja consiste en que el mantenimiento del programa puede realizarse por varios programadores ya que no importa cómo lo hace sino qué hace cada módulo. La modularidad permite entonces crear una serie de procedimientos que pueden incorporarse en cualquier algoritmo que lo necesite, es decir, se tiene una biblioteca de módulos o procedimientos.

Dentro del programa ensamblador del sistema UNIMAT cada módulo va acompañado de un comentario que indica qué es lo que hace.

- Optimización del código, se refiere al proceso de refinamiento del programa, es decir, una vez que se ha resuelto el problema específico (o módulo), se procede a simplificar el número de instrucciones implicadas en el procedimiento a fin de ahorrar espacio de memoria y de aumentar su velocidad de operación.

El mapa de memoria se muestra en la figura No.21.

\$0000	Variables y Stack
\$03FF	
\$1000	Registros de configuración
\$105F	
\$2000	Memoria de datos de códigos CNC
\$3FFF	
\$BFF0	Memoria EEPROM
\$BFFF	
\$E000	Memoria de programa y vectores de interrup.
\$FFFF	

Fig. No. 21.- Mapa de memoria del sistema UNIMAT

La transferencia de información entre los diferentes módulos del programa se realiza por medio de variables que residen en la ZONA DE VARIABLES Y STACK de la memoria RAM interna del microcontrolador, lo cual permite un manejo global de la información utilizando un direccionamiento a base de apuntadores a cada una de las variables.

El software de control desarrollado está basado en el modo de posicionamiento punto a punto debido a que únicamente se especifican los puntos a los que tiene que dirigirse la herramienta de corte sin importar la trayectoria que ésta tome.

Básicamente el programa se divide en dos módulos:

- El Módulo automático o Stand-by.
- El Módulo de operación manual.

El módulo automático o stand-by consiste de un programa principal el cual recibe los comandos de la línea de comunicación serial por medio de una interrupción. Estos comandos incluyen borrado de memoria de programa CNC (CLEAR), verificar comunicación, ejecutar programa, almacenamiento de un programa CNC en la memoria, inicialización de los ejes (Reset ejes) y modo de operación manual. El programa indica que está listo para recibir comandos por medio de un LED parpadeante. Los comandos respectivos son enviados al controlador por medio de la computadora, éste los decodifica y realiza la operación correspondiente:

**Etapas de borrado de memoria.** Consiste en eliminar cualquier programa CNC que se encuentre en la memoria RAM de programa, así como la inicialización de las variables correspondientes al administrador de la memoria.

**Etapas de verificación de la comunicación.** Esta etapa permite al controlador verificar si existe comunicación con la computadora y viceversa utilizando un protocolo de comunicación sencillo que consiste en la retransmisión del byte recibido hasta tres veces. Este protocolo es controlado por software en la PC.

**Etapas de almacenamiento de un programa.** Se encarga del almacenamiento del programa CNC en la memoria RAM externa ubicada entre las direcciones 2000H y 3FFFH. El modo de almacenamiento está basado en un sistema FIFO (First-In, First-Out) en forma de buffer circular, lo cual permite que se almacenen programas de tamaño mayor a 8K bytes. Sin embargo si el programa es mayor a 8K bytes, la transmisión se interrumpe hasta que se hayan ejecutado los 8K bytes almacenados.

**Etapas de ejecución de un programa.** Se encarga de ejecutar el programa CNC almacenado en la memoria RAM externa efectuando los movimientos programados de los ejes de la máquina-herramienta.

**Etapas de inicialización de los ejes de la máquina.** Esta etapa se encarga del posicionamiento de los ejes X, Y, Z en su posición respectiva cero máquina  $X_{home}$ ,  $Y_{home}$ ,  $Z_{home}$ , así como de inicializar las variables respectivas a cada eje.

**Etapas de operación manual.** Se encarga de preparar la máquina para su operación manual (módulo de operación manual) utilizando el teclado de la IBM-PC.

El diagrama de flujo correspondiente al módulo automático se muestra en el apéndice #1 con el título "AUTO.FLW".

El módulo de operación manual permite un control completo de la máquina, similar al que ofrece el modo automático, por medio del teclado de la PC. Estas operaciones incluyen además posicionamiento de la herramienta, encendido del husillo de la herramienta y movimiento de los ejes. Este modo de operación requiere de una comunicación permanente con la computadora.

El diagrama de flujo correspondiente al módulo automático se muestra en el apéndice #1 con el título "MANUAL.FLW".

El programa en lenguaje ensamblador se muestra en el apéndice #1 con el nombre de "CONTROL.ASM".

### **3.2.- Diseño del software**

El sistema UNIMAT cuenta con un programa de alto nivel el cual tiene los siguientes objetivos:

- Proporcionar un ambiente interactivo entre el usuario y la máquina
- Permitir la edición de programas CNC.
- Controlar y administrar las funciones de la máquina en forma manual o automática
- Traducir los códigos de programación y generar un programa ejecutable por el controlador
- Establecer y controlar la comunicación con el controlador en forma serial compatible con la norma RS232C
- Establecer los parámetros de configuración del sistema que se requieran
- Permitir el mantenimiento del programa fácilmente

La interfaz con el usuario se lleva al cabo por medio de un sistema de menús de acuerdo a la tecnología IDE ( Integrated Development Environment), el cual permite tener un control de las diferentes funciones del programa por medio de menús de persiana y ventanas que indican las diferentes operaciones del sistema. Inicialmente se había planteado una interfaz bajo un ambiente gráfico o GUI (Graphical User Interface) pero los requerimientos no exigían de un ambiente de trabajo de esas características.

El lenguaje de programación utilizado es el TURBO C++ de Borland el cual permite una lógica de programación estructurada u orientada a objetos (OOP) además de ser compatible con el lenguaje "C" de Kernigham y Ritchie. Se decidió utilizar este lenguaje debido a que es un lenguaje de nivel medio utilizado por los profesionales del software, el cual permite controlar el hardware de la computadora, por su velocidad, por su tamaño del código generado y por su direccionamiento dinámico de datos.

**UNIMAT.EXE** está diseñado con base en la programación estructurada y modularizada lo cual permite su fácil mantenimiento. El lenguaje ensamblador está mezclado en algunas rutinas para aumentar la velocidad de direccionamiento y optimización del código.

Una de las partes más importantes del programa es el editor. El editor se encarga de leer los programas CNC y colocarlos en su memoria de edición. Sin embargo, el tamaño de los programas es variable lo cual no permite que se fije un espacio específico de memoria para el editor, sino que debe utilizar la memoria necesaria según se requiera. Lo anterior se obtiene por medio de un almacenamiento dinámico de información que permite utilizar toda la memoria disponible de la computadora si es necesario o utilizar solamente una parte de ella. El resultado es un aprovechamiento eficiente de los recursos de memoria de la computadora.

Otro aspecto importante en el desarrollo del software es el sistema operativo MS-DOS. Debido a las características del programa, es importante entender cómo MS-DOS administra los diferentes recursos de la computadora. Entender además el sistema de interrupciones hardware, de software y la forma de direccionamiento de la memoria.

La comunicación entre la computadora y el controlador se controla por medio del sistema **UNIMAT.EXE**. Este se encarga de controlar el flujo de información y de la configuración de los parámetros de comunicación serial. La comunicación serial se realiza por medio de la tarjeta adaptadora de comunicaciones asíncronas el cual contiene el circuito integrado 8250 o 16550 el cual es una versión más avanzada del 8250. Debido a que este no es un sistema de comunicaciones estrictamente hablando, y el flujo de información no es masivo, dejamos el manejo del hardware al BIOS del sistema por medio de la interrupción 14H, lo cual resulta más sencillo.

Las comunicaciones entre el controlador y la IBM-PC se realizan utilizando la norma RS232C para comunicaciones seriales en formato NULL-MODEM, lo cual deriva en un cable trenzado de tres hilos, Tx, Rx y GND. La velocidad de transmisión es de 2400 bauds para 8 bits de información, 1 bit de inicio, 1 bit de parada y no paridad. La velocidad de transmisión puede aumentar hasta 9600 bauds, sin embargo consideramos suficiente una velocidad de 2400 bauds debido a las características del controlador y de la información que se transmite.

El flujo de información se controla por medio del siguiente protocolo:

- 1 - Se envía el byte de información
- 2 - El controlador lo recibe y lo decodifica. La computadora espera 10 ms
- 3 - Si transcurren 10 ms se retransmite el byte. Se repite el paso #2
- 4 - Se realiza el paso #3 tres veces.
- 5 - Si después de tres veces no hay respuesta, se manda un mensaje de error y se cancela la comunicación.
- 6 - El controlador regresa un "echo" del byte recibido.
- 7 - La computadora lo decodifica y continúa la transmisión. Salta al paso #1

El programa del sistema UNIMAT se encarga de leer programas CNC de los diferentes medios de almacenamiento de la PC y colocarlos en su memoria de edición. También es capaz de permitir la edición de un programa y almacenarlo. Los programas deben estar escritos en formato CNC utilizando los códigos normalizados y que se encuentren disponibles en el controlador. Una vez terminada la edición se encarga de generar un código ejecutable y optimizado de los códigos editados, los cuales serán enviados al controlador por el puerto serial. Cuando se genera el código ejecutable, se verifican los errores los códigos no existentes o no implementados en el controlador.

El sistema permite también el control manual de la máquina-herramienta. En esta etapa se presenta una pantalla en la computadora que muestra el estado de la máquina desplegando la siguiente información:

- Sistema de unidades empleado: milímetros o pulgadas
  - Posición del eje X en pulgadas o en milímetros
  - Posición del eje Y en pulgadas o en milímetros
  - Incremento de avance en los ejes X-Y-Z
  - Estado del husillo de la herramienta de corte: ON- OFF
  - Menú de opciones para la operación manual de la máquina y de sus movimientos

Además el programa tiene una opción de aplicación específica para la utilización del programa TANGO. Básicamente el proceso de operación de UNIMAT.EXE es el siguiente.

- 1.- En el símbolo del sistema escriba lo siguiente  
CAUNIMAT [enter]
- 2.- UNIMAT presenta una pantalla de presentación. Pulse cualquier tecla
- 3.- Se muestra el ambiente de trabajo con la barra de menú en la parte superior de la pantalla.
- 4.- Seleccione el menú *EDITAR*.
- 5.- Escriba el programa CNC que desea
- 6.- Una vez terminada la edición del programa, pulse la tecla ESC.
- 7.- Seleccione el menú *ARCHIVO* y elija la opción GRABAR.
- 8.- Seleccione el menú *CODIGO* y elija la opción TRANSMITIR
- 9.- Encienda el controlador y espere a que el LED empiece a parpadear
- 10.- Conéctelo al puerto serial COM1 de la computadora
- 11 - Pulse ENTER
- 12 - El programa se transmite y se ejecuta

Para mayor información de la utilización del sistema UNIMAT.EXE consulte el MANUAL DEL USUARIO DEL SISTEMA UNIMAT que se incluye en el apéndice #2.

### 3.3.- Etapa de potencia

La etapa de potencia del sistema UNIMAT consiste en el conjunto de circuitos electrónicos y eléctricos utilizados para proporcionar movimiento a la máquina-herramienta. La etapa de potencia recibe las señales de control provenientes del controlador para realizar las diferentes funciones de la máquina-herramienta.

Para el movimiento de los ejes X, Y y Z se utilizaron motores de pasos cuyas características son las siguientes:

- Número de fases: 4 Fases
- Polarización: 5 Vdc @ 1.0 Amp / Fase
- Número de pasos : 200 Pasos / Rev
- Resolución:  $1.8^\circ$  por paso

La resolución del motor de pasos se obtiene a partir de lo siguiente:

Avance del tornillo: 1mm / Rev  
Resolución mínima: 0.05 mm  
 $1 \text{ mm} / 0.05 \text{ mm} = 20 \text{ divisiones}$   
 $1 \text{ Rev} = 360^\circ$

de donde

$1 \text{ mm} = 360^\circ$   
 $0.05 \text{ mm} = X$

lo que resulta en

$$X = 18^\circ$$

Los motores son operados con un desplazamiento de paso completo a una frecuencia aproximada de 250Hz. Cada motor se encuentra alimentado por una fuente de potencia independiente esto para aumentar el rendimiento de la fuente además de no someter a una sola fuente toda la carga del sistema de potencia ( Aprox. 35 Watts) Las fases del motor son controladas por un amplificador de corriente construido con amplificadores darlington trabajando entre corte y saturación, los cuales a su vez, se encuentran optoacoplados para evitar que cualquier sobrecarga dañe las salidas de la etapa del controlador. El amplificador no requiere disipación, aunque por seguridad se utiliza un ventilador.



El circuito controlador del encendido del husillo de la herramienta de corte consiste en un relevador. Se utiliza un relevador ya que la velocidad de conmutación es variable pero no alcanza velocidades muy grandes además de que no requiere disipación.

El diagrama eléctrico de la etapa de potencia se muestra en el apéndice #1 con el nombre "POTENCIA.SCH"

### 3.4.- Etapa y acoplamiento mecánico

El diseño del acoplamiento mecánico resultó el más complicado debido a que no se partió del diseño completo de una máquina-herramienta sino que fue necesario adaptar la máquina a los requerimientos necesarios. El diseño se realizó atendiendo básicamente los aspectos siguientes:

- Modificar lo menos posible la máquina-herramienta
- Obtener los requerimientos de precisión
- Garantizar los movimientos de la máquina a una velocidad aceptable
- Proporcionar soporte mecánico a las diferentes etapas del sistema

Para lograr lo anterior se realizó un estudio exhaustivo de los posibles mecanismos y elementos mecánicos disponibles. Sin embargo nos enfrentamos al problema de la falta de partes en México. En E.U.A. e Inglaterra por ejemplo existen tiendas en donde se pueden conseguir todo tipo de piezas mecánicas por medio de catálogo (muy al estilo de Radio Shack) para la realización de prototipos y proyectos, pero en México desafortunadamente no, lo cual complicó esta etapa.

Los motores de los ejes X-Y están acoplados por medio de un sistema elevador de velocidad de engranes lo cual evita oscilaciones entre los ejes del motor y el tornillo sin fin. La relación de transmisión es de 0.53499. La figura No. 22 muestra el arreglo final de engranes para el movimiento de los ejes X-Y.

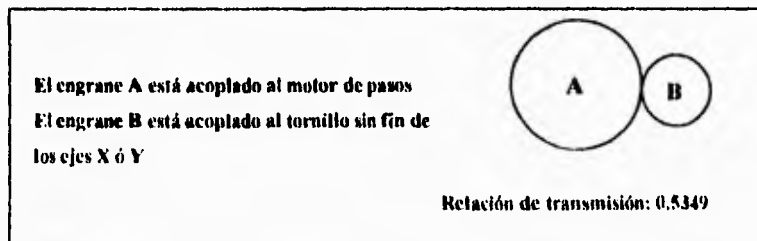


Figura No. 22.- Acoplamiento de los ejes X - Y a los motores

El motor del eje Y está montado sobre un carrito el cual tiene un riel para desplazarse a lo largo del eje X debido a que la bancada del eje Y se mueve en las direcciones -X y +X, de donde el motor del eje Y no puede estar fijo.

Es necesario aclarar que el acoplamiento de engranes proporciona mayor velocidad a los desplazamientos de los ejes pero el par aplicado disminuye, aunque para la operación como taladro CNC el par resultante a lo largo de los ejes X-Y es suficiente ya que la herramienta realiza su movimiento de corte en forma vertical y giratoria. Ambos ejes (X -Y) tienen un sensor de posición Home el cual permite al controlador fijar el punto de reposo o Cero Máquina.

El eje Z proporciona movimiento a la herramienta de corte (taladro) en sentido vertical. El mecanismo de este eje debe ser capaz de proporcionar la fuerza suficiente para penetrar la superficie a taladrar, de desplazar la herramienta con su fuerza motriz y de mantener en una posición la herramienta.

El mecanismo del eje Z consiste en un sistema de transmisión reductor acoplado al motor correspondiente por medio de engranes. Los engranes de acoplamiento entre el motor de pasos y el sistema de transmisión se encuentran colocados del mismo modo que para los ejes X-Y. La transmisión tiene una relación de reducción 1.24. El tiempo de bajada y subida del taladro depende de la carrera máxima fijada con los sensores Zhome y Zfin, siendo el tiempo máximo de aproximadamente 15 segundos.

La máquina está montada sobre una base de madera la cual proporciona soporte mecánico tanto a la máquina como a los motores y el riel.

Es necesario mencionar que aunque los acoplamientos mecánicos presentan juego (backlash) y los sistemas de transmisión no son los ideales, el sistema funciona perfectamente.

La Fig. No. 23 muestra la disposición final de la máquina-herramienta del sistema UNIMAT.

La posición del Cero Máquina corresponde a la posición en la que se encuentra la herramienta de corte cuando los ejes X, Y y Z son llevados a su posición HOME; debido a las características de la máquina-herramienta del sistema UNIMAT, el Cero Máquina está definido como se muestra en la figura No. 24.

La figura No. 24 muestra el sistema de ejes definido mismo que deberá ser utilizado como referencia para la programación de la máquina-herramienta. Nótese que no se muestra el mecanismo del eje Z para mayor claridad. La posición Zhome se localiza en el punto en donde la herramienta de corte se encuentra contraída en dirección vertical y hacia arriba. Figura No. 25

La lista de componentes mecánicos se muestra en el apéndice #2 así como ilustraciones que muestran el sistema UNIMAT

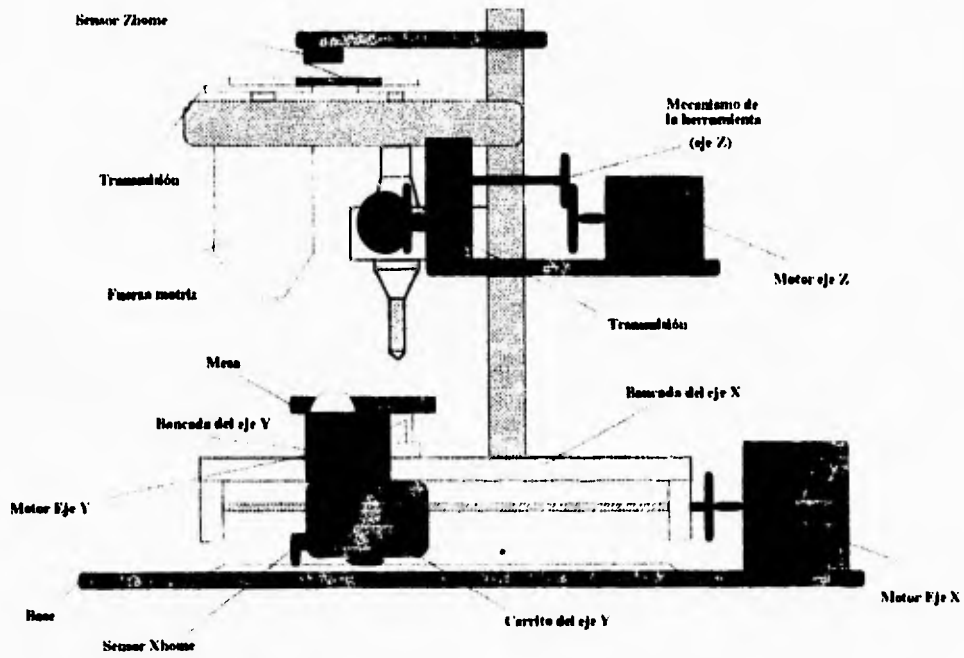


Fig.23.- Diagrama de la máquina UNIMAT (vista lateral)

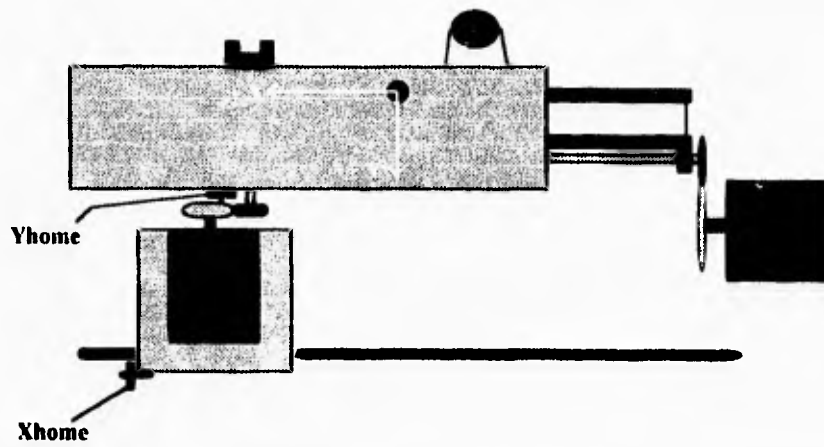


Figura No. 24.- Vista superior de la máquina-herramienta en donde se muestra la posición Cero Máquina.

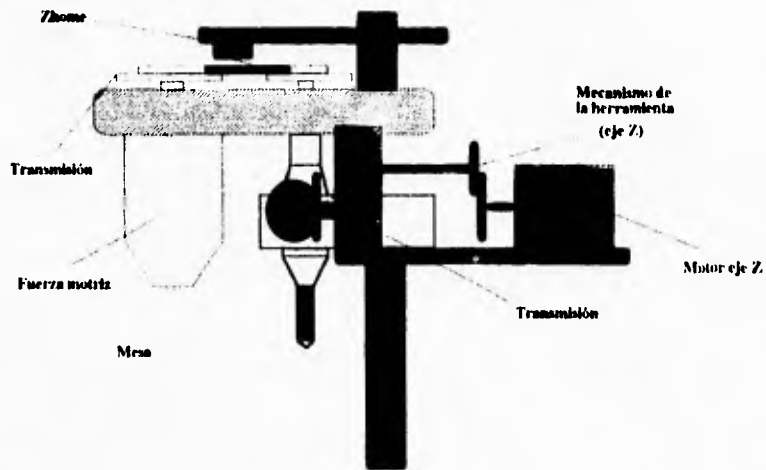


Figura No. 25.- Vista lateral de la máquina-herramienta que muestra la posición Zhome

## Capítulo 4

### Conclusiones

Después de haber realizado numerosas pruebas, ajustes y modificaciones, el objetivo fue alcanzado. Sin embargo es necesario recalcar que el sistema UNIMAT no es el fin de una idea sino el principio de algo que puede y debe ser perfeccionado.

El diseño de este sistema deja como experiencia la necesidad de fomentar el desarrollo tecnológico a nivel nacional ya que la dependencia tecnológica desde el punto de vista de insumos, es excesivamente grande lo cual limita el desarrollo industrial de nuestro país. Es curioso pero la información tecnológica se encuentra "limitada" por los países industrializados con el fin de reservarla para sí mismos. Cuando se empieza a buscar información importante es común encontrarse con detalles como "No está documentado", con cajas negras que solamente nos dicen qué debe hacerse pero no cómo se hace o que la información relevante no está actualizada, por ello es necesario emplear al máximo el ingenio y los conocimientos propios para la solución de los problemas.

Sin embargo queda claro que México cuenta con personal capacitado para afrontar los retos de productividad y competitividad que se requieren a nivel mundial, el cual debe ser aprovechado para aumentar el nivel de bienestar de nuestro país.

El sistema UNIMAT fue diseñado cuidadosamente, ninguna decisión fue tomada al vapor sino que se hicieron los estudios necesarios para lograr un diseño balanceado y funcional. El hecho de que el sistema abarque tres disciplinas interrelacionadas es importante, ya que sienta las bases del diseño de maquinaria automatizada y de manera particular, de máquinas con control numérico por computadora.

Este sistema puede ser utilizado para planificar y proyectar la fabricación en serie de este tipo de máquinas a nivel nacional con el objeto de reducir los costos de la automatización, así como de la dependencia tecnológica.

A nivel del sistema UNIMAT, es posible realizar modificaciones posteriores como son el uso de nuevos códigos de programación, nuevas funciones, entre otras, con la ventaja de que éstas pueden ser realizadas con un mínimo de modificaciones al hardware, lo cual facilita su modernización.

Otro aspecto importante es el hecho de que, aunque el sistema UNIMAT es prácticamente una máquina a nivel experimental, realiza las operaciones básicas que cualquier máquina de taladrado con CNC puede realizar, con la precisión y confiabilidad requeridas.

Asimismo, es necesario entender que aunque la automatización presenta grandes ventajas, el ingeniero no debe olvidar el carácter social de su profesión, lo cual implica que en el caso de que no sea estrictamente necesario, el diseño de un sistema automatizado no debe estar concebido con el objetivo de desplazar al ser humano, sino de ayudarlo a eficientar su desempeño y de aumentar su seguridad en el trabajo, dejando a las máquinas lo que el hombre no puede realizar y no al contrario.

Para mí es un orgullo haber realizado el sistema UNIMAT ya que las dificultades encontradas a lo largo de su desarrollo, los dolores de cabeza, los sustos, y otros pormenores, me han dejado claro que las cosas no salen de la nada, que es necesario esforzarse día a día, que es responsabilidad de un profesional respaldar su profesión con conocimientos sólidos y bien fundamentados.

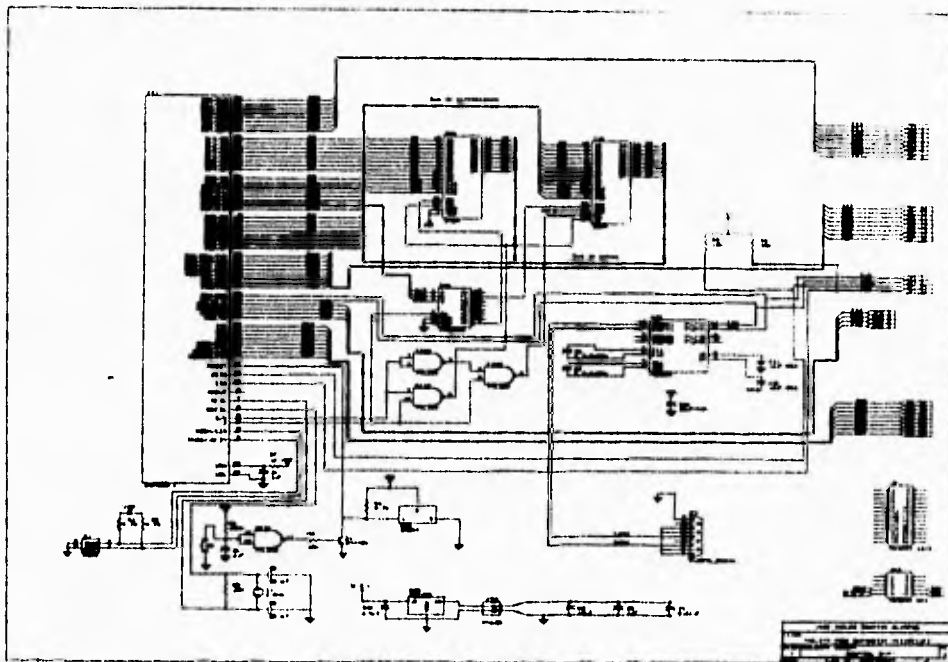
UNIMAT es un sistema que demuestra que, con las limitaciones que presenta el mercado y nivel tecnológico nacionales, es posible realizar ingeniería de excelente nivel y es una prueba de que en México la ingeniería es rentable y que no todo viene del exterior como muchas personas piensan.

La tecnología debe ser una posibilidad de conocimiento para toda la población, por ello es necesario abrir el conocimiento y ponerlo al alcance de todos, esa es la clave del desarrollo de nuestro grande y hermoso país.

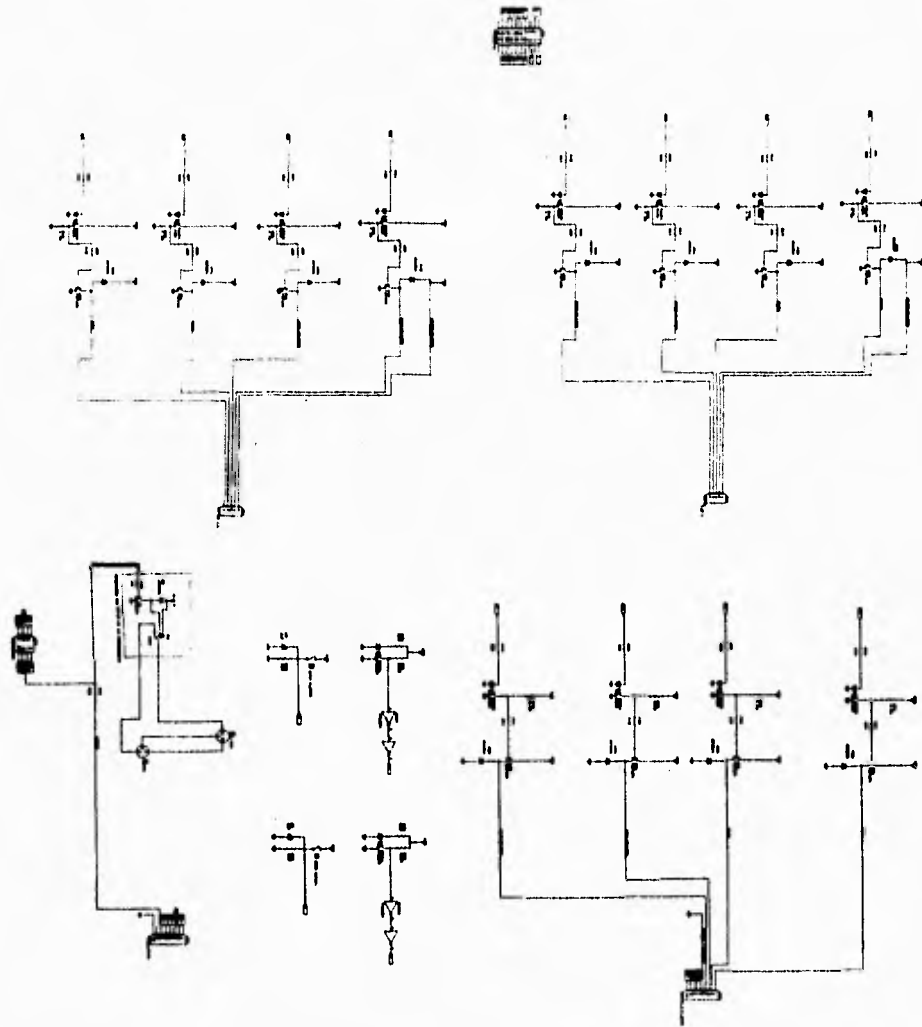
José Carlos Benítez Álvarez

## Apéndice #1

### Diagramas y programas

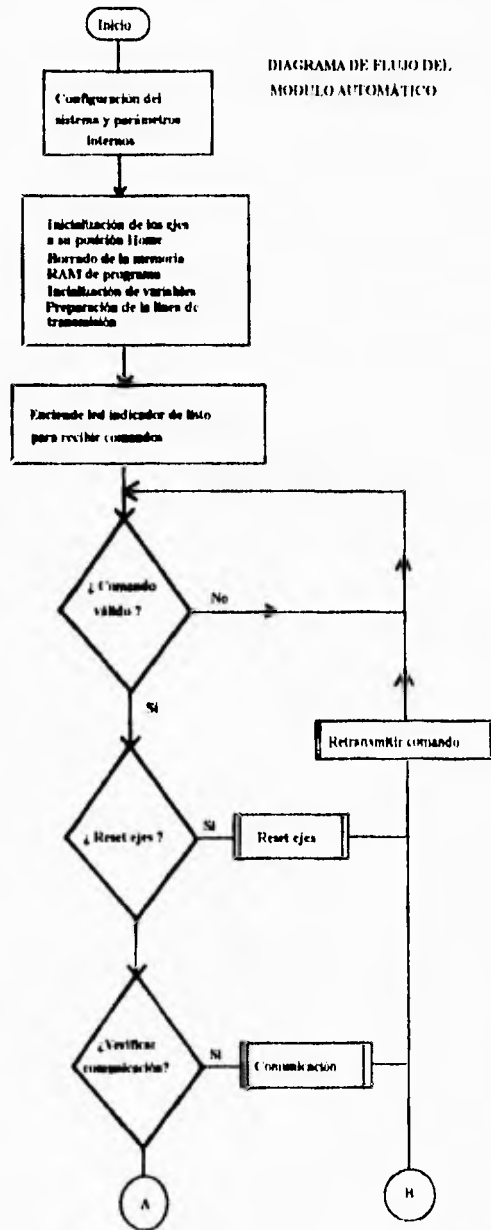


CONTROL. SCH



POTENCIA.SCH





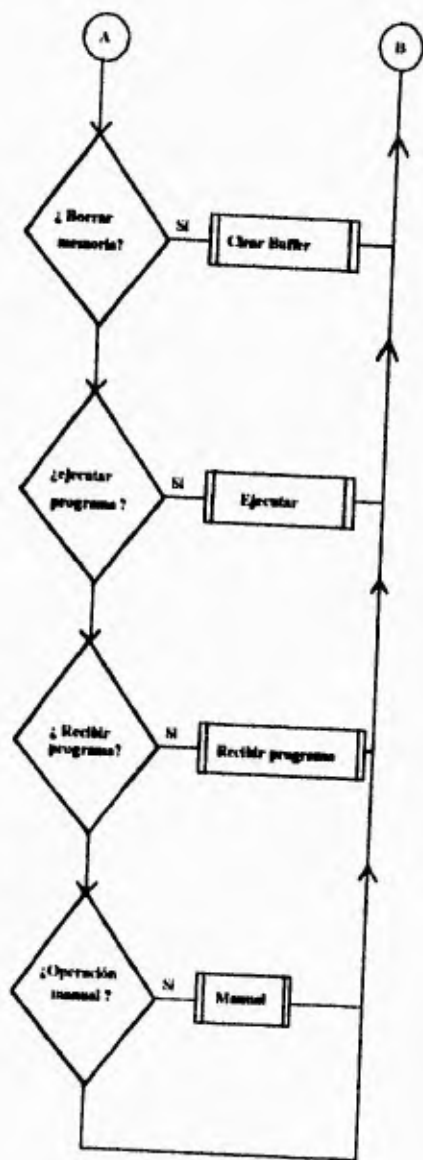
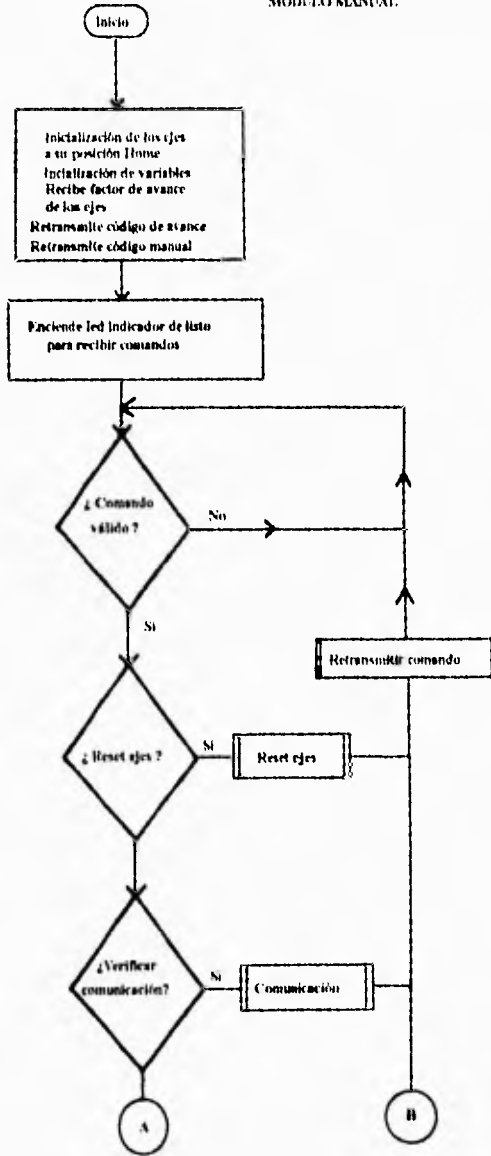
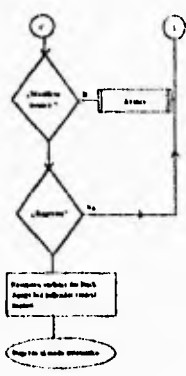
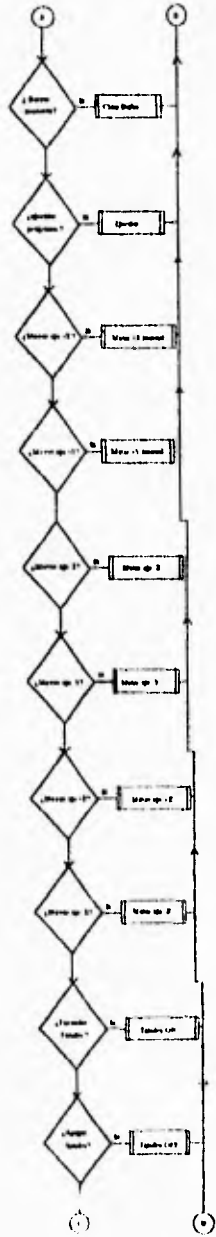


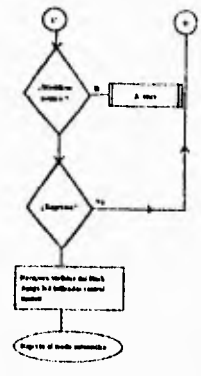
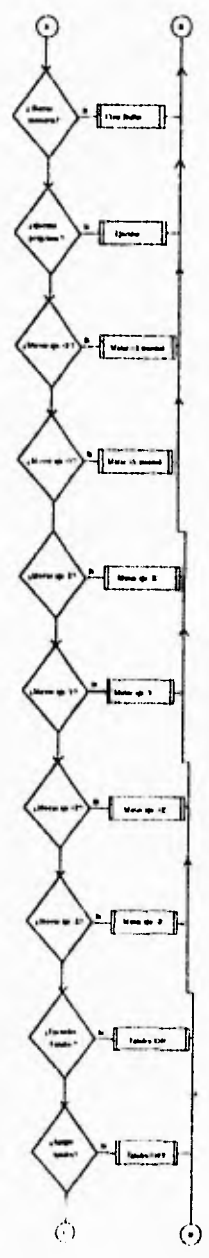
DIAGRAMA DE FLUJO DEL  
MÓDULO MANUAL.



Apéndice #1: Diagramas y programas 66



Apéndice #1: Diagramas y programas 66



**Programa en lenguaje ensamblador para el módulo de control utilizando un microcontrolador 68HC11F1 : CONTROL.ASM**

\*\*\*\*\*

**\* DEFINICION DE CONSTANTES**

\*\*\*\*\*

```

PORTA EQU $1000
DDRA EQU $1001
PORTG EQU $1002
DDRG EQU $1003
PORTB EQU $1004
PORTF EQU $1005
PORTC EQU $1006
DDRC EQU $1007
PORTD EQU $1008
DDRD EQU $1009
PORTE EQU $100A
CSTL EQU $105D
CSGSIZ EQU $105F
CSGADR EQU $105E
TMSK2 EQU $1024
TFLG2 EQU $1025
PACTL EQU $1026
SCDR EQU $102F
TRUE EQU $01
FALSE EQU $00
INICIO EQU $2000
FIN EQU $4000
VACIO EQU $00
CUENTA EQU $0F ,DEBE SER $0F
    
```

\*\*\*\*\*

**\* DEFINICION DE VARIABLES**

\*\*\*\*\*

```

CLOCK EQU $00
PULSO EQU $01
GPOINTER EQU $03
VEL11 EQU $05
VELL EQU $06
WORDX EQU $07
WORDY EQU $09
NSEGS EQU $0F
XABS EQU $10
XACTUAL EQU $12
XBYTE EQU $14
    
```

```

YABS      EQU $15
YACTUAL   EQU $17
YBYTE     EQU $19
LLEN0     EQU $1A
RXBYTE    EQU $1B
TXBYTE    EQU $1C
NPOINTER  EQU $1D
PPOINTER  EQU $1F
LAST_BYTE EQU $21
TRAMA     EQU $22
TRAMA2    EQU $23
GCODE     EQU $24
SERIAL    EQU $25
WORDZ     EQU $26
ZBYTE     EQU $28
ZBAS      EQU $2B
ZACTUAL   EQU $2D
AVANCE    EQU $30
EJECUTANDO EQU $31
    ORG $E000
    LDS #$03FF
    LDAA #$0
    STAA CSTL
    STAA CSGADR
    LDAA #$07      ; DESACTIVA CSGEN
    STAA CSGSIZ
    LDAA #$FF
    STAA DDRG
    STAA DDRA
    CLR PORTA
    LDAA #$FE
    STAA DDRD
    LDX #$1000
    LDAA #$00
    STAA $2B,X
    STAA $2C,X
    JSR DELAY
    LDAA #$2C      ; HABILITA SCI-> RIE,Tx,Rx
    STAA $102D
    BSET $2B,X,$32 ; PONE REGISTRO BAUD A 2400
    JSR DELAYS
    CLR SCDR      ; LIMPIA SCI DATA REGISTER
    LDAA #$40      ; HABILITA LA RTII
    STAA TMSK2

```

```

LDAA #$03
STAA PACTL      ; CONFIGURA RTI A 32.77 ms
LDAA #FALSE
STAA PULSO
LDAA #CUENTA
STAA CLOCK
LDAA #$03      ; xbyte = 3
STAA XBYTE
STAA YBYTE      ; ybyte = 3
STAA ZBYTE      ; zbyte = 3
JSR XYPORT
JSR RESET_Z
JSR RESET_Y
JSR RESET_X
JSR RESET_BUFFER
CLR LIENO
CLR GCODE
CLR TRAMA
CLR TRAMA2
CLR SERIAL
CLR PORTA
CLR PORTG
CLR PORTE
CLR RXBYTE
LDAA #'B'
STAA TXBYTE
JSR TRANS
LDAA #$80      ;habilita interrupciones
TAP
MAINI:  LDAA RXBYTE
        CMPA #'B'
        BNE CASEb
        STAA TXBYTE
        CLR RXBYTE
        CLR SERIAL
        JSR TRANS
        BRA ENDCASE
CASEb:  CMPA #'b'
        BNE CASEc
        JSR RECIBIR_PROG
        BRA ENDCASE
CASEc:  CMPA #'c'
        BNE CASEr
        JSR RESET_BUFFER
    
```



```

STAA TXBYTE
CLR RXBYTE
CLR SERIAL
JSR TRANS
BRA ENDCASE
CASEr:   CMPA #'r'
        BNE CASEe
        JSR RESET_Z
        JSR RESET_Y
        JSR RESET_X
        STAA TXBYTE
        CLR RXBYTE
        CLR SERIAL
        JSR TRANS
        BRA ENDCASE
CASEe:   CMPA #'e'
        BNE CASEm
        JSR EJECUTAR
        STAA TXBYTE
        CLR RXBYTE
        CLR SERIAL
        CLR PORTG
        CLR PORTA
        JSR TRANS
        BRA ENDCASE
CASEm:   CMPA #'m'
        BNE ENDCASE
        JSR MANUAL
        STAA TXBYTE
        CLR RXBYTE
        CLR SERIAL
        CLR PORTG
        CLR PORTA
        JSR TRANS
ENDCASE: JMP MAIN1
MOTORX:  PSHA
        PSIB
        PSHX
        PSHY
        LDAA #504 ;enciende el led de motor x activo
        ORAA PORTD ;e indica que se está moviendo
        STAA PORTD
        LDAA XBYTE
        LDX WORDX ;wordx = getw(del buffer)

```

```

STAA TXBYTE
CLR RXBYTE
CLR SERIAL
JSR TRANS
BRA ENDCASE
CASEr:   CMPA #'r'
        BNE CASEe
        JSR RESET_Z
        JSR RESET_Y
        JSR RESET_X
        STAA TXBYTE
        CLR RXBYTE
        CLR SERIAL
        JSR TRANS
        BRA ENDCASE
CASEe:   CMPA #'e'
        BNE CASEm
        JSR EJECUTAR
        STAA TXBYTE
        CLR RXBYTE
        CLR SERIAL
        CLR PORTG
        CLR PORTA
        JSR TRANS
        BRA ENDCASE
CASEm:   CMPA #'m'
        BNE ENDCASE
        JSR MANUAL
        STAA TXBYTE
        CLR RXBYTE
        CLR SERIAL
        CLR PORTG
        CLR PORTA
        JSR TRANS
ENDCASE: JMP MAIN1
MOTORX:  PSHA
        PSIB
        PSHX
        PSHY
        LDAA #$04      ;enciende el led de motor x activo
        ORAA PORTD    ;e indica que se está moviendo
        STAA PORTD
        LDAA XBYTE
        LDX WORDX      ;wordx = getw(del buffer)

```

```

STX XABS      ;xabs = getw(wordx)
LDX XACTUAL
CPX XABS
BEQ CONTINUEX      ;if (xactual == xabs) continue
BLO DO1X          ;if (xactual < xabs) do{
BRA ELSEX        ;else{
CONTINUEX: LDAA PORTD      ;apaga el led de motor x activo
ANDA #$EB      ;y el led de resetting ejes
STAA PORTD
PULY
PULX
PULB
PULA
RTS
DO1X:  LSR      ;xbyte >>=1
      CMPA #101 ;if (xbyte == 1) xbyte = 9
      BEQ NINEX
      CMPA #104 ;if (xbyte == 4) xbyte = 12
      BEQ DOCEX
      BRA OUTX1
NINEX:  LDAA #109      ;xbyte = 9
      BRA OUTX1
DOCEX:  LDAA #112      ;xbyte = 12
OUTX1:  STAA XBYTE     ;actualiza xbyte
      JSR XYPOR
      JSR DELAY
      INX      ;xactual +=1
      CPX XABS
      BLO DO1X      ;while(xactual < xabs)
      LDX XABS
      STX XACTUAL   ;xactual = xabs
      STAA XBYTE
      JMP CONTINUEX ;}
ELSEX:  NOP
DO2X:  LSL      ;xbyte <<=1
      CMPA #124 ;if (xbyte == 24) xbyte = 9
      BEQ NUEVEX
      CMPA #118 ;if (xbyte == 18) xbyte = 3
      BEQ TRESX
      BRA OUTX2
NUEVEX: LDAA #109      ;xbyte = 9
      BRA OUTX2
TRESX:  LDAA #103      ;xbyte = 3
OUTX2:  STAA XBYTE

```

```

JSR XYPORT
JSR DELAY
DEX                ;xactual--1
CPX XABS
BHI DO2X          ;while(xactual > xabs)
LDX XABS
STX XACTUAL      ;xactual = xabs
STAA XBYTE      ;actualiza xbyte
JMP CONTINUEX   ;}

MOTORY:  PSHA
        PSHB
        PSHX
        PSHY
        LDAA #$08    ;enciende el led de motor y activo
        ORAA PORTD   ;y el led de rescting ejes
        STAA PORTD
        LDAB YBYTE
        LDX WORDY    ;wordy = getw(del buffer)
        STX YABS     ;yabs = getw(wordy)
        LDX YACTUAL
        CPX YABS
        BEQ CONTINUEY ;if (yactual == yabs) continue
        BLO DO1Y     ;if (yactual < yabs) do{
        BRA ELSEY    ;else{

CONTINUEY: LDAA PORTD ;apaga el led de motor y activo
          ANDA #$E7   ;y el led de rescting ejes
          STAA PORTD
          PULY
          PULX
          PULB
          PULA
          RTS

DO1Y:    LSRB        ;ybyte >>=1
          CMPB #101  ;if (ybyte == 1) ybyte = 9
          BEQ NINEY
          CMPB #104  ;if (ybyte == 4) ybyte = 12
          BEQ DOCEY
          BRA OUTY1

NINEY:   LDAB #109   ;ybyte = 9
          BRA OUTY1

DOCEY:   LDAB #112   ;ybyte = 12
OUTY1:   STAB YBYTE  ;actualiza ybyte
        JSR XYPORT
        JSR DELAY

```

```

    INX          ;yactual+=1
    CPX YABS
    BLO DO1Y    ;while(yactual < yabs)
    LDX YABS
    STX YACTUAL ;yactual = yabs
    STAB YBYTE
    JMP CONTINUEY ;}
ELSEY:  NOP
DO2Y:  LSLB    ;ybyte <<=1
        CMPB #!24 ;if (ybyte == 24) ybyte = 9
        BEQ NUEVEY
        CMPB #!18 ;if (ybyte == 18) ybyte = 3
        BEQ TRESY
        BRA OUTY2
NUEVEY: LDAB #!09 ;ybyte = 9
        BRA OUTY2
TRESY:  LDAB #!03 ;ybyte = 3
OUTY2:  STAB YBYTE ;actualiza ybyte
        JSR XYPORT
        JSR DELAY
        DEX      ;yactual-=1
        CPX YABS
        BHI DO2Y ;while(yactual > yabs)
        LDX YABS
        STX YACTUAL ;yactual = yabs
        STAB YBYTE
        JMP CONTINUEY ;}
MOTORZ: PSHA
        PSHB
        PSHX
        PSHY
        LDAA #$20 ;enciende el led de taladro activo
        ORAA PORTD ;indica que se está taladrando un punto
        STAA PORTD
        LDAB ZBYTE
BAJARZ: LSRB    ;zbyte >>=1
        CMPB #!01 ;if (zbyte == 1) zbyte = 9
        BEQ NINEZ
        CMPB #!04 ;if (zbyte == 4) zbyte = 12
        BEQ DOCEZ
        BRA OUTZ1
NINEZ:  LDAB #!09 ;zbyte = 9
        BRA OUTZ1
DOCEZ:  LDAB #!12 ;zbyte = 12

```

```

    INX          ;yactual+=1
    CPX YABS
    BLO DO1Y    ;while(yactual < yabs)
    LDX YABS
    STX YACTUAL ;yactual = yabs
    STAB YBYTE
    JMP CONTINUEY ;}
ELSEY:  NOP
DO2Y:   LSLB    ;ybyte <<=1
        CMPB #124 ;if (ybyte == 24) ybyte = 9
        BEQ NUEVEY
        CMPB #118 ;if (ybyte == 18) ybyte = 3
        BEQ TRESY
        BRA OUTY2
NUEVEY: LDAB #109 ;ybyte = 9
        BRA OUTY2
TRESY:  LDAB #103 ;ybyte = 3
OUTY2:  STAB YBYTE ;actualiza ybyte
        JSR XYPORT
        JSR DELAY
        DEX      ;yactual-=1
        CPX YABS
        BHI DO2Y ;while(yactual > yabs)
        LDX YABS
        STX YACTUAL ;yactual = yabs
        STAB YBYTE
        JMP CONTINUEY ;}
MOTORZ: PSHA
        PSHB
        PSHX
        PSHY
        LDAA #$20 ;enciende el led de taladro activo
        ORAA PORTD ;indica que se está taladrando un punto
        STAA PORTD
        LDAB ZBYTE
BAJARZ: LSRB    ;zbyte >>=1
        CMPB #101 ;if (zbyte == 1) zbyte = 9
        BEQ NINEZ
        CMPB #104 ;if (zbyte == 4) zbyte = 12
        BEQ DOCEZ
        BRA OUTZ1
NINEZ:  LDAB #109 ;zbyte = 9
        BRA OUTZ1
DOCEZ:  LDAB #112 ;zbyte = 12

```

```

OUTZ1:  STAB ZBYTE      ;actualiza zbyte
        JSR ZPORT
        JSR DELAY
        LDAA PORTE
        ANDA #$20
        TSTA
        BEQ BAJARZ
SUBIRZ:  LSLB           ,zbyte <<=1
        CMPB #!24      ;if (zbyte == 24) zbyte = 9
        BEQ NUEVEZ
        CMPB #!18      ;if (zbyte == 18) zbyte = 3
        BEQ TRESZ
        BRA OUTZ2
NUEVEZ:  LDAB #!09     ;zbyte = 9
        BRA OUTZ2
TRESZ:   LDAB #!03     ;zbyte = 3
OUTZ2:   STAB ZBYTE    ;actualiza zbyte
        JSR ZPORT
        JSR DELAY
        LDAA PORTE
        ANDA #$04
        TSTA
        BEQ SUBIRZ
        LDAA PORTD     ;apaga el led de motor z activo
        ANDA #$CF      ;y el led de reseting ejes
        STAA PORTD
        PULY
        PULX
        PULB
        PULA
        RTS
XYPORT:  PSHA
        CLRA
        LDAA YBYTE
        LSLA
        LSLA
        LSLA
        LSLA
        ORAA XBYTE
        STAA PORTG     , DEBE SER PORTG
        PULA
        RTS
ZPORT   PSHA
        LDAA PORTA
    
```

```

        ANDA #$F0
        ORAA ZBYTE
        STAA PORTA      ; DEBE SER PORTA
        PULA
        RTS
DELAY   PSHA
        PSHX
        LDAA #$02      ; DEBE SER $02
DLOOP1  LDX #$02A0    ; DEBE SER $02A0
DLOOP2  DEX
        BNE DLOOP2
        DECA
        BNE DLOOP1
        PULX
        PULA
        RTS
DELAYS  PSHA
        PSHX
        LDX #3504
WLOOP   DEX
        BNE WLOOP
        PULX
        PULA
        RTS
*****
* SUBROUTINA QUE LIMPIA EL BUFFER CIRCULAR *
*****
RESET_BUFFER: PSHA
          PSHX
          LDX #INICIO
          STX GPOINTER
          STX PPOINTER
          LDAA #VACIO
CLEAR:    STAA 0,X
          INX
          CPX #FIN
          BNE CLEAR
          LDX #$00
          STX NPOINTER
          CLR LLEN0
          CLR GCODE
          CLR TRAMA
          CLR TRAMA2
          LDX #$0000
    
```



STX WORDX  
 STX WORDY  
 STX XABS  
 STX YABS  
 STX XACTUAL  
 STX YACTUAL  
 PULX  
 PULA  
 RTS

\*\*\*\*\*  
 \* POSICIONA EL EJE X EN Xo \*  
 \*\*\*\*\*

```

RESET_X:   PSHA
           PSHB
           PSHX
           LDAA #$14      ;enciende el led de motor x activo
           ORAA PORTD     ;y el led de reseting ejes
           STAA PORTD
           LDAA XBYTE
           LDAB PORTE     ;verifica si está activado el
           ANDB #$01     ;sensor de Xhome. Si no está
           TSTB          ;activado entonces sólo lo busca
           BEQ BUSCAX0   ;de lo contrario hace lo siguiente
           LDX #$042E    ;Mueve los ejes +5 mm
           STX WORDX     ;y luego los resetea para fijar
           LDX #$0000    ;siempre el mismo Home
           STX XABS
           STX XACTUAL   ;xactual = 0
           JSR MOTORX
           LDAA #$14     ;enciende el led de motor x activo
           ORAA PORTD   ;y el led de reseting ejes
           STAA PORTD
           LDAA XBYTE
BUSCAX0:   LDAB PORTE
           ANDB #$01
           TSTB
           BNE FINRESX
           LSLA          ;xbyte <<= 1
           CMPA #!24    ;if (xbyte == 24) xbyte = 9
           BEQ NEUFX
           CMPA #!18    ;if (xbyte == 18) xbyte = 3
           BEQ TROISX
    
```

```

        BRA SORTEX2
NEUFX:  LDAA #109      ;xbyte = 9
        BRA SORTEX2
TROISX: LDAA #103      ;xbyte = 3
SORTEX2: STAA XBYTE    ;actualiza xbyte
        JSR XYPORT
        JSR DELAY
        JMP BUSCAX0
FINRESX: LDAA PORTD    ;apaga el led de motor x activo
        ANDA #$EB      ;y el led de reseting ejes
        STAA PORTD
        LDX #$0000
        STX WORDX
        STX XABS        ;xabs = 0
        STX XACTUAL     ;xactual = 0
        PULX
        PULB
        PULA
        RTS
    
```

```

RESET_Y: PSHA
        PSHB
        PSHX
        LDAA #$18      ;enciende el led de motor y activo
        ORAA PORTD     ;y el led de reseting ejes
        STAA PORTD
        LDAA YBYTE
        LDAB PORTE     ;verifica si está activado el
        ANDB #$02      ;sensor de Yhome. Si no está
        TSTB          ;activado entonces sólo lo busca
        BEQ BUSCAY0    ;de lo contrario hace lo siguiente
        LDX #$042E     ;Mueve el eje +5 mm para tener
        STX WORDY      ;siempre el mismo Home
        LDX #$0000
        STX YABS
        STX YACTUAL     ;yactual = 0
        JSR MOTORY
        LDAA #$18      ;enciende el led de motor y activo
        ORAA PORTD     ;y el led de reseting ejes
        STAA PORTD
        LDAA YBYTE
BUSCAY0: LDAB PORTE
        ANDB #$02
    
```

```

TSTB
BNE FINRESY
LSLA          ;ybyte <<=1
CMPA #!24    ;if (ybyte == 24) ybyte = 9
BEQ NEUFY
CMPA #!18    ;if (ybyte == 18) ybyte = 3
BEQ TROISY
BRA SORTEY2
NEUFY:      LDAA #!09          ;ybyte = 9
           BRA SORTEY2
TROISY:    LDAA #!03          ;ybyte = 3
SORTEY2:   STAA YBYTE        ;actualiza ybyte
           JSR XYPORT
           JSR DELAY
           JMP BUSCAZO
FINRESY:   LDAA PORTD        ;apaga el led de motor y activo
           ANDA #$E7          ;y el led de reseting ejes
           STAA PORTD
           LDX #$0000
           STX WORDY
           STX YABS          ;yabs = 0
           STX YACTUAL       ;yactual = 0
           PULX
           PULB
           PULA
           RTS

RESET_Z:   PSHA
           PSHB
           LDAA #$30         ;enciende el led de taladro activo activo
           ORAA PORTD        ;y el led de reseting ejes
           STAA PORTD
           LDAA #$03
BUSCAZO:   LDAB PORTE
           ANDB #$04
           TSTB
           BNE FINRES_Z
           LSLA              ;zbyte <<=1
           CMPA #!24         ;if (zbyte == 24) zbyte = 9
           BEQ NEUFZ
           CMPA #!18         ;if (zbyte == 18) zbyte = 3
           BEQ TROISZ
           BRA SORTEZ1
NEUFZ:    LDAA #!09          ;zbyte = 9

```

```

        BRA SORTEZ1
TROISZ:  LDAA #103      ;zbyte = 3
SORTEZ1:  STAA ZBYTE    ;actualiza zbyte
        JSR ZPORT
        JSR DELAY
        JMP BUSCAZO
FINRES_Z: LDAA PORTD    ;apaga el led de motor z activo
        ANDA #$CF      ;y el led de reseting ejes
        STAA PORTD
        PULB
        PULA
        RTS
    
```

```

*****
* LLENA EL BUFFER CON EL PROGRAMA Y MANDA UN CARACTER DE WAIT "w" *
* SI EL PROGRAMA ES MAYOR QUE 8K. CUANDO OCURRE ESTO SE EJECUTA *
* EL PROGRAMA Y LUEGO SE CONTINUA LA RECEPCION DEL PROGRAMA Y SE *
* TERMINA DE EJECUTAR.(PARA RESUMIR EL PROGRAMA REGRESA EL ULTIMO *
* BYTE RECIBIDO EN LA VARIABLE TXBYTE. *
*****
    
```

```

RECIBIR_PROG: PSHA
        PSHB
        PSHX
        PSHY
        LDAB #$40
        ORAB PORTA    ; enciende LED de recibiendo byte A6
        STAB PORTA
        STAA TXBYTE   ; 'b' AL HOST
        LDAA #$0C     ; INHABILITA SCI
        STAA $102D
        CLR SERIAL
        LDAA LLENO
        CMPA #TRUE
        BEQ BUFFERLLENO ; SI BUFFER LLEN, BUFFER_LLENO
        LDAA #$2C     ; SI NOT, HABILITA SCI
        STAA $102D
        LDAB #$00     ; ESTABLECE CONTADOR DE TRAMAS EN EL REG_B
LLENADO: JSR TRANS    ; REGRESA EL RXBYTE TO HOST ('b' INCLUSIVE)
        PSHA        ; SALVA TEMPORALMENTE ACC A EN EL STACK
RDBYTE:  LDAA SERIAL  ; LEE CARACTER DEL PUERTO SERIAL
        CMPA #TRUE   ; ESPERA LA INTERRUPCION DEL PUERTO SERIE
        BNE RDBYTE
        CLR SERIAL
        PULA        ; RECUPERA ACC A DEL STACK
    
```

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

```

INCB          ; INCREMENTA CONTADOR DE TRAMA
CMPB #S04    ; VERIFICA EL TERCER CARACTER
BNE ALMACENAR ; ALMACENAR SI NO ES EL FIN DE TRAMA
LDAB RXBYTE  ; LEE EL CARACTER DEL BUFFER SERIAL
CMPB #f      ; LO COMPARA CON FIN DE ARCHIVO
BNE RESCONT  ; SI NOT EOF, THEN RESET CONT_TRAMA
JSR FILL_BUFFER
JSR TRANS    ; REGRESA FIN DE ARCHIVO AL HOST
LDX PPOINTER
DEX
STX PPOINTER
RETURN:      LDAB #SBF
ANDB PORTA  ; APAGA LED DE RECIBIENDO BYTE
STAB PORTA
PULY
PULX
PULB
PULA
RTS
RESCONT:    LDAB #S01 ; RESETS CONTADOR DE TRAMA
ALMACENAR: STAB TRAMA ; SALVA CONTADOR DE TRAMA
            LDAA LLENO
            CMPA #TRUE
            BEQ BUFFERLLENO
            JSR FILL_BUFFER
            BRA LLENADO
BUFFERLLENO: LDAA #S0C ; INHABILITA SCI
            STAA $102D
            LDAA RXBYTE ; SALVA EL ULTIMO BYTE RECIBIDO
            STAA LAST_BYTE
            LDAA #'w' ; MANDA UN WAIT AL HOST 'w'
            STAA TXBYTE
            JSR TRANS ; TRANSMITE EL "w" AL HOST
            JSR EJECUTAR ; EJECUTA LA PRIMERA PARTE DEL PROGRAMA
            LDAA LAST_BYTE ; RECUPERA EL ULTIMO CARACTER A ENVIAR
            JSR FILL_BUFFER ; ALMACENA EL ULTIMO BYTE RECIBIDO
            STAA TXBYTE ; RESUME COMUNICACIÓN CON LA PC
            LDAB TRAMA ; RECUPERA EL CONTADOR DE LA TRAMA
            LDAA #S2C ; HABILITA SCI
            STAA $102D
            JMP LLENADO

FILL_BUFFER: PSHA
            PSMB

```

```

PSHX
PSHY
LDX PPOINTER. ; <-ESTE PARÁMETRO SE PASA EN EL STACK
LDAA RXBYTE ; CON LA SUBROUTINA RECIBIR_PROG
STAA 0,X
INX
CPX #FIN
BEQ RESET_PPOINTER
STX PPOINTER
STAA TXBYTE ; PREPARA BYTE PARA REGRESARLO AL HOST
CLR LLENO ; lleno = false
LDAA #$7F
ANDA PORTA ; APAGA LED DE BUFFER LLENO
STAA PORTA
BRA RESTORE
RESET_PPOINTER: LDX #INICIO ; SI EL BUFFER LLENO, RESET VARIABLES
STX PPOINTER ; AL INICIO, ESTO SE HACE UNA VEZ QUE
STAA TXBYTE
INC LLENO ; SE ALMACENA EL ÚLTIMO BYTE,LLENO = TRUE
LDAA #$80
ORAA PORTA ; enciende led de buffer lleno A7
STAA PORTA
RESTORE: PULY
PULX
PULB
PULA
RTS

EJECUTAR: PSHA
PSHB
PSHX
PSHY
CLR SERIAL
LDX GPOINTER
LDY NPOINTER
LDAA GCODE
CMPA #'Y'
BEQ CASEY
CMPA #'Z'
BEQ CASEZJ
CASEX: LDAA TRAMA2
CMPA #$01
BEQ GCODE_X2
CMPA #$02

```

```

    BEQ GCODE_X3
    CMPA #$03
    BEQ GCODE_XYZ1

GCODE_XYZ1:  LDAA SERIAL      ;LEE CARACTER DEL PUERTO SERIAL.
              CMPA #FALSE    ;PARA VER SI NO HAY FIN DE EJECUCIÓN
              BEQ CONT_EJEC
              CLR SERIAL
              LDAA RXBYTE
              CMPA #'E'
              BEQ END1

CONT_EJEC:   CLR TRAMA2
              LDAA 0,X        ;GET(GCODE) FROM BUFFER
              STAA GCODE     ;SALVA EL CODIGO ACTUAL
              CMPA #'P'      ;¿EOF?
              BEQ END1
              INC TRAMA2
              CMPA #'Y'
              BEQ GCODE_Y2
              CMPA #'Z'
              BEQ GCODE_Z2

GCODE_X2:    INX
              CPX #FIN
              BEQ END1
              LDAB 0,X        ;GET(XCOOR_NIBBLE BAJO) FROM BUFFER
              INC TRAMA2

GCODE_X3:    INX
              CPX #FIN
              BEQ END1
              LDAA 0,X        ;GET(XCOOR_NIBBLE ALTO) FROM BUFFER
              STD NPOINTER
              INC TRAMA2
              STD WORDX       ;ACTUALIZA COORDENADA X
              JSR MOTORX
              LDY #$0000
              STY NPOINTER
              INX
              CPX #FIN
              BEQ END1
              BRA GCODE_XYZ1

END1        JMP END_EJECUTAR
CASEZJ      JMP CASEZ

```

```

CASEY:    LDAA TRAMA2
          CMPA #$01
          BEQ GCODE_Y2
          CMPA #$02
          BEQ GCODE_Y3
          CMPA #$03
          BEQ GCODE_XYZ1
GCODE_Y2: INX
          CPX #FIN
          BEQ END_EJECUTAR
          LDAB 0,X      ,GET(YCOOR_NIBBLE BAJO) FROM BUFFER
          INC TRAMA2
GCODE_Y3: INX
          CPX #FIN
          BEQ END_EJECUTAR
          LDAA 0,X     ,GET(YCOOR_NIBBLE ALTO) FROM BUFFER
          INC TRAMA2
          STD WORDY    ,ACTUALIZA COORDENADA X
          JSR MOTORY
          LDY #$0000
          STY NPOINTER
          INX
          CPX #FIN
          BEQ END_EJECUTAR
          JMP GCODE_XYZ1
    
```

```

CASEZ:    LDAA TRAMA2
          CMPA #$01
          BEQ GCODE_Z2
          CMPA #$02
          BEQ GCODE_Z3
          CMPA #$03
          BEQ XYZ1
GCODE_Z2: INX
          CPX #FIN
          BEQ END_EJECUTAR
          LDAB 0,X      ,GET(YCOOR_NIBBLE BAJO) FROM BUFFER
          INC TRAMA2
GCODE_Z3: INX
          CPX #FIN
          BEQ END_EJECUTAR
          LDAA 0,X     ,GET(YCOOR_NIBBLE ALTO) FROM BUFFER
          INC TRAMA2
          STD WORDZ    ,ACTUALIZA COORDENADA X
    
```



```

JSR MOTORZ
LDY #$0000
STY NPOINTER
INX
CPX #FIN
BEQ END_EJECUTAR
XYZI:    JMP GCODE_XYZI
    
```

```

END_EJECUTAR: PSHA
              LDAA LLENO
              CMPA #TRUE
              BEQ CIRCULAR      ;if(LLENO) then guarda variables
              CLR TRAMA2       ;reset al contador trama2(trama no trunca)
              CLR GCODE        ;y todas las demás variables que
              LDX #INICIO      ;intervienen en la rutina ejecutar
              STX GPOINTER     ;ya que se terminó el programa sin
              LDX #$0000
              STX NPOINTER     ;circular el buffer
              PULA
              PULY
              PULX
              PULB
              PULA
              RTS
    
```

```

CIRCULAR:    PULA
              CLR LLENO
              LDX #INICIO
              STX PPOINTER
              STD NPOINTER
              DEX
              STX GPOINTER
              PULY
              PULX
              PULB
              PULA
              RTS
    
```

```

MANUAL:     PSIIA
            PSII B
            PSII X
            PSII Y
            LDAA RXBYTE      ;REGRESA AL HOST EL BYTE 'm'
    
```

```

STAA TXBYTE
JSR TRANS
CLR SERIAL
RBYTEM: LDAA SERIAL      ;LEE CARACTER DEL AVANCE DEL PUERTO SERIAL
        CMPA #TRUE      ;ESPERA LA INTERRUPCION DEL PUERTO SERIE
        BNE RBYTEM
        CLR SERIAL
        LDAA RXBYTE
        STAA AVANCE     ;ACTUALIZA LA VARIABLE DE AVANCE
        STAA TXBYTE     ;LA REGRESA AL HOST
        JSR TRANS
        LDAA #$10      ;ENCIENDE EL LED DE OPERACIÓN MANUAL
        ORAA PORTA
        STAA PORTA
        JSR RESET_Z    ;POSICIONA LA MAQUINA EN ZERO MACHINE
        JSR RESET_Y
        JSR RESET_X
        LDAA #'m'      ;MANDA AVISO DE ACTIVO AL HOST
        STAA TXBYTE
        JSR TRANS
        CLR SERIAL     ;LIMPIA LA m QUE PRODUJO LA LLAMADA DE LA RUTINA
        CLR RXBYTE
RDBYTEM: LDAA SERIAL    ;LEE CARACTER DEL PUERTO SERIAL
        CMPA #TRUE     ;ESPERA LA INTERRUPCION DEL PUERTO SERIE
        BNE RDBYTEM
        CLR SERIAL
        LDAA RXBYTE    ;LEE EL CARACTER DEL BUFFER SERIAL
        CMPA #'D'      ;EJE X A LA IZQUIERDA +X
        BEQ CASE_DI
        CMPA #'I'      ;EJE X A LA DERECHA -X
        BEQ CASE_I1
        CMPA #'F'      ;EJE Y PENETRA +Y
        BEQ CASE_F1
        CMPA #'R'      ;EJE Y PENETRA -Y
        BEQ CASE_R1
        CMPA #'U'      ;EJE Z Arriba +Z
        BEQ CASE_U1
        CMPA #'L'      ;EJE Z ABAJO -Z
        BEQ CASE_L1
        CMPA #'O'      ;TALADRO ON
        BEQ CASE_O1
        CMPA #'S'      ;TALADRO OFF
        BEQ CASE_S1
        CMPA #'r'      ;RESET EJES
    
```

```

BEQ CASE_rel
CMPA #'c' ;CLEAR BUFFER
BEQ CASE_cl
CMPA #'B' ;ESTABLECER COMUNICACIÓN
BEQ CASE_B1
CMPA #'e' ;EJECUTAR
BEQ CASE_cl
CMPA #'a'
BEQ CASE_al ;AVANCE
CMPA #'Q' ;QUIT EJECUTAR
BNE CASE_ELSE
LDAA #$EF ;APAGA EL LED DE OPERACIÓN MANUAL
ANDA PORTA
STAA PORTA
PULY
PULX
PULB
PULA
RTS

```

```

CASE_D1: JMP CASE_D
CASE_I1: JMP CASE_I
CASE_F1: JMP CASE_F
CASE_R1: JMP CASE_R
CASE_U1: JMP CASE_U
CASE_L1: JMP CASE_L
CASE_O1: JMP CASE_O
CASE_S1: JMP CASE_S
CASE_rel: JMP CASE_re
CASE_cl: JMP CASE_c
CASE_B1: JMP CASE_B
CASE_cl: JMP CASE_e
CASE_al: JMP CASE_a
CASE_ELSE: JMP END_CASEM

```

```

CASE_I: LDY XACTUAL ;VERIFICA SI ESTA EN POSICIÓN CERO MÁQUINA
CPY #$0000 ;SI ESTA EN X0 THEN NO MOVER ELSE MOVER UN PASO
BEQ END_1
LDY WORDX ;MUEVE MOTOR_X A LA IZQUIERDA 0.01 MM(AVANCE DEL
PSIBB
LDAB AVANCE
BEQ CONT 1
WHILE 1: DEX
BEQ CONT 1

```

Apéndice #1: Diagramas y programas 87

```
    DECB
    CMPB #$00
    BNE WHILE_I
CONT_I: PULB
        STX WORDX
        JSR MOTORX
END_I:  JMP END_CASEM
CASE_D: LDX WORDX
        PSHB
        LDAB AVANCE
        BEQ CONT_D
WHILE_D: INX
        DECB
        CMPB #$00
        BNE WHILE_D
CONT_D: PULB
        STX WORDX
        JSR MOTORX
        JMP END_CASEM
CASE_F: LDX WORDY      ;MUEVE MOTOR_Y A LA DERECHA 0.05 MM(AVANCE DEL
        PSHB
        LDAB AVANCE
        BEQ CONT_F
WHILE_F: INX
        DECB
        CMPB #$00
        BNE WHILE_F
CONT_F: PULB
        STX WORDY
        JSR MOTORY
        BRA END_CASEM
CASE_R: LDX YACTUAL    ;VERIFICA SI ESTA EN POSICIÓN CERO MÁQUINA
        CPX #$0000    ;SI ESTA EN Y0 THEN NO MOVER ELSE MOVER UN PASO
        BEQ END_CASEM
        LDX WORDY
        PSHB
        LDAB AVANCE
        BEQ CONT_R
WHILE_R: DEX
        BEQ CONT_R
        DECB
        CMPB #$00
        BNE WHILE_R
CONT_R: PULB
```

```

STX WORDY
JSR MOTORY
BRA END_CASEM
CASE_U: PSHB
        LDAB AVANCE
UP:    JSR SUBIRZ_MANUAL
        DECB
        BNE UP
        PULB
        BRA END_CASEM
CASE_L: PSHB
        LDAB AVANCE
LOW:   JSR BAJARZ_MANUAL
        DECB
        BNE LOW
        PULB
        BRA END_CASEM
CASE_O: LDA #20 ;enciende el led de taladro activo
        ORAA PORTD ;indica que se está taladrando un punto
        STAA PORTD
        BRA END_CASEM
CASE_S: LDA PORTD ;apaga el led de motor z activo
        ANDA #SCF ;y el led de reseting ejes
        STAA PORTD
        BRA END_CASEM
CASE_re: JSR RESET_Z
        JSR RESET_Y
        JSR RESET_X
        BRA END_CASEM
CASE_c: JSR RESET_BUFFER
        BRA END_CASEM
CASE_a: STAA TXBYTE
        JSR TRANS
        CLR SERIAL
ARBYTEM: LDA SERIAL ;LEE CARACTER DEL AVANCE DEL PUERTO SERIAL
        CMPA #TRUE ;ESPERA LA INTERRUPCION DEL PUERTO SERIE
        BNE ARBYTEM
        CLR SERIAL
        LDA RXBYTE
        STAA AVANCE ;ACTUALIZA LA VARIABLE DE AVANCE
        BRA END_CASEM
CASE_e JSR EJECUTAR
CASE_B: NOP
END_CASEM: STAA TXBYTE
    
```

```

JSR TRANS
JMP RDBYTEM
SUBIRZ_MANUAL:PSHA
    PSHB
    PSHX
    PSHY
    LDAB ZBYTE
    LDAA PORTE    ;verifica si esta en la posición Zo
    ANDA #$04
    TSTA
    BNE MOUTZ2    ;si está en Zo no subir y regresar
    LSLB          ;zbyte <<=1
    CMPB #124     ;if (zbyte == 24) zbyte = 9
    BEQ MNUEVEZ
    CMPB #118     ;if (zbyte == 18) zbyte = 3
    BEQ MTRESZ
    BRA MOUTZ2
MNUEVEZ: LDAB #109    ;zbyte = 9
    BRA MOUTZ2
MTRESZ:  LDAB #103    ;zbyte = 3
MOUTZ2:  STAB ZBYTE   ;actualiza zbyte
    JSR ZPORT
    JSR DELAY
    PULY
    PULX
    PULB
    PULA
    RTS

```

```

BAJARZ_MANUAL:PSHA
    PSHB
    PSHX
    PSHY
    LDAB ZBYTE
    LDAA PORTE    ;verifica si esta en la posición Zf
    ANDA #$20
    TSTA
    BNE MOUTZ1    ;si está en Zo no subir y regresar
    LSRB          ;zbyte >>=1
    CMPB #101     ;if (zbyte == 1) zbyte = 9
    BEQ MNINEZ
    CMPB #104     ;if (zbyte == 4) zbyte = 12
    BEQ MDOCEZ
    BRA MOUTZ1

```

Apéndice #1: Diagramas y programas 90

```

MNINEZ:  LDAB #109      ;zbyte = 9
          BRA MOUTZ1
MDOCEZ:  LDAB #112     ;zbyte = 12
MOUTZ1:  STAB ZBYTE    ;actualiza zbyte
          JSR ZPORT
          JSR DELAY
          PULY
          PULX
          PULB
          PULA
          RTS

TRANS  PSHA
       PSHX
       TPA
       ORAA #$10      ;inhabilita interrupciones
       TAP
       LDAA TXBYTE
       LDX #$1000
       BRCLR $2E,X,$80,*
       STAA $102F
       PULX
       PULA
       CLI           ;habilita interrupciones
       RTS

SCI    LDX #$1000
       BRCLR $2E,X,$20,*
       LDAA $102F
       STAA RXBYTE   ;SERVICIO DE INTERRUPCION PUERTO SERIE
       LDAA #TRUE
       STAA SERIAL   ;HABILITA LA BANDERA DE CARACTER SERIAL
       RTI

RTIME: LDAA #$40     ;CLEAR RTIF
       STAA TFLG2
       DEC CLOCK
       LDAA CLOCK
       CMPA #$00
       BEQ LED
       RTI           ;TODAVIA NO ES EL TIEMPO

LED:   LDAA #CUENTA
       STAA CLOCK   ;RESET CLOCK
       LDAA PULSO   ;IF PULSO = 0 THEN LED = ON, PULSO = 1

```

```
      CMPA #FALSE
      BEQ ON
OFF:  CLRA      ; ELSE (PULSO=1,LED=ON) PULSO = 0 LED = OFF
      STAA PULSO
      LDAA PORTA
      ANDA #$DF
      STAA PORTA
      RTI
ON:   LDAA #TRUE
      STAA PULSO
      LDAA PORTA
      ORAA #$20
      STAA PORTA
      RTI
```

```
IRQINT: LDAA #$0F
        ORAA PORTG
        STAA PORTG
        RTI
```

```
XIRINT: LDAA #$F0
        ORAA PORTG
        STAA PORTG
        RTI
```

```
*****
* VECTORES DE INTERRUPCION
*****
```

```
ORG $FFF0
FDB RTIME
ORG $FFD6
FDB SCI
ORG $FFF2
FDB IRQINT
ORG $FFF4
FDB XIRINT
ORG $FFFE
FDB $E000
```

```
END
```



# ***UNIMAT***

***CNC Drilling Machine***

***MECATRONICA***

## Apéndice #2

### SISTEMA UNIMAT, Manual del usuario

El Sistema UNIMAT es un sistema de taladrado con control numérico que incluye tres partes :

- El ambiente integrado de desarrollo: UNIMAT.EXE
- El controlador
- La máquina-herramienta

#### Ambiente integrado de desarrollo: UNIMAT.EXE

UNIMAT.EXE es un programa que nos permite editar, verificar y ejecutar programas escritos en formato CNC estándar. También podemos controlar manualmente los movimientos de la máquina-herramienta utilizando el teclado de la computadora. Constituye, entonces, la interfaz entre la máquina y el usuario.

#### Instalación

Los requerimientos del sistema para poder instalar UNIMAT.EXE son los siguientes:

- Una computadora IBM-PC o compatible
- Un procesador 8086 o superior
- Sistema Operativo DOS 3.1 o superior
- 640 Kb de memoria convencional
- 640 Kb de espacio en disco

Para instalar UNIMAT.EXE, realice los siguientes pasos:

- 1.- En su disco duro cree el directorio UNIMAT

```
c:>mkdir unimat [enter]
```

- 2.- Coloque su disco original de UNIMAT en el drive A
- 3.- Copie el sistema al directorio UNIMAT de su disco duro:

```
c:>cd unimat [enter]  
c:>\unimat>copy a:\*. * [enter]
```

- 4.- Retire su disco original de la unidad A y guárdelo en un lugar seguro

Los archivos contenidos en su disco original del sistema UNIMAT son los siguientes:

- UNIMAT.EXE: Ambiente integrado de desarrollo del sistema
- DEMO.CNC: Programa fuente CNC de demostración del sistema

#### Utilización del ambiente integrado de desarrollo

Para iniciar un sesión con el sistema UNIMAT realice lo siguiente:

- 1.- En el directorio en donde se encuentre el sistema, escriba:

```
c:\>unimat\>unimat [leer.cnc] [salida.cod] (enter)
```

los parámetros entre [] son opcionales e indican lo siguiente:

**leer.cnc:** Indica el nombre del archivo a leerse en forma automática al iniciar el sistema y colocarse en la memoria de edición.

**salida.cod:** Indica el nombre del archivo de salida, en donde serán colocados los códigos G optimizados.

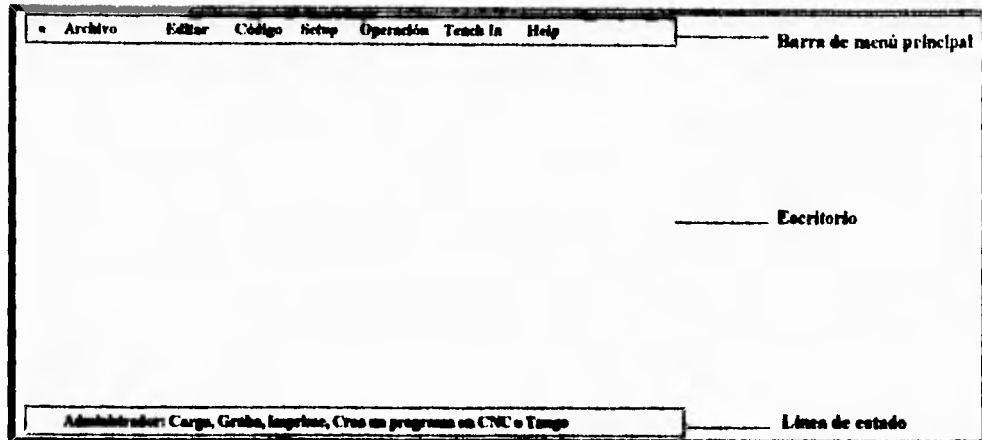
UNIMAT presenta la siguiente bienvenida:

**UNAM. Facultad de Ingeniería  
Centro de Diseño y Manufactura  
Departamento de Mecatrónica  
Laboratorio de Electrónica  
UNIMAT: CNC Drilling Machine  
Versión 1.0  
José Carlos Benítez Álvarez  
Copyright (C) CDM, México 1995**

**Pulse cualquier tecla ...**

Pulse cualquier tecla para ingresar a la zona de trabajo del sistema.

La zona de trabajo se muestra en la siguiente figura:



La barra de menú principal proporciona acceso a las diferentes herramientas del sistema utilizadas para crear, editar, generar y ejecutar programas CNC así como para controlar manualmente la máquina-herramienta.

La barra de estado indica la acción que el sistema realizará, así como lo que realiza según sea el caso, además de que proporciona las teclas disponibles para cancelar el proceso que se realiza.

El escritorio es la zona en donde se editan los programas CNC y en donde se muestran las diferentes ventanas utilizadas para desplegar menús, mensajes e instrucciones según sea la actividad que se realiza.

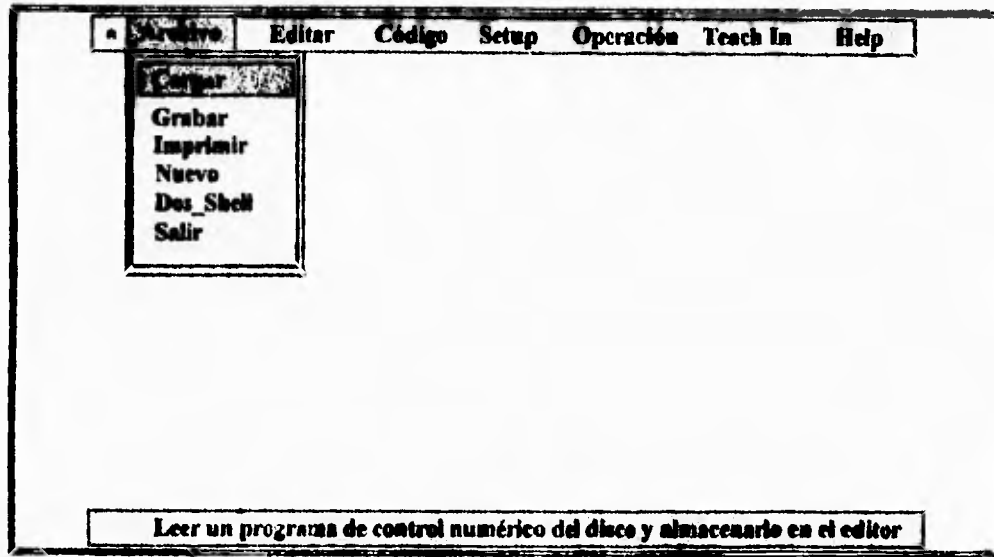
Para seleccionar las herramientas del menú principal teclee la letra mayúscula que se presenta en color amarillo, de cada título en el menú principal o utilice las teclas del cursor para posicionarse en la herramienta deseada y pulse enter o la tecla bajar del cursor.

#### **Símbolo de activo (\*)**

Indica que el sistema está activo y que el protector de pantalla está instalado. El protector de pantalla se habilita cada 30 segs. aproximadamente.

**El menú Archivo**

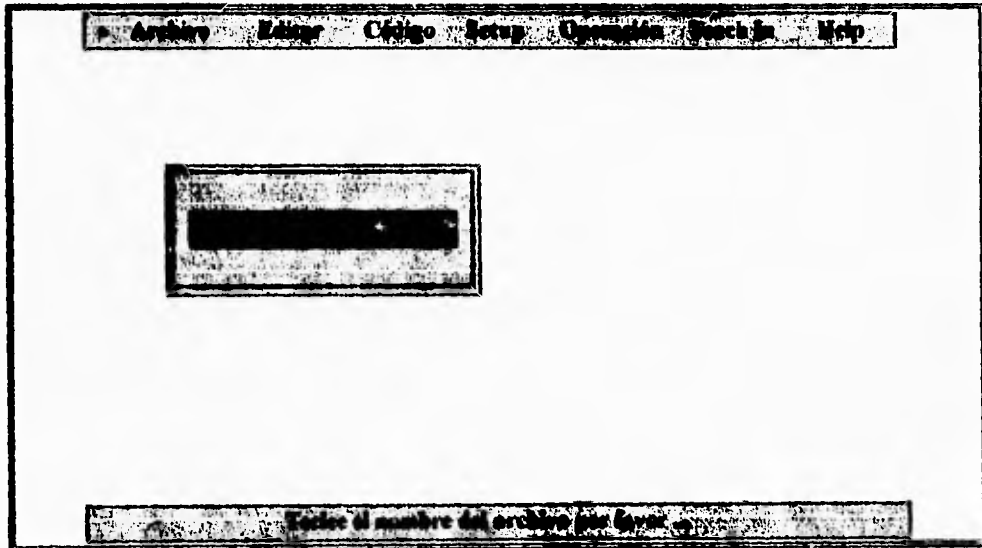
Cuando selecciona el menú Archivo del menú principal, se muestra la siguiente pantalla



El menú Archivo tiene las siguientes opciones:

Opción	Acción
Cargar	Leer un programa CNC del disco y almacenarlo en la memoria del editor
Grabar	Graba el programa en el editor en el disco
Imprimir	Imprime el programa activo en la memoria del editor en la impresora
Nuevo	Prepara la memoria del editor para editar un programa
Dos_Shell	Permite salir temporalmente al sistema operativo
Salir	Sale de UNIMAT.EXE

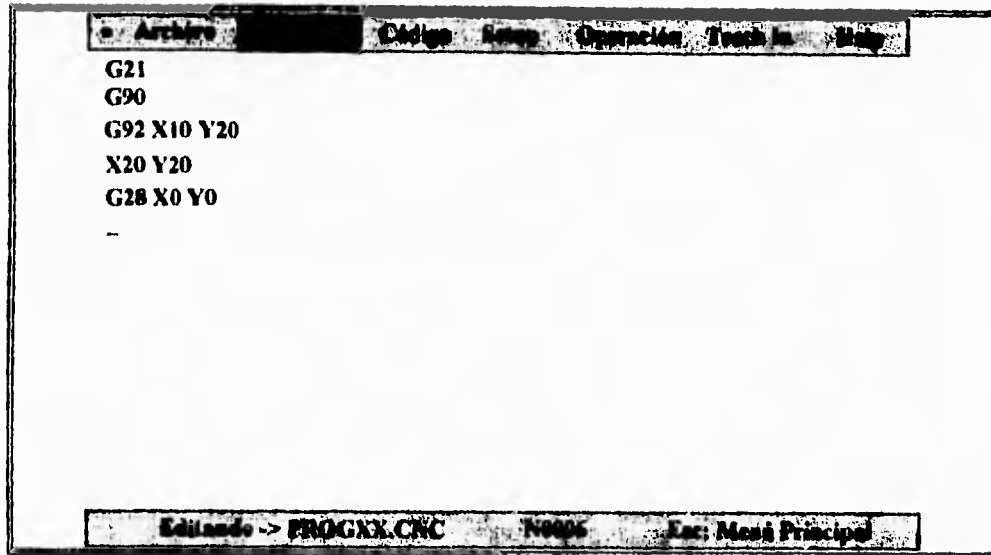
Por ejemplo cuando desea leer un programa del disco, seleccione el menú *ARCHIVO-CARGAR*, y el sistema le preguntará el nombre de archivo a leer, escríbalo y pulse enter, como se muestra en la siguiente figura:



Una vez que se ha leído el archivo, se muestra en el escritorio.

### **El menú Editar**

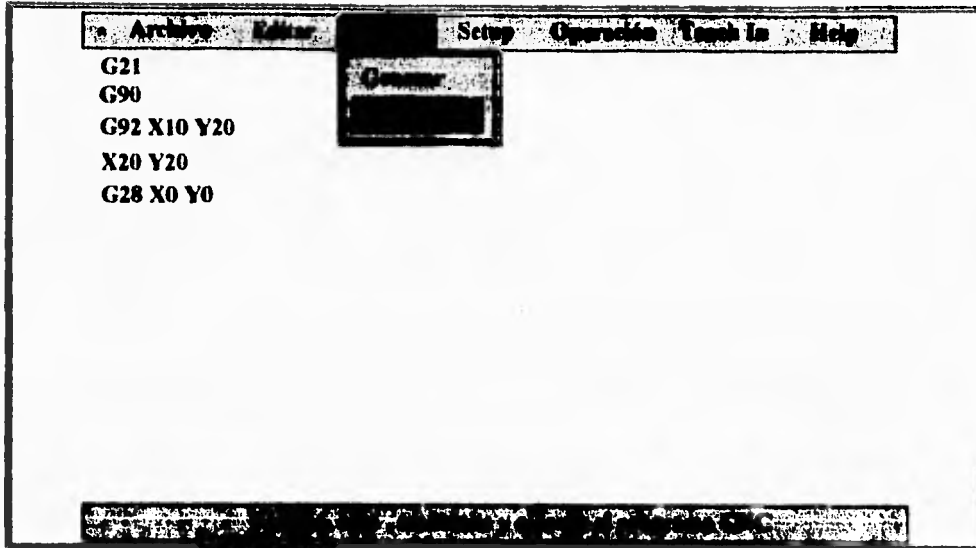
Cuando selecciona el menú *Editar* del menú principal, puede comenzar la edición de un nuevo programa y del programa que leyó del disco, como se muestra en la siguiente figura:



Una vez que haya terminado la edición del programa, pulse la tecla **Esc** para regresar al menú principal.

#### **El menú Código**

El menú **Código** se emplea para la verificación, optimización y ejecución del programa activo en el editor, como muestra la siguiente figura.



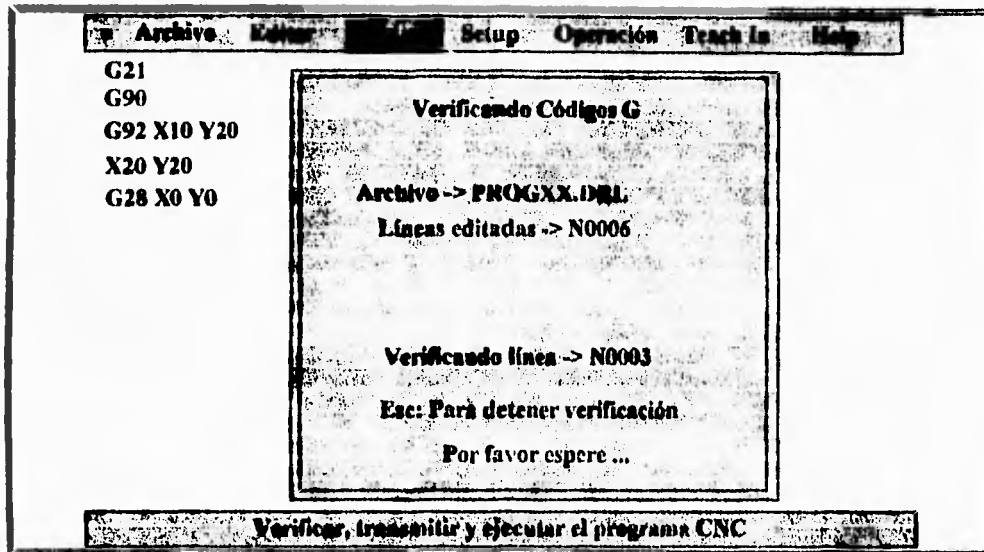
El menú Código tiene la siguientes opciones



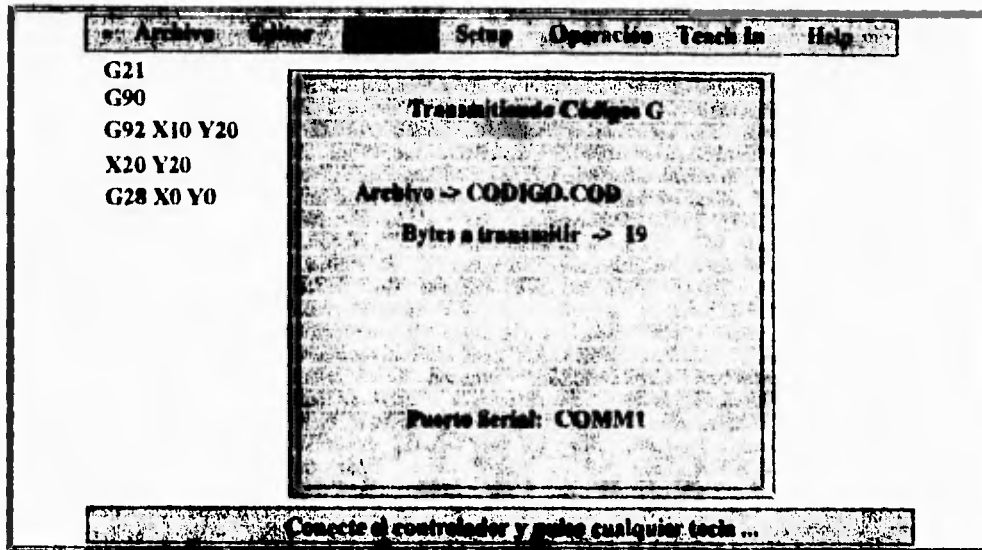
Opción	Acción
Generar	Verifica y genera el código CNC optimizado del programa en el editor
Transmitir	Verifica, genera y transmite el código CNC optimizado al controlador

Cuando selecciona el menú *Código-Generar*, se verifican los códigos y los parámetros correspondientes y se genera un archivo con extensión .COD en el disco duro, el cual contiene el código ejecutable por el controlador, como se muestra en la figura siguiente:





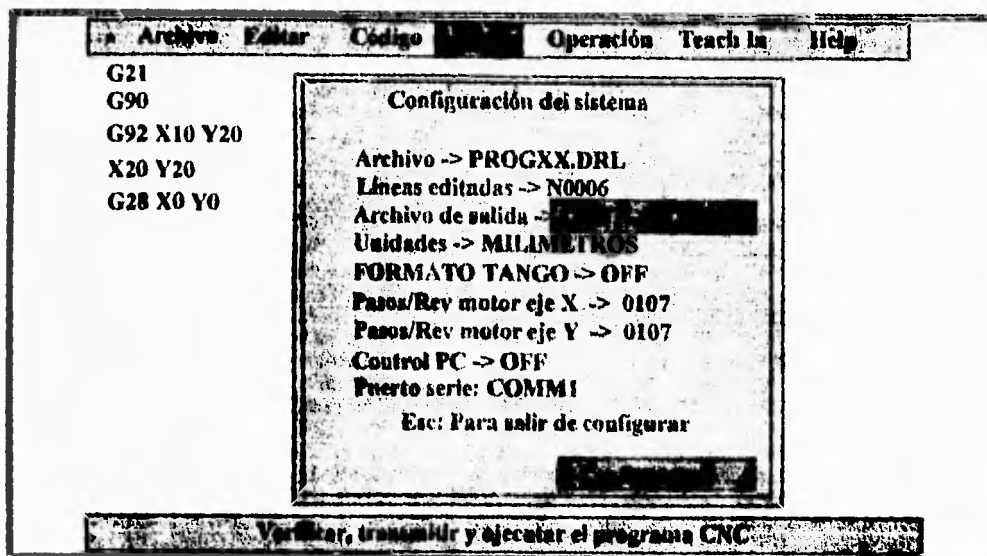
Si existe algún error, el sistema muestra el respectivo error en la pantalla. Si no existe ningún error en el programa, seleccione *Código-Transmitir* y el programa será transmitido al controlador para su ejecución. Puede utilizar la opción *Transmitir* para realizar el proceso de generación del código y transmitirlo directamente al controlador.



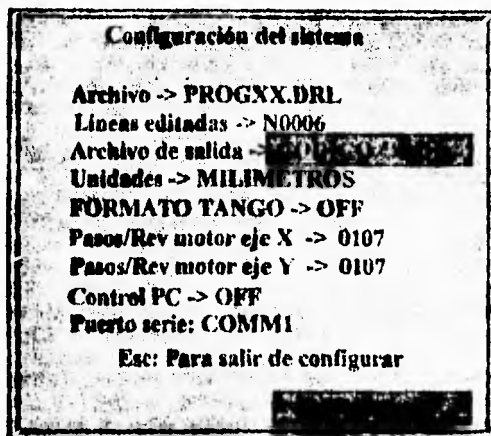
Cuando se ejecuta el programa, el sistema inicializa los ejes coordenados a su posición Home y comienza la ejecución del programa. Puede detener la ejecución del programa pulsando Esc o puede continuar con cualquier otra actividad en su computadora pulsando la tecla Return. Si desea puede desconectar el controlador del puerto serial. **Cuando pulsa Esc para detener la ejecución del programa, el controlador realiza la última instrucción que estaba realizando y se detiene.** En caso de requerir un paro de emergencia, pulse la tecla R ubicada en el panel del controlador.

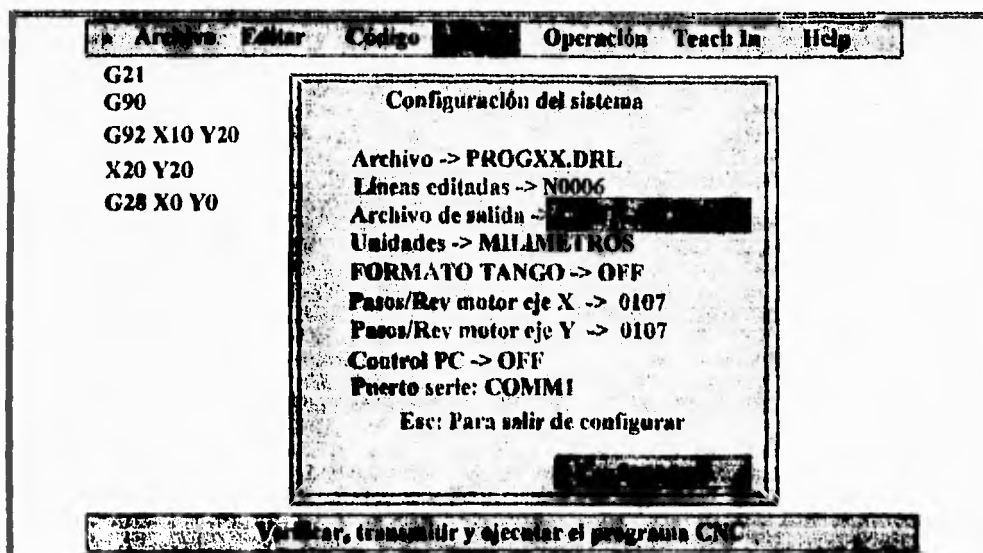
#### El menú de configuración Setup

Este menú contiene los parámetros de configuración del sistema que son necesarios para la realización del código ejecutable que será enviado al controlador así como otros parámetros como se muestra en la siguiente figura:

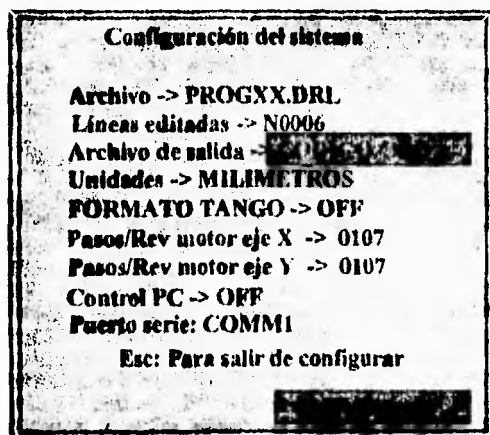


Los diferentes parámetros se explican a continuación:





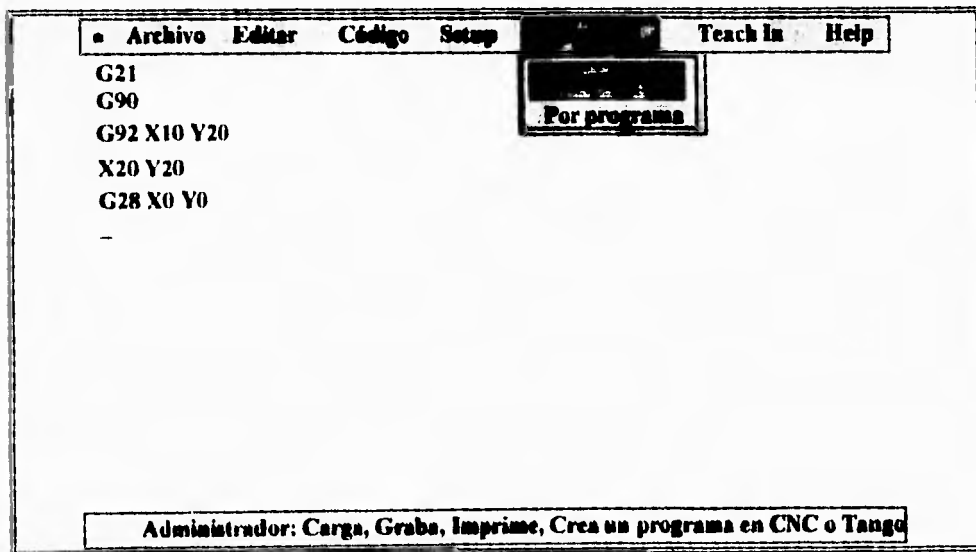
Los diferentes parámetros se explican a continuación:



Parámetro	Significado
Archivo	Archivo actual en la memoria de edición
Líneas editadas	Número de líneas editadas del programa
Archivo de salida	Archivo de salida del código ejecutable, debe especificarse aquí
Unidades	Unidades utilizadas: Milímetros o pulgadas (ajustable por programa)
Formato Tango	Indica si se programa en formato estándar o en formato TANGO
Pasos/Rev motor eje X	Relación de transmisión en pasos de motor entre el eje X y el motor X
Pasos/Rev motor eje Y	Relación de transmisión en pasos de motor entre el eje Y y el motor Y
Control PC	Permite controlar la máquina con la computadora (Reservado a JCBA)
Puerto serie	Especifica el puerto de comunicaciones utilizado: COM1 o COMM2

### El menú Operación

El menú Operación permite especificar el tipo de control de la máquina como se muestra en la siguiente figura:



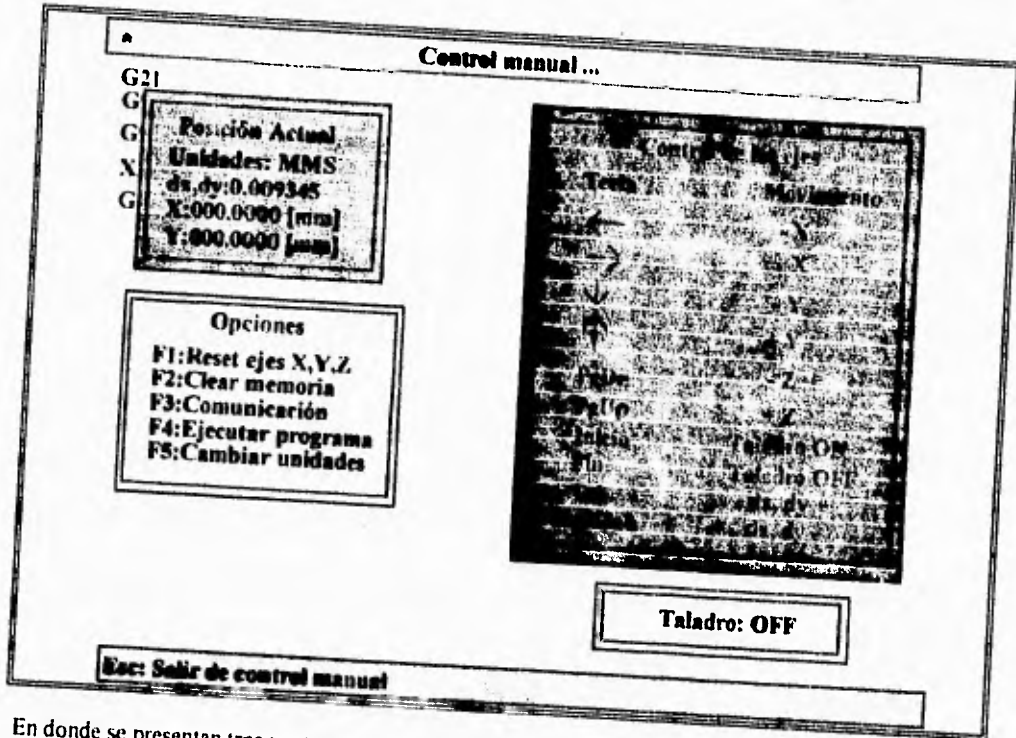
Este menú tiene las siguientes opciones:



Que incluyen lo siguiente:

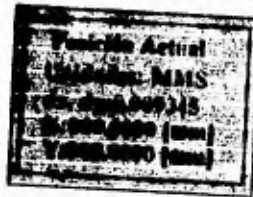
Opción	Acción
Manual	Permite el control de la máquina-herramienta por medio del teclado de la PC
Por programa	Especifica los parámetros necesarios para controlar la máquina por programa

Para controlar la máquina por medio del control manual seleccione el menú *Operación-Manual*, el sistema establece comunicación con el controlador, inicializa los ejes y muestra una pantalla como la siguiente



En donde se presentan tres ventanas que contienen lo siguiente.

**Ventana de Posición Actual**

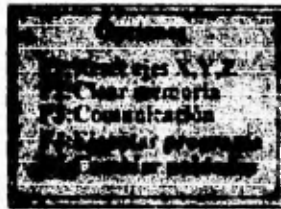


Opción	Parámetro
Unidades	Unidades utilizadas para el despliegue de la información
dx, dy	Tamaño del incremento para mover los ejes X-Y El eje Z es proporcional
X	Posición de la herramienta en el eje X
Y	Posición de la herramienta en el eje Y

El eje Z se mueve en forma proporcional a  $dx.dy$  ya que su movimiento está determinado por una trayectoria fija por los sensores  $Z_{home}$  y  $Z_{fin}$ .

### La ventana Opciones

Esta ventana permite realizar las operaciones siguientes



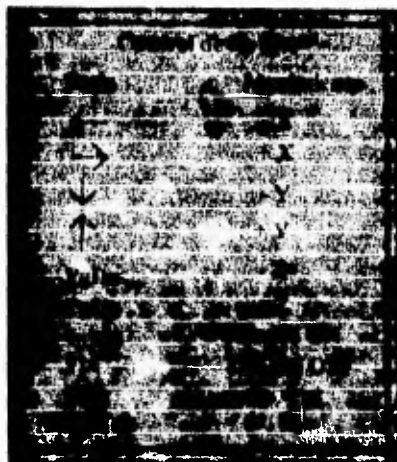
los cuales incluyen lo siguiente

Opción	Acción
F1	Inicializa los ejes X, Y, Z a su posición Home y detiene el husillo del taladro
F2	Limpiar la memoria RAM del controlador. Si existe un programa lo elimina
F3	Verifica el estado de la línea de comunicación
F4	Ejecuta el programa activo (si existe alguno) en la memoria RAM del controlador
F5	Cambia las unidades empleadas para el despliegue de la posición

### La ventana de Control de los ejes

Esta ventana muestra el conjunto de teclas disponibles para controlar los movimientos de la máquina-herramienta como se muestra en la siguiente figura





Las teclas operan como se indica en la siguiente tabla:

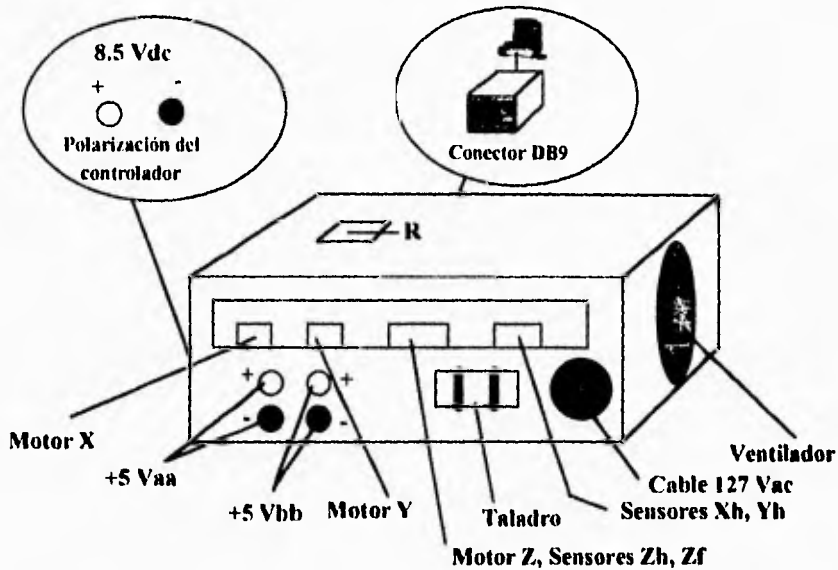
Tecla	Acción
Cursor Izq	Mueve la herramienta en la dirección -X según el incremento especificado
Cursor Der	Mueve la herramienta en la dirección +X según el incremento especificado
Cursor Abajo	Mueve la herramienta en la dirección -Y según el incremento especificado
Cursor Arriba	Mueve la herramienta en la dirección +Y según el incremento especificado
Avanzar Página	Mueve la herramienta en la dirección -Z proporcional al incremento especificado
Regresar Página	Mueve la herramienta en la dirección +Z proporcional al incremento especificado
Inicio	Enciende el husillo del taladro
Fin	Apaga el husillo del taladro
Tab	Aumenta dx, dy en 0.009345 mm ( 0.000367 in)
Shift Tab	Disminuye dx, dy en 0.009345 mm ( 0.000367 in)

Los menús **Teach In** (enseñar) y **Help** (ayuda) no están implementados y están reservados para modificaciones posteriores.

## 2.- El controlador

El controlador del sistema lo constituyen los circuitos electrónicos encargados de suministrar las señales de control a la máquina-herramienta. Estos se encuentran contenidos en un gabinete el cual permite el conexionado de los motores y de los sensores de la máquina, así como del sistema de ventilación.

Además permite la conexión con la IBM-PC y compatibles por medio de un cable RS232C compatible. Para el conexionado de la máquina refiérase a la siguiente figura:



**Notas:**

- 1.- Las fuentes de potencia Vaa, Vbb deben estar reguladas a 5 V y proporcionar una corriente de 2.5 Amp como mínimo cada una. Puede ser una sola fuente de 5 Vdc @ 5 Amp mínimo, pero no es recomendable.
- 2.- La fuente de potencia de 8.5 Vdc debe suministrar 1 Amp mínimo. Debe ser completamente independiente a las fuentes de los motores, esto con el fin de aislar al circuito de control del circuito de potencia.
- 3.- El botón "R" proporciona el Reset al sistema de control y sirve como paro de emergencia.
- 4.- Cada vez que se encienda el controlador pulse el botón R para proporcionar un reset al sistema si éste no responde al encendido de la fuente de potencia.
- 5.- Para realizar las conexiones de los motores y sensores, puede abrir el gabinete del controlador y hacer las conexiones de tal manera que los cables correspondientes salgan por la ventana de conexiones.

**Códigos disponibles en el controlador CNC**

Los códigos CNC disponibles en el controlador son los siguientes:

**CÓDIGOS DISPONIBLES EN EL CONTROLADOR**

<b>Código</b>	<b>Acción</b>	<b>Tipo</b>
<b>G20</b>	Introducción de dimensiones en <b>pulgadas</b>	Modal
<b>G21</b>	Introducción de dimensiones en <b>milímetros</b>	Modal
<b>G28</b>	Movimiento de la herramienta de corte sin taladrar	No modal
<b>G90</b>	Programación absoluta	Modal
<b>G91</b>	Programación incremental	Modal
<b>G92</b>	Definición del cero pieza o cero programa	Modal
<b>X__</b>	Especifica el valor de la coordenada X a taladrar	No modal
<b>Y__</b>	Especifica el valor de la coordenada Y a taladrar	No modal
<b>%</b>	Limitador de programa	-
<b>;</b>	Introducción de comentarios en el programa	-

**3.- La máquina-herramienta**

La máquina-herramienta la constituye la máquina UNIMAT de EMCO Co. con los componentes mecánicos agregados para convertirla en una máquina CNC. En conjunto la máquina-herramienta tiene las siguientes características:

- Peso: 20 kg
- Dimensiones: Largo : 70.0 cm ( 27.5590 in )  
Ancho: 35.0 cm ( 13.7795 in )  
Altura: 50.0 cm ( 19.6850 )
- Longitud del eje X: 220 mm ( 8.6614 in )
- Longitud del eje Y: 45 mm ( 1.7716 in )
- Longitud del eje Z: 35 mm ( 1.3779 in )

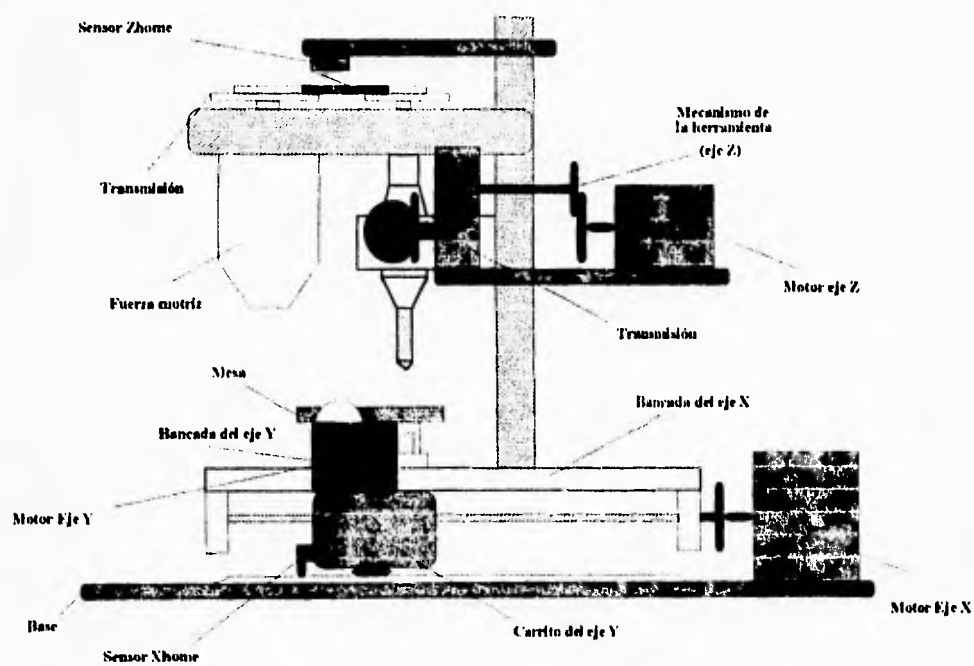
Todas las partes mecánicas se encuentran montadas sobre una base de madera que le permite absorber las vibraciones además de que proporciona apoyo a la máquina-herramienta.

Para operar la máquina-herramienta, es recomendable utilizar unos lentes protectores para proteger la vista en caso de ruptura de la herramienta de corte, además de colocarla en un lugar adecuado que soporte el peso y los esfuerzos durante el taladrado.

La máquina-herramienta es capaz de perforar materiales tales como:

- Aluminio
- Madera
- Acrílico
- Baquelita
- Plástico

La disposición final de la máquina se muestra en la siguiente figura:



## **Bibliografía**

Kassab, Vincent. **Technical C Programming**  
E.U.A, Prentice-Hall, 1989.

Schildt, Herber. **The Art of C , Elegant programming solutions**  
E.U.A., McGraw-Hill, 1991.

Schildt, Herbert. **Turbo C, Programación Avanzada**  
México, McGraw-Hill, 1990.

Jamsa, Kris, **Turbo C programmers library**  
E.U.A., McGraw-Hill, 1988.

Hansen. **C programming**  
E.U.A., Adison Wesley, 1990.

Kernighan, Brian W; Ritchie, Dennis M. **El lenguaje de programación C**. 2a. edición. México, Prentice-Hall, 1991.

Terrier, François. **Borland C++, mode d'emploi**. Vol 1.  
France, Sybex, 1991.

Yahouédéou, Janvier. **C et Turbo C, Structures de Données**  
France, Sybex, 1989.

Rojas, Alberto Ponce. **Ensamblador Básico**.  
México, Computec, 1993.

Goffon, Peter W. **Mastering Serial Communications**.  
2nd. edition. E.U.A, Sybex, 1994.

Robinson;Pollack. **Computer Numerical Control**  
E.U.A., Prentice-Hall, 1990.

Valentino, James V.; Goldenberg, Joseph.  
**Introduction to Computer Numerical Control**.  
E.U.A., Prentice-Hall, 1993.

## Bibliografía

---

Lynch, Mike. **Computer Numerical Control for Machining.**  
E.U.A., Mc-Graw-Hill, 1992.

Vergnas, Jean. **Máquinas-herramientas con Control Numérico**  
España, Urmo S.A. de Ediciones, 1990

Gonzalez Nuñez, Juan.  
**El control numérico en las máquinas-herramienta**  
México, C.E.C.S.A., 1990

**Modern Machine Shop (Revista)**  
April 1994

Sokoloff, Leonard; Gayakwad, Ramakant.  
**Analog and Digital Control Systems.**  
E.U.A., Prentice-Hall, 1988.

Curtin. **Microprocessors, software and applications**  
E.U.A., Prentice-Hall, 1990

Motorola. **MC68HC11F1, User's manual.**  
E.U.A., Motorola, 1988

Amstead, B.H., Ostwald, Phillip F., Begeman, Myron L.  
**Procesos de Manufactura, versión SI.**  
México, C.E.C.S.A., 1979.