

**DESARROLLO DE UN SISTEMA DE CONTROL PARA UNA
MICROMÁQUINA HERRAMIENTA BASADO EN CÓDIGO ESTÁNDAR PARA
CNC**

A u t o r e s :

Alberto Caballero Ruiz, Leopoldo Ruiz Huerta, Héctor Hugo Silva López.

(Informe Técnico)

D e p a r t a m e n t o d e a d s c r i p c i ó n :

Instrumentación y Medición

G r u p o :

Micromecánica y Mecatrónica, GMM.

Ciudad Universitaria, 5 de diciembre de 2010.

ÍNDICE.

Resumen.....	1
1. Antecedentes.....	1
1.1 Micromecánica en el mundo.....	3
1.2 Micromecánica en México.....	4
1.3 Sistemas de control implementados en el GMM..... anteriores a este trabajo.	6
1.4 Sistemas CNC.....	8
1.4.1 Definición de CNC.....	8
1.4.2 Clasificación del CNC.....	9
1.4.3 Ventajas y desventajas del CNC.....	9
1.4.4 Programación y códigos.....	10
2. Objetivo y alcances.....	11
2.1 Definición del problema.....	11
2.2 Objetivo.....	12
3. Marco teórico.....	12
3.1 Introducción a la programación en sistemas CNC.....	13
3.2 Programación en sistemas CNC.....	13
3.2.1 Estructura de un programa para CNC.....	13
3.2.2 Principales códigos de programación CNC.....	14
3.3 Técnicas de control en sistemas CNC.....	15
3.3.1 Técnica por referencia de pulsos.....	15
3.3.2 Técnica por muestreo de datos.....	15
3.4 Interpolación por DDA.....	16
3.4.1 Interpolación lineal.....	16
3.4.2 Interpolación circular.....	18
3.5 Interpolación incremental (ó por escalones).....	19
3.6 Interpolación por búsqueda directa.....	20
4. Desarrollo del sistema de control.....	22
4.1 Caso de estudio.....	22
4.2 Características esperadas.....	23
4.3 Propuesta de control.....	23
4.3.1 Características del sistema de control.....	25
4.3.1a Modos de operación.....	25
4.3.1b Parámetros monitoreados.....	26
4.3.2 Códigos G y M soportados.....	26
4.3.2a Descripción y sintaxis de los códigos...	27
4.4 Simulación de los métodos de interpolación.....	27
4.5 Evaluación de los métodos de interpolación.....	30
4.6 Selección del método de interpolación.....	31
4.7 Programa de control.....	32
4.8 Interfaz computacional.....	33
4.9 Ejemplos del proceso de control.....	36
4.9.1 Ejemplo (a).....	36
4.9.2 Ejemplo (b).....	38
4.10 Hardware electrónico.....	40
4.10.1 Circuito PBL3717A.....	40
4.10.2 Circuito A3988.....	42

4.10.3 Evaluación y selección del circuito..... de potencia	43
5. Pruebas y resultados.....	45
5.1 Pruebas realizadas al <i>hardware</i> electrónico.....	46
5.2 Maquinado de figuras simples.....	50
5.3 Maquinado con interpolación en dos ejes.....	51
5.4 Ejecución de un programa en código estándar G y M.....	54
Conclusiones.....	58
Referencias.....	60
Anexos.....	62
A.1 Descripción y sintaxis de los códigos.....	62
A.2 Comandos para activar modos de operación..... en la interfaz computacional.	64
A.3 Diagrama y tarjeta electrónica de potencia.....	65
A.4 Código del programa de control grabado en el Microcontrolador PIC18F8720.	68

RESUMEN.

En las últimas décadas, los microsistemas se han convertido en un área de interés para el sector científico y tecnológico a nivel mundial. En consecuencia, se han desarrollado una gran variedad de microdispositivos enfocados a diversos campos del sector científico; uno de esos campos es el de la manufactura de piezas con dimensiones pequeñas, donde el uso de máquinas herramienta de tamaño convencional en la manufactura de piezas con tamaños entre unos milímetros y unas centenas de micrómetros representa un alto costo en el equipo empleado, consumos de energía y de espacio. Hasta este momento se han desarrollado diversos prototipos a nivel mundial, los cuales cuentan con interfaces de control hechas acorde a las características de cada equipo y grupo de trabajo. En este trabajo se presenta el desarrollo de un sistema de control CNC (Control Numérico asistido por Computadora) para una micromáquina herramienta que permita emplear código estándar en su programación. Como parte de dicho desarrollo se considerarán tres algoritmos de interpolación de los cuales se seleccionará el más adecuado mediante matrices de decisión, con el fin de generar el maquinado de figuras de dimensiones menores al milímetro.

1. ANTECEDENTES.

Actualmente a nivel mundial, se aprecia una tendencia en el desarrollo de nuevas tecnologías enfocadas a la miniaturización de los sistemas, ejemplos de estas tecnologías son los *MicroElectroMechanical Systems* (MEMS), desarrollados principalmente en los Estados Unidos de América, *MicroSystems Technology* (MST), cuyo mayor auge está en los países europeos; y por último los *MicroMachine Technology* (MMT) desarrollados en Japón [Kussul et al, 1996].

El principal interés en el desarrollo de nuevas tecnologías para la fabricación de elementos micromecánicos, obedece a las restricciones causadas por los materiales y geometrías de las piezas [Ruiz, 2005]. Como ya se ha mencionado anteriormente, el motivo de hacer uso de las técnicas de la mecánica convencional para la obtención de micromecanismos tiene como fin adoptar procedimientos conocidos que nos permitan obtener las tolerancias requeridas.

En el diagrama de la figura 1, se muestra (en naranja) el camino que el Grupo de Micromecánica y Mecatrónica (GMM) ha seguido desde el año 1999 para el desarrollo de microequipo. Por un lado tenemos la tecnología de los MEMS, la cual puede ser utilizada para crear dispositivos de bajas dimensiones con componentes mecánicos simples generados a partir de técnicas de fabricación de circuitos integrados, sin embargo estos dispositivos poseen prácticamente dos dimensiones debido a su escaso espesor [Ruiz, 2000].

Debido a que para el desarrollo de sistemas micromecánicos (basados en mecánica convencional), como: MicroMáquinas Herramienta (MMH), micromanipuladores y microrobots, se requieren de estructuras mecánicas con las siguientes características [Kussul et al, 1996]:

- Tres dimensiones reales.
- Formas complejas.
- Posibilidades de ensamble y movimiento.
- Uso de diferentes materiales.

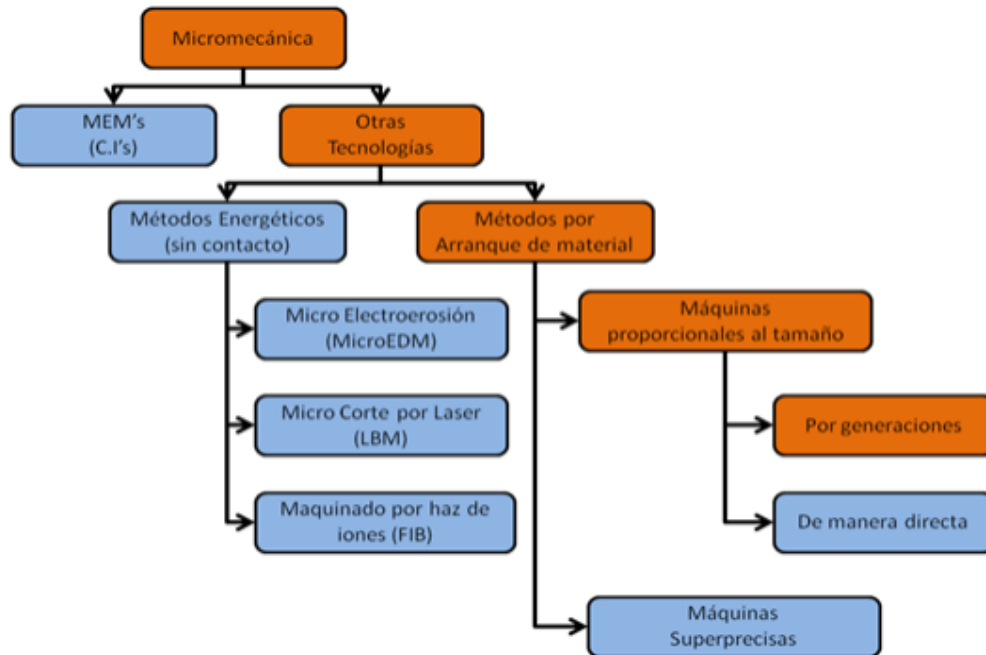


Figura 1. Desarrollo de la Micromecánica en el GMM [Ruiz y Caballero, 2006].

Esto nos deja ver que es necesario buscar otro camino para el desarrollo de microdispositivos complejos. Así pues, otras tecnologías existentes difieren en el aspecto de desbaste, dividiéndose básicamente en métodos energéticos (sin contacto) y por arranque de material. Actualmente, los prototipos de MMH manufacturados en el GMM, adoptan la metodología por arranque de material, sin implicar esto que no se esté interactuando con métodos energéticos.

Por lo tanto, para generar microequipo se tienen dos alternativas a seguir: la primera de ellas consiste en el empleo de máquinas superprecisas, mientras que la segunda se basa en la disminución de tamaño de máquinas herramienta (MH) y manipuladores [German, 2007].

El empleo de maquinaria superprecisa no es del todo recomendable para una aplicación comercial, ya que requiere de una gran inversión debido a que su costo es elevado. Mientras que, la disminución en tamaño de MH y manipuladores, resulta en un decremento en los costos de producción de dispositivos comerciales [Ruiz, 2000].

El principio básico propuesto por el GMM para generar microequipo a bajo costo es desarrollar generaciones de microequipo. En consecuencia, el GMM desarrolló máquinas generacionales, con dimensiones proporcionales a las piezas de trabajo.

Esta última decisión se basa en el método conocido como Tecnología de Microequipo (MET), el cual consiste en desarrollar microdispositivos generacionales; es decir, se desarrollan máquinas que posteriormente se utilizarán para el desarrollo de generaciones subsecuentes de menor tamaño [Kussul et al, 1996].

1.1. Micromecánica en el mundo.

El auge por el desarrollo de microdispositivos se inició a mediados de los años 80's; por ejemplo, en el campo de la medicina la demanda de microequipo aumenta con aplicaciones en microscopía, diagnóstico, intervenciones quirúrgicas poco invasivas, tratamientos, etc. La reducción de dimensiones en el instrumental médico permite que la inspección en el cuerpo humano sea menos riesgosa, así mismo, eleva la precisión y eficiencia en los diagnósticos [Ruiz, 2000].



Figura 2. Microrobots como ayudantes en intervenciones quirúrgicas¹.

Como ejemplos de las diversas aplicaciones de los sistemas micromecánicos están dos pequeños robots presentados en la figura 2, los cuales están destinados a la inspección en ojos (izquierda) y espalda (derecha). En la figura 3a, se muestran un microrobot para tareas de inspección desarrollado por laboratorios *Sandia* de 40x19x17mm con menos de 30gr de peso. Por otro lado, en la figura 3b, se muestra el microrobot volador desarrollado por *Seiko-Epson Co.* de 85mm de alto por 136mm de diámetro con un peso de 13gr y autonomía de vuelo por 3 minutos.

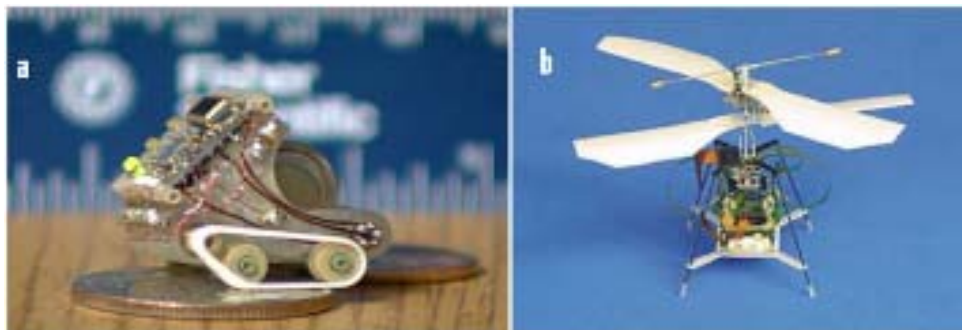


Figura 3. (a) Microrobot para tareas de inspección² (b) Microrobot volador³.

¹ <http://www.faulhaber-group.com/n380604/n.html>

En lo que respecta al desarrollo de MMH, en la figura 4 se muestran tres desarrollos importantes. En la figura 4a, se muestra un microtorno desarrollado en el *Mechanical Engineering Laboratory* del *National Institute of Advanced Industrial Science and Technology* de Japón con dimensiones 30x30x30mm [Okazaki, 2000]. En la figura 4b, se muestra una MMH desarrollada en el *International Research and Training Center of UNESCO/IIP of Information Technologies and Systems, National Academy of Science of Ukraine* [Kussul y Kasatkina, 1998]. Mientras que en la figura 4c, se muestra una microfábrica automatizada desarrollada en el año 2000 por investigadores del *Mechanical Engineering Laboratory* del *National Institute of Advanced Industrial Science and Technology* en Japón [Tanaka, 2001].

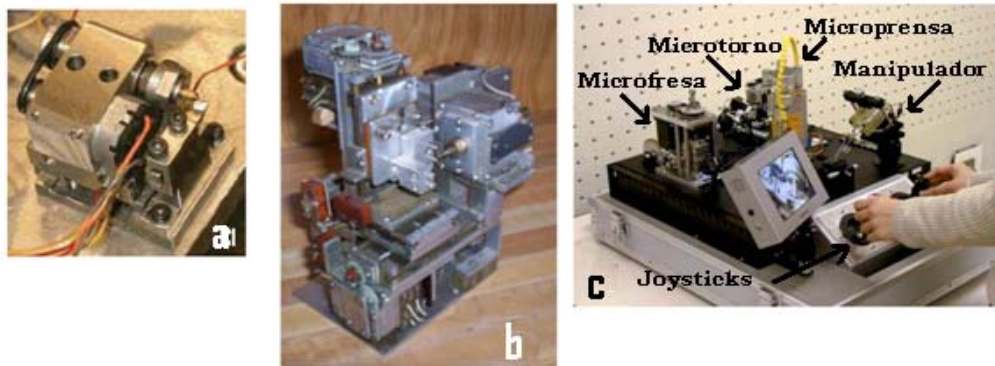


Figura 4. (a) Microtorno japonés, (b) Microcentro de maquinado ucraniano, (c) Microfábrica japonesa automatizada.

1.2. Micromecánica en México.

En México, el GMM en su línea de investigación referente al desarrollo de tecnología de punta para la fabricación de micropiezas, da inicio a la creación de microequipo generacional presentando en el año 2000 (dentro del *First International Conference on Mechatronics and Robotics* en San Petersburgo, Rusia) el primer prototipo de MMH, mostrado en la figura 5, con capacidad de realizar procesos de torneado, fresado y taladrado. En consecuencia, se da inicio a la primera generación de equipo micromecánico automatizado de bajo costo desarrollado totalmente en México [Ruiz, 2005].

Las principales características de esta MMH son: dimensiones de 130x160x85mm, tres grados de libertad que permiten desplazamientos lineales y uno rotacional, como actuadores se utilizan motores a pasos desarrollados en la misma entidad, resolución de 1.87µm lineales por paso del motor, y capacidad para producir piezas en un intervalo de los 100µm a los 5mm.

² <http://www.sandia.gov/media/NewsRel/NR2001/minirobot.htm>

³ <http://www.linuxelectronics.com/news/general/epson-announces-advanced-modelworlds-lightest-micro-flying-robot>

Como resultado de la experiencia que se obtuvo durante el diseño y construcción del prototipo de la figura 5, el GMM, en el año 2005 presenta el segundo prototipo de MMH de primera generación, basado en mecanismos de palancas y paralelogramos, lo cual permite que los movimientos desarrollados tengan una mejor resolución pero generando como movimiento secciones circulares.



Figura 5. Primer prototipo de MMH fabricado en México.

Este segundo prototipo mostrado en la figura 6, tiene dimensiones de 180x210x135mm, cuatro grados de libertad, los actuadores empleados son motores a pasos, la resolución lineal es de $0.596\mu\text{m}$ y tiene la capacidad de producir piezas en un intervalo de $50\mu\text{m}$ hasta 5mm. Este prototipo está diseñado para operar mediante el control de un sistema de cómputo convencional.



Figura 6. Segundo prototipo de MMH fabricado en México.

1.3. Sistemas de control implementados en el GMM anteriores a este trabajo.

Actualmente, el GMM cuenta con dos sistemas de control que han arrojado resultados valiosos para la automatización de equipo micromecánico [Caballero, 2000], [Yañez, 2007]. El primer sistema de control implementado para el prototipo de MMH de la figura 5, fue diseñado con el objetivo de controlarlo mediante una computadora personal (PC), eliminando en medida de lo posible el hardware de control externo. El sistema está compuesto de un amplificador de DC de 16 canales, y cuatro acondicionadores de señal para los sensores de contacto colocados al principio de carrera de cada eje [Caballero, 2000].

La MMH cuenta con un conector DB25, el cual contiene las conexiones a los motores y sensores de posición de origen mediante contacto físico, y posición de materia prima por contacto eléctrico. Además, el *software* de control empleado, mostrado en la figura 7, fue desarrollado en Borland C++, y es mediante esta herramienta que se monitorea el estado del sistema micromecánico a través de dos puertos paralelos que se encuentran divididos en tres secciones, dos de ellas dedicadas al movimiento de los actuadores y la última a las señales de retroalimentación provenientes de los sensores de posición por contacto físico y por contacto eléctrico [Caballero, 2000].

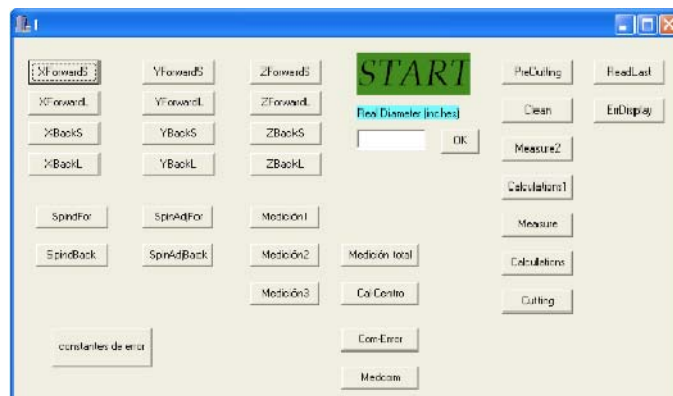


Figura 7. Software de control empleado en el primer prototipo.

Los resultados obtenidos con este sistema de control, son los mostrados en la figura 8. En donde podemos apreciar las piezas fabricadas de forma automática con este sistema de control.

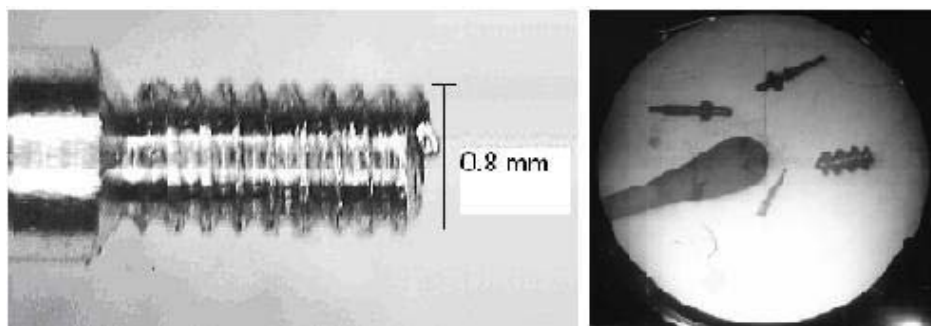


Figura 8. Micropiezas fabricadas con el primer prototipo [Reséndiz, 2005].

El segundo sistema de control, y más reciente, empleado en el prototipo de MMH de la figura 5, lo conforma una aplicación computacional desarrollada en Visual Basic [Yañez, 2007], la cual es capaz de interpretar código estándar para CNC. Este sistema de control hace uso de los mismos recursos (en cuanto a *hardware*) que el primer sistema de control (Figura 9).

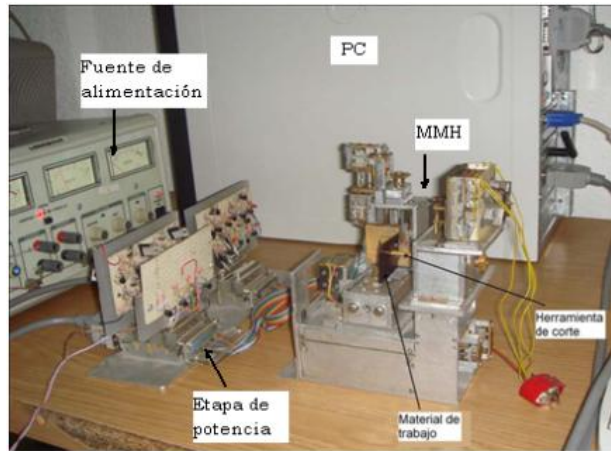


Figura 9. Recursos empleados por la aplicación computacional.

El procedimiento para manufacturar una pieza en este sistema de control, comienza después de haber generado el dibujo en un software tipo CAD, y por medio de un software CAM generar el código correspondiente para la manufactura, dicho código se importa por medio de la aplicación que se muestra en la figura 10.

Los resultados de esta aplicación computacional se muestran en la figura 11, los cuales consistieron básicamente en maquinados de figuras simples, figuras en donde se empleaban interpolaciones en dos ejes, y la ejecución de dos programas para CNC, que representaban el escudo de los pumas con dimensiones resultantes de 4mm y el logo del GMM dimensionado en 12 mm.

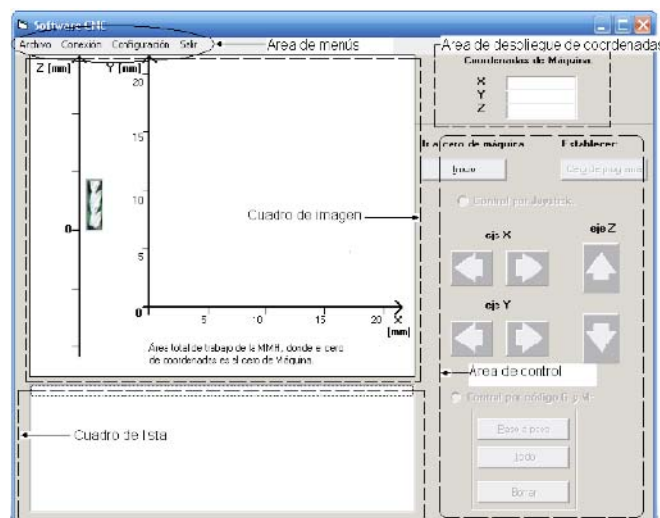


Figura 10. Pantalla principal de la interfaz computacional para el control de MMH [Yañez, 2007].

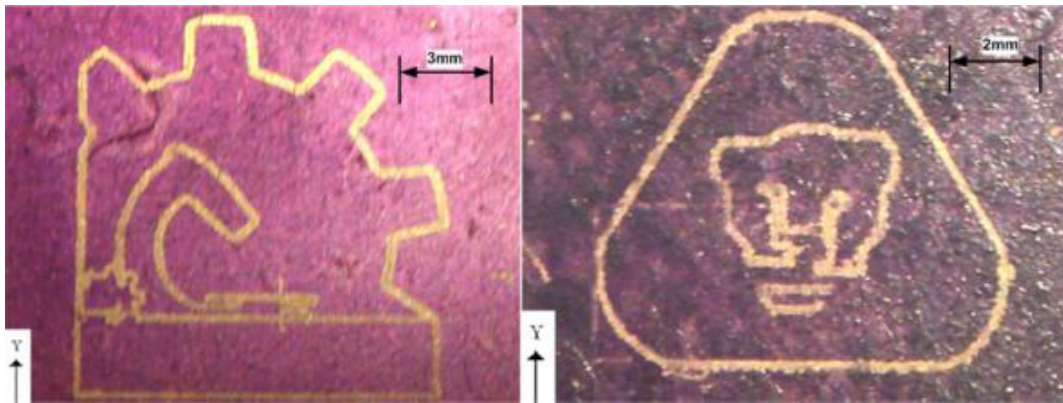


Figura 11. Resultados obtenidos por la aplicación computacional programada en *Visual Basic* [Yañez, 2007].

En lo que respecta al segundo prototipo de MMH mostrado en la figura 6, y con el fin de tener una estimación del posible comportamiento de éste, se tiene desarrollado un programa (en plataforma *Borland C++ Builder*), con soluciones por métodos numéricos, de las ecuaciones relacionadas con las trayectorias de los casquetes semicirculares, generados por el desplazamiento en cada uno de los ejes [Ruiz, 2005].

Así pues, durante el desarrollo de estos sistemas de control se han obtenido conclusiones importantes, por ejemplo, la interfaz computacional de la figura 10 presenta una ventaja fundamental en el desarrollo de micropartes respecto al primer sistema de control mostrado en la figura 7. Mientras que en el primer sistema de control el proceso de manufactura se programa en lenguaje C++, en el segundo sistema de control se reconoce código estándar para CNC. El interpretar código estándar, reduce y simplifica considerablemente el tiempo de diseño y manufactura de micropiezas.

1.4. Sistemas CNC.

Hoy en día, las máquinas herramienta de control numérico asistido por computadora (CNC) son empleadas con el fin de llevar a cabo la transformación física de materia prima para generar una geometría específica de manera automatizada. Esto se debe, a que estas máquinas cuentan con computadoras internas que permiten controlar el proceso de maquinado, así como el cambio de herramientas y el orden en la ejecución de los programas [Ferré, 1999]. De tal forma, es necesario definir conceptualmente este tipo de sistemas para su mejor comprensión.

1.4.1 Definición de CNC.

Podemos definir a un CNC como un sistema de automatización programable que se basa en una serie de instrucciones codificadas que gobiernan todas las acciones de una máquina herramienta [Ferré, 1999], haciendo que ésta desarrolle una secuencia de operaciones y movimientos previamente establecidos. En la figura 12, se muestran los componentes básicos de un

CNC, los cuales son: *un programa*, que contiene toda la información de las acciones a ejecutar; la *unidad de control*, que interpreta las instrucciones y las convierte en señales para los actuadores de la máquina; y la *máquina*, que ejecuta las operaciones previstas [Ferré, 1999].

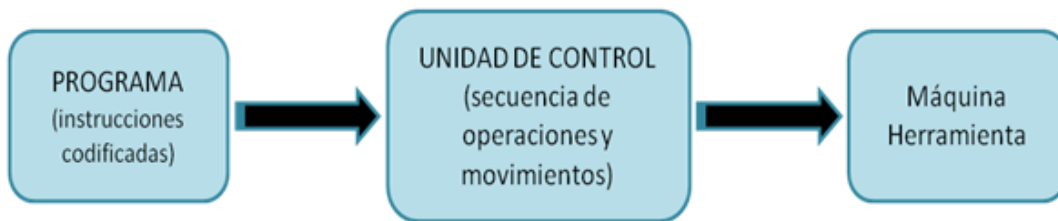


Figura 12. Componentes básicos de un CNC.

1.4.2 Clasificación del CNC.

La clasificación de los sistemas CNC varía de acuerdo a ciertas características de los mismos, como son: el tipo de sistema de referencia que manejan, el tipo de trayectorias, la capacidad para controlar un número determinado de ejes simultáneamente, etc. [Smid, 2003]. Esta clasificación está descrita en la tabla 1.

De acuerdo a:	Tipos
Sistema de referencia	Fijo Flotante
Trayectorias	Punto a punto Paraxial Continuo
Control simultáneo de ejes	2D 2D y ½ 3D
Accionamiento	Hidráulico Eléctrico Neumático
Técnica de control	Lazo Abierto (LA) Lazo Cerrado (LC)

Tabla 1. Clasificación de un CNC [Caballero y Ruiz, 2005].

1.4.3 Ventajas y desventajas del CNC.

La conveniencia de utilizar un CNC sobre un equipo convencional, estriba en ciertos puntos como lo son: el tipo de producción que se tendrá con dichos sistemas (lotes medianos), complejidad de las geometrías a manufacturar, el número de operaciones a procesar, por cambios frecuentes en el diseño, requerimientos de precisión y fiabilidad, altos costos por falla en el maquinado, etc. [Smid, 2003]. Es por eso, que en la tabla 2 se denotan las ventajas y desventajas al usar este tipo de sistemas.

Ventajas:
- Mejor planeación de operaciones.
- Mayor flexibilidad en maquinado.
- Reducción en tiempo de programación.
- Mejor control de procesos en maquinados.
- Disminución en costos de herramientas.
- Se incrementa seguridad en usuario.
- Reduce el tiempo de flujo de material.
- Reduce el manejo de la pieza de trabajo.
- Aumento en la productividad.
- Se incrementa la precisión.
Desventajas:
- Alto costo de inversión.
- Personal especializado en programación.
- Alto costo en mantenimiento.
- Requiere tiempo de adaptación.
- Se requiere un volumen alto de producción.

Tabla 2. Ventajas y desventajas de un CNC [Caballero y Ruiz, 2005].

1.4.4 Programación y códigos.

La manufactura de una determinada pieza sigue un proceso el cual es representado en el diagrama a bloques de la figura 13. Como primer paso es necesario elaborar un diseño de la geometría a manufacturar mediante el empleo de un determinado *software*. Enseguida, y de acuerdo con la naturaleza de la pieza, se selecciona la MH a emplear y, con base en esto, se definen las herramientas adecuadas para cada una de las tareas así como la secuencia de uso de las mismas.

Enseguida se establecen las secuencias de corte, para después obtener determinados parámetros como coordenadas, velocidades, avances y profundidades de corte. Una vez teniendo éstos, es posible generar un programa de cada una de las operaciones por medio de un código propio de manufactura, con el fin de llevar a cabo una simulación del proceso a desarrollar.

El objetivo de una simulación, es detectar posibles malfuncionamientos en un proceso. Por último, se procede a ejecutar el programa.

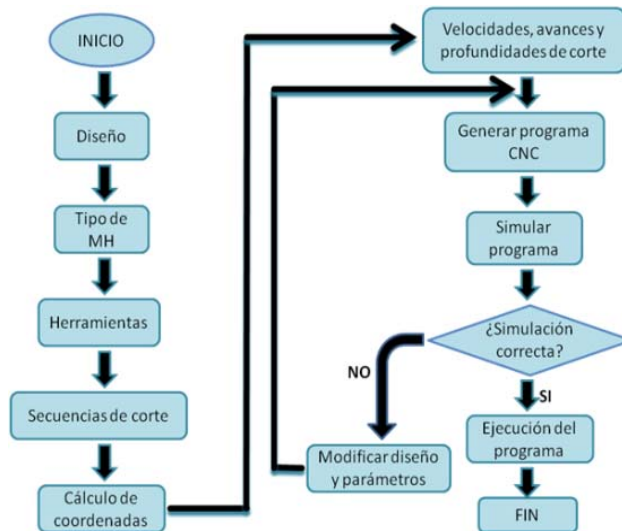


Figura 13. Proceso de manufactura en un CNC [Jiménez, 2002].

2. OBJETIVO Y ALCANCES.

2.1 Definición del problema.

Un diagrama detallado de un sistema CNC es mostrado en la figura 14. En éste podemos apreciar un medio de transferencia de información, el cual es conectado directamente al sistema. Éste, almacena en un bloque de memoria central el programa a ejecutar, para después ser leído y procesado por una rutina del programa de control, generando las señales requeridas para la comparación con una señal de corrección, y como resultado se generen las señales requeridas por los actuadores de la máquina en cuestión.

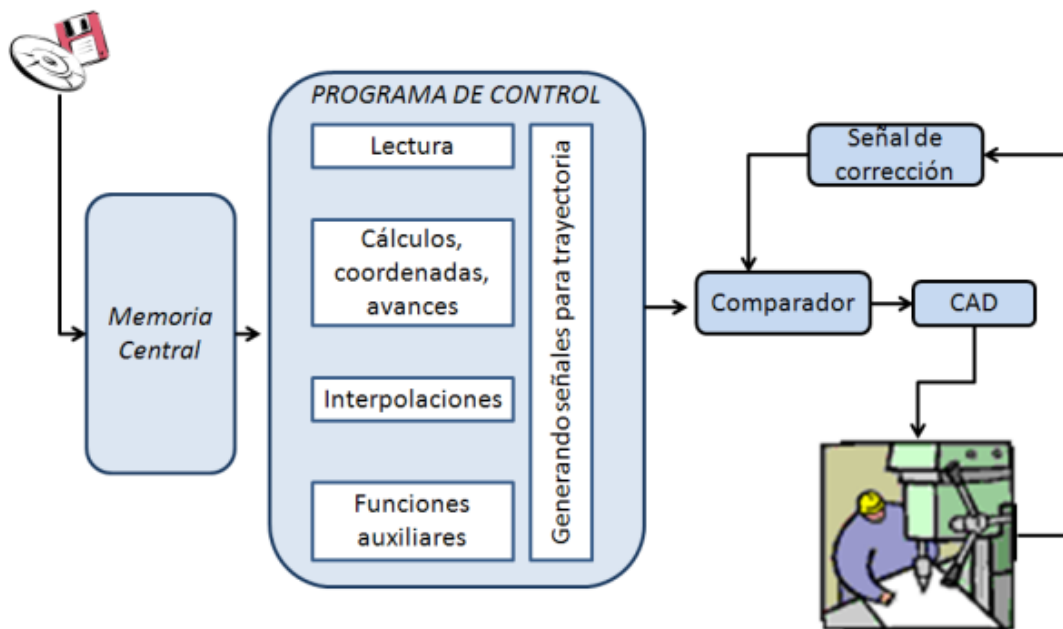


Figura 14. Diagrama general de un sistema CNC.

Analizando los beneficios proporcionados por este tipo de sistemas, el GMM se ha inclinado por implementar (gradualmente) características propias de sistemas CNC a sus sistemas de control, con el fin de simplificar y mejorar sus métodos de programación para la manufactura de micropiezas.

En consecuencia, en el trabajo *Desarrollo de una interfaz computacional para el control de MMH* [Yañez, 2007], aún cuando cumple con el propósito de interpretar código estándar G y M logrando así reducir considerablemente el tiempo de manufactura de piezas, sigue presentando algunos inconvenientes, como por ejemplo, las interpolaciones lineales y circulares se desarrollan punto a punto, lo cual es ineficiente y causa un incremento en el código de programa. Así mismo, la aplicación se ejecuta en ambiente *Windows*, lo que limita las frecuencias de accionamiento de los actuadores, y hace que el sistema de control sea propenso a fallas por interrupciones del sistema operativo.

Otra desventaja notoria, es que se sigue requiriendo de dos puertos paralelos en la PC para llevar a cabo el control de una MMH con cuatro grados de libertad [Caballero et al, 2007]. Además, el seguimiento que hace la interfaz al proceso de manufactura, grafica una vez terminada la trayectoria, es decir, la geometría a desarrollar se representa mediante curvas ideales.

Por lo tanto, y debido a estos inconvenientes, en este trabajo se plantea una propuesta para tratar de solucionar estos problemas en la generación de microequipo del GMM.

2.2 Objetivo.

Analizar y evaluar algoritmos de interpolación aplicados al desarrollo de un sistema de control para una MMH de primera generación basado en código estándar para CNC.

3. MARCO TEÓRICO.

En un controlador CNC, tal como se muestra en la figura 15, se pueden distinguir tres bloques fundamentales: un *programa de la pieza*, un *programa de servicio*, y un *programa de control*. El *programa de la pieza* contiene la descripción de la geometría de la pieza a manufacturar y las condiciones de corte, tales como la velocidad del husillo, avance de la herramienta, profundidad de corte, etc. El *programa de servicio* es usado para verificar, editar y corregir el programa de la pieza (en caso de ser necesario). Por último, el *programa de control* interpreta las instrucciones del programa de la pieza con el fin de producir las señales requeridas por los actuadores empleados [Koren, 1979].

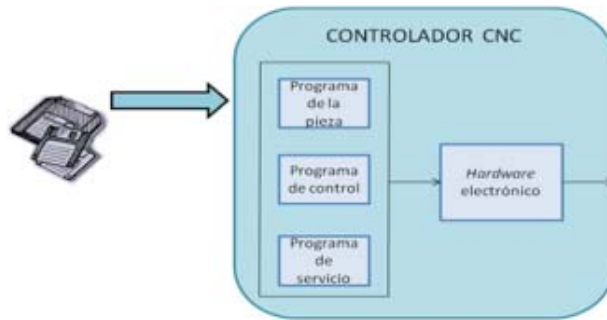


Figura 15. Bloques fundamentales de un controlador CNC.

3.1 Introducción a la programación en sistemas CNC.

Para realizar un programa en código estándar para CNC debemos tener en cuenta varios factores. Estos factores los podemos dividir en geométricos y tecnológicos. Los factores de geometría de la pieza contienen datos sobre sus dimensiones (plano de taller); además de tolerancias, acabados, superficies de referencia, origen de movimientos, etc. Mientras que los factores tecnológicos hacen referencia al material de la pieza a manufacturar, tipo de mecanizado, velocidad y profundidad de corte, revoluciones de la pieza o herramienta, lubricante, etc. [Escalona, 2002].

Una vez atendidos estos factores se puede dar inicio a la programación de la pieza a manufacturar. Un programa de código estándar para CNC es una lista secuencial de instrucciones de maquinado que son ejecutadas por una máquina CNC [Smid, 2003]. A las instrucciones se les conoce como código de CNC, las cuales deben contener toda la información requerida para lograr el maquinado de la pieza.

3.2 Programación en sistemas CNC.

Se pueden utilizar dos métodos para llevar a cabo la programación en sistemas CNC, la programación manual y la programación automática. En la programación manual, el programa se escribe únicamente por medio de razonamientos y cálculos que realiza un operario. Mientras que en la programación automática, los cálculos los realiza un equipo de cómputo que suministra el programa de la pieza en lenguaje máquina, es por eso que recibe también el nombre de programación asistida por computadora [Ferré, 1999].

3.2.1 Estructura de un programa para CNC.

En un programa para CNC cada movimiento o acción se realiza secuencialmente, por esta razón en cada bloque debe haber ciertas características; por ejemplo, cada movimiento o acción se realiza secuencialmente, por lo que cada bloque debe ser enumerado y usualmente contener un solo comando. Estos comandos pueden ser códigos G ó M. La

mayoría de los códigos G y M contienen variables (direcciones) definidas por el programador para cada función específica [Jiménez, 2002]. En la tabla 3, se describen cada una de las posibles variables que pueden acompañar, a un código G o a un código M.

La estructura de una línea de código para CNC está compuesta básicamente por cuatro elementos: una *Dirección* (N01) seguida de un *Comando* (G ó M) y después una *Palabra* compuesta de coordenadas (X10.0 Y5 Z0) para terminar con *Parámetros de manufactura* tales como velocidades, avances y herramientas.

Letra	Descripción
N	Número de secuencia
X	Comando del eje X
Y	Comando del eje Y
Z	Comando del eje Z
R	Radio desde el centro especificado
I	Coordenada X del centro del arco
J	Coordenada Y del centro del arco
F	Avance
S	Velocidad de husillo
T	Número de la herramienta

Tabla 3. Variables de la programación CNC [Jiménez, 2002].

3.2.2 Principales códigos de programación CNC.

Los códigos G que se refieren a movimientos elementales son: el G00 (movimiento rápido), G01 (avance lineal), G02 y G03 (avances en segmentos circulares). Mientras que los códigos M comúnmente empleados son: M02 (fin del programa), M03 y M04 (rotación del husillo), M05 (paro del husillo) y M06 (cambio de herramienta) [Ferré, 1999].

Existen más códigos G y M relacionados con otras tareas, por ejemplo, la selección del plano de trabajo, pausas, sistema de unidades, retorno a posición de origen, compensaciones, sistema coordinado, activación de refrigerante, etc.

A su vez, se tienen códigos de tareas especializadas que resultan de combinaciones con los códigos elementales, a éstos se les denomina ciclos enlatados y entre ellos se encuentran: el G71 (ciclo de desbaste exterior), G72 (ciclo de desbaste y acabado vertical), G74 (ciclo de barrenado), G81, G82 y G83 (ciclos de taladrado profundo y con pausa), G92 (ciclo para hacer cuerdas), G94 (ciclo para hacer careado), etc. [Smid, 2003] Todos éstos representan ciclos que simplifican la programación de una pieza, combinando una gran cantidad de operaciones y disminuyendo los pasos de programación simplificando las operaciones matemáticas y optimizando las condiciones de corte [Caballero y Ruiz, 2005].

Como se ha mencionado al principio de este apartado, el *programa de control* es el encargado de interpretar todos estos códigos para producir las señales

adecuadas que activarán los actuadores de la máquina en cuestión. Es por esto necesario investigar más acerca de las técnicas de control empleadas en sistemas CNC, así como el tipo de algoritmos implementados en estos sistemas.

3.3 Técnicas de control en sistemas CNC.

De todos los bloques de un sistema CNC, la rutina principal en el programa de control es el algoritmo de interpolación, el cual coordina el movimiento de los ejes de la máquina en cuestión, para generar el maquinado de la ruta requerida [Ferré, 1999]. Es importante mencionar que dicha ruta es una combinación de segmentos lineales y circulares, y de acuerdo con esto, el programa de control debe contener rutinas capaces de producir movimientos simultáneos en los ejes para desarrollar este tipo de trayectorias.

Básicamente, un sistema CNC tiene dos técnicas para transmitir las señales de referencia generadas por el programa de control, una es basada en una secuencia de pulsos (técnica por referencia de pulsos) y la otra es mediante palabras binarias (técnica por muestreo de datos) [Koren, 1979].

3.3.1 Técnica por referencia de pulsos.

En los sistemas que utilizan este tipo de técnica, la computadora produce una secuencia de pulsos, como señal de referencia, para el movimiento de cada eje. Cada pulso generado produce un movimiento equivalente a la resolución de la MMH. El número acumulado de pulsos representa la posición, mientras que la frecuencia de los mismos es proporcional a la velocidad en cada eje [17, 18] (figura 16).

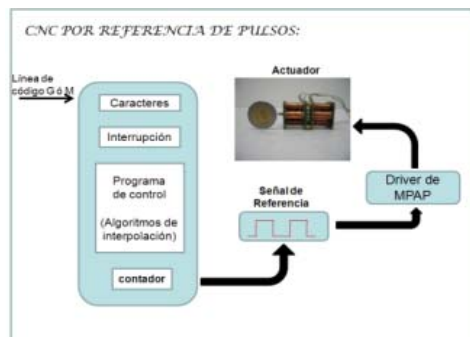


Figura 16. Sistema por referencia de pulsos.

3.3.2 Técnica por muestreo de datos.

En este tipo de técnica (figura 17), el lazo de control es cerrado mediante *software* por medio de la misma computadora. El programa de control compara la palabra binaria generada con la señal realimentada produciendo una señal de error. Este error es enviado a un convertidor digital-analógico, el cual a su vez proporciona un nivel de voltaje proporcional requerido para la velocidad en cada uno de los ejes [Koren y Masory, 1982].

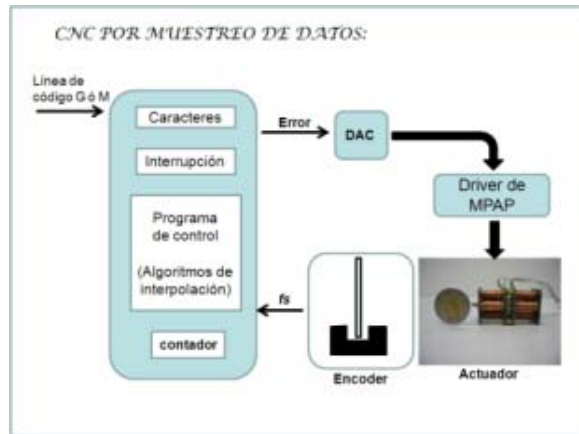


Figura 17. Sistema por referencia de palabra.

Ambas técnicas, requieren distintas rutinas de interpolación en el programa de control para generar sus correspondientes señales de referencia. En todos los sistemas de control desarrollados en el GMM hasta el momento se ha optado por la *técnica de referencia de pulsos*, debido a la simplicidad para ser programados, y su fácil implementación para el tipo de actuadores empleados (motores a pasos). Sin embargo, hay que tener en cuenta que en este tipo de técnica hay restricción en cuanto a la máxima velocidad alcanzada durante la interpolación [Koren, 1981].

Los métodos de interpolación empleados en la técnica por referencia de pulsos están basados en iteraciones controladas por la ejecución de una interrupción, es decir, cada vez que se ejecuta ésta se produce una iteración del método de interpolación, generando un pulso como salida hacia el(los) actuador(es) correspondiente(s) [Koren, 2003]. La máxima velocidad será proporcional a la velocidad a la cual se puede ejecutar el algoritmo de interpolación. Los algoritmos de interpolación por referencia de pulsos comúnmente empleados en sistemas CNC son básicamente tres:

1. Interpolación por analizador diferencial digital, de sus siglas en inglés (*DDA-Digital Differential Analyzer*).
2. Interpolación por escalones ó incremental.
3. Interpolación por búsqueda directa, de sus siglas en inglés (*DSM-Direct Search Method*).

3.4 Interpolación por DDA.

3.4.1 Interpolación lineal.

Este método de interpolación, está basado en adiciones, subtracciones y comparaciones, y consiste en la obtención de las principales componentes de una trayectoria rectilínea (X y Y). Suponiendo dos ejes coordenados, los valores absolutos de estas componentes, denotadas con a y b en el diagrama de flujo mostrado en la figura 18, se almacenan en los registros correspondientes de acuerdo a la proyección del eje coordenado, P_x y P_y [4].

Durante cada iteración, los valores de las proyecciones almacenados en P_x y P_y serán sumados al contenido de otro registro, denotado Q . El desborde de cada uno de estos registros (uno por cada eje coordenado, Q_x y Q_y) consiste en sobrepasar la magnitud de la hipotenusa L (condición de desborde). Además de esto, se verificarán los valores de los contadores de pasos, $CntX$ y $CntY$, cuyo valor de desborde está determinado por el cociente de las proyecciones entre la resolución del sistema.

Cuando el desborde de los registros Q_x y Q_y ocurra, un pulso es generado en la dirección correspondiente donde se originó el desborde. Una vez generado el pulso, la condición de desborde L es restada al registro suma Q donde se generó el desborde. De esta forma se actualizará el registro contador de pasos correspondiente, $CntX$ ó $CntY$.

La frecuencia a la cual se ejecuta el algoritmo está dado por:

$$f = \frac{d * F * 0.1}{60} \dots\dots\dots (1)$$

donde el significado de las variables es el siguiente:

d : es el número de pasos necesarios para recorrer un milímetro.

F : es el avance de la herramienta durante el maquinado.

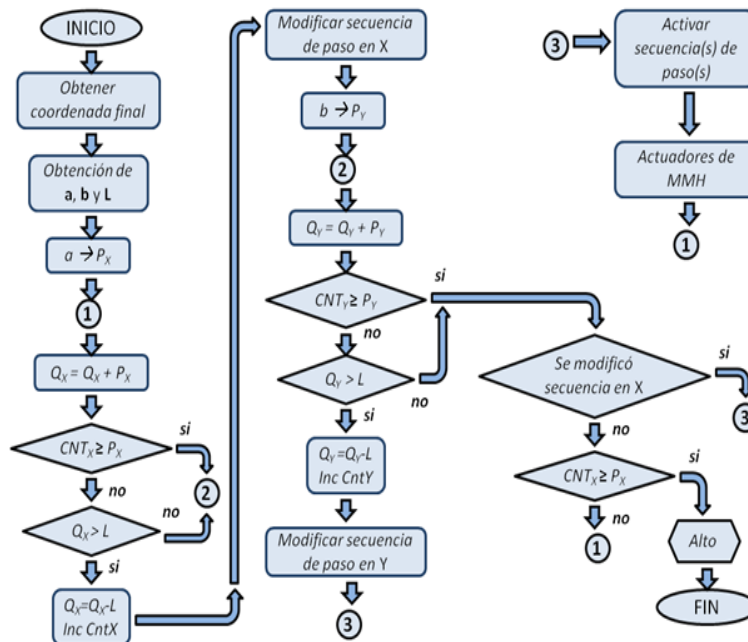


Figura 18. Algoritmo DDA para segmentos rectilíneos.

La forma empleado para obtener d será:

$$d = \frac{1mm}{Resol} \dots\dots\dots (2)$$

mientras, que la condición de desborde está dada por:

$$L = \sqrt{a^2 + b^2} \dots\dots\dots (3)$$

Este algoritmo también se puede aplicar a segmentos circulares, la diferencia existente respecto al caso de segmentos rectilíneos radica en que los valores de los registros P_x y P_y varían constantemente.

3.4.2 Interpolación circular.

Para este caso, el método requiere de dos operaciones más en cada iteración (una adición y una substracción). Sin embargo, la metodología es muy parecida al caso lineal. La principal diferencia radica, en que al ser constante la condición de desborde (dada por el radio) las proyecciones de los registros P_x y P_y cambian al ir avanzando sobre la circunferencia. En la figura 19 se muestra un diagrama a bloques de este procedimiento.

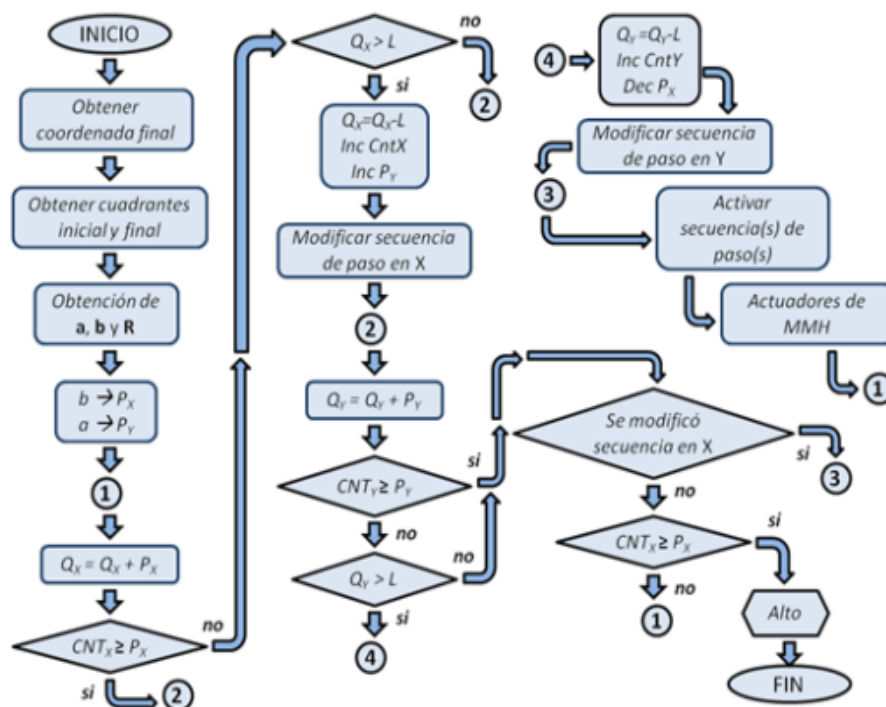


Figura 19. Algoritmo DDA para segmentos circulares (1er cuadrante).

Otras diferencias no menos importantes son: que las proyecciones X y Y (suponiendo dos ejes coordenados) formadas por los extremos del segmento circular con el centro del mismo se almacenan en registros contrarios (X en P_y y Y en P_x). Así mismo, dependiendo del cuadrante en el que nos encontremos, al generarse un pulso en la dirección del eje X (mediante el mismo procedimiento que en el caso lineal) disminuye en una unidad el registro P_y , y si fuese un pulso para el eje Y aumenta en una unidad el registro P_x .

3.5 Interpolación incremental (ó por escalones).

Este método, también denominado *aproximación por escalones*, realiza una aproximación a una trayectoria circular mediante incrementos (pasos) en un solo eje a la vez. En cada iteración se avanza en un solo sentido hasta alcanzar el punto final. Esto se logra mediante una variable de comparación D , dada por:

$$D_f = X_f^2 + Y_f^2 - R^2 \quad \dots\dots\dots (4)$$

Como se aprecia, en la expresión (4) se requiere de un multiplicador que podría aumentar el tamaño de la palabra y la velocidad de procesamiento del algoritmo. En consecuencia, existe una mejora a este método que consiste en modificar y añadir un índice más a la expresión anterior, resultando:

$$D_{i,j} = X_i^2 + Y_j^2 - R^2 \quad \dots\dots\dots (5)$$

Donde i y j representan los pasos completados en cada una de las direcciones, X y Y . Ahora bien, para el caso del primer cuadrante cuando se ejecuta un paso en una determinada dirección (por ejemplo en X) la expresión (4) queda de la siguiente manera:

$$D_{i+1,j} = (X_i + 1)^2 + Y_j^2 - R^2 = D_{i,j} + 2X_i + 1 \quad \dots\dots (6)$$

Estableciéndose un incremento representado por:

$$\Delta X_i = 2X_i + 1 \quad \dots\dots\dots (7)$$

Y reemplazando (7) en la expresión (6), resulta:

$$D_{i+1,j} = D_{i,j} + \Delta X_i \quad \dots\dots\dots (8)$$

Así, para el siguiente paso en X se tiene que:

$$D_{i+2,j} = D_{i+1,j} + \Delta X_{i+1} \quad \dots\dots\dots (9)$$

donde

$$\Delta X_{i+1} = 2X_{i+1} + 1 = 2(X_i + 1) + 1 = \Delta X_i + 2 \quad \dots\dots (10)$$

y similarmente para el eje Y , tenemos:

$$D_{i,j+1} = D_{i,j} + \Delta Y_j \quad \dots\dots\dots (11)$$

y para:

$$\Delta Y_{j+1} = \Delta Y_j + 2 \quad \dots\dots\dots (12)$$

donde:

$$\Delta Y_j = -2Y_j + 1 \quad \dots\dots\dots (13)$$

Con esto, se mejora la velocidad de procesamiento al tener expresiones más sencillas para el valor de D (expresiones 8 y 11), en las cuales no intervienen potencias.

En la figura 20, se puede apreciar que la dirección en la que se dé el paso dependerá del signo de la variable D y del cuadrante de la circunferencia en el que se encuentre la herramienta. Así pues, si se trata del primer cuadrante y $D < 0$ (el punto se encuentra dentro del círculo) el incremento será en la dirección $X+$ (positiva), y si $D > 0$ (el punto se encuentra fuera del círculo) el incremento será en la dirección $Y-$ (negativa). El caso $D = 0$ (el punto está sobre la circunferencia) puede ser incluido en cualquiera de los dos casos anteriores, y todo esto hasta alcanzar el punto final (X_f, Y_f) .

Por último, cabe aclarar que el valor de D que se obtiene en cada iteración es evaluado con ayuda de su valor previo y los respectivos incrementos, ΔX y ΔY , los cuales son calculados mediante las ecuaciones (7) y (13).

La simplicidad de cada una de las ecuaciones anteriores (en este método) provoca que la velocidad de ejecución del algoritmo aumente. Sin embargo, la incapacidad de este método para originar movimientos simultáneos, provoca que más iteraciones (de las realizadas por el método DDA) sean requeridas para describir un círculo.

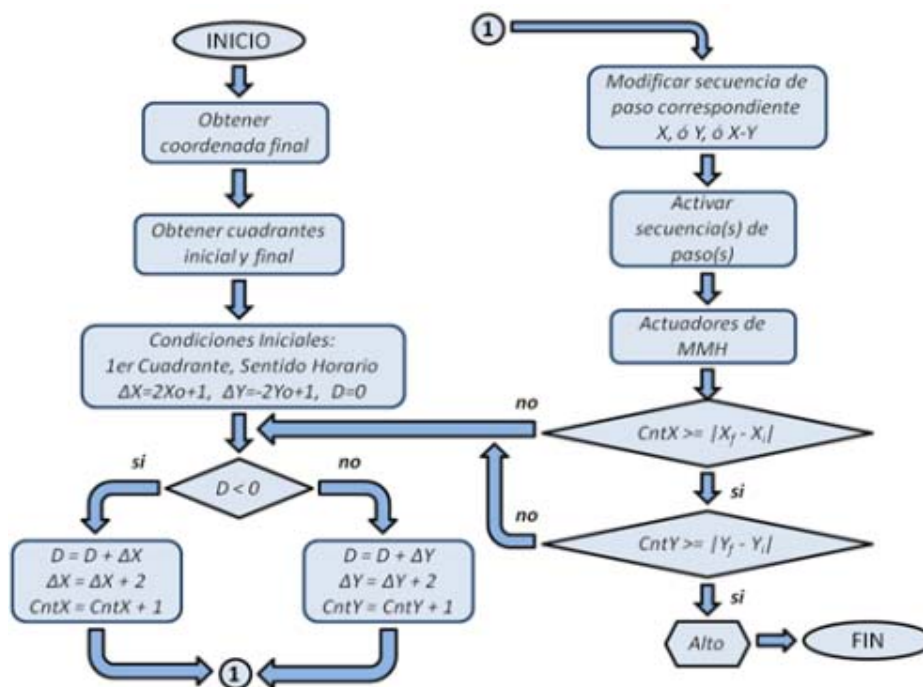


Figura 20. Algoritmo por interpolación incremental.

3.6 Interpolación por búsqueda directa.

Este método aplica la misma técnica en los cálculos que el método de *interpolación por escalones*, con la diferencia que este método de interpolación

si permite el movimiento simultáneo en ambos ejes, lo cual trae como resultado una mejor interpolación del segmento circular [Koren, 2003].

Esta mejora es alcanzada mediante la evaluación de la ecuación (5) en tres posibles puntos. Estos resultados de la variable D son comparados entre sí, con la finalidad de obtener la dirección del error mínimo. La elección de estos puntos depende del sentido de giro y del cuadrante de la circunferencia al que esté referida la trayectoria.

Por ejemplo, para un caso de primer cuadrante y sentido horario, se llevan a cabo tres pruebas para determinar la ruta del error mínimo. En la primera, una unidad (resolución) es sumada al subíndice i , teniendo que evaluar las ecuaciones (7) y (8) para obtener la correspondiente $D_{(i+1,j)}$. En la segunda prueba, se hace lo propio con el subíndice j , obteniendo de las ecuaciones (11) y (13) el valor $D_{(i,j+1)}$, mientras que en la última prueba se aplican los dos procedimientos mediante la siguiente expresión:

$$D_{(i+1,j+1)} = D_{i,j} + \Delta X_i + \Delta Y_j \quad \dots\dots\dots (14)$$

Con la ecuación (14) es posible obtener el tercer valor $D_{(i+1,j+1)}$ y comenzar la comparación para obtener la ruta del error mínimo. El procedimiento, se ilustra en el diagrama de la figura 21. Una vez hecha la comparación de estos tres valores se procede a generar el pulso correspondiente en la dirección resultante, además de actualizar los correspondientes valores de ΔX y ΔY , tal como se hace en el método por escalones.

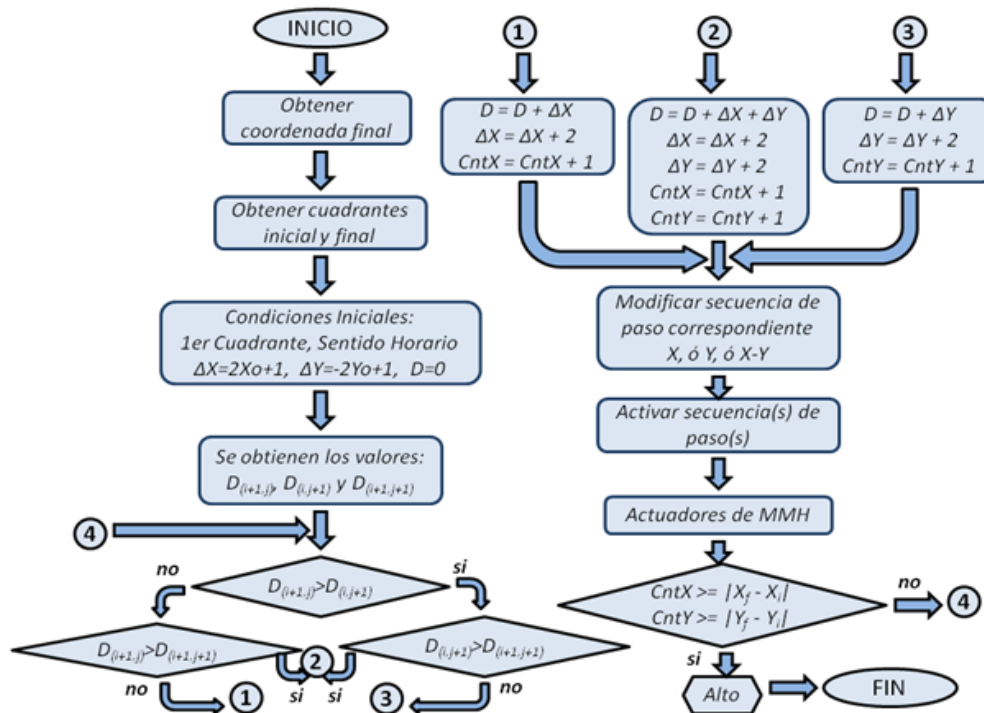


Figura 21. Algoritmo por el método de búsqueda directa.

4. DESARROLLO DEL SISTEMA DE CONTROL.

En el GMM se busca que las especificaciones en el desarrollo de sus sistemas de control sean compatibles a los sistemas CNC comerciales, con el objetivo de mejorar el nivel de automatización de sus prototipos micromecánicos. Antes de adentrarnos en las características que deberá tener este sistema de control, describiremos primeramente el prototipo de MMH a controlar, haciendo énfasis en su estado actual y principales componentes.

4.1 Caso de estudio.

Actualmente, se cuenta con una nueva versión del primer prototipo de MMH mostrado en la figura 5. En esta nueva versión se aplicaron sistemas CAD y CAM con la finalidad de mejorar la reproducibilidad del prototipo, además de facilitar la manufactura y ensamble del centro de maquinado mostrado en la figura 22.

Esta nueva versión de MMH tiene dimensiones de 120x160x85mm, tres ejes traslacionales y uno rotacional. La transmisión de engranes tienen una relación de 1:128, mientras que sus actuadores son motores a pasos con 28 pasos por revolución (figura 23). La resolución de este prototipo es de 140[nm] por paso del motor [German, 2007].



Figura 22. Nueva versión del primer prototipo de MMH desarrollado en México.

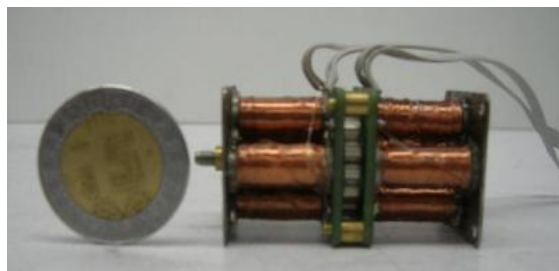


Figura 23. Motor a pasos bipolar de magneto permanente.

4.2 Características esperadas.

- ☐ El sistema de control propuesto debe de reconocer las principales instrucciones G y M para procesos de fresado descritas posteriormente en el apartado 4.3.2.
- ☐ Los algoritmos de control deben ser programados en un dispositivo externo a la PC con el fin de reducir la dependencia de la misma.
- ☐ El algoritmo de interpolación programado debe ser aquel que ofrezca los mejores parámetros en su ejecución (aproximación, iteraciones, complejidad, etc.).
- ☐ La interpretación gráfica (en la interfaz) del proceso de manufactura debe monitorearse en cada paso del motor.
- ☐ El sistema de potencia debe ser capaz de proporcionar las características eléctricas necesarias para el accionamiento de los actuadores.
- ☐ La interfaz de usuario programada debe monitorear, en sus diferentes modos de operación, parámetros básicos de manufactura (avance, revoluciones por minuto, posición y proceso de maquinado).

4.3 Propuesta de control.

Como ejemplo de un sistema de control para un MH convencional, el GMM cuenta con una fresadora CNC de la empresa DynaMechtronics mostrada en la figura 24. Esta MH lleva a cabo un control por procesamiento distribuido mediante Microcontroladores de T.I (*Texas Instruments*). Estos dispositivos son generalmente empleados en tareas de automatización debido a su bajo costo y amplia flexibilidad para diseño e implementación. Entre otras características, esta fresadora permite ser operada por desplazamientos continuos, por rutinas de pasos establecidas, por posicionamiento a través de una perilla, por una línea de código G ó M, y por un programa para CNC.



Figura 24. Fresadora CNC del GMM [German, 2007].

En el diagrama de la figura 25 se muestra una propuesta de control para una MMH de primera generación que, mediante algoritmos de control programados en un dispositivo externo a la PC, disminuirá la dependencia que se ha tenido con ésta en los últimos sistemas de control desarrollados. La tarjeta controladora contará con ciertos procesos programados, tales como atención a interrupción, recepción de cadenas, cálculo de parámetros, interpolaciones, etc.

Todo esto con el fin de que el *hardware* electrónico (potencia) proporcione los niveles de corriente adecuados a las señales de referencia generadas (control) para el accionamiento de los actuadores.

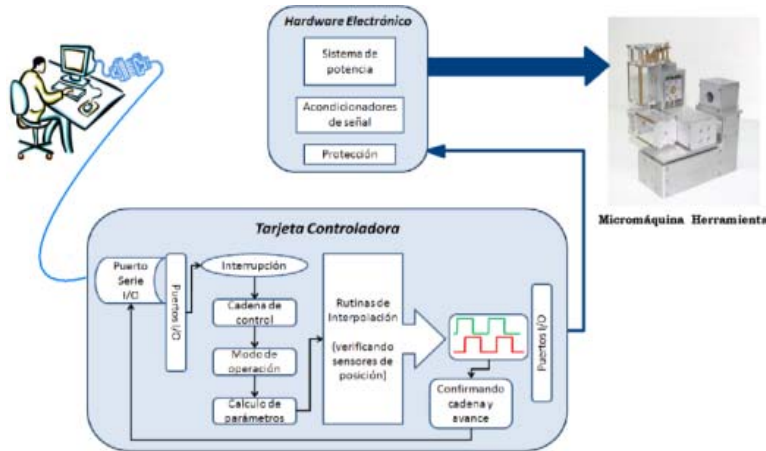


Figura 25. Sistema de control propuesto.

Así pues, el sistema será capaz de satisfacer especificaciones tales como: gran capacidad de memoria programa, entradas y salidas para controlar más de cuatro grados de libertad, procesamiento a 20MHz (ó más), set instrucciones elaborado, cierto grado de conocimiento para su programación, etc.

El sistema propuesto se implementó en una tarjeta de *Microchip* con un microcontrolador PIC18F8720. Las características de esta tarjeta se muestran en la tabla 4. Mientras que los recursos del microcontrolador PIC se resumen en la tabla 5.

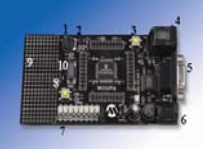
Tarjeta electrónica empleada	
	
Potenciómetro conectado al CA/D de 10bits.	(1)
Sensor de temperatura: Microchip TC74.	(2)
Pulsador de <i>reset</i> .	(3)
Conector para MPLAB ICD2.	(4)
Conector RS232.	(5)
Terminal de alimentación.	(6)
Hilera de 8 <i>leds</i> , con <i>jumper</i> para desconectar.	(7)
Pulsador para modalidad de programación.	(8)
Área de propósito general.	(9)
Cristal.	(10)

Tabla 4. Características de la tarjeta de control.

Microcontrolador.	PIC18F8720
Tipo de encapsulado.	80pines-TQFP
Frecuencia de operación [MHz].	20
Memoria de programa (bytes).	128K
Memoria de datos (bytes).	3840
Memoria EEPROM (bytes).	1024
Interfaz para memoria externa.	Si

Fuentes de interrupción.	18
Puertos de entrada/salida.	A, B, C, D, E, F, G, y J
Pines de entrada/salida.	68
Temporizadores.	5
Módulos CCP.	5
Módulos de comunicación serial.	MSSP, USART (2)
Módulos de comunicación paralela.	PSP
Módulos CAD.	10bits / 16 canales
Set de instrucciones.	77

Tabla 5. Recursos del microcontrolador PIC18F8720.

4.3.1 Características del sistema de control.

En el desarrollo del este sistema de control se abordaron tres aspectos: La PC debe estar limitada a desarrollar tareas de monitoreo en los diferentes modos de control del sistema. El programa de control debe ser capaz de calcular y ejecutar interpolaciones de tipo lineal y circular para obtener movimientos suaves y precisos. Por último, el sistema reconocerá código estándar G y M para tareas de fresado.

4.3.1a Modos de operación.

Se propone que el sistema de control a desarrollar cuente con cuatro modos de operación para accionar el movimiento en la MMH:

- ☐ por cursor de mando (*Joystick*)
- ☐ por una secuencia de pasos
- ☐ por ejecución de una línea de código G y M
- ☐ por ejecución de un programa de código G y M

En el modo por *Joystick* la interfaz computacional a desarrollar permitirá que, mediante un cursor de mando programado, el usuario pueda desplazar los carros de la MMH en todo su espacio de trabajo, logrando ubicar la herramienta dentro del volumen de trabajo además de variar la velocidad de desplazamiento como se considere adecuado.

En el modo *Por Pasos*, los avances en la MMH serán por rutinas cuyos valores son los siguientes: 1 paso, que representa el mínimo desplazamiento del sistema, 14 y 28 pasos, que representan media vuelta y una vuelta, respectivamente, 10, 50, 100, y 1000 pasos, rutinas que permiten completar desplazamientos poco críticos en el posicionamiento de la herramienta.

Los dos modos de operación restantes que ejecutan una línea de código G ó M y un programa son muy parecidos aunque con propósitos diferentes. Mientras que el modo de ejecución de una línea de código es empleado en cualquier instante de un proceso de manufactura, o bien, para establecer el cero de programa; en el modo de ejecución de programa se ejecuta consecutivamente cada una de las líneas del programa CNC dando por hecho que todas las condiciones para el proceso de maquinado están bien definidas.

4.3.1b Parámetros monitoreados.

Durante la ejecución de cada uno de los modos de operación del sistema de control es necesario que el usuario tenga en cuenta cómo varían ciertos parámetros que influyen en el proceso de maquinado. Los parámetros que este sistema de control monitorea en todas las modalidades de funcionamiento, son:

- ☐ Posición.
- ☐ Velocidad (Hz y RPM's).
- ☐ Avance (mm/min).
- ☐ Cadena de control enviada.
- ☐ Interpretación gráfica.

La posición está representada por tres tipos de coordenadas, las que reflejan la posición respecto al cero de máquina (X_m , Y_m , y Z_m), otras que dan la posición respecto al cero de pieza (X_p , Y_p , y Z_p), y las últimas que nos dan las coordenadas del cero de pieza (X_{cp} , Y_{cp} , y Z_{cp}). Mientras tanto, la velocidad y el avance serán parámetros referidos a los actuadores y a la herramienta empleada en la MMH, y estarán restringidos de acuerdo a sus límites de operación.

En cuanto al envío de información, se confirmarán las cadenas de control enviadas entre ambas etapas, interfaz computacional y tarjeta de control, dando la posibilidad al usuario en todo momento de localizar fallas de sintaxis. Por último, la interpretación gráfica consiste en generar un gráfico representativo de lo manufacturado, y a diferencia del sistema de control anterior denominado *Desarrollo de una interfaz computacional para el control de MMH*, el avance de cada actuador se monitorea en todo instante lo que permite tener una representación gráfica paso a paso del proceso de manufactura.

4.3.2 Códigos G y M soportados.

Los sistemas de CNC se gobiernan por un conjunto de aproximadamente 70 instrucciones de código *G* y 25 instrucciones de código *M* [Smid, 2003]. La cantidad de instrucciones *G* y *M* pueden llegar a variar dependiendo del fabricante del sistema CNC. Para el caso del sistema propuesto se contempla que contenga las instrucciones *G* y *M* estándares para llevar a cabo tareas de fresado en dos y medio ejes, es decir, el movimiento de un punto a otro se realizará moviendo tres ejes pero sólo dos de manera simultánea. Los códigos estándares *G* y *M* empleados, más otros códigos elementales para tareas de fresado son:

1. G00 Posicionamiento rápido.
2. G01 Interpolación lineal.
3. G02 Interpolación circular en sentido horario.
4. G03 Interpolación circular en sentido anti-horario.
5. G04 Parada temporizada.
6. G28 Retorno automático a posición cero de máquina.

7. G54 Localidad para almacenar y referenciar el cero de pieza.
8. G90 Sistema coordenado absoluto.

Mientras que las funciones *M* estándares empleadas son:

1. M30 Fin de programa.
2. M03 Activación del husillo en sentido horario.
3. M04 Activación del husillo en sentido anti-horario.
4. M05 Paro del husillo.
5. M06 Cambio de herramienta.

4.3.2a Descripción y sintaxis de los códigos.

La descripción y sintaxis de cada una de las instrucciones del apartado anterior se muestran en la sección de *A1 de los Anexos* de este trabajo. En la mayoría de los códigos *G* listados en el apartado anterior se requiere el traslado de la herramienta de un punto a otro, esto hace necesario la ejecución de algún método de interpolación. En el marco teórico se estudiaron tres métodos de interpolación empleados en sistemas CNC. En el siguiente apartado se muestran simulaciones como resultado de haber programado los tres métodos de interpolación en un *software* computacional.

4.4 Simulación de los métodos de interpolación.

Con el fin de analizar los métodos de interpolación hasta ahora expuestos, se programaron y simularon los tres métodos en un programa computacional. El *software* empleado para esta tarea fue *Visual Basic*. En las figuras de la 26 a la 29 se puede apreciar el resultado de estas simulaciones.

La aplicación programada mostrada en la figura 26 es capaz de simular trayectorias circulares (de 50 μ m de radio máximo) empleando cualquiera de los tres métodos de interpolación hasta ahora descritos, a su vez estas simulaciones son comparadas con una trayectoria ideal. Además, esta aplicación permite dibujar las tres trayectorias empalmadas con el fin de comparar la ejecución de los tres algoritmos.

Otro parámetro que se maneja en la aplicación empleada para generar la trayectoria mostrada en la figura 26, es el número de iteraciones que lleva a cabo cada método de interpolación. Este parámetro, se emplea para determinar cuál de los métodos propuestos requiere el menor tiempo para describir la misma trayectoria.

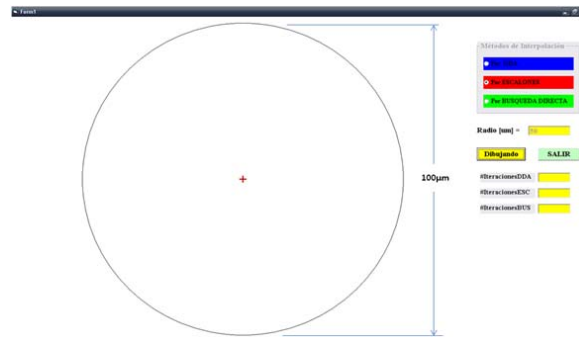


Figura 26. Pantalla del programa que simula los métodos de interpolación.

En la figura 27 se puede apreciar la simulación del método de *interpolación por escalones* para una circunferencia de 30µm de radio. En esta representación gráfica se observa la ruta generada por este algoritmo (en rojo), y lo burdo de su aproximación en las zonas 1, 2, 3 y 4 de la circunferencia, es decir, en las fronteras de cada uno de los cuadrantes.

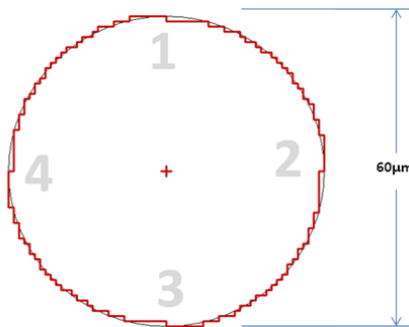


Figura 27. Simulación del método por escalones.

Por otra parte, en la figura 28 se puede observar la simulación (en azul) del método de *interpolación por DDA*, para la misma circunferencia que el método por *escalones*. A diferencia del método por *escalones*, en esta representación gráfica se aprecia una mejora en la descripción de la curva en las zonas 1, 2, 3 y 4 de la circunferencia. También, se observa en la simulación que este método tiene una mejor aproximación en el resto de la circunferencia ideal debido a la implementación de pasos simultáneos.

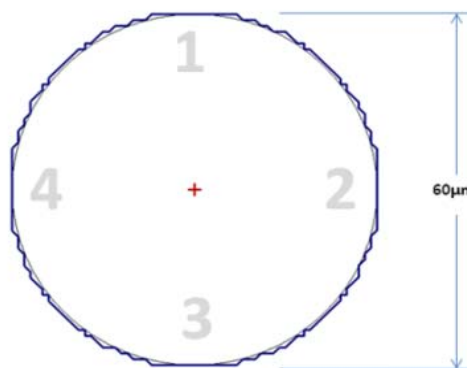


Figura 28. Simulación del método por DDA.

En la tercera simulación mostrada en la figura 29 se observa la trayectoria generada (en verde) por el método de *interpolación por búsqueda directa*, para una circunferencia de $30\mu\text{m}$ de radio. En esta representación, se aprecia como el seguimiento a la circunferencia ideal es más apegado que en los dos métodos anteriores. Los resultados de estas tres simulaciones para una circunferencia de $30\mu\text{m}$ de radio son mostrados en la tabla 7.

Por último, en la figura 30 se observa la simulación de las tres trayectorias generadas por los tres métodos de interpolación (empalmadas). En esta simulación se aprecia como el método por *búsqueda directa* (en verde) optimiza la trayectoria calculando el error generado en las tres posibles direcciones dependiendo del cuadrante. Mientras que en la figura 31 se muestran tres acercamientos correspondientes a las zonas 1, 5 y 2 de la figura 30.

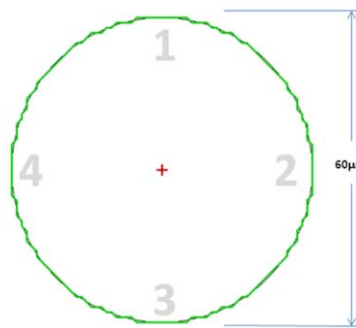


Figura 29. Simulación del método por Búsqueda Directa.

Otro parámetro importante a considerar en estos algoritmos es el tiempo que tardan en ejecutar una iteración con el fin de tomar la decisión respecto a la dirección en la cual hay que avanzar.

Como se aprecia de la tabla 6, el método por escalones al no implementar pasos simultáneos requiere de menos comparaciones para determinar una ruta de avance, esto conlleva a tener el menor tiempo consumido por iteración resultando en $3.2\mu\text{s}$. Por otro lado, el método por búsqueda directa al ser prácticamente una mejora del método por escalones, ya que implementa pasos simultáneos, tiene un consumo de tiempo de $8.6\mu\text{s}$ el cual es mayor al registrado por los demás. Finalmente, y debido al número de comparaciones realizadas, el método por *DDA* le lleva $6.8\mu\text{s}$ el ejecutar una iteración para decidir la dirección de avance.

Algoritmo	Número de Iteraciones (CIRCUNFERENCIA)	Error radial teórico máximo [nm]	Tiempo por iteración [µs]
Por escalones	240	140	3.2
Por DDA	188	140	6.8
Por búsqueda directa	168	70	8.6

Tabla 6. Resultados obtenidos de las simulaciones a los algoritmos propuestos.

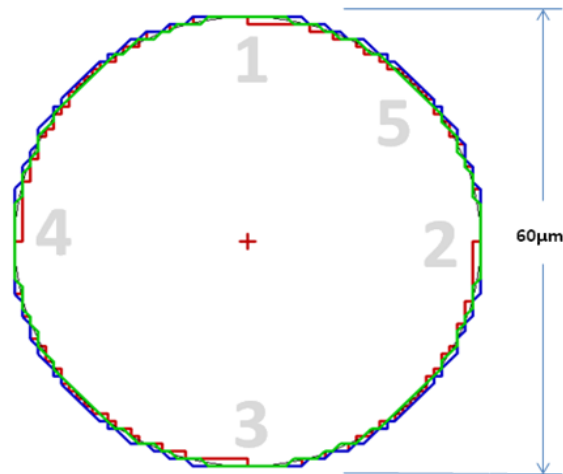


Figura 30. Simulación de los tres métodos de interpolación.

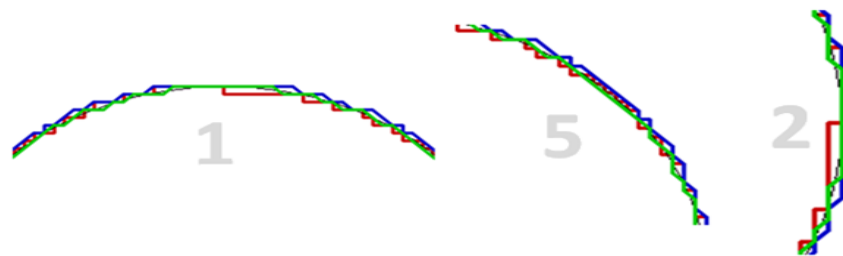


Figura 31. Acercamientos a las zonas 1, 5 y 2 de la figura 32.

4.5 Evaluación de los métodos de interpolación.

Hasta el momento se han descrito tres métodos de interpolación para sistemas CNC, y en este apartado se evalúan con el fin de sentar una base para la selección apropiada del método de interpolación a emplear. El proceso de evaluación utilizado, denominado *Matriz de decisión*, pone especial énfasis en lo que se busca (objetivo). La aplicación de este proceso se describe en los siguientes pasos:

- I. Establecer los parámetros de importancia (objetivos, rangos ó criterios).
- II. Ordenar los parámetros de manera jerárquica.
- III. Asignar valores relativos a cada parámetro.
- IV. Establecer el rendimiento para cada solución propuesta (calificar la propuesta).
- V. Calcular y comparar en función de los resultados [Ruiz, 2005], [Atila y Jesse, 1993].

Desarrollando los pasos I, II y III, se obtiene la tabla 7 donde se enuncian y ordenan jerárquicamente los parámetros de importancia, así como los valores relativos asignados a cada parámetro. El nivel de importancia de cada valor relativo está asignado de la siguiente manera: el valor 1 significa *poco*

importante, el valor 2 significa *importante*, mientras que el valor 3 significa *muy importante*. La suma de los valores relativos deberá corresponder a 10.

Parámetros de importancia	Valor relativo asignado
Mejor Simulación.	3
Error radial teórico máximo.	3
Número de iteraciones (CIRCUNFERENCIA).	3
Tiempo por iteración.	1

Tabla 7. Parámetros de importancia y valores relativos asignados.

4.6 Selección del método de interpolación.

Para establecer el rendimiento de cada solución propuesta (calificación) se emplea una escala del uno al diez. Enseguida se multiplica la calificación otorgada por el valor relativo asignado (factor de importancia) a cada parámetro, esto conlleva a obtener un valor máximo de hasta 100 puntos por cada solución propuesta. La suma más alta de estos *productos* determina la opción a escoger. Los resultados del proceso de evaluación a estos algoritmos son mostrados en la tabla 8.

El rendimiento de cada solución (calificación), de acuerdo a cada parámetro, se realizó de la siguiente manera: para el parámetro *Mejor Simulación*, se consideraron las simulaciones mostradas en las figuras de la 27 a la 31, y se estableció un calificación de 10 a la circunferencia ideal. Además, se tomó en cuenta el grado de aproximación en las zonas 1, 2, 3 y 4 de las figuras mencionadas anteriormente. En cuanto al parámetro *Error radial teórico máximo*, se determinó que la calificación de 10 correspondiera a un error cero, y que la calificación fuese disminuyendo de acuerdo a las unidades de longitud básica del sistema (resolución) que fuese incrementando el error a partir de cero.

Método de Interpolación	Mejor Simulación n (3)		Error radial teórico max. (3)		Número de Iteraciones (CIRCUNFERENCIA) (3)		Tiempo por iteración [μs] (1)		TOTAL
DDA	8	24	9	27	8.8	26.4	7	7	84.4
Escalera	6	18	9	27	5.7	17.1	10	10	72.1
Búsqueda Directa	9	27	9.5	28.5	10	30	5	5	90.5

Tabla 8. Evaluación de los algoritmos de interpolación.

Mientras tanto, en el parámetro *Número de Iteraciones* se seleccionó como referencia el que realizó menor número de iteraciones (para una calificación de 10), y dicho número de iteraciones se dividió entre 100. Por último, en el parámetro de *Tiempo por iteración* se decidió seleccionar como referencia una calificación de 5 para el que mayor tiempo requirió para una iteración, y después dividir esa magnitud entre 10 para ir incrementando la calificación en las demás lecturas.

Concluyendo con base en los resultados de la tabla 8, el método de interpolación con mayor puntuación fue el de *búsqueda directa* logrando una

suma de 90.5 puntos. Su principal ventaja fue la de generar una trayectoria simulada con una mejor aproximación a la curva ideal que los otros dos métodos. Aunque el tiempo de cada iteración es aproximadamente 2.7 veces el tiempo ejecutado por el método de escalera, compensa este defecto con el número de iteraciones realizadas para describir una circunferencia.

Una vez seleccionado el algoritmo de interpolación a emplear, se procede a plantear el proceso de control que permita reconocer los códigos propuestos.

4.7 Programa de control.

En la figura 32 se muestra un diagrama de flujo que hace mención a las principales rutinas del programa de control. En este diagrama se aprecia que después de las tres primeras rutinas el programa se ramifica hacia sub-rutinas de acuerdo a la cadena de control recibida.

En el primer bloque, *INICIO*, se declaran e inicializan todas las variables del programa, además de configurar la interrupción por recepción serial, la transmisión de datos, los puertos de entrada y salida, etc.

El bloque *RECEPCIÓN* se refiere a la rutina encargada de almacenar la cadena de control enviada por la interfaz computacional. Para asegurar que la cadena se haya recibido completa y correctamente el programa de control envía un eco a la interfaz, y ésta la despliega en una casilla dedicada a este propósito cuyo nombre es *CONFIRMACIÓN*.

Por otro lado, el bloque *MODALIDAD*, analiza el primer carácter de la cadena recibida y con base en esto genera la ruta de procesamiento de acuerdo al modo descifrado. Son 11 los posibles caminos en este programa de control:

- 1) Rutina de *alto*.
- 2) Rutina para modificar el juego mecánico (*backlash*).
- 3) Rutina para avanzar con un cursor programado (*joystick*).
- 4) Rutina para avanzar moviendo dos ó más ejes.
- 5) Rutina para avanzar con una secuencia de *pasos*.
- 6) Rutina para accionar el motor a una frecuencia determinada.
- 7) Rutina para regresar los carros al origen de máquina (*home*).
- 8) Rutinas para incrementar y disminuir la velocidad (sólo disponibles en uso por joystick).
- 9) Rutina para almacenar el cero de pieza.
- 10) Rutina para interpretar una línea de código G.
- 11) Rutina para interpretar una línea de código M.

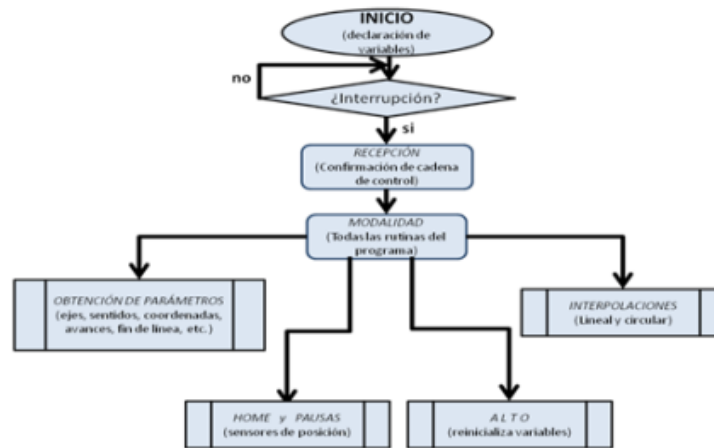


Figura 32. Rutinas y sub-rutinas principales del programa de control.

Los últimos cuatro bloques hacen referencia a todas las sub-rutinas listadas arriba.

4.8 Interfaz computacional.

La interfaz computacional desarrollada en *Visual Basic* consta de dos ventanas: una relativa a la presentación del proyecto llamada *pantalla de inicio* la cual se muestra en la figura 33, y otra dedicada a la operación del prototipo de MMH llamada *interfaz de usuario* la cual se muestra en la figura 34.

En la *pantalla de inicio* esencialmente se observa información relacionada con la dependencia de origen, título de proyecto, autores, país, fecha, y dos botones. El botón *INICIAR* manda a llamar la ventana de la interfaz, con la cual el usuario puede comenzar a operar la MMH. Mientras que el botón *SALIR* cierra la aplicación.



Figura 33. Pantalla de inicio de la interfaz computacional.

En la *interfaz de usuario* mostrada en la figura 34 se muestran nueve secciones. En la sección número 1, tenemos un cuadro programado referente a

las modalidades en las que puede operar el prototipo: por joystick, por pasos, por línea y por programa, la selección de cualquiera de estas opciones habilitará los cuadros correspondientes a estas modalidades.

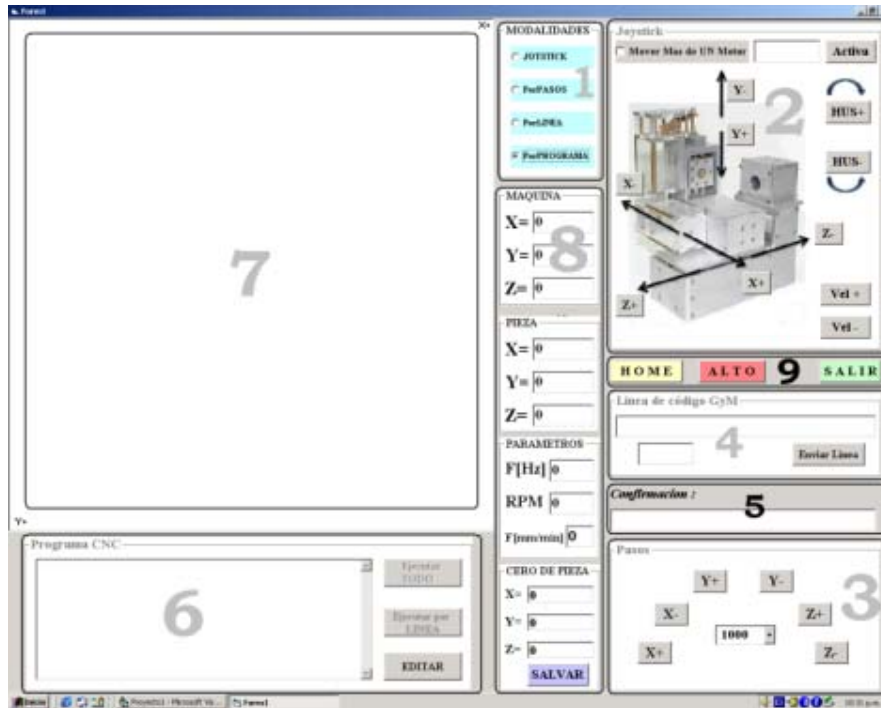


Figura 34. Secciones de la interfaz computacional.

La sección número 2 se refiere a la modalidad por joystick, y en ella se aprecia una imagen del prototipo de MMH a controlar. Sobre dicha imagen, se muestran una serie de botones con los cuales se le indica al usuario de qué eje y en qué dirección se trasladará el carro ($X_{\pm}$, $Y_{\pm}$, $Z_{\pm}$ y $HUS_{\pm}$), además de dos botones que permitirán incrementar ó disminuir la velocidad durante el desplazamiento (Vel+ y Vel-). También se logra distinguir en la parte superior una casilla de verificación (*Mover más de un motor*) que tendrá que ser seleccionada para activar el movimiento de ejes simultáneos, dicho movimiento se ejecutará a través del botón *Activa*, esto ocurrirá siempre y cuando se introduzca la cadena con la sintaxis adecuada (ver sección A.2 de los Anexos) dentro de la casilla próxima al botón rotulado *Activa*.

Por otro lado, la sección 3 es una modalidad dedicada a los avances que se encuentran previamente definidos por 6 secuencias de pasos contenidas en una ventana de selección (1, 14, 28, 10, 50, 100, y 1000), y se pueden ejecutar pulsando los botones que se encuentran etiquetados de la misma forma que en la sección de joystick, con el fin de que el usuario tenga conocimiento de la dirección y sentido de desplazamiento.

En la sección 4 se dispone de una casilla para que el usuario introduzca una línea de código G ó M, ó bien una cadena que active alguna de las rutinas listadas en la sección 4.3.2 de este capítulo. La cadena introducida en esta casilla es enviada al dar *click* en el botón *Enviar línea*, lo cual genera que se envíe carácter por carácter vía puerto serie, y de la misma forma es recibida de

vuelta para su confirmación, y desplegada en la casilla de la sección 5 con el fin de que el usuario verifique la línea enviada.

En la sección 6 se tiene una caja de texto en la que se introduce un programa CNC con líneas que obedezcan la sintaxis mostrada en las secciones A1 y A2 de los anexos. En esta sección, al dar click en el botón *EDITAR*, se le permite al usuario modificar la sintaxis del programa; mientras que el botón *Ejecutar por línea* permite correr el programa CNC línea por línea. En la sección 7 se representa gráficamente el resultado de cada una de las modalidades descritas hasta el momento, abarcando un área de 1.3mmX1.3mm de la totalidad del área de trabajo. Los movimientos realizados en el modo *joystick* y por *pasos* serán graficados en color gris. Los movimientos rápidos referentes a los códigos *G00* y *G28*, además del regreso a cero de máquina, serán graficados en color amarillo. Mientras que las interpolaciones representadas por los códigos *G01*, *G02*, y *G03* serán graficadas en color azul.

La sección 8 es empleada para tareas de monitoreo. En esta sección se registran tres tipos de coordenadas: las primeras son relativas al cero de máquina (X_m, Y_m, Z_m), otras son con referencia al cero de pieza (X_p, Y_p, Z_p), y las últimas correspondientes al cero de pieza (X_{cp}, Y_{cp}, Z_{cp}). Dentro de esta misma sección existen casillas dedicadas al monitoreo de ciertos parámetros como son: el avance, frecuencia y revoluciones por minuto (RPM's) a las que funcionan los actuadores de la MMH. El botón *SALVAR* alojado en esta sección permite establecer el cero de pieza dentro del área de trabajo.

Por último, la sección 9 aloja tres botones, el primero de ellos "*HOME*" ejecuta una rutina que desplaza los carros de la MMH al origen de máquina, esta rutina es la primera en ejecutarse al iniciar la aplicación. El regreso de cada uno de los carros a la posición cero de máquina, por razones de seguridad, se realiza primero en el eje Z, enseguida el eje X, y por último el eje Y. Estos movimientos cesarán hasta detectar la señal arrojada por los sensores de contacto eléctrico (*push buttons*) ubicados al final de carrea de cada uno de los ejes. Después de haber sentido este regreso se ejecuta una rutina de 400 pasos en dirección contraria para cada uno de los ejes, esto es como medida de seguridad.

También es importante mencionar que cada vez que exista un cambio de dirección en cualquier eje se ejecuta una rutina que compense el juego mecánico (*backlash*). El botón rotulado como *ALTO* envía una cadena que al ser descifrada por el Microcontrolador cesa cualquier tipo de movimiento, almacenando el estado actual en el que quedaron energizados los actuadores. El botón *SALIR* tiene la misma función que en la pantalla de inicio.

En consecuencia y una vez que se han descrito las secciones elementales de la interfaz de usuario, se propone la siguiente secuencia para una mejor operación de la interfaz descrita:

1. Se ejecuta la rutina de *HOME*.
2. Una vez terminada, se da *click* de nuevo al botón *HOME*.
3. Se elige modalidad de operación (sección 1).
4. Se opera la MMH en la sección deseada.

5. Para cambiar de modalidad, se recurre a la sección 1.
6. En cualquier momento estarán disponibles los botones *HOME*, *ALTO* y *SALIR*.

Ahora, se proponen dos ejemplos que muestren el funcionamiento del programa de control.

4.9 Ejemplos del proceso de control.

La forma de procesar una cadena de control mediante este sistema, se presenta en dos ejemplos. Estos dos ejemplos fueron propuestos debido a que son representativos de los cuatro modos de operación del sistema. La *Cadena1* ejemplifica los modos *joystick* y *por pasos* (modos muy similares), mientras que la *Cadena2* ejemplifica los modos *por línea* y *por programa* (modos similares). Las cadenas propuestas son las siguientes:

- a) Cadena1: JX+;
- b) Cadena2: G02X02969.00Y03400.00I03012.00J03040.00F08;

Estas cadenas son enviadas por la interfaz de usuario y recibidas vía puerto serie por la tarjeta de control, a través del módulo *USART (Universal Synchronous Asynchronous Receiver Transmitter)* del microcontrolador PIC.

4.9.1 Ejemplo (a).

Primeramente, se ejecuta la rutina de *HOME* con el fin de llevar a una posición conocida los carros de la MMH. Se le da *click* de nuevo al botón *HOME* para habilitar el recuadro de la sección 1. Enseguida, se selecciona el modo *joystick* que habilita la sección 2 de la interfaz. Después se da *click* al botón que activa el movimiento del eje X, X+, lo cual produce el envío de la cadena "JX+;".

Una vez enviada la Cadena1 (JX+), ésta es recibida por la tarjeta de control en la cual se atiende una interrupción, almacenando en un vector llamado *CadenaControl* de 35 elementos la cadena recibida. El primer carácter de la cadena es almacenado en el elemento cero del vector, el segundo en el elemento uno y así sucesivamente. En el momento en el que se detecte el carácter fin de línea (;) se termina el proceso de almacenamiento. La forma de procesar la *Cadena1* una vez almacenada se muestra en el diagrama de la figura 35.

Para decodificar la cadena de control, se ejecuta la rutina de modalidad. Ésta lee el primer elemento del vector y descifra el modo en el cual se quiere operar la MMH. En este caso, la letra J indica que los desplazamientos se harán por medio de un *joystick*. Aparte de descifrar la modalidad de desplazamiento, genera la ruta de procesamiento correspondiente a esa modalidad, es decir, predice que parámetros son los próximos a decodificar. Para este caso en particular, se espera que se defina el grado de libertad en el cual se desplazara la MMH, para lo cual se invoca a la rutina *obténde eje*.

Cabe aclarar que si el carácter leído no se puede descifrar, es decir, no se encuentra dentro de la rutina esperada, el programa interpreta que la sintaxis de la cadena de control es errónea y no ejecuta tarea alguna.

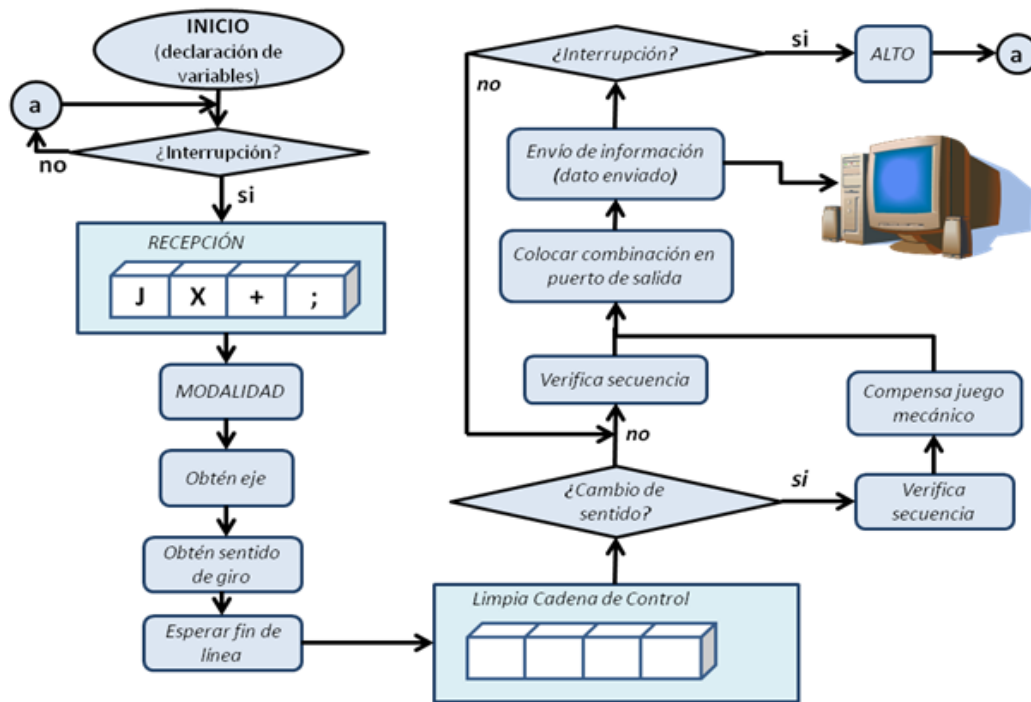


Figura 35. Proceso de control para la cadena "JX+;".

Después de que el programa ha descifrado que los desplazamientos se harán en el eje X, el proceso necesita saber el sentido de giro del actuador para lo cual se ejecuta una rutina llamada *obtén sentido de giro*. En esta rutina se pregunta si el sentido corresponde al carácter "-" (guión medio), el cual es interpretado como un sentido antihorario, en caso de no ser así, el programa interpreta que se trata de un sentido horario. Para finalizar esta decodificación se ejecuta la rutina *esperar fin de línea*, la cual espera el carácter ";" (punto y coma) y de no ser así, el código es erróneo.

Ya que ha sido descifrada la modalidad de control se limpia el vector *CadenaControl* para una nueva recepción, y se ejecuta inmediatamente después la rutina que pregunta si hay *cambio de sentido*. En esta rutina se compara el estado de dos banderas: *sentidoactual* y *sentidoanterior*. El propósito de esta comparación es compensar el juego mecánico de la MMH, por lo tanto, si el estado de estas dos banderas resulta ser diferente entonces se ejecuta una rutina que compensa el juego mecánico.

Para cualquier tipo de movimiento a realizar en la MMH es obligatorio ejecutar una rutina denominada *verifica secuencia*. Ésta se encarga de seguir un orden para energizar las fases del actuador con el propósito de no ejecutar pasos de más, o avanzar en sentido incorrecto.

La combinación generada (byte) se graba en el puerto D (salidas a actuadores) del microcontrolador, esta combinación permanece grabada en el puerto durante un determinado tiempo (temporización), este tiempo está de acuerdo a

la frecuencia en la que se esté ejecutando la modalidad en cuestión. En el modo *joystick* la frecuencia de activación de los actuadores es de 170Hz.

Una vez que el actuador giró en la dirección y sentido correctos, se ejecuta una rutina denominada *envío de información* en la cual informamos a la interfaz, mediante un variable denominada *dato enviado*, que se ha efectuado un movimiento. En la variable usada para este propósito va implícita la dirección y sentido del movimiento. Cabe mencionar que cualquier avance realizado en la MMH será transmitido a la interfaz de usuario para su correspondiente registro: coordenadas y representación gráfica.

Por último, se verifica el estado de la bandera de interrupción por recepción serial para una posible decodificación de una nueva cadena de control. En caso de estar activada se detiene el avance de cualquier actuador mediante la rutina *Alto*, y se vuelve a ejecutar la rutina de *Recepción*. De no estar activada la bandera de interrupción, se prosigue en la siguiente combinación para energizar el actuador mediante la rutina llamada *verifica la secuencia*.

4.9.2 Ejemplo (b).

Para este ejemplo se repite la primera parte del ejemplo anterior, se ejecuta la rutina de *HOME* con el fin de llevar a una posición conocida los carros de la MMH. Se le da *click* de nuevo al botón *HOME* para habilitar el recuadro de la sección 1. Enseguida, se selecciona el modo *por línea* que habilita la sección 4 de la interfaz.

Después se introduce la Cadena2 en la casilla de ésta sección (G02X02969.00Y03400.00I03012.00J03040.00F08;). Enseguida se da *click* al botón *ENVIAR* que produce el envío serial de la Cadena 2.

Ahora bien, en este ejemplo la cadena a manejar será un código G correspondiente a una interpolación circular en sentido horario cuya sintaxis es como sigue: *G02X02969.00Y03400.00I03012.00J03040.00F08;*. El procedimiento a seguir es mostrado en la figura 36 y es muy parecido al del ejemplo (a), con la diferencia que algunas rutinas cambiarán debido a que esta modalidad requiere de parámetros diferentes.

Hasta el bloque de modalidad todo resulta igual. Después de éste, se genera una ruta diferente ya que la modalidad descifrada es distinta, la cual en este caso es *línea de código G*. En consecuencia, se invoca una rutina llamada *obtención de 2 dígitos*, esta rutina es la encargada de descifrar y almacenar el número de código G en una variable denominada *NoCodigoG*, para después pasar a la rutina de *obtención de eje* donde el procedimiento a seguir resulta ser el mismo que en el ejemplo anterior, hasta el bloque de *sentido de giro*.

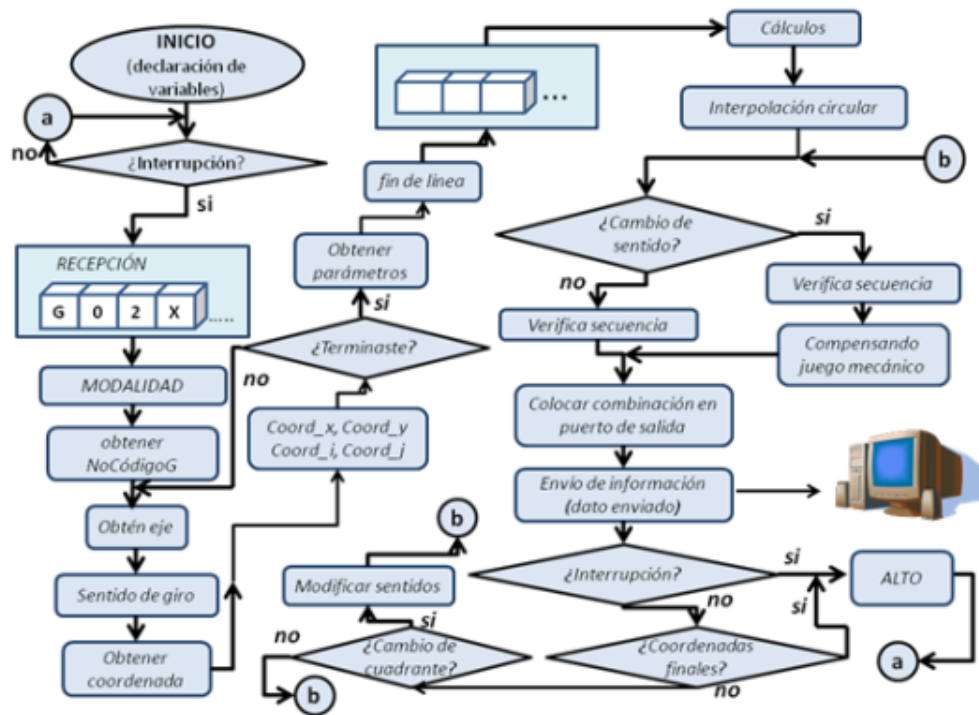


Figura 36. Proceso de control para un código G02.

Una vez que se ha determinado el eje y el sentido de giro, se procede a obtener la magnitud de la coordenada en cuestión, esto se logra a través de dos rutinas, una para obtener la magnitud entera de cinco dígitos y otra para obtener la magnitud decimal, de dos dígitos. Debido a que este código G requiere de 4 coordenadas en total, los procesos de obtención de eje, sentido de giro, obtención de magnitud entera y decimal se repiten cuatro veces.

Después del bucle generado para la obtención de coordenadas, el programa ejecuta la rutina *obtención de parámetros*. En ésta, se tiene la posibilidad de descifrar algunos parámetros como son: avances, revoluciones por minuto, herramienta, y pausas. En el caso del código a tratar (G02) se requiere el parámetro *avance*, para lo cual se ejecuta la rutina de obtención de 2 dígitos, esta rutina almacena la magnitud de este parámetro en una variable denominada *Avance*.

Sin más parámetros requeridos para esta interpolación, se verifica el carácter fin de línea y, en caso de presentarse, se limpia el vector *CadenaControl* y se pasa a una rutina de cálculos, propios de la interpolación a realizar.

La rutina de cálculos dicta las condiciones iniciales bajo las cuales comenzará el método de interpolación (cuadrante inicial, cuadrante final, radio, frecuencia, distancias, etc.). Para el caso de interpolación circular el método a emplear es el de *Búsqueda Directa*. Durante la ejecución de éste método de interpolación se estarán procesando continuamente los bloques restantes, a partir del bloque *cambio de sentido* hasta el bloque para verificar el estado de la bandera por interrupción.

Por último, después de ejecutarse cada iteración del algoritmo, se verifica si alcanzo las coordenadas finales y si hay cambio en el número de cuadrante,

con el propósito de parar la ejecución del algoritmo de interpolación ó continuar con el mismo, modificando sentidos de giro.

4.10 Hardware electrónico.

En este trabajo, se proponen dos diferentes circuitos integrados de potencia, el PBL3717A de la compañía *ST Microelectronics* y el A3988 de *Allegro MicroSystems*. La selección del C.I. a emplear es realizada con el mismo método empleado para el algoritmo de interpolación (matriz de decisión). Para la aplicación de éste método se practican los pasos enunciados en el apartado *Comparación de los métodos de interpolación y Selección del método de interpolación* de este capítulo.

4.10.1 Circuito PBL3717A.

El PBL3717A es un circuito capaz de controlar una fase de un motor a pasos bipolar, este circuito cuenta con control de corriente (por chopper) que limita ésta última a tres niveles seleccionables a través de dos entradas lógicas, I_0 e I_1 (pines 7 y 9). Cuando estas dos entradas están a un nivel alto el dispositivo está deshabilitado.

La sección de potencia está compuesta por un puente H bipolar con cuatro diodos internos de conmutación rápida, los cuales protegen los devanados del motor de posibles regresos de corriente. Este circuito es capaz de proporcionar una corriente de hasta 1.2A, con un voltaje de carga que puede ir de los 10 a los 50V. El diagrama interno que describe el funcionamiento de este circuito es mostrado en la figura 37.

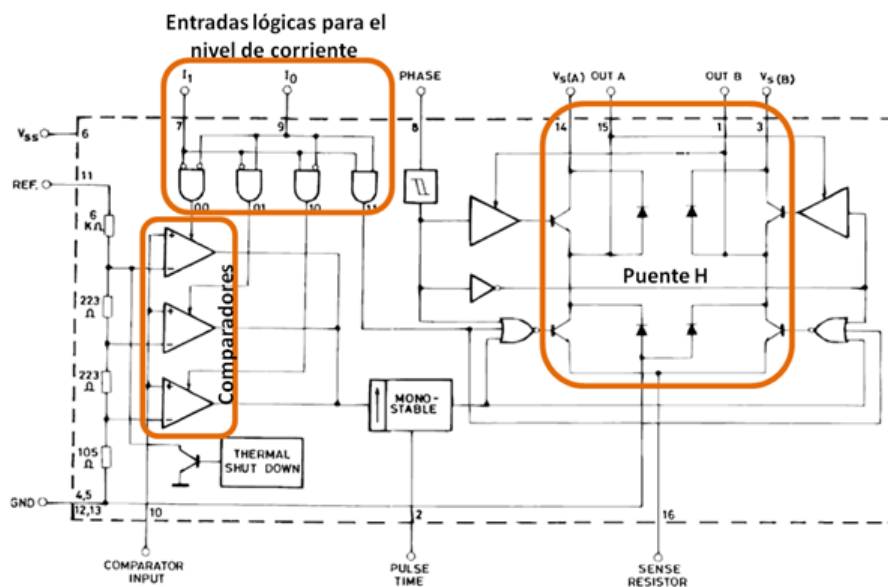


Figura 37. Circuito interno del PBL3717A.

A este tipo de circuito se le denomina de *tiempo de desconexión constante*, es decir, una vez alcanzado el valor de la corriente del devanado (previamente establecida), ésta se interrumpe durante un tiempo fijo. Una red externa RC conectada al pin 10 determina el tiempo de apagado de los transistores de potencia inferiores del puente H, tal y como lo describe la ecuación 15.

$$T_{OFF} = 0.69RC \quad \dots\dots\dots (15)$$

Los pulsos detectados por los transistores de potencia son generados por un circuito monoestable que se encuentra a la salida de los comparadores.

Los valores de las entradas lógicas I_0 y I_1 , y el valor del voltaje de referencia del pin 11, determinan el voltaje a compararse con el voltaje generado desde la resistencia de sensado en el pin 16, que a su vez está conectada a los emisores de los transistores inferiores de cada puente H. El valor de la corriente pico máxima a controlar se determina por la siguiente expresión:

$$I_{pico} = 0.083 V_{ref} / R_{sen} \quad \dots\dots\dots (16)$$

donde:

I_{PICO} : Corriente pico máxima.

V_{REF} : Voltaje de referencia.

R_{SEN} : Resistencia de sensado.

El sentido en el que circula la corriente en el devanado, se controla por medio de la entrada digital *FASE* (pin 8). El circuito mostrado en la figura 38, permite operar un motor a pasos bipolar de manera bidireccional y de acuerdo a la secuencia formada por las señales de entrada *FASE A* y *FASE B* (PhA y PhB).

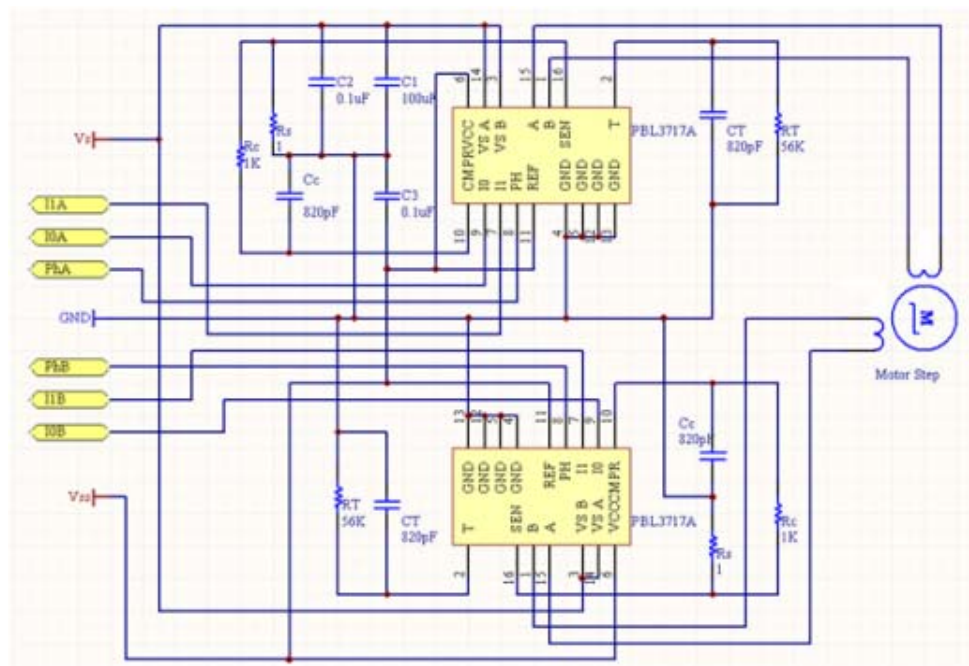


Figura 38. Circuito para controlar un motor a pasos bipolar de manera bidireccional.

El PBL3717A permite implementar micropasos, pero a expensas de un menor torque. De tal forma, el motor a pasos puede operar en modo de *paso completo*, *medios pasos* y *cuartos de paso*. Para controlar este circuito, las entradas lógicas I_{0A} , I_{1A} , Ph_A , I_{0B} , I_{1B} y Ph_B mostradas en la figura 40, son generadas a través de la tarjeta controladora (PIC18F8720).

Por último, en la entrada FASE de cada circuito se incorpora un *schmitt trigger* con el fin de proporcionar inmunidad al ruido. Además, este circuito cuenta con una protección por exceso de temperatura que inhibe la operación a máxima corriente.

4.10.2 Circuito A3988.

El A3988 es un circuito diseñado para operar dos motores a pasos bipolar ó cuatro motores de DC. La corriente máxima es de 1.2A con un voltaje de carga que va de los 8 a los 36V. Esta corriente es proporcionada por cada uno de los puentes H (de transistores DMOS) y regulada a través de un circuito interno PWM, éste usa un circuito disparador que controla el tiempo que los *drivers* permanecen apagados, internamente este tiempo es fijado a 30µs. La corriente pico en cada puente es determinada a través de resistencias sensoras (R_{SENS1} , R_{SENS2} , R_{SENS3} y R_{SENS4}) y de un voltaje de referencia (V_{REF}). El diagrama interno de éste circuito y su conexión a motores de pasos es mostrado en la figura 39.

Las entradas lógicas de este circuito son las mismas que en el caso del PBL3717A ($PHASE$, I_0 e I_1) y se recomienda que de ser alimentadas a un nivel lógico alto se haga a través de resistencias de *pull-up*, con el fin de limitar la corriente de entrada y evitar sobrevoltajes. Además, la configuración de estas entradas permite establecer los modos de funcionamiento del motor: paso completo, medios pasos y cuartos de paso.

Al igual que el circuito anterior, el A3988 cuenta con un circuito que al detectar un sobrecalentamiento deshabilita las salidas del dispositivo hasta que las condiciones mejoren. Este circuito se tiene en dos matrículas diferentes, A3988EV y A3988JP, ambas son de montaje superficial con un área de contacto expuesta para un mejor desempeño térmico.

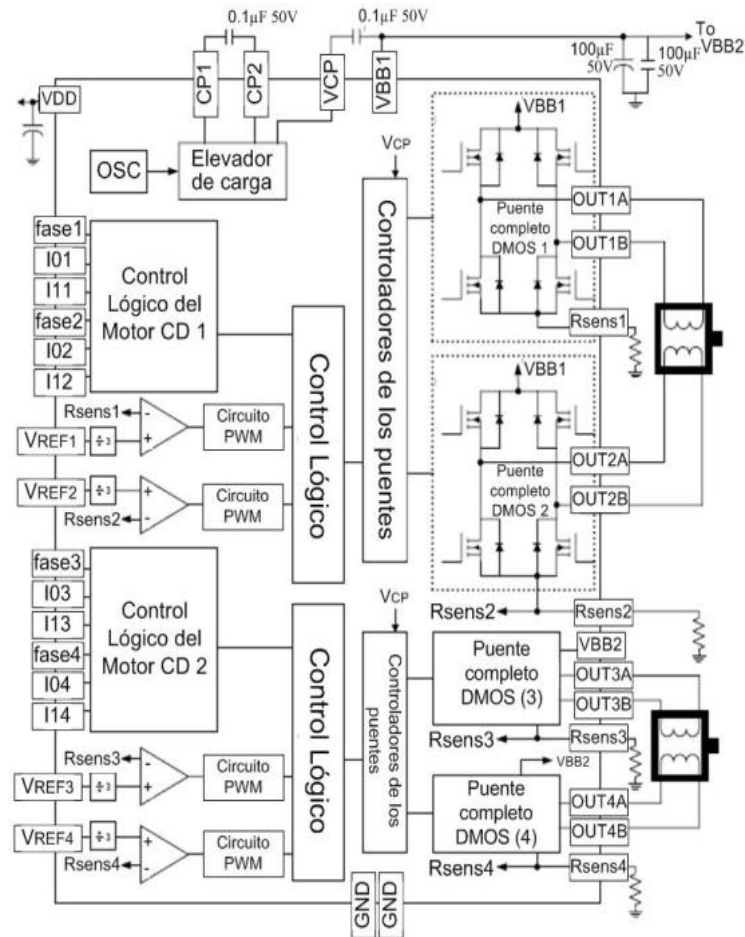


Figura 39. Circuito interno del A3988 para controlar dos motores a pasos bipolares.

4.10.3 Evaluación y selección del circuito de potencia.

En la tabla 9 se ordenan jerárquicamente los parámetros de importancia para la evaluación de los circuitos de potencia. Los valores relativos asignados a los parámetros manejan los mismos niveles de importancia que en la sección 4.5.

Parámetros de importancia	Valor relativo asignado
Corriente proporcionada.	3
Voltaje de carga.	3
Recursos para implementar.	3
Elementos externos requeridos.	1

Tabla 9. Parámetros de importancia en la comparación de circuitos de potencia.

En la tabla 10 se muestra el resultado de la evaluación de los circuitos con respecto a cada uno de los parámetros listados en la tabla 9. En *cursivas* se muestra el resultado de multiplicar la calificación por el factor de importancia de cada parámetro. La forma de realizar esta evaluación fue de la siguiente manera: para la *corriente*, se calificó con 10 a ambos circuitos debido a que la magnitud proporcionada de 1.2A fue suficiente, tomando en cuenta que la resistencia en cada bobina es de 29Ω y el intervalo de voltaje en la operación de los motores a pasos fue de los 8 los 14V. En cuanto al *voltaje*, se le otorgó

una mayor calificación al PBL3717A debido a la posibilidad de operarlo en un mayor intervalo de voltaje.

Circuito de potencia	Corriente (3)		Voltaje (3)		Recursos para implementar (3)		Elementos externos requeridos (1)		Total
PBL3717A	10	30	10	30	10	30	8	7	97
A3988	10	30	9	27	5	15	10	10	82

Tabla 10. Evaluación de los circuitos de potencia.

Como resultado de la evaluación realizada en la tabla 10, el circuito PBL3717A es implementado como el sistema de potencia del sistema de control propuesto. En la figura 40 se muestra una imagen del circuito impreso correspondiente a una etapa de potencia para dos motores a pasos. Esta tarjeta consta de 4 conectores molex (de cuatro y tres pines), uno para la alimentación, otro para las señales de control provenientes de la tarjeta controladora (Microcontrolador PIC), y dos más para las salidas a los motores a pasos. También cuenta con un circuito de aislamiento (PC847) con el fin de proteger a la tarjeta controladora. De tal forma, esta tarjeta pretende mejorar los problemas relacionados con la limitante respecto al número de actuadores controlados por el sistema de control [Okazaki, 2000].



Figura 40. Sistema de potencia para dos actuadores de la MMH.

El esquemático y diseño del circuito impreso de la tarjeta mostrada en la figura 40, están reportados en la sección A.3 de los Anexos de este trabajo.

5. PRUEBAS Y RESULTADOS.

En este apartado se presentan las pruebas que se realizaron al sistema de control descrito anteriormente. En primer lugar se obtuvieron las señales generadas por la tarjeta controladora, y el circuito de potencia seleccionado. Después se realizaron tres tipos de pruebas al sistema micromecánico en

cuestión. La primera prueba consistió en maquinar figuras simples como rectas y cuadrados, en donde los movimientos realizados son en un eje a la vez. El objetivo de esta prueba fue determinar el juego mecánico, que existe en los ejes X y Y de la MMH, y compensarlo de la mejor manera posible.

Posteriormente se maquinaron figuras que requerían interpolación en dos ejes (circunferencias concéntricas); finalmente, se desarrollaron pruebas que consistieron en la ejecución de programas en código estándar G y M.

Para el desarrollo de las pruebas enunciadas anteriormente, se emplearon los siguientes elementos: una PC con Windows 2000 y puerto serie, la MMH de la figura 24, láminas delgadas de acero y latón bañadas en tinta azul para metal, como herramienta de desbaste fue empleada una punta de acero, un osciloscopio de cuatro canales, una fuente de alimentación dual de 30V.

El sistema propuesto y cada una de sus partes se muestran en la figura 41. El maquinado de las figuras comienza a partir del momento en el que la punta hace contacto con la lámina, para después comenzar con el seguimiento de las trayectorias establecidas dentro del código programado.

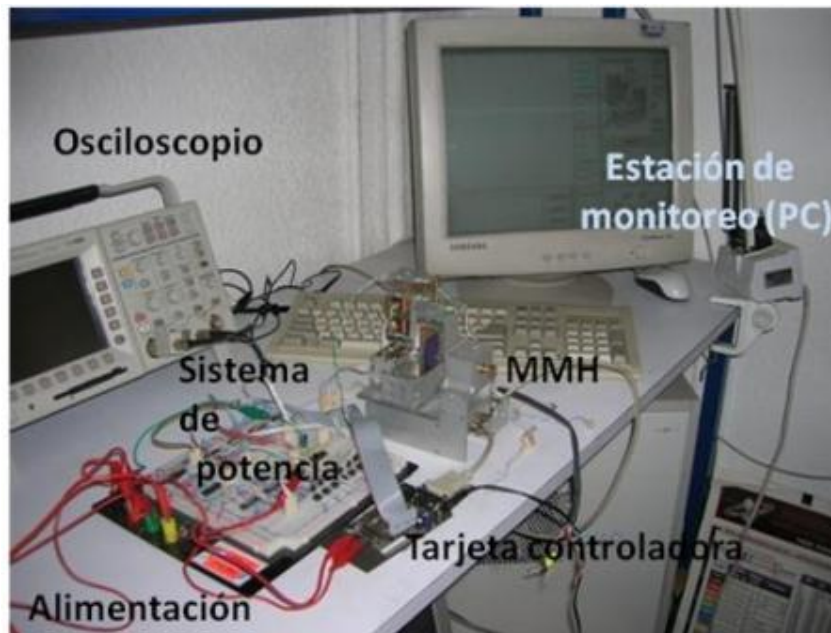


Figura 41. Sistema propuesto para las pruebas al sistema de control.

Las mediciones de los maquinados realizados en este trabajo se llevaron a cabo por medio de un comparador óptico de marca “Nikon Profile V-16D” con resolución de $1\mu\text{m}$, el cual cuenta con un sistema de proyección de tres lentes de aumento 20x, 50x y 100x y un área de trabajo de 50mm X 50mm. Este comparador óptico es mostrado en la figura 42, y se encuentra en el Laboratorio de Metrología del CCADET, UNAM.

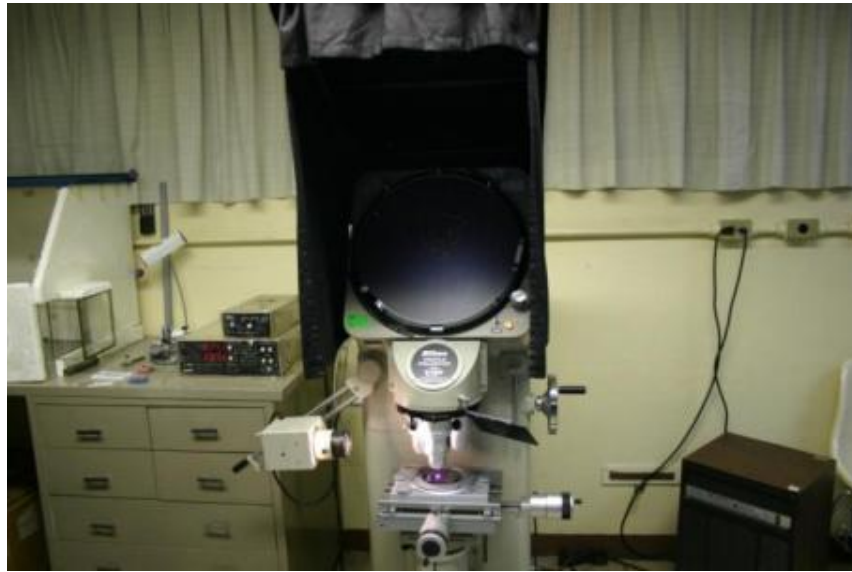


Figura 42. Comparador óptico del Laboratorio de Metrología.

5.1 Pruebas realizadas al *hardware* electrónico.

Las pruebas realizadas al *hardware* electrónico consistieron primeramente en la obtención de las señales generadas por la tarjeta de control. En la figura 43, se muestran los trenes de pulsos generados para desplazar los carros en cada uno de los ejes X, Y y Z de la MMH, a una frecuencia de 42Hz con un de voltaje lógico TTL.

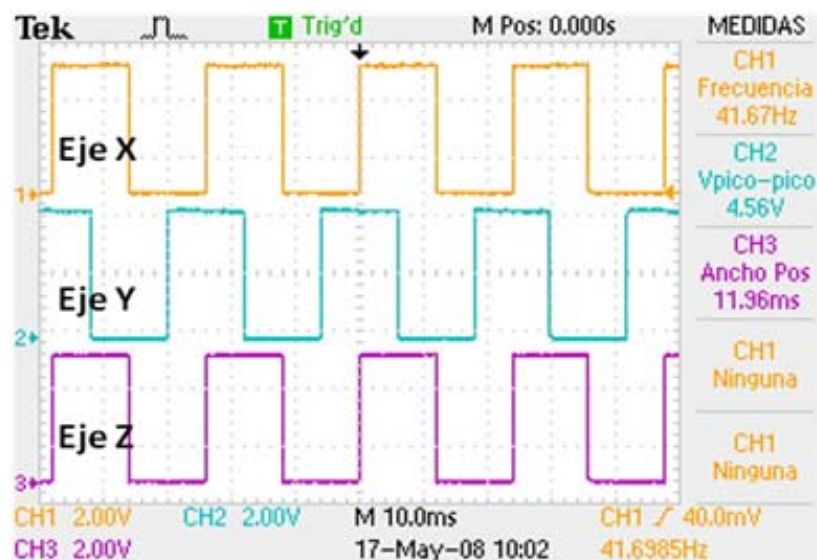


Figura 43. Señales generadas por la tarjeta de control para los tres ejes de la MMH.

Por otro lado, en la figura 44 se observan las señales generadas por el sistema de potencia (sin carga) al proporcionarle las señales de la figura 43, con un

voltaje de carga de 11.5V registrándose un consumo de todo el sistema de 1.2A (tres grados de libertad).

En la figura 45, se tienen las señales generadas por el sistema de potencia al conectarles los motores de la MMH.

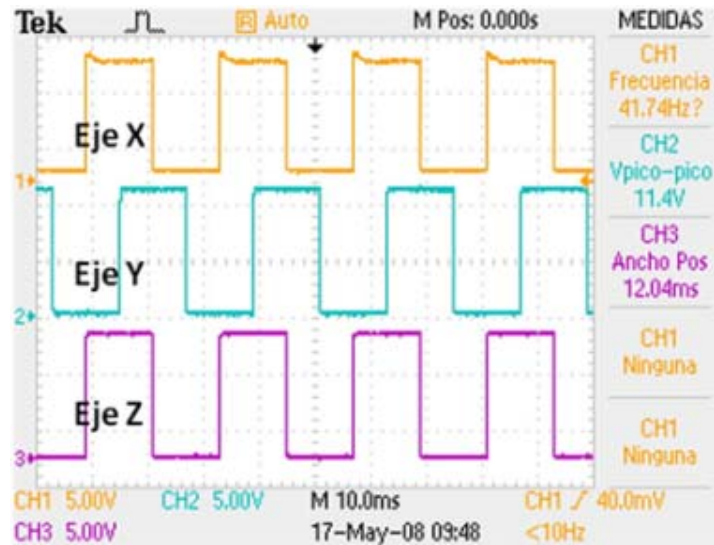


Figura 44. Señales generadas por la etapa de potencia(sin carga).

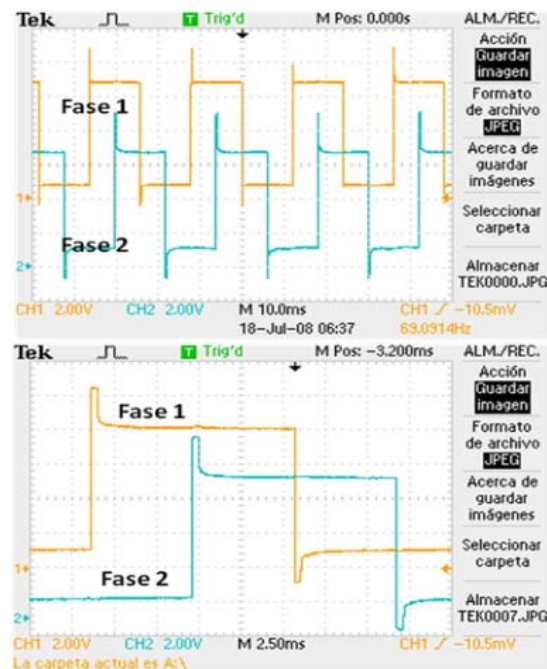


Figura 45. Señales generadas por la etapa de potencia con carga inductiva.

En estas señales se aprecian unos picos debido a efectos inductivos provocado por las fases del motor al ser conectado al sistema de potencia. Para mejorar estas señales se propone un circuito LC, mostrado en la figura 46, a la salida de cada una de las fases de un motor (circuito PBL3717A). Para determinar el

valor de la inductancia y capacitancia del arreglo LC, se aplican las siguientes ecuaciones:

$$L \approx \frac{1}{10} L_M \quad \dots\dots\dots (17)$$

$$C \approx \frac{4 \cdot 10^{-10}}{L} \quad \dots\dots\dots (18)$$

donde:

L_M : Inductancia de la fase del motor.

Aplicando las ecuaciones (17) y (18), para un valor de $L_M = 18\text{mH}$, se obtuvo una inductancia $L = 1.8\text{mH}$, y consecuentemente una $C = 220\text{nF}$.

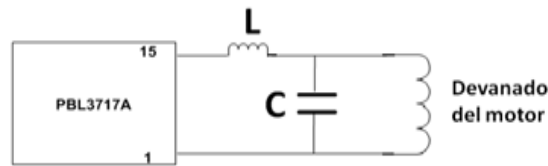


Figura 46. Circuito LC para reducir efectos inductivos en las fases del motor.

Después de haber implementado el arreglo LC, se puede apreciar en la imagen (de mayor periodo) de la figura 47, que el área formada por los picos disminuyó, es decir los picos son más angostos. Como parte de este ejercicio, se implementó un arreglo LC doble en serie (segundo orden) en una sola fase del motor (fase 1), mientras que en la fase 2 no se tiene circuito LC. Las señales obtenidas son mostradas en la figura 48.

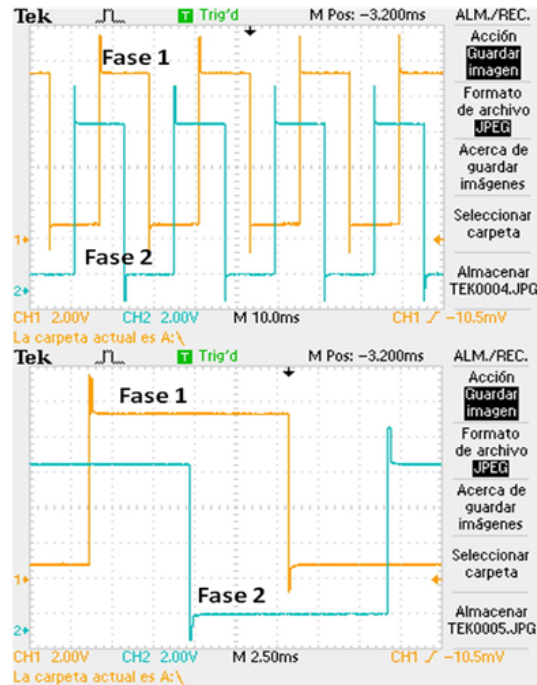


Figura 47. Señales generadas por el sistema de potencia con circuito LC.

En estas imágenes se observa cómo se reduce, aún mas, el área formada por los picos, al aumentar el orden del arreglo. Esto representa un menor consumo de corriente y menos calentamiento en los devanados del motor.

Por último, en la figura 49 se tienen registradas las señales de los ejes X y Y, al procesar un código G02 (interpolación circular en sentido horario). Estas señales corresponden a la transición del cuadrante 2 al cuadrante 3 en una circunferencia de un radio de 100 μm .

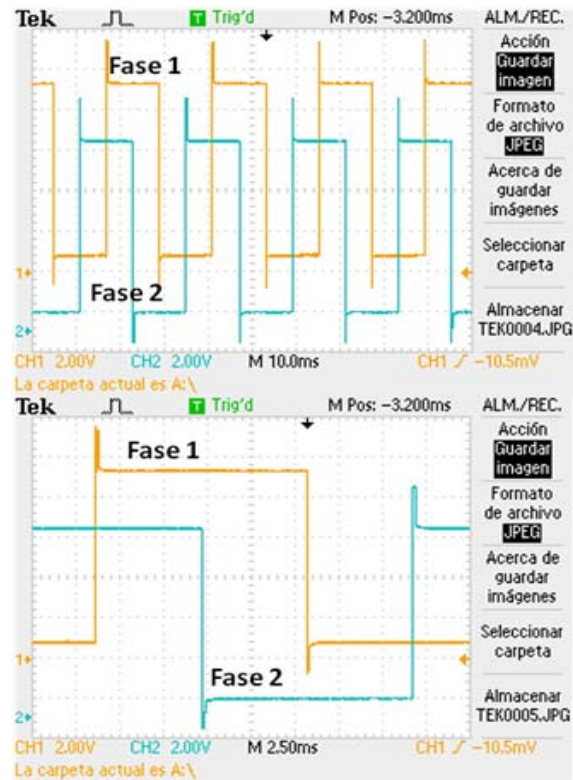


Figura 48. Señales generadas con circuito LC de segundo orden en fase 1.

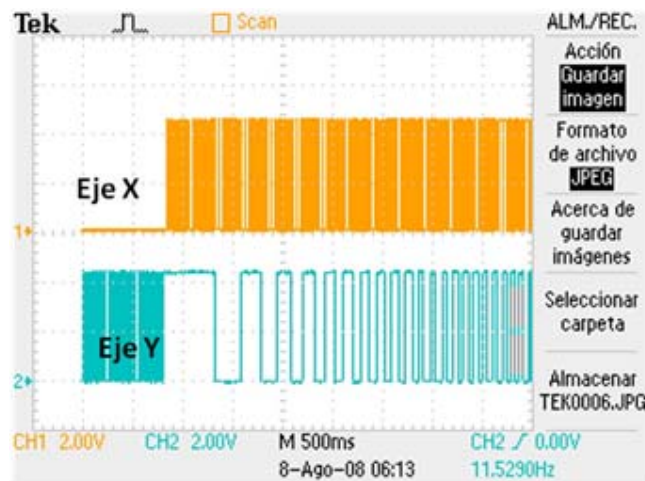


Figura 49. Señales generadas por un código G02.

5.2 Maquinado de figuras simples.

Los primeros maquinados llevados a cabo consistieron en dos rectas paralelas a cada uno de los ejes en cuestión X y Y, unidas en uno de sus extremos por otra recta. Con el fin de medir el juego mecánico en toda el área de trabajo, se dividió esta última, en tres regiones horizontales (H1, H2 y H3) y tres verticales (V1, V2 y V3). Las dimensiones del área de trabajo son: 10mm en el eje X y 7mm en el eje Y. La diferencia de las longitudes de las dos rectas, nos permitirá tener una mejor evaluación del juego mecánico existente en cada una de las tres regiones verticales y horizontales del área de trabajo. En la figura 50, se muestran los resultados obtenidos de estos maquinados mientras que, en la tabla 11 se muestran los datos arrojados por estas pruebas.

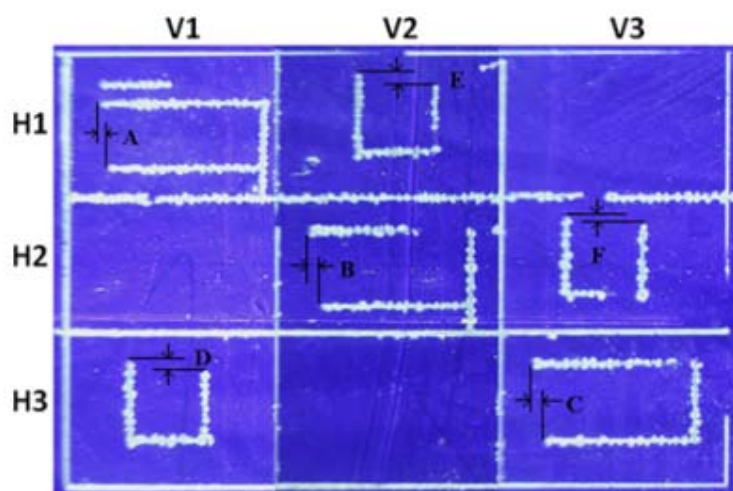


Figura 50. Maquinados para determinar el juego mecánico en toda el área de trabajo.

Tal como lo muestra la figura 50, así como las magnitudes A, B, C, D, E y F colocados en la tabla 11, el juego mecánico no es uniforme en toda el área de trabajo. De tal forma, se obtuvo la media aritmética (para X y Y) con estos valores, resultando un juego mecánico por compensar para el eje X de 192.5 μ m, mientras que en el eje Y fue de 168.5 μ m.

Figura	Dimensiones teóricas [mm]	Dimensiones experimentales [mm]	Juego mecánico [μ m]	Media Aritmética [μ m]
A→ejeX (región H1)	2.5	2.509	190	---
B→ejeX (región H2)	2.5	2.503	201	---
C→ejeX (región H3)	2.5	2.502	187	= 192.5
D→ejeY (región V1)	1.2	1.205	152	---
E→ejeY (región V2)	1.2	1.203	180	---
F→ejeY (región V3)	1.2	1.199	173	= 168.5

Tabla 11. Magnitudes del juego mecánico para los ejes X y Y.

Con el fin de corroborar los valores de compensación del juego mecánico obtenidos en la tabla 11, se maquinaron tres cuadrados de 2.5mm en las tres diferentes regiones del área de trabajo. En la figura 51 se muestran los cuadrados maquinados, mientras que en la tabla 12 se muestran los resultados de forma cuantitativa.

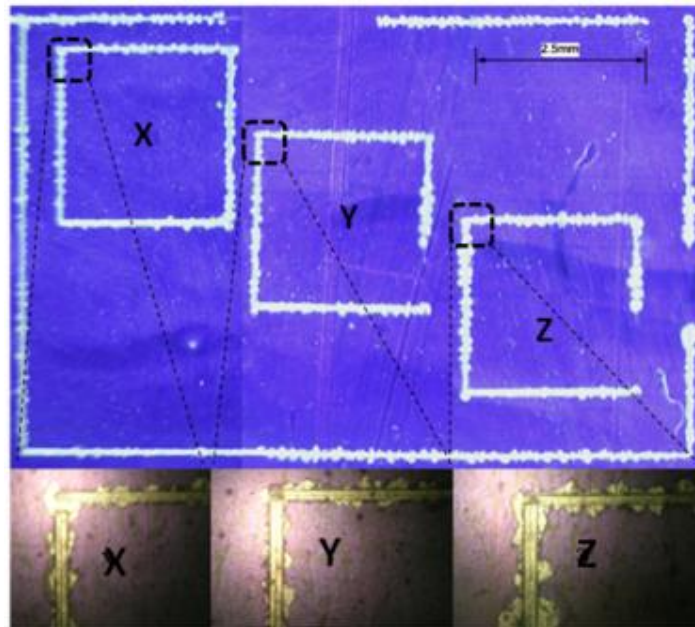


Figura 51. Maquinados para corroborar la compensación del juego mecánico.

Figura	Dimensiones Teóricas [mm]	Dimensiones experimentales [mm]	Juego mecánico [μm]	Media aritmética [μm]
Cuadrado_X	2.5	Dx = 2.489 Dy = 2.519	X = 12 Y = 19	= 11.5
Cuadrado_Y	2.5	Dx = 2.484 Dy = 2.497	X = 17 Y = 5	= 9.5
Cuadrado_Z	2.5	Dx = 2.497 Dy = 2.494	X = 6 Y = 5	

Tabla 12. Resultados de los maquinados mostrados en la figura 53.

Como se observar en la figura 51, la compensación realizada con el promedio obtenido de los valores de la tabla 13, aún no compensa la totalidad del juego mecánico. De tal forma, se obtuvo de nuevo la media aritmética de las micras restantes obtenidas en la tabla 12.

En consecuencia, la media aritmética resultante de la tabla 12, en X, es sumada a la ya existente resultando en una nueva compensación en el eje X de 204 μm , mientras que en el eje Y fue de 178 μm . Con estos nuevos valores, se procedió a realizar el maquinado de geometrías que requiriesen la interpolación en ambos ejes.

5.3 Maquinado con interpolación en dos ejes.

El fin de estas pruebas es cuantificar el nivel de aproximación de los algoritmos de control programados en el Microcontrolador PIC a una geometría dada, de tal forma que, tanto de manera visual y por dimensionamiento de la geometría propuesta, se comparen los resultados obtenidos en este trabajo con los del sistema de control anterior (*Desarrollo de una interfaz computacional para el*

control de Micromáquinas-Herramienta). Esto permite saber si es posible la implementación de estos algoritmos de control en otros prototipos micromecánicos desarrollados.

Para el desarrollo de estas pruebas se propusieron como geometrías de maquinado tres círculos concéntricos (a), (b) y (c) de 3, 2 y 1mm de radio respectivamente. Estas geometrías tienen como fin verificar la capacidad del sistema propuesto para conservar un determinado diámetro, así como el centro de las geometrías. Los maquinados resultantes (realizados en sentido horario) se muestran en la parte alta de la figura 52, y las magnitudes de los diámetros ($\phi 1$, $\phi 2$, $\phi 3$, $\phi 4$, $\phi 5$, $\phi 6$, $\phi 7$ y $\phi 8$) correspondientes a cada circunferencia son registradas en la tabla 13.

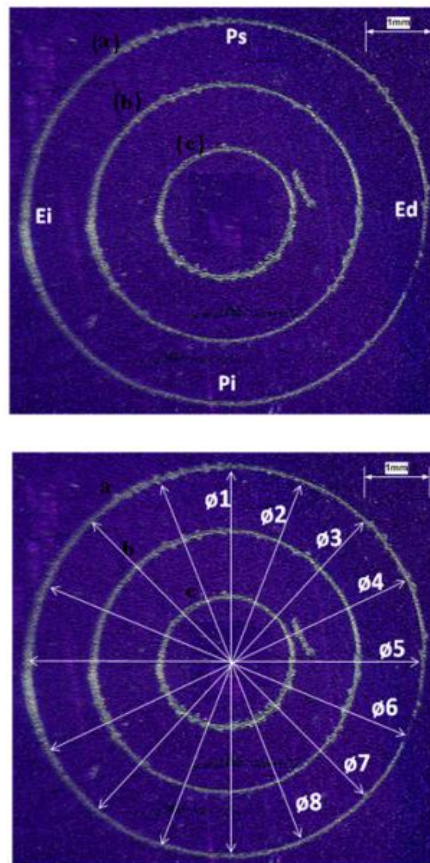


Figura 52. Maquinados con interpolación circular en ejes X y Y.

Por otro lado, en las figuras 53 y 54 se muestran unos acercamientos correspondientes a las zonas Ps, Pi, Ei y Ed de cada circunferencia de la figura 52. En la figura 53, se observa que la compensación al juego mecánico, del extremo Ei, está sobrada por $21\mu\text{m}$ en las circunferencias (a) y (b) y $18\mu\text{m}$ en la (c). Mientras que en la zona Ed, se compensaron 51, 25 y $12\mu\text{m}$ de más en las circunferencias (a), (b) y (c) respectivamente.

Por último, en la figura 54 se aprecia que en la zona Pi existe una compensación de más en el eje Y, cuya magnitud es de $30\mu\text{m}$ en la circunferencia (a), mientras que en la zona Ps compensa $9\mu\text{m}$ de más en las circunferencias (b) y (c).

Circulo	Diámetro teórico [mm]	Diámetros medidos [mm]	Juego mecánico en los extremos [μm]	Tiempo [min]	Media aritmética y Desviación estándar [μm]
A	6	Ø1=5.950 Ø2=5.980 Ø3=6.001 Ø4=6.003 Ø5=5.961 Ø6=5.926 Ø7=5.904 Ø8=5.940	Ei = 21 Ed = 51 Ps = 8 Pi = 30	23	=0.035
B	4	Ø1=3.987 Ø2=3.991 Ø3=3.998 Ø4=4.018 Ø5=3.990 Ø6=3.961 Ø7=3.947 Ø8=3.969	Ei = 21 Ed = 25 Ps = 9 Pi = 13	14	=0.022
C	2	Ø1=1.991 Ø2=2.001 Ø3=2.017 Ø4=2.017 Ø5=2.001 Ø6=1.971 Ø7=1.970 Ø8=1.967	Ei = 18 Ed = 12 Ps = 9 Pi = 3	8	=0.020

Tabla 13. Resultados de los maquinados mostrados en la figura 54.

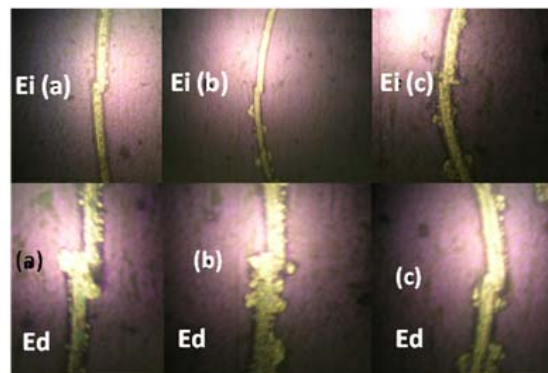


Figura 53. Errores en la compensación del juego mecánico (Extremos izquierdo y derecho).

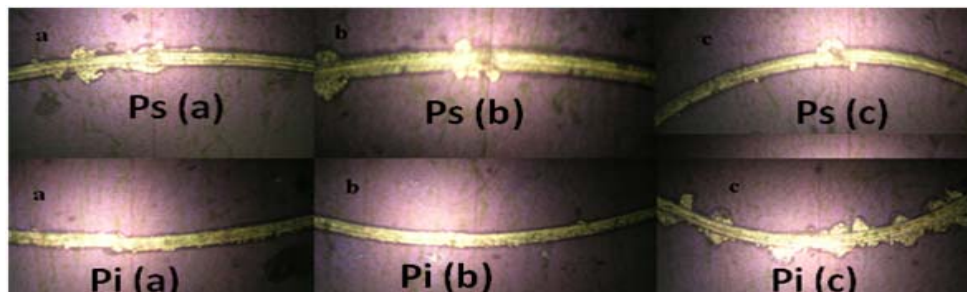


Figura 54. Errores en la compensación del juego mecánico (Polos superior e inferior).

Con el fin de obtener un error promedio de las mediciones realizadas, se analizaron estadísticamente los diámetros registrados en la tabla 13 mediante la desviación estándar que proporciona información sobre un 62.8% de la muestra, por lo que fue necesario multiplicarlo por tres para tener el error en un porcentaje del 99.3% de la muestra [mathWorld, 1997].

Por lo tanto, los errores de estas mediciones son:

$$3\sigma_a = 0.105\mu\text{m} \quad \text{Error}_{(a)} = \pm 105\text{nm}$$

$$3\sigma_b = 0.066\mu\text{m} \quad \text{Error}_{(b)} = \pm 66\text{nm}$$

$$3\sigma_c = 0.060\mu\text{m} \quad \text{Error}_{(c)} = \pm 60\text{nm}$$

Con los resultados obtenidos, es posible deducir que el juego mecánico, aún existente en el desarrollo de estas geometrías, es debido a la variación de éste mismo parámetro dentro de las regiones propuestas en la figura 50.

Conservando las magnitudes de compensación del juego mecánico, se procedió a desarrollar geometrías con un mayor número de instrucciones. Para evaluar este aspecto se seleccionó el escudo de del equipo de futbol de los pumas de la UNAM.

5.4 Ejecución de un programa en código estándar G y M.

El procedimiento realizado para la elaboración de las geometrías propuestas fue el siguiente:

1. Dibujar ambas figura en *software* CAD.
2. Generar el proceso de manufactura en una plataforma CAM.
3. Generar el código G y M.
4. Depurar en forma manual el código obtenido en el punto anterior.
5. Colocar el código depurado en la ventanilla correspondiente de la interfaz computacional desarrollada.
6. Ejecutarlo línea por línea.
7. Representar gráficamente del progreso de manufactura.

En la figura 55 se muestran dos imágenes de dimensiones similares como resultado del maquinado del escudo de los pumas. A la izquierda se muestra el maquinado de un puma de 4mm, realizado con *la interfaz computacional para el control de Micromáquinas-Herramienta* [Yañez, 2007]. En la imagen de la derecha tenemos el maquinado del puma de 4.8x4.96mm que se llevó a cabo con el sistema de control propuesto en este trabajo. Además, en la figura 56 se muestra una fotografía referente a uno de los procesos de maquinado de esta geometría; y en la tabla 14 se muestran los resultados cuantitativos de las dimensiones, así como el error relativo de partes de la figura 55.

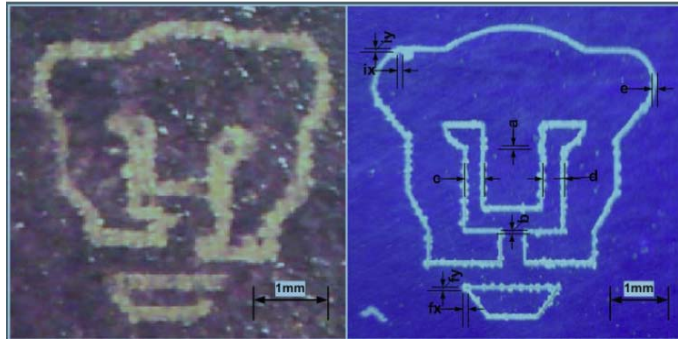


Figura 55. Puma maquinado con interfaz computacional [Yañez, 2007] (izq), y Puma maquinado con sistema de control mediante Microcontrolador PIC (der).

Las magnitudes a , b , e , i_x , i_y , f_x y f_y de la figura 55 derecha, representan los errores ocasionados por la compensación hecha al juego mecánico durante el maquinado de la geometría propuesta. El error relativo obtenido representa el porcentaje de desviación del valor medido respecto al valor real de la geometría.

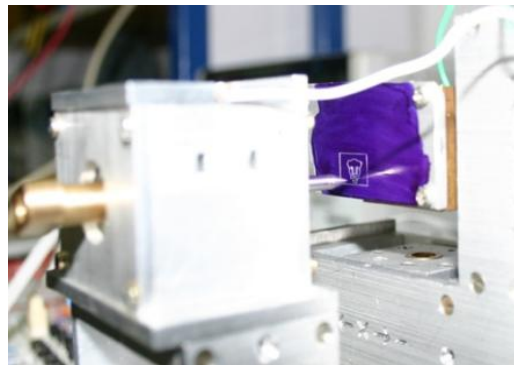


Figura 56. Proceso de maquinado del logo de los Pumas de 5mm.

Mediciones del logo de los pumas	Dimensiones teóricas [mm]	Dimensiones del maquinado [mm]	Juego mecánico [μ m]	Error relativo [%]
Longitud en X	4.86	4.830	---	0.61
Longitud en Y	4.96	4.947	---	0.26
a	---	---	Y = 11	---
b	---	---	Y = 20	---
C	X = 350×10^{-3}	X _c = 330×10^{-3}	X = 20	5.71
D	X = 350×10^{-3}	X _d = 370×10^{-3}	X = 20	5.71
E	---	---	X = 15	---
I	---	---	i _x = 12 i _y = 10	---
F	---	---	f _x = 15 f _y = 10	---

Tabla 14. Dimensiones, juego mecánico y error relativo del Puma de 5mm.

Por otro lado, en la figura 57 se muestran dos maquinados correspondientes un puma de 1.17x1.19mm. En el puma de la izquierda, el maquinado comienza en el punto denotado como **I** (cero de pieza) para terminar en el punto **F**. Mientras que el puma mostrado a la derecha, se maquinó en espejo, es decir, maquinando primero el lado derecho a partir del punto **le** para terminar en el punto **Fe** y comenzar con el lado izquierdo a partir del mismo punto **le**. El objetivo de cambiar el orden de maquinado de la figura 57 derecha, respecto de la izquierda, es mostrar que se pueden minimizar los errores ocasionados por la compensación del juego mecánico que aún existe en nuestro sistema.

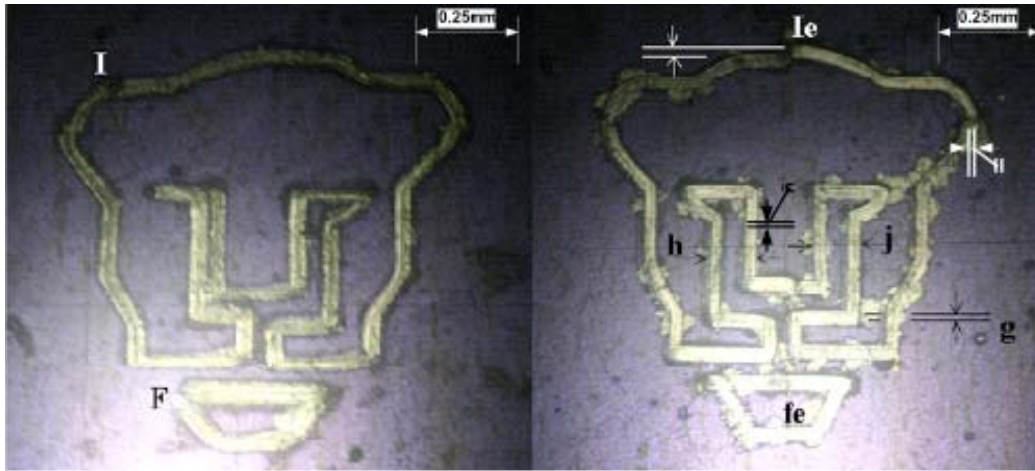


Figura 57. Maquinado continuo de un Puma de 1.2mm (izq); Puma maquinado en espejo de 1.2mm (der).

En la tabla 15 se aprecian las magnitudes g , h , j e le registradas de la figura 57 derecha. En esta tabla se observa que la diferencia entre las magnitudes h y j sólo es de $3\mu\text{m}$, menor que la magnitud de la figura de la izquierda de un valor de $59\mu\text{m}$. En la figura 58 se muestra una foto del proceso de maquinado de la figura 57 derecha.

Mediciones del logo de los pumas	Dimensiones teóricas [mm]	Dimensiones del maquinado [mm]	Juego mecánico [μm]	Error relativo [%]
Longitud en X	1.17	1.165	---	0.42
Longitud en Y	1.19	1.2	---	0.84
le	---	---	Y = 29	---
ñ	---	---	Y = 2	---
ll	---	---	Y = 24	---
g	---	---	Y = 3	---
h	X = 80×10^{-3}	X _h = 100×10^{-3}	X = 20	25
j	X = 80×10^{-3}	X _j = 97×10^{-3}	X = 17	21.2

Tabla 15. Dimensiones y juego Mecánico del Puma de 1.2mm.

Por último, en la figura 59 se muestran dos maquinados de un puma de dimensiones inferiores al milímetro: 970X990 μ m. El puma que se muestra a la izquierda fue maquinado en la parte central superior del área de trabajo, mientras que el puma de la derecha se maquinó en otra zona con el objetivo de verificar la calidad de maquinados en el espacio de trabajo de la MMH.

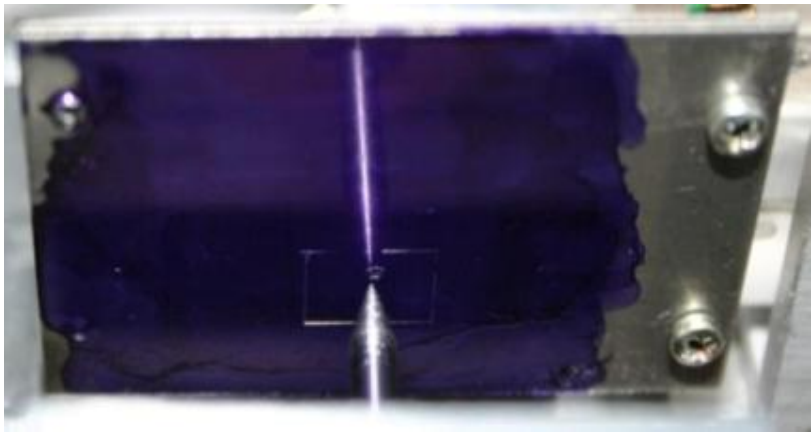


Figura 58. Proceso de maquinado del Puma de 1.2mm.

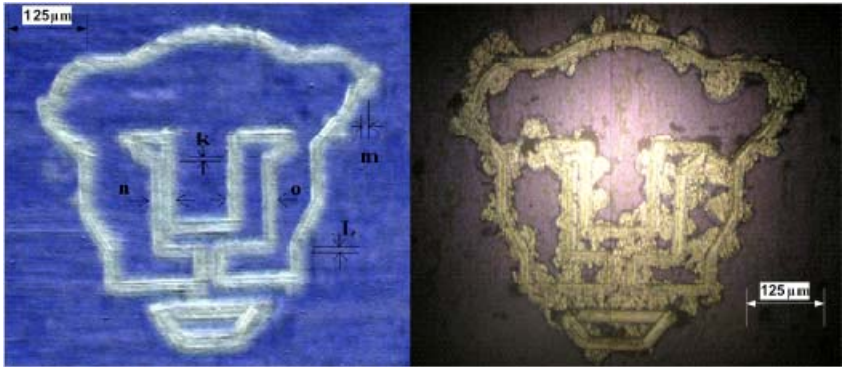


Figura 59. Dos maquinados de un Puma de 1mm.

Además, en la tabla 16 se muestran los resultados cuantitativos de las dimensiones marcadas en la figura 59 izquierda. Estas magnitudes dejan ver que los errores en la compensación del juego mecánico afectan de manera considerable a las geometrías con dimensiones inferiores al milímetro, resultando una enorme desventaja para el maquinado de micropiezas.

Mediciones del logo de los pumas	Dimensiones teóricas [μ m]	Dimensiones del maquinado [μ m]	Juego mecánico [μ m]	Error relativo [%]
Longitud en X	970	968	---	0.2
Longitud en Y	990	994	---	0.4
k	---	---	Y = 6	---
L	---	---	Y = 5	---
m	---	---	X = 29	---
n	X = 70	X _n = 45	X = 25	35.7
o	X = 70	X _o = 103	X = 33	47.1

Tabla 16. Dimensiones y Juego Mecánico del Puma de 1mm.

CONCLUSIONES.

Se analizaron y evaluaron tres algoritmos de interpolación (DDA, Escalera y DSM) para el desarrollo de geometrías con trayectorias rectas y circulares. Los métodos de interpolación seleccionados fueron: el algoritmo por DDA para segmentos rectilíneos, y el algoritmo por búsqueda directa (DSM) para trayectorias circulares, ya que éste último resultó el de menor número de iteraciones para desarrollar circunferencias, menor error radial teórico (70nm), mejor simulación, y aunque el tiempo de cada iteración es 2.7 veces el tiempo ejecutado por el método de escalera (3.2 μ s), compensa este defecto con el número de iteraciones realizadas.

El sistema de control implementado interpreta un total de 13 códigos para CNC, 8 códigos G relativos a tareas tales como: movimientos rápidos (G00 y G28), interpolación lineal y circular (G01 y G02, G03), temporizaciones (G04) y referencias (G54 y G90). Mientras que los códigos M fueron 5, relativos a: giro del husillo (M03 y M04), paro del husillo (M05), cambio de herramienta (M06) y fin de programa (M30). Estos comandos permiten operar una MMH de primera generación en modalidad de fresadora.

También se justificó la tendencia a implementar gradualmente características propias de sistemas CNC a los sistemas de control desarrollados por el GMM, esto con el fin de simplificar y mejorar sus métodos de programación para la manufactura de micropiezas.

Por otra parte, se propusieron e implementaron soluciones a ciertos inconvenientes que presenta el sistema de control basado en *la interfaz computacional para el control de Micromáquinas-Herramienta* [Yáñez, 2007], como lo son: las interrupciones generadas por el sistema operativo del equipo de cómputo empleado. Para este inconveniente se propuso el uso de un dispositivo de control externo a la PC, siendo propiamente el Microcontrolador PIC18F8720, del cual se utilizaron 16 pines de E/S repartidos en 3 puertos (D, E y J), un módulo USART para la comunicación serial, una interrupción para recepción de datos y 47Kb de memoria programa. Con esto también se da paso a controlar más de cuatro grados de libertad sin la necesidad de emplear otro equipo de cómputo.

Se programó una interfaz en *VisualBasic* que permite operar la MMH en cuatro modos: por *joystick*, por pasos, por línea de código, y por programa CNC. Además, esta interfaz registra tres tipos de coordenadas: unas relativas al cero de máquina, otras al cero de pieza y las coordenadas del cero de pieza. También indica el avance, las RPM y la frecuencia a la que operan los actuadores de la MMH. Esto último lo hace de acuerdo a una previa caracterización del sistema, ya que como se pudo apreciar durante el desarrollo de este trabajo, no se cuenta con elementos de realimentación para cerrar un lazo de control.

Se realizaron cuatro tipos de pruebas: las primeras fueron al *hardware* electrónico, enseguida se realizaron maquinados de figuras simples para la

compensación del juego mecánico, después maquinados con interpolaciones circulares, y por último la ejecución de programas en código estándar G y M.

Con el maquinado de figuras simples se obtuvo una magnitud promedio de juego mecánico por compensar en el eje X de $204\mu\text{m}$, mientras que en el eje Y resultaron $178\mu\text{m}$. Por otro lado, en los maquinados con interpolaciones en dos ejes (circunferencias), se concluye un error promedio de $\pm 77\text{nm}$. Además, las magnitudes E y P de las circunferencias (*a*, *b* y *c*) maquinadas demuestran que aún existen errores por compensar, de lo cual se concluye que el juego mecánico varía en toda el área de trabajo. El error correspondiente a estos errores por compensar es de $\pm 33\mu\text{m}$ en el eje X, y $\pm 27\mu\text{m}$ en el eje Y.

De la ejecución de programas en código estándar, se concluyen buenos resultados teniendo un error relativo máximo en el eje X de 0.9% y 1.86% en el eje Y respecto a sus medidas de diseño en CAD.

Por último, con este sistema de control se demostró que el maquinado en espejo de las figuras (que su geometría permita) puede minimizar los errores ocasionados por el juego mecánico.

REFERENCIAS.

[Atila y Jesse, 1993] Atila Ertas, Jesse C. Jones. *The Engineering Design Process*. Department of Mechanical Engineering Texas Tech University. Editorial Jhon Wiley & Sons, Inc. 1993.

[Caballero et al, 2007] A. Caballero Ruiz, H. Silva López, L. Ruiz Huerta. *Desarrollo de un sistema de control para una micromáquina herramienta basado en código estándar para CNC*. 8vo. Congreso Iberoamericano de Ingeniería Mecánica. Cusco, Perú. Octubre, 2007.

[Caballero y Ruiz, 2005] Alberto Caballero R. Leopoldo Ruiz H. *Apuntes de diseño y construcción de prototipos*. Programa de estudios de maestría, periodo escolar 2005-2. Ciudad Universitaria.

[Caballero, 2000] Alberto Caballero Ruiz. *Caracterización de un microcentro de maquinado de bajo costo*. División de estudios de posgrado, Facultad de Ingeniería, 2000. Tesis de maestría. Ingeniería mecánica.

[Caballero, 2005] Alberto Caballero Ruiz. *Metodología de evaluación para microequipo*. División de estudios de posgrado, Facultad de Ingeniería, 2005. Tesis de doctorado. Área mecánica.

[Escalona, 2002] Iván Escalona. *Diseño y manufactura asistidos por computadora. Introducción al CNC*. Ingeniería industrial, UPIICSA-IPN, México D.F. 2002. www.monografias.com

[Ferré, 1999] Ferré Masip R. *Como programar un control numérico*. Editorial Alfaomega. México, D.F. 1999. pags 155.

[German, 2007] German Herrera Granados. *Aplicación de sistemas CAD y CAM para el desarrollo de una micromáquina herramienta*. Facultad de Ingeniería. Ciudad Universitaria. Julio, 2007. Tesis de Licenciatura en Ingeniería mecánica.

[Jiménez, 2002] Ricardo Jiménez. *Control Numérico por Computadora (CNC)*. Ingeniería de Manufactura, 2002.

[Koren y Masory, 1982] Y, Koren. O, Masory. *Reference - Word Circular Interpolators for CNC Systems*. Faculty of Mechanical Engineering, Technion-Israel Institute of Technology, Haifa, Israel. Transaction of the ASME, Vol.104, pag. 400, November 1982.

[Koren, 1976] Y, Koren. *Interpolator for a Computer Numerical Control System*. IEEE Trans. On Computer, Vol. 25, No. 1, Jan. 1976, pp. 32 – 37.

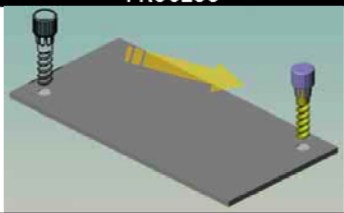
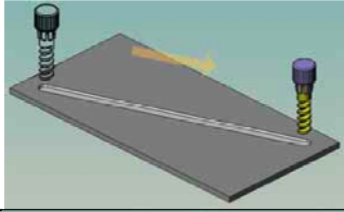
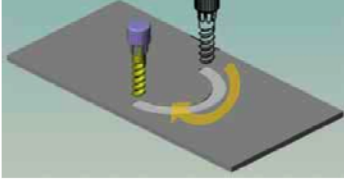
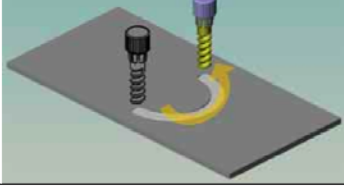
[Koren, 1979] Y, Koren. *Design of Computer Control for Manufacturing Systems*. Faculty of Mechanical Engineering, Technion - Israel Institute of Technology, Haifa, Israel. Transactions of the ASME, Vol. 101, pag.326 August 1979.

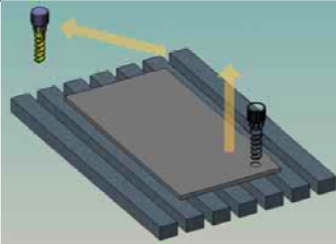
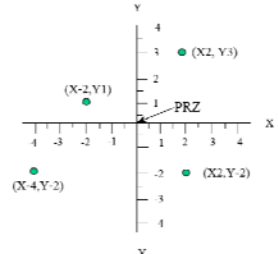
- [Koren, 1981] Y. Koren, O. Masory. *Reference - Pulse Circular Interpolators for CNC Systems*. Faculty of Mechanical Engineering, Technion-Israel Institute of Technology, Haifa, Israel. *Journal of Engineering for Industry*, Vol. 103, pag. 131, February 1981.
- [Koren, 2003] Y. Koren. *Computer Control of Manufacturing Systems*. Technion Israel Institute of Technology. Editorial McGraw-Hill, Inc. México, 2003.
- [Kussul et al, 1996] E. Kussul, D. Rachkovskij, T. Baydik and S. Talayev 1996. *Micromechanical engineering: a basis for the low cost manufacturing of mechanical microdevices using microequipment*. J. Micromech. Microeng.
- [Kussul et al, 2002] Kussul, E., Baydik, T., Ruiz, L., Caballero, A., Velazco, G. *Development of Low-cost Microequipment*. En 2002 IEEE International Symposium on Micromechatronics and Human Science. Nagoya Municipal Industrial Research Institute (Eds.), Nagoya, Japan.
- [Kussul y Kasatkina, 1998] E. M. Kussul, L. M. Kasatkina. *Los problemas de desarrollo de máquinas herramienta controladas con computadora para fábricas micro mecánicas sobremesa. // Sistemas de control y computadoras*. 1998, N5, V.28, pp 32-39.
- [Okazaki, 2000] Okazaki, Kitahara. *Micro-machine Tool to machine micro-parts*, Proc. ASPE Annual meeting, 2000.
- [Reséndiz, 2005] José Reséndiz Sánchez. *Investigación del sistema de control de una MMH y propuesta de optimización del sistema*. Facultad de Ingeniería. Ciudad Universitaria, Marzo, 2005. Tesis de Licenciatura en Ingeniería mecánica electricista.
- [Ruiz y Caballero, 2006] Leopoldo Ruiz Huerta, Alberto Caballero Ruiz. *Presentación del GMM, 2006-2007, CCADET, UNAM*.
- [Ruiz, 2000] Leopoldo Ruiz Huerta. *Diseño y construcción de un microcentro de maquinado de bajo costo*. División de estudios de posgrado, Facultad de Ingeniería, 2000. Tesis de maestría, Ingeniería mecánica.
- [Ruiz, 2005] Leopoldo Ruiz Huerta. *Desarrollo de microequipo para células de manufactura*. División de estudios de posgrado, Facultad de Ingeniería, 2005. Tesis de doctorado. Área mecánica.
- [Smid, 2003] Smid P. *A Comprehensive Guide to Practical CNC Programming. CNC PROGRAMMING HANDBOOK*. Editorial Industrial Press, Inc. Segunda edición. Nueva York, EUA, 2003.
- [Tanaka, 2001] M. Tanaka. *Development of desktop machining microfactory*. Riken Review. No. 34, Abril, 2001. Japón.

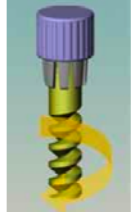
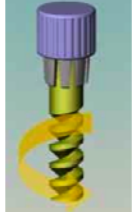
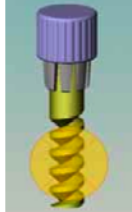
[Yáñez, 2007] Alfredo Yáñez Soto. *Desarrollo de una interfaz computacional para el control de Micromáquinas-Herramienta*. Facultad de Ingeniería. Ciudad Universitaria, Marzo, 2007. Tesis de Licenciatura en Ingeniería en computación.

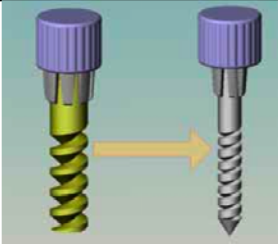
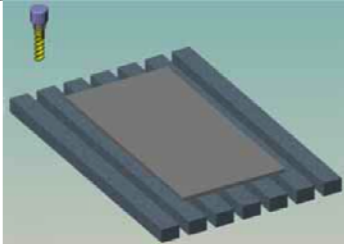
ANEXOS

A.1 Descripción y sintaxis de los códigos.

CÓDIGOS "G" y SINTAXIS	DESCRIPCIÓN	PROCESO
G00X±____Y±____;	Este comando desplaza la herramienta al punto indicado por las coordenadas "X" y "Y" (de 5 dígitos enteros y 2 decimales), a la velocidad máxima de desplazamiento de la máquina y en línea recta. En su recorrido no se efectúa arranque de material.	
G01X±____Y±____F____;	El comando genera el movimiento simultáneo de uno ó dos ejes, desde su posición actual hasta alcanzar la posición final, especificada por la coordenadas "X" y "Y" (de 5 dígitos enteros y 2 decimales) mediante una trayectoria rectilínea, y a una velocidad de trabajo determinada por el avance "F" (de 2 dígitos).	
G02X±____Y±____I±____J±____F____;	Este comando se utiliza para efectuar un movimiento en arco de circunferencia, en sentido de las manecillas del reloj, a partir del punto donde se encuentra hasta el punto final indicado por las coordenadas "X" y "Y" (de 5 dígitos enteros y 2 decimales), y tomando como referencia el centro del arco denotado por las coordenadas "I" y "J" (de 5 dígitos). La velocidad de desplazamiento estará sujeta al avance "F" (de 2 dígitos).	
G03X±____Y±____I±____J±____F____;	Comando que al igual que G02, se utiliza para efectuar un movimiento en arco de circunferencia, en sentido contrario al de las manecillas del reloj, a partir del punto donde se encuentra hasta el punto final indicado por las coordenadas "X" y "Y" (de 5 dígitos enteros y 2 decimales), y tomando como referencia el centro del arco denotado por las coordenadas "I" y "J" (de 5 dígitos). La velocidad de desplazamiento estará sujeta al avance "F" (de 2 dígitos).	

G04P_;	Con esta instrucción se obtiene una pausa durante un tiempo especificado por "P" (en segundos y de 2 dígitos). Una vez transcurrido este tiempo la máquina continúa automática el proceso que sigue.	
G28X0Y0; ó G28Z0;	Este comando permite trasladar automáticamente la herramienta a la posición cero de máquina en el(los) eje(s) correspondiente(s). Sus movimientos describen trayectorias rectilíneas a máxima velocidad. Generalmente es empleado en los cambios de herramienta.	
G54;	Reserva una localidad de memoria en la cual se almacenará el cero de pieza, y a la cual se referirán todos los datos de posición de la pieza a manufacturar.	
G90;	Establece al sistema con dimensiones en modo absoluto. Este modo utiliza como punto de referencia el cero de pieza almacenado por G54.	

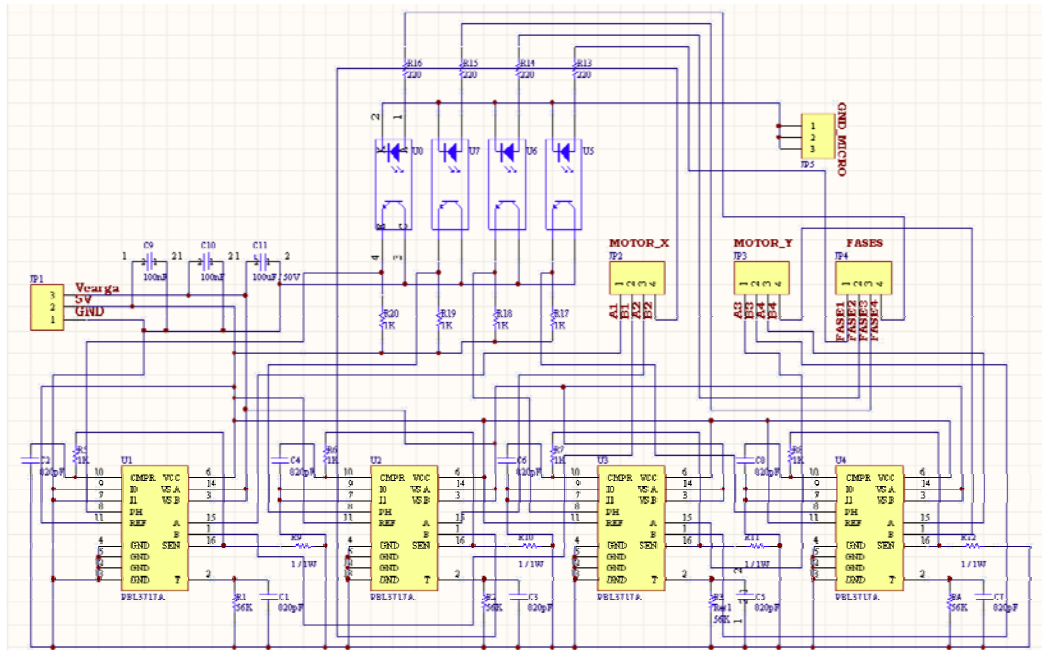
CÓDIGOS "M" y SINTAXIS	DESCRIPCIÓN	PROCESO
M03S_;	Comando que activa el giro del husillo principal en sentido horario. La velocidad del husillo, estará determinada por las revoluciones dadas en la misma sintaxis del comando (4 dígitos).	
M04S_;	Este comando activa el giro del husillo principal en sentido contrario a las manecillas del reloj. La velocidad del husillo, estará determinada por las revoluciones dadas en la misma sintaxis del comando (4 dígitos).	
M05;	Comando que detiene el giro del husillo. Anula las funciones M03 y M04.	

M06T_;	Procedimiento que prepara a la máquina para llevar a cabo un cambio de herramienta, ubicada en la posición identificada con el número proporcionado en la sintaxis (2 dígitos), del almacén del magazine de herramientas.	
M30;	Termina y reinicia el programa CNC de vuelta a la secuencia No.1.	

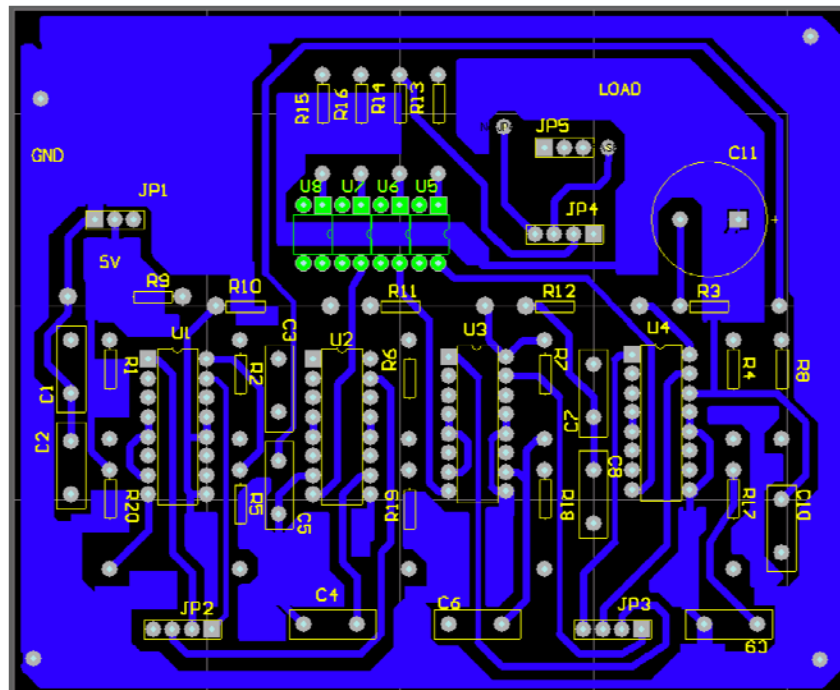
A.2 Comandos para activar modos de operación en la interfaz computacional.

Modalidad de operación	Sintaxis de la cadena de control
Por joystick	JX±; (modo joystick/eje/sentido/fin de línea)
Por joystick (movimiento simultáneo de dos o más ejes)	&X±Y±Z±; (más de un eje/eje1/sentido/.../eje3/sentido/fin de línea)
Por pasos	PX±1000; (modo pasos/eje/sentido/cifra cuatro dígitos/fin de línea)
Por línea y Por programa	En este modo la cadena de control es ocupada por cualquiera de los comandos descritos en las tablas 11 y 12 de esta misma sección.
Para mandar a HOME	H; (comando de home/fin de línea)
Para activar el ALTO	A; (comando de alto/fin de línea)
Para determinar el juego mecánico	BX200; (comando para modificar el juego mecánico/eje/cifra de tres dígitos/fin de línea)
Para activar el movimiento a una frecuencia determinada.	FX±250; (comando para activar el movimiento a una frecuencia /eje/sentido/fin de línea)
Para el incremento de velocidad	W; (comando para incrementar velocidad/fin de línea)
Para el decremento de velocidad	V; (comando para disminuir velocidad/fin de línea)
Para almacenar el cero de pieza	Q; (comando para almacenar cero de pieza/fin de línea)

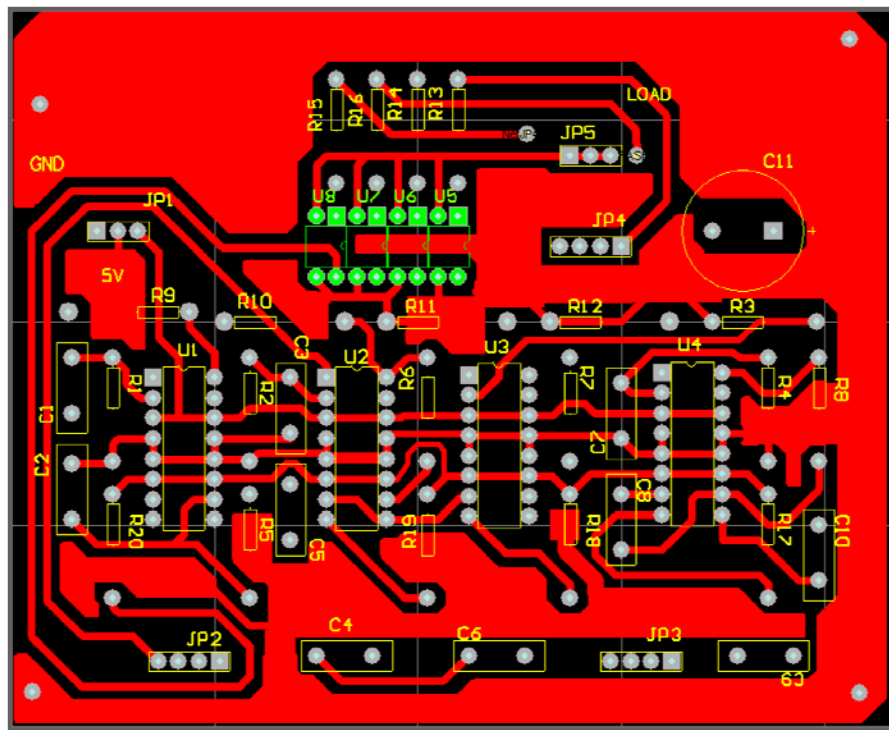
A.3 Diagrama y tarjeta electrónica de potencia.



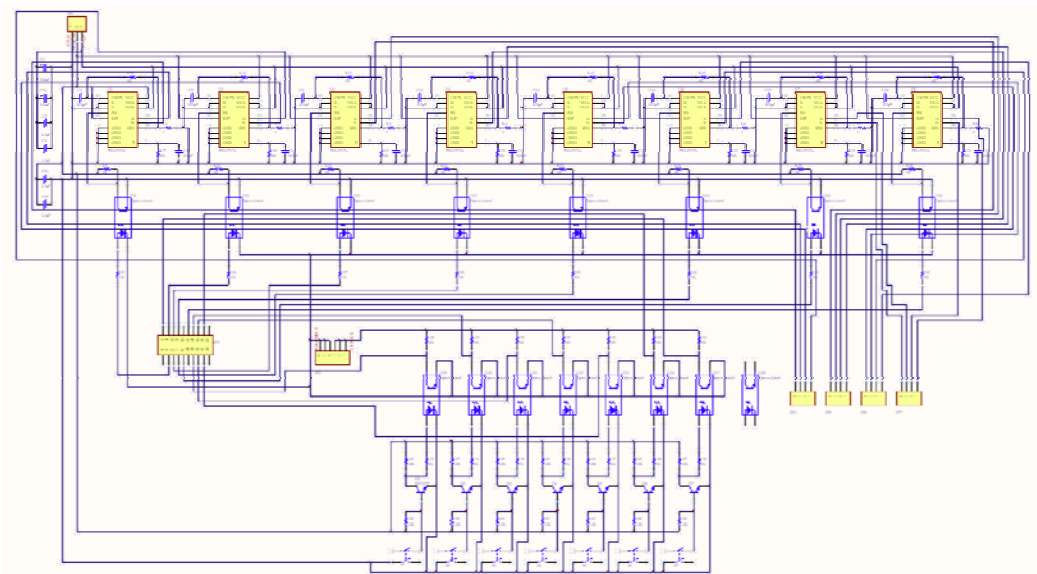
Esquemático del sistema de potencia para dos motores.



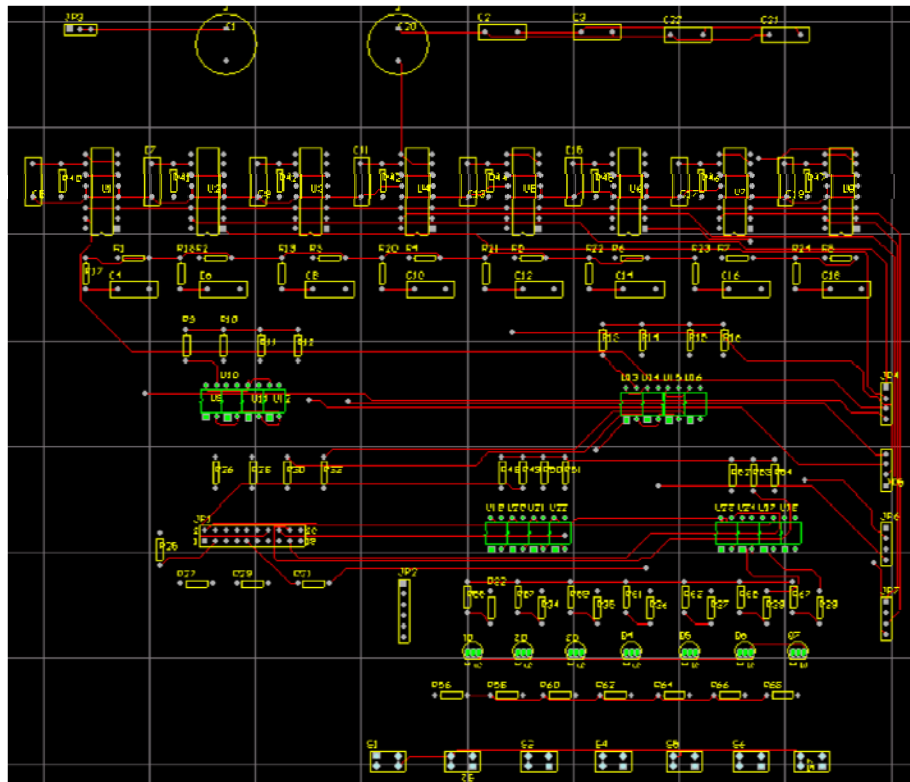
Cara inferior del circuito impreso para el sistema de potencia.



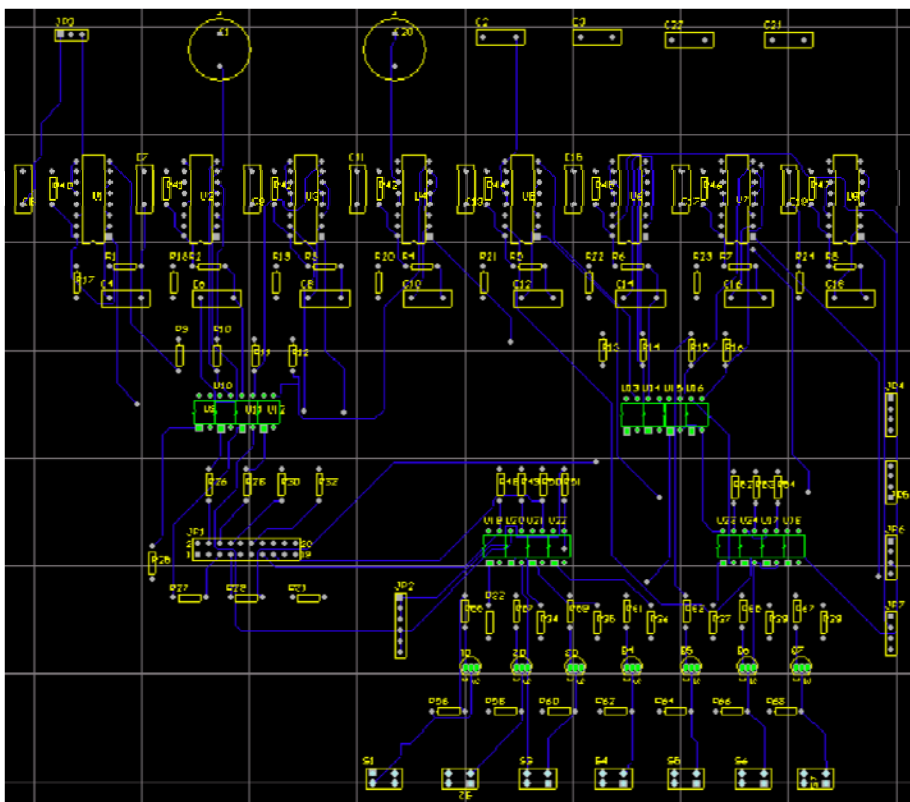
Cara posterior del circuito impreso para el sistema de potencia.



Esquemático del sistema de potencia para cuatro motores.



Cara posterior del circuito impreso para el sistema de potencia implementado.



Cara inferior del circuito impreso para el sistema de potencia implementado.

A.4 Código del programa de control grabado en el Microcontrolador PIC18F8720.

```
*****
*****
' * Name      : Programa TesisMaestria (resol=140nm) compensando
Backlash.BAS *
' * Author    : Ing. Héctor Hugo Silva López
*
' * Colaboración : Dr. Alberto Caballero Ruiz y Dr. Leopoldo Ruiz
Huerta.      *
' *           : All Rights Reserved
*
' * Date      : 01/03/2008
*
' * Version   : 1.0
*
' * Notes     : Desarrollo de un sistema de control para una
Micromáquina Herra-- *
' *           : mienta de primera generación basado en código
estándar para CNC.   *
*****
*****

' ***** CONFIGURACIÓN DEL PIC Y DECLARACIÓN DE VARIABLES *****
*****

Device 18F8720
XTAL = 20MHZ

Symbol TXEN = TXSTA1.5
Symbol SYNC = TXSTA1.4
Symbol BRGH = TXSTA1.2
Symbol SPEN = RCSTA1.7
Symbol CREN = RCSTA1.4
Symbol RC1IF = PIR1.5
Symbol RC1IE = PIE1.5
Symbol PEIE = INTCON.6
Symbol GIE = INTCON.7

Symbol PushBiniX = PORTE.4
Symbol PushBiniY = PORTE.5
Symbol PushBiniZ = PORTE.6
Symbol PushBTool = PORTE.7
Symbol PushBfinX = PORTE.3
Symbol PushBfinY = PORTE.2
Symbol PushBfinZ = PORTJ.1

' ***** DECLARANDO BANDERAS *****
*****

Dim joystick As Bit
Dim porpasos As Bit
Dim porfrecuencia As Bit
Dim home As Bit
Dim homex As Bit
Dim mascarahomex As Bit
Dim homey As Bit
Dim mascarahomey As Bit
Dim homez As Bit
Dim mascarahomez As Bit
Dim ejex As Bit
Dim mascaraejex As Bit
Dim ejey As Bit
```

```

Dim mascaraejey As Bit
Dim ezej As Bit
Dim mascaraejez As Bit
Dim ejeh As Bit
Dim coordenadai As Bit
Dim mascaracoordi As Bit
Dim coordenadaj As Bit
Dim mascaracoordj As Bit
Dim sentidox As Bit
Dim sentidoy As Bit
Dim sentidoz As Bit
Dim sentidoxant As Bit
Dim sentidoyant As Bit
Dim sentidozant As Bit
Dim iniciodecomando As Bit
Dim partedecimal As Bit
Dim sentidoh As Bit
Dim sentidoi As Bit
Dim sentidoj As Bit
Dim cerodepieza As Bit
Dim lineacodigog As Bit
Dim lineacodigom As Bit
Dim diopasoenx As Bit
Dim diopasoeny As Bit
Dim diopasoenz As Bit
Dim diopasoenh As Bit
Dim numerocodigo As Bit
Dim masdeuneje As Bit
Dim noavance As Bit
Dim sistemaabsoluto As Bit
Dim NoPausa As Bit
Dim Sangria As Bit
Dim mascaraSangria As Bit
Dim backlash As Bit
Dim JgoMecanico As Bit
Dim solouneje As Bit
Dim Positivo As Bit

```

```

! ***** DECLARANDO VARIABLES
*****

```

```

Dim RESOL As Float
Dim Fcia As DWord
Dim Pdo As DWord
Dim Xm As Float
Dim Ym As Float
Dim Zm As Float
Dim Xp As Float
Dim Yp As Float
Dim Zp As Float
Dim Xcp As Float
Dim Ycp As Float
Dim Zcp As Float
Dim dummy10 As Byte
Dim dummy11 As Byte
Dim dummy12 As Byte
Dim POSICION As DWord
Dim ESTADOx As Byte
Dim ESTADOy As Byte
Dim dummy5 As Byte
Dim dummy6 As Byte
Dim ValorCinco As DWord

```

```

Dim ValorCuatro As DWord
Dim ESTADOz As Byte
Dim ValorDos As DWord
Dim ESTADOh As Byte
Dim Xdda As Float
Dim COMBINACION As Byte
Dim Ydda As Float
Dim DATOenviado As Byte
Dim PPR As DWord
Dim DATOentrada As Byte
Dim PX As DWord
Dim CadenadeControl[55] As Byte
Dim PY As DWord
Dim n As Byte
Dim HIPO As DWord
Dim cnt5 As Byte
Dim QX As DWord
Dim cnt4 As Byte
Dim QY As DWord
Dim cnt3 As Byte
Dim ValorTres As DWord
Dim cnt2 As Byte
Dim ContadorX As DWord
Dim NoCodigoG As Byte
Dim ContadorY As DWord
Dim dummy16 As Byte
Dim Zdda As Float
Dim dummy1 As Byte
Dim PZ As DWord
Dim dummy9 As Byte
Dim Idda As Float
Dim CUADRANTE As Byte
Dim Jdda As Float
Dim NoCodigoM As Byte
Dim Radio As DWord
Dim Xo As DWord
Dim dummy13 As Byte
Dim dummy14 As Byte
Dim dummy15 As Byte
Dim Yo As DWord
Dim dummy2 As Byte
Dim dummy3 As Byte
Dim dummy4 As Byte
Dim dummy7 As Byte
Dim dummy8 As Byte
Dim D As DWord
Dim D1 As DWord
Dim D2 As DWord
Dim D3 As DWord
Dim DeltaX As DWord
Dim DeltaY As DWord
Dim PASOS As DWord
Dim JgoMecanicoX As DWord
Dim JgoMecanicoY As DWord
Dim JgoMecanicoZ As DWord
Dim PXdec As Float
Dim PYdec As Float
Dim PZdec As Float
Dim Xodec As Float
Dim Yodec As Float
Dim Pausa As Word

```

```

Dim Avance As Float
Dim CUADRANTEFINAL As Byte

'***** ESTABILIZANDO AL MICROCONTROLADOR PIC *****
DelayMS 600

ON_INTERRUPT GoTo RECEPCION

GoTo INICIO

'***** RUTINA DE RECEPCIÓN DE CADENA DE CONTROL *****
RECEPCION:      btfss    RC1IF
                retfie   fast
                movf     RCREG1,0
                movwf    DAT0entrada

                If DAT0entrada = 59 Then                                'Fin
de linea? (;)
                    CadenadeControl[n] = DAT0entrada
                    Clear n
                    movf    DAT0entrada,0
                    movwf   TXREG1
                    GoTo    MODALIDAD
                Else
                    CadenadeControl[n] = DAT0entrada
                    Inc n
                    movf    DAT0entrada,0
                    movwf   TXREG1
                    retfie fast
                EndIf

'***** DESCIFRANDO EL MODO DE OPERACIÓN *****
MODALIDAD:      Select CadenadeControl[n]
                Case 65
'Modalidad de Alto (A)
                    Clear n
                    GoTo ALTO
                Case 66
'Determinando el Juego Mecánico (B)
                    Set backlash
                    Inc n
                    GoTo OBTENEJE
                Case 74
'Modalidad de avance por Joystick (J)
                    Set joystick
                    Fcia = 170
                    Pdo = 1000000 / Fcia
                    Inc n
                    GoTo OBTENEJE
                Case 38
'Modalidad de avance por Joystick con MAS DE UN EJE (&)
                    Set masdeuneje
                    Inc n
                    Fcia = 170
                    Pdo = 1000000 / Fcia
                    GoTo OBTENEJE
                Case 80
'Modalidad de avance por Pasos (P)

```

```

        Set porpasos
        Inc n
        Fcia = 170
        Pdo = 1000000 / Fcia
        GoTo OBTENEJE
    Case 70
        'Modalidad de encendido por Frecuencia (F)
        Set porfrecuencia
        Inc n
        GoTo OBTENEJE
    Case 72
        'Modalidad de regreso a Home (H)
        Inc n
        Set home
        Set homez
        Set ejez
        Set mascaraejez
        Set sentidoz
        Clear sentidox
        Clear sentidoy
        Fcia = 170
        Pdo = 1000000 / Fcia
        GoTo FINDELINEA
    Case 87
        para incrementar la velocidad (W)
        Clear n
        Fcia = Fcia + 4
        If Fcia > 500 Then
            Fcia = 500
            Pdo = 1000000 / Fcia
            GoTo VERIFICASECUENCIA
        EndIf
        Pdo = 1000000 / Fcia
        GoTo VERIFICASECUENCIA
    Case 86
        para disminuir la velocidad (V)
        Clear n
        Fcia = Fcia - 4
        If Fcia < 16 Then
            Fcia = 16
            Pdo = 1000000 / Fcia
            GoTo VERIFICASECUENCIA
        EndIf
        Pdo = 1000000 / Fcia
        GoTo VERIFICASECUENCIA
    Case 81
        para almacenar el cero de pieza (Q)
        Clear solouneje
        Clear n
        Clear lineacodigog
        Set cerodepieza
        Xcp = Xm
        Ycp = Ym
        Zcp = Zm
        Clear Xp
        Clear Yp
        Clear Zp
        retfie fast
    Case 71
        'Modalidad por línea de códigoG (G)
        Set lineacodigog

```

```

        Set numerocodigo
        Inc n
        GoTo OBTEN2DIG
    Case 77
        'Modalidad por línea de códigoM (M)
        Set lineacodigom
        Set numerocodigo
        Inc n
        GoTo OBTEN2DIG
    Case Else
        retfie fast
    EndSelect
    retfie fast

'***** RUTINA PARA OBTENER LOS EJES *****
OBTENEJE:    Select CadenadeControl[n]
              Case 88
                  'Eje "x" (X)
                      Inc n
                      Set ejex
                      Set mascaraejex
                      If backlash = 1 Then GoTo OBTEN3DIG
                      If solouneje = 1 Then
                          Clear solouneje
                      Else
                          Set solouneje
                      EndIf
                      GoTo SENTIDOGIRO
              Case 89
                  'Eje "y" (Y)
                      Inc n
                      Set ejey
                      Set mascaraejey
                      If backlash = 1 Then GoTo OBTEN3DIG
                      If solouneje = 1 Then
                          Clear solouneje
                      Else
                          Set solouneje
                      EndIf
                      GoTo SENTIDOGIRO
              Case 90
                  'Eje "z" (Z)
                      Inc n
                      Set ejez
                      Set mascaraejez
                      If backlash = 1 Then GoTo OBTEN3DIG
                      GoTo SENTIDOGIRO
              Case 72
                  'Husillo (H)
                      Inc n
                      Set ejeh
                      GoTo SENTIDOGIRO
              Case 73
                  'Coordenada "i" (I)
                      Inc n
                      Set coordenadai
                      Set mascaracoordi
                      GoTo SENTIDOGIRO
              Case 74
                  'Coordenada "j" (J)

```

```

        Inc n
        Set coordenadaj
        Set mascaracoordj
        GoTo SENTIDOGIRO
    Case Else
        If NoCodigoG = 4 Then GoTo PARAMETROS
        If NoCodigoG = 1 Then GoTo PARAMETROS
        solouneje = 1
        GoTo FINDELINEA
    EndSelect

***** RUTINA PARA OBTENER SENTIDO DE GIRO *****
SENTIDOGIRO:      Select CadenadeControl[n]
                  Case 45
'Sentido antihorario (-)
                    If ejex = 1 Then
                        Inc n
                        Positivo = 0
                        If joystick = 1 Then sentidox = 0
                        If masdeuneje = 1 Then sentidox = 0
                        If porpasos = 1 Then sentidox = 0
                        If porfrecuencia = 1 Then sentidox = 0
                    ElseIf ejey = 1 Then
                        Inc n
                        Positivo = 0
                        If joystick = 1 Then sentidoy = 0
                        If masdeuneje = 1 Then sentidoy = 0
                        If porpasos = 1 Then sentidoy = 0
                        If porfrecuencia = 1 Then sentidoy = 0
                    ElseIf ejez = 1 Then
                        Inc n
                        Positivo = 0
                        If joystick = 1 Then sentidoz = 0
                        If masdeuneje = 1 Then sentidoz = 0
                        If porpasos = 1 Then sentidoz = 0
                        If porfrecuencia = 1 Then sentidoz = 0
                    ElseIf ejeh = 1 Then
                        Inc n
                        Positivo = 0
                        If joystick = 1 Then sentidoh = 0
                        If masdeuneje = 1 Then sentidoh = 0
                        If porpasos = 1 Then sentidoh = 0
                        If porfrecuencia = 1 Then sentidoh = 0
                    ElseIf coordenadai = 1 Then
                        Inc n
                        Positivo = 0
                    ElseIf coordenadaj = 1 Then
                        Inc n
                        Positivo = 0
                    Else
                        retfie fast
                    EndIf
                Case Else
'Sentido horario (+)
                    If ejex = 1 Then
                        Positivo = 1
                        If joystick = 1 Then sentidox = 1
                        If masdeuneje = 1 Then sentidox = 1
                        If porpasos = 1 Then sentidox = 1
                        If porfrecuencia = 1 Then sentidox = 1

```

```

ElseIf ejey = 1 Then
    Positivo = 1
    If joystick = 1 Then sentidoy = 1
    If masdeuneje = 1 Then sentidoy = 1
    If porpasos = 1 Then sentidoy = 1
    If porfrecuencia = 1 Then sentidoy = 1
ElseIf ejez = 1 Then
    Positivo = 1
    If joystick = 1 Then sentidoz = 1
    If masdeuneje = 1 Then sentidoz = 1
    If porpasos = 1 Then sentidoz = 1
    If porfrecuencia = 1 Then sentidoz = 1
ElseIf ejeh = 1 Then
    Positivo = 1
    If joystick = 1 Then sentidoh = 1
    If masdeuneje = 1 Then sentidoh = 1
    If porpasos = 1 Then sentidoh = 1
    If porfrecuencia = 1 Then sentidoh = 1
ElseIf coordenadai = 1 Then
    Positivo = 1
ElseIf coordenadaj = 1 Then
    Positivo = 1
Else
    retfie fast
EndIf
EndSelect

```

***** *RUTINA PARA VERIFICAR EL FIN DE LA LÍNEA* *****

```

FINDELINEA:      If CadenadeControl[n] = 59 Then
'Fin de línea? (;)
    If joystick = 1 Then
        Clear n
        If ejex = 1 Then
            GoSub CompensaJgoMecanicoX
            GoTo VERIFICASECUENCIA
        ElseIf ejey = 1 Then
            GoSub CompensaJgoMecanicoY
            GoTo VERIFICASECUENCIA
        ElseIf ejez = 1 Then
            GoSub CompensaJgoMecanicoZ
            GoTo VERIFICASECUENCIA
        Else
            Clear n
            retfie fast
        EndIf
    ElseIf backlash = 1 Then
        Clear n
        Clear backlash
        GoTo ALTO
    ElseIf porfrecuencia = 1 Then
        Fcia = Fcia * 4
        Pdo = 1000000 / Fcia
        Clear n
        If ejex = 1 Then
            GoSub CompensaJgoMecanicoX
            GoTo VERIFICASECUENCIA
        ElseIf ejey = 1 Then
            GoSub CompensaJgoMecanicoY
            GoTo VERIFICASECUENCIA
        ElseIf ejez = 1 Then

```

```

        GoSub CompensaJgoMecanicoZ
        GoTo VERIFICASECUENCIA
    Else
        Clear n
        retfie fast
    EndIf
ElseIf porpasos = 1 Then
    Clear n
    If ejex = 1 Then
        GoSub CompensaJgoMecanicoX
        PASOS = ValorCuatro
        GoTo VERIFICASECUENCIA
    ElseIf ejey = 1 Then
        GoSub CompensaJgoMecanicoY
        PASOS = ValorCuatro
        GoTo VERIFICASECUENCIA
    ElseIf ejez =1 Then
        GoSub CompensaJgoMecanicoZ
        PASOS = ValorCuatro
        GoTo VERIFICASECUENCIA
    Else
        Clear n
        retfie fast
    EndIf
ElseIf masdeuneje = 1 Then
    If ejex = 1 Then
        GoSub CompensaJgoMecanicoX
        diopasoex = 1
        Clear ejex
        Clear mascaraejex
        If diopasoey = 1 Then
            Set ejey
            Set mascaraejey
            GoSub CompensaJgoMecanicoY
            diopasoey = 1
            Clear ejey
            Clear mascaraejey
        EndIf
        If diopasoenz = 1 Then
            Set ejez
            Set mascaraejez
            GoSub CompensaJgoMecanicoZ
            diopasoenz = 1
            Clear ejez
            Clear mascaraejez
        EndIf
        If diopasoex = 1 Then Set mascaraejex
        If diopasoey = 1 Then Set mascaraejey
        If diopasoenz = 1 Then Set mascaraejez
        Clear n
        GoTo SECUENCIA
    ElseIf ejey = 1 Then
        GoSub CompensaJgoMecanicoY
        diopasoey = 1
        Clear ejey
        Clear mascaraejey
        If diopasoex = 1 Then
            Set ejex
            Set mascaraejex
            GoSub CompensaJgoMecanicoX
            diopasoex = 1

```

```

        Clear ejex
        Clear mascaraejex
    EndIf
    If diopasoenz = 1 Then
        Set ejez
        Set mascaraejez
        GoSub CompensaJgoMecanicoZ
        diopasoenz = 1
        Clear ejex
        Clear mascaraejez
    EndIf
    If diopasoenx = 1 Then Set mascaraejex
    If diopasoeny = 1 Then Set mascaraejey
    If diopasoenz = 1 Then Set mascaraejez
    Clear n
    GoTo SECUENCIAIX
ElseIf ejez = 1 Then
    GoSub CompensaJgoMecanicoZ
    diopasoenz = 1
    Clear ejex
    Clear mascaraejez
    If diopasoenx = 1 Then
        Set ejex
        Set mascaraejex
        GoSub CompensaJgoMecanicoX
        diopasoenx = 1
        Clear ejex
        Clear mascaraejex
    EndIf
    If diopasoeny = 1 Then
        Set ejey
        Set mascaraejey
        GoSub CompensaJgoMecanicoY
        diopasoeny = 1
        Clear ejey
        Clear mascaraejey
    EndIf
    If diopasoenx = 1 Then Set mascaraejex
    If diopasoeny = 1 Then Set mascaraejey
    If diopasoenz = 1 Then Set mascaraejez
    Clear n
    GoTo SECUENCIAIX
Else
    Clear n
    retfie fast
EndIf
ElseIf homez = 1 Then
    Clear n
    Set iniciodecomando
    GoTo VERIFICACEROMAQUINA
ElseIf NoCodigoG = 0 Then
    If ejez = 1 Then
        Clear n
        Clear lineacodigog
        GoTo DDALINEALZ
    ElseIf solouneje = 1 Then
        If mascaraejex = 1 Then
            Set ejex
            Clear n
            Clear lineacodigog
            Clear iniciodecomando

```

```

        GoTo DDALINEALX
    ElseIf mascaraejey = 1 Then
        Set ejey
        Clear n
        Clear lineacodigog
        Clear iniciodecomando
        GoTo DDALINEALY
    Else
        Clear n
        retfie fast
    EndIf
Else
    Clear n
    Clear lineacodigog
    GoTo DDALINEAL
EndIf
ElseIf NoCodigoG = 1 Then
    If ejez = 1 Then
        Clear n
        Clear lineacodigog
        GoTo DDALINEALZ
    ElseIf solouneje = 1 Then
        If mascaraejex = 1 Then
            Set ejex
            Clear n
            Clear lineacodigog
            Clear iniciodecomando
            GoTo DDALINEALX
        ElseIf mascaraejey = 1 Then
            Set ejey
            Clear n
            Clear lineacodigog
            Clear iniciodecomando
            GoTo DDALINEALY
        Else
            Clear n
            retfie fast
        EndIf
    Else
        Clear n
        Clear lineacodigog
        GoTo DDALINEAL
    EndIf
ElseIf NoCodigoG = 28 Then
    If solouneje = 1 Then
        If mascaraejex = 1 Then
            Xdda = 0
            Set ejex
            Clear n
            Clear lineacodigog
            GoTo DDALINEALX
        ElseIf mascaraejey = 1 Then
            Ydda = 0
            Set ejey
            Clear n
            Clear lineacodigog
            GoTo DDALINEALY
        Else
            Clear n
            retfie fast
        EndIf
    EndIf

```

```

ElseIf ejez = 1 Then
    Zdda = 0
    Clear n
    Clear lineacodigog
    GoTo DDALINEALZ
Else
    Clear n
    Clear lineacodigog
    GoTo DDALINEAL
EndIf
ElseIf NoCodigoG = 2 Then
    Clear n
    Clear lineacodigog
    GoTo DSMCIRCULAR
ElseIf NoCodigoG = 3 Then
    Clear n
    Clear lineacodigog
    GoTo DSMCIRCULAR
ElseIf NoCodigoG = 54 Then
    Clear solouneje
    Clear n
    Clear lineacodigog
    Set cerodepieza
    Xcp = Xm
    Ycp = Ym
    Zcp = Zm
    Xp = 0
    Yp = 0
    Zp = 0
    retfie fast
ElseIf NoCodigoG = 90 Then
    Clear solouneje
    Clear n
    Clear lineacodigog
    Set sistemaabsoluto
    retfie fast
ElseIf NoCodigoG = 4 Then
    Clear n
    Clear lineacodigog
    GoTo EjecutaPausa
ElseIf NoCodigoM = 30 Then
    Clear n
    Clear lineacodigom
    Set home
    Set iniciodecomando
    Set homez
    Set ejez
    Set mascaraejez
    Set sentidoz
    Clear sentidox
    Clear sentidoy
    Fcia = 170
    Pdo = 1000000 / Fcia
    GoTo VERIFICACEROMAQUINA
Else
    Clear n
    retfie fast
EndIf
ElseIf masdeuneje = 1 Then
    If ejex = 1 Then
        diopasoenx = 1
    
```

```

        Clear ejex
        Clear mascaraejex
        GoTo OBTENEJE
    ElseIf ejey = 1 Then
        diopasoeny = 1
        Clear ejey
        Clear mascaraejey
        GoTo OBTENEJE
    ElseIf ejez = 1 Then
        diopasoenz = 1
        Clear ejez
        Clear mascaraejez
        GoTo OBTENEJE
    Else
        Clear n
        retfie fast
    EndIf
    ElseIf porpasos = 1 Then
        GoTo OBTEN4DIG
    ElseIf porfrecuencia = 1 Then
        GoTo OBTEN3DIG
    ElseIf backlash = 1 Then
        GoTo OBTEN3DIG
    ElseIf lineacodigog = 1 Then
        GoTo OBTEN5DIG
    ElseIf lineacodigom = 1 Then
        GoTo OBTEN4DIG
    Else
        Clear n
        retfie fast
    EndIf

'***** RUTINA PARA OBTENER LOS PARÁMETROS DE
CONTROL *****
PARAMETROS:      Select CadenadeControl[n]
                  Case 70
'Declarar avance (F)
                        Inc n
                        Set noavance
                        GoTo OBTEN2DIG
                  Case 80
'Determinar pausa (P)
                        Inc n
                        Set NoPausa
                        GoTo OBTEN2DIG
                  Case 46
'Verificar si es un punto (.)
                        Inc n
                        GoTo VERIFICADIGDECIMAL
                  Case 83
'Establecer velocidad del husillo (S)
                        Clear n
                        retfie fast
                  Case 84
'Determinar el número de herramienta (T)
                        Clear n
                        retfie fast
                  Case Else
                        GoTo FINDELINEA
    EndSelect

```

```

***** RUTINA PARA VERIFICAR LA SECUENCIA DE PASO *****
*****

```

```

VERIFICASECUENCIA:      If ejex = 1 Then
                        If sentidox = 1 Then
                            If PushBfinX = 0 Then GoTo ALTO
                        Else
                            If PushBiniX = 0 Then
                                If homex = 1 Then GoTo
HabilitaHomeY
                                GoTo ALTO
                            EndIf
                        EndIf
                        diopasoex = 1
                        GoTo SECUENCIAX
ElseIf ejey = 1 Then
    If sentidoy = 1 Then
        If PushBfinY = 0 Then GoTo ALTO
    Else
        If PushBiniY = 0 Then
            If homey = 1 Then GoTo
HabilitaSangriaZ
            GoTo ALTO
        EndIf
    EndIf
    diopasoey = 1
    GoTo SECUENCIAX
ElseIf ejez = 1 Then
    If sentidoz = 1 Then
        If PushBiniZ = 0 Then
            If homez = 1 Then GoTo
HabilitaHomeX
            GoTo ALTO
        EndIf
    Else
        If PushBTool = 0 Then GoTo ALTO
        If PushBfinZ = 0 Then GoTo ALTO
    EndIf
    diopasoenz = 1
    GoTo SECUENCIAX
ElseIf ejeh = 1 Then
    diopasoenh = 1
    GoTo SECUENCIAH
Else
    Clear n
    retfie fast
EndIf

```

```

***** SECUENCIA DE CONTROL PARA CADA EJE *****
*****

```

```

SECUENCIAX:            If mascaraejex = 0 Then GoTo SECUENCIAY
                        If sentidox = 0 Then GoTo ACTUALIZAESTADOX
'sentidox = 1
                        If ESTADOX = 1 Then
                            ESTADOX = 4
                            GoTo COMBINACIONESX
                        Else
                            Dec ESTADOX
                            GoTo COMBINACIONESX
                        EndIf
ACTUALIZAESTADOX:      If ESTADOX = 4 Then
                            ESTADOX = 1

```

```

        GoTo COMBINACIONESX
    Else
        Inc ESTADOx
        GoTo COMBINACIONESX
    EndIf
COMBINACIONESX:
    Select ESTADOx
        Case 1
            Set COMBINACION.0
            Clear COMBINACION.1
            GoTo SECUENCIAIY
        Case 2
            Set COMBINACION.0
            Set COMBINACION.1
            GoTo SECUENCIAIY
        Case 3
            Clear COMBINACION.0
            Set COMBINACION.1
            GoTo SECUENCIAIY
        Case Else
            Clear COMBINACION.0
            Clear COMBINACION.1
            GoTo SECUENCIAIY
    EndSelect

SECUENCIAIY:
    If mascaraeje = 0 Then GoTo SECUENCIAZ
    If sentidoy = 1 Then GoTo ACTUALIZAESTADY
    If ESTADY = 1 Then
        ESTADY = 4
        GoTo COMBINACIONESY
    Else
        Dec ESTADY
        GoTo COMBINACIONESY
    EndIf
ACTUALIZAESTADY:
    If ESTADY = 4 Then
        ESTADY = 1
        GoTo COMBINACIONESY
    Else
        Inc ESTADY
        GoTo COMBINACIONESY
    EndIf
COMBINACIONESY:
    Select ESTADY
        Case 1
            Set COMBINACION.2
            Clear COMBINACION.3
            GoTo SECUENCIAZ
        Case 2
            Set COMBINACION.2
            Set COMBINACION.3
            GoTo SECUENCIAZ
        Case 3
            Clear COMBINACION.2
            Set COMBINACION.3
            GoTo SECUENCIAZ
        Case Else
            Clear COMBINACION.2
            Clear COMBINACION.3
            GoTo SECUENCIAZ
    EndSelect

```

```

SECUENCIAZ:      If mascaraejez = 0 Then GoTo
VERIFICACAMBIODESENTIDO
                  If sentidoz = 0 Then GoTo ACTUALIZAESTADOZ
                  If ESTADOz = 1 Then
                      ESTADOz = 4
                      GoTo COMBINACIONESZ
                  Else
                      Dec ESTADOz
                      GoTo COMBINACIONESZ
                  EndIf
ACTUALIZAESTADOZ: If ESTADOz = 4 Then
                  ESTADOz = 1
                  GoTo COMBINACIONESZ
                  Else
                      Inc ESTADOz
                      GoTo COMBINACIONESZ
                  EndIf
COMBINACIONESZ:  Select ESTADOz
                  'Secuencia de paso en "Z"
                  Case 1
                      Set COMBINACION.4
                      Clear COMBINACION.5
                      If iniciodecomando = 1 Then
                          GoTo VERIFICACAMBIODESENTIDO
                      ElseIf JgoMecanico = 1 Then
                          If ejex = 1 Then
                              GoTo CompensaBacklashX
                          ElseIf ejey = 1 Then
                              GoTo CompensaBacklashY
                          ElseIf ejez = 1 Then
                              GoTo CompensaBacklashZ
                          Else
                              Clear n
                              retfie fast
                          EndIf
                      ElseIf NoCodigoG = 28 Then
                          NoCodigoG = 99
                          GoTo AVANCEPORJOYSTICK
                      ElseIf NoCodigoG = 0 Then
                          NoCodigoG = 99
                          GoTo AVANCEPORJOYSTICK
                      Else
                          GoTo AVANCEPORJOYSTICK
                      EndIf
                  Case 2
                      Set COMBINACION.4
                      Set COMBINACION.5
                      If iniciodecomando = 1 Then
                          GoTo VERIFICACAMBIODESENTIDO
                      ElseIf JgoMecanico = 1 Then
                          If ejex = 1 Then
                              GoTo CompensaBacklashX
                          ElseIf ejey = 1 Then
                              GoTo CompensaBacklashY
                          ElseIf ejez = 1 Then
                              GoTo CompensaBacklashZ
                          Else
                              Clear n
                              retfie fast
                          EndIf
                      ElseIf NoCodigoG = 28 Then

```

```

        NoCodigoG = 99
        GoTo AVANCEPORJOYSTICK
    ElseIf NoCodigoG = 0 Then
        NoCodigoG = 99
        GoTo AVANCEPORJOYSTICK
    Else
        GoTo AVANCEPORJOYSTICK
    EndIf
Case 3
    Clear COMBINACION.4
    Set COMBINACION.5
    If iniciodecomando = 1 Then
        GoTo VERIFICACAMBIODESENTIDO
    ElseIf JgoMecanico = 1 Then
        If ejex = 1 Then
            GoTo CompensaBacklashX
        ElseIf ejey = 1 Then
            GoTo CompensaBacklashY
        ElseIf ejez = 1 Then
            GoTo CompensaBacklashZ
        Else
            Clear n
            retfie fast
        EndIf
    ElseIf NoCodigoG = 28 Then
        NoCodigoG = 99
        GoTo AVANCEPORJOYSTICK
    ElseIf NoCodigoG = 0 Then
        NoCodigoG = 99
        GoTo AVANCEPORJOYSTICK
    Else
        GoTo AVANCEPORJOYSTICK
    EndIf
Case Else
    Clear COMBINACION.4
    Clear COMBINACION.5
    If iniciodecomando = 1 Then
        GoTo VERIFICACAMBIODESENTIDO
    ElseIf JgoMecanico = 1 Then
        If ejex = 1 Then
            GoTo CompensaBacklashX
        ElseIf ejey = 1 Then
            GoTo CompensaBacklashY
        ElseIf ejez = 1 Then
            GoTo CompensaBacklashZ
        Else
            Clear n
            retfie fast
        EndIf
    ElseIf NoCodigoG = 28 Then
        NoCodigoG = 99
        GoTo AVANCEPORJOYSTICK
    ElseIf NoCodigoG = 0 Then
        NoCodigoG = 99
        GoTo AVANCEPORJOYSTICK
    Else
        GoTo AVANCEPORJOYSTICK
    EndIf
EndSelect

```

SECUENCIAH:

If sentidoh = 1 Then GoTo ACTUALIZAESTADOH

```

If ESTADOh = 1 Then
    ESTADOh = 4
    GoTo COMBINACIONESH
Else
    Dec ESTADOh
    GoTo COMBINACIONESH
EndIf
ACTUALIZAESTADOH: If ESTADOh = 4 Then
    ESTADOh = 1
    GoTo COMBINACIONESH
Else
    Inc ESTADOh
    GoTo COMBINACIONESH
EndIf
COMBINACIONESH: Select ESTADOh
    'Secuencia de paso en "H"
    Case 1
        Set COMBINACION.6
        Clear COMBINACION.7
        GoTo AVANCEPORJOYSTICK
    Case 2
        Set COMBINACION.6
        Set COMBINACION.7
        GoTo AVANCEPORJOYSTICK
    Case 3
        Clear COMBINACION.6
        Set COMBINACION.7
        GoTo AVANCEPORJOYSTICK
    Case Else
        Clear COMBINACION.6
        Clear COMBINACION.7
        GoTo AVANCEPORJOYSTICK
EndSelect

'***** RUTINA PARA VERIFICAR EL CAMBIO DE SENTIDO EN
ALGÚN EJE *****
VERIFICACAMBIODESENTIDO: If iniciodecomando = 1 Then
    If homez = 1 Then
        If sentidoz = sentidozant Then
            Clear iniciodecomando
            GoTo AVANCEPORJOYSTICK
        Else
            retfie fast
        EndIf
    ElseIf homex = 1 Then
        If sentidox = sentidoxant Then
            Clear iniciodecomando
            GoTo AVANCEPORJOYSTICK
        Else
            retfie fast
        EndIf
    ElseIf homey = 1 Then
        If sentidoy = sentidoyant Then
            Clear iniciodecomando
            GoTo AVANCEPORJOYSTICK
        Else
            retfie fast
        EndIf
    ElseIf Sangria = 1 Then
        If ejez = 1 Then
            If sentidoz = sentidozant Then

```

```

        Clear iniciodecomando
        GoTo AVANCEPORJOYSTICK
    Else
        Clear iniciodecomando
        Set JgoMecanico
        PASOS = JgoMecanicoZ
    EndIf
    ElseIf ejex = 1 Then
        If sentidox = sentidoxant Then
            Clear iniciodecomando
            GoTo AVANCEPORJOYSTICK
        Else
            Clear iniciodecomando
            Set JgoMecanico
            PASOS = JgoMecanicoX
        EndIf
    ElseIf ejey = 1 Then
        If sentidoy = sentidoyant Then
            Clear iniciodecomando
            GoTo AVANCEPORJOYSTICK
        Else
            Clear iniciodecomando
            Set JgoMecanico
            PASOS = JgoMecanicoY
        EndIf
    Else
        retfie fast
    EndIf
Else
    retfie fast
EndIf
ElseIf JgoMecanico = 1 Then
    If ejex = 1 Then
        GoTo CompensaBacklashX
    ElseIf ejey = 1 Then
        GoTo CompensaBacklashY
    ElseIf ejez = 1 Then
        GoTo CompensaBacklashZ
    Else
        Clear n
        retfie fast
    EndIf
ElseIf NoCodigoG = 0 Then
    NoCodigoG = 99
    GoTo AVANCEPORJOYSTICK
ElseIf NoCodigoG = 1 Then
    NoCodigoG = 99
    GoTo AVANCEPORJOYSTICK
ElseIf NoCodigoG = 28 Then
    If ejez = 1 Then
        NoCodigoG = 99
        GoTo AVANCEPORJOYSTICK
    Else
        GoTo AVANCEPORJOYSTICK
    EndIf
Else
    GoTo AVANCEPORJOYSTICK
EndIf
EndIf

```

***** RUTINA PARA COMPENSAR EL JUEGO MECÁNICO EN
 "X" *****

```

CompensaJgoMecanicoX:      If sentidox = sentidoxant Then
                             Clear iniciodecomando
                             Return
                             Else
                             Clear iniciodecomando
                             Set JgoMecanico
                             PASOS = JgoMecanicoX
                             sentidoxant = sentidox
                             GoTo VERIFICASECUENCIA
                             EndIf
CompensaBacklashX:        PORTD = COMBINACION
                             DelayUS Pdo
                             Dec PASOS
                             If PASOS = 0 Then
                             Clear JgoMecanico
                             If home = 1 Then
                             PASOS = 3000
                             GoTo VERIFICASECUENCIA
                             Else
                             PASOS = PX
                             Return
                             EndIf
                             Else
                             btfss    RC1IF
                             GoTo VERIFICASECUENCIA
                             retfie fast
                             EndIf

***** RUTINA PARA COMPENSAR EL JUEGO MECÁNICO EN "Y" *****
CompensaJgoMecanicoY:      If sentidoy = sentidoyant Then
                             Clear iniciodecomando
                             Return
                             Else
                             Clear iniciodecomando
                             Set JgoMecanico
                             PASOS = JgoMecanicoY
                             sentidoyant = sentidoy
                             GoTo VERIFICASECUENCIA
                             EndIf
CompensaBacklashY:        PORTD = COMBINACION
                             DelayUS Pdo
                             Dec PASOS
                             If PASOS = 0 Then
                             Clear JgoMecanico
                             If home = 1 Then
                             PASOS = 3000
                             GoTo VERIFICASECUENCIA
                             Else
                             PASOS = PY
                             Return
                             EndIf
                             Else
                             btfss    RC1IF
                             GoTo VERIFICASECUENCIA
                             retfie fast
                             EndIf

***** RUTINA PARA COMPENSAR EL JUEGO MECÁNICO EN "Z" *****
CompensaJgoMecanicoZ:      If sentidoz = sentidozant Then

```

```

        Clear iniciodecomando
        Return
    Else
        Clear iniciodecomando
        Set JgoMecanico
        PASOS = JgoMecanicoZ
        sentidozant = sentidoz
        GoTo VERIFICASECUENCIA
    EndIf
CompensaBacklashZ:
    PORTD = COMBINACION
    DelayUS Pdo
    Dec PASOS
    If PASOS = 0 Then
        Clear JgoMecanico
        If home = 1 Then
            PASOS = 3000
            GoTo VERIFICASECUENCIA
        Else
            PASOS = PZ
            Return
        EndIf
    Else
        btfss RC1IF
        GoTo VERIFICASECUENCIA
        retfie fast
    EndIf

    ***** RUTINA PARA AVANZAR EN LA MODALIDAD DE
    JOYSTICK *****
    AVANCEPORJOYSTICK: PORTD = COMBINACION
    DelayUS Pdo

    If diopasoex = 1 Then
        Set DAT0enviado.0
        If sentidox = 1 Then
            Set DAT0enviado.4
            'X+" ----> 65
            If cerodepieza = 1 Then
                Xm = Xm + RESOL
                Xp = Xp + RESOL
            Else
                Xm = Xm + RESOL
            EndIf
        Else
            Clear DAT0enviado.4
            'X-" ----> 1
            If cerodepieza = 1 Then
                Xm = Xm - RESOL
                Xp = Xp - RESOL
            Else
                Xm = Xm - RESOL
            EndIf
        EndIf
    EndIf

    If diopasoey = 1 Then
        Set DAT0enviado.1
        If sentidoy = 1 Then
            Set DAT0enviado.5
            'Y+" ----> 130
            If cerodepieza = 1 Then

```

```

        Ym = Ym + RESOL
        Yp = Yp + RESOL
    Else
        Ym = Ym + RESOL
    EndIf
Else
    Clear DATOenviado.5
    '"Y-" ----> 2
    If cerodepieza = 1 Then
        Ym = Ym - RESOL
        Yp = Yp - RESOL
    Else
        Ym = Ym - RESOL
    EndIf
EndIf
EndIf

If diopasoenz = 1 Then
    Set DATOenviado.2
    If sentidoz = 1 Then
        Set DATOenviado.6
        '"Z+" ----> 36
        If cerodepieza = 1 Then
            Zm = Zm + RESOL
            Zp = Zp + RESOL
        Else
            Zm = Zm + RESOL
        EndIf
    Else
        Clear DATOenviado.6
        '"Z-" ----> 4
        If cerodepieza = 1 Then
            Zm = Zm - RESOL
            Zp = Zp - RESOL
        Else
            Zm = Zm - RESOL
        EndIf
    EndIf
EndIf

If diopasoenh = 1 Then
    Set DATOenviado.3
    If sentidoh = 1 Then
        Set DATOenviado.7
        '"HUS+" ----> 24
    Else
        Clear DATOenviado.7
        '"HUS-" ----> 8
    EndIf
EndIf

If joystick = 1 Then
    movf    DATOenviado,0
    movwf   TXREG1
    Clear   DATOenviado
    btfss   RC1IF
    GoTo    VERIFICASECUENCIA
    retfie  fast
ElseIf masdeuneje = 1 Then
    movf    DATOenviado,0
    movwf   TXREG1

```

```

Clear    DATOenviado
If mascaraejex = 1 Then
    If sentidox = 1 Then
        If PushBfinX = 0 Then GoTo ALTO
    Else
        If PushBiniX = 0 Then GoTo ALTO
    EndIf
EndIf
If mascaraejez = 1 Then
    If sentidoy = 1 Then
        If PushBfinY = 0 Then GoTo ALTO
    Else
        If PushBiniY = 0 Then GoTo ALTO
    EndIf
EndIf
If mascaraejez = 1 Then
    If sentidoz = 1 Then
        If PushBiniZ = 0 Then GoTo ALTO
    Else
        If PushBTool = 0 Then GoTo ALTO
        If PushBfinZ = 0 Then GoTo ALTO
    EndIf
EndIf
btfss    RC1IF
GoTo SECUENCIAX
retfie fast
ElseIf porfrecuencia = 1 Then
    movf    DATOenviado,0
    movwf   TXREG1
    Clear    DATOenviado
    btfss    RC1IF
    GoTo VERIFICASECUENCIA
    retfie fast
ElseIf homez = 1 Then
    movf    DATOenviado,0
    movwf   TXREG1
    Clear    DATOenviado
    Clear    diopasoenz
    btfss    RC1IF
    GoTo VERIFICASECUENCIA
    retfie fast
HabilitaHomeX:
    Zm = 0
    Clear    homez
    Clear    ejex
    Clear    mascaraejez
    Clear    diopasoenz
    sentidozant = sentidoz
    Set    ejex
    Set    mascaraejex
    Set    homex
    Set    iniciodecomando
    GoTo VERIFICACEROMQUINA
ElseIf homex = 1 Then
    movf    DATOenviado,0
    movwf   TXREG1
    Clear    DATOenviado
    Clear    diopasoenx
    btfss    RC1IF
    GoTo VERIFICASECUENCIA
    retfie fast
HabilitaHomeY:
    Xm = 0

```

```

Clear homex
Clear ejex
Clear mascaraejex
Clear diopasoenx
sentidoxant = sentidox
Set homey
Set ejey
Set mascaraejey
Set iniciodecomando
GoTo VERIFICACEROMAQUINA
ElseIf homey = 1 Then
    movf    DAT0enviado,0
    movwf   TXREG1
    Clear   DAT0enviado
    Clear   diopasoeny
    btfss   RC1IF
    GoTo    VERIFICASECUENCIA
    retfie  fast
HabilitaSangriaZ:
    Ym = 0
    Clear   homey
    Clear   ejey
    Clear   mascaraejey
    Clear   diopasoeny
    sentidoyant = sentidoy
    Set     ejez
    Set     mascaraejez
    Set     iniciodecomando
    GoTo    VERIFICACEROMAQUINA
ElseIf NoCodigoG = 0 Then
    If ejez = 1 Then
        Dec PASOS
        movf    DAT0enviado,0
        movwf   TXREG1
        Clear   DAT0enviado
        Clear   diopasoenz
        sentidozant = sentidoz
        If PASOS = 0 Then
            sentidozant = sentidoz
            GoTo ALTO
        EndIf
        btfss   RC1IF
        GoTo    VERIFICASECUENCIA
        retfie  fast
    Else
        movf    DAT0enviado,0
        movwf   TXREG1
        Clear   DAT0enviado
        Clear   diopasoenx
        Clear   diopasoeny
        If Xp = Xdda Then
            If Yp = Ydda Then
                sentidoxant = sentidox
                sentidoyant = sentidoy
                GoTo ALTO
            EndIf
        EndIf
        sentidoxant = sentidox
        sentidoyant = sentidoy
        If sentidox = 1 Then
            If PushBfinX = 0 Then GoTo ALTO
        Else

```

```

        If PushBiniX = 0 Then GoTo ALTO
    EndIf
    If sentidoy = 1 Then
        If PushBfinY = 0 Then GoTo ALTO
    Else
        If PushBiniY = 0 Then GoTo ALTO
    EndIf
    btfss    RC1IF
    GoTo INTERLINEALX
    retfie fast
EndIf
ElseIf NoCodigoG = 1 Then
    movf    DAT0enviado,0
    movwf   TXREG1
    Clear   DAT0enviado
    Clear   diopasoenx
    Clear   diopasoeny
    sentidoxant = sentidox
    sentidoyant = sentidoy
    If sentidox = 1 Then
        If PushBfinX = 0 Then GoTo ALTO
    Else
        If PushBiniX = 0 Then GoTo ALTO
    EndIf
    If sentidoy = 1 Then
        If PushBfinY = 0 Then GoTo ALTO
    Else
        If PushBiniY = 0 Then GoTo ALTO
    EndIf
    btfss    RC1IF
    GoTo INTERLINEALX
    retfie fast
ElseIf NoCodigoG = 28 Then
    If solouneje = 1 Then
        If ejex = 1 Then
            Dec PASOS
            movf    DAT0enviado,0
            movwf   TXREG1
            Clear   DAT0enviado
            If PASOS = 0 Then GoTo ALTO
            btfss    RC1IF
            GoTo VERIFICASECUENCIA
            retfie fast
        ElseIf ejey = 1 Then
            Dec PASOS
            movf    DAT0enviado,0
            movwf   TXREG1
            Clear   DAT0enviado
            If PASOS = 0 Then GoTo ALTO
            btfss    RC1IF
            GoTo VERIFICASECUENCIA
            retfie fast
        Else
            Clear n
            retfie fast
        EndIf
    ElseIf ejez = 1 Then
        Dec PASOS
        movf    DAT0enviado,0
        movwf   TXREG1
        Clear   DAT0enviado
    
```

```

    If PASOS = 0 Then GoTo ALTO
    btfss    RC1IF
    GoTo VERIFICASECUENCIA
    retfie   fast
Else
    movf     DATOenviado,0
    movwf    TXREG1
    Clear    DATOenviado
    Clear    diopasoenx
    Clear    diopasoeny
    If sentidox = 1 Then
        If PushBfinX = 0 Then GoTo ALTO
    Else
        If PushBiniX = 0 Then GoTo ALTO
    EndIf
    If sentidoy = 1 Then
        If PushBfinY = 0 Then GoTo ALTO
    Else
        If PushBiniY = 0 Then GoTo ALTO
    EndIf
    btfss    RC1IF
    GoTo INTERLINEALX
    retfie   fast
EndIf
ElseIf NoCodigoG = 2 Then
    movf     DATOenviado,0
    movwf    TXREG1
    Clear    DATOenviado
    diopasoenx = 0
    diopasoeny = 0

    If CUADRANTE = CUADRANTEFINAL Then
        If CUADRANTEFINAL = 1 Then
            If Yp = Ydda Then GoTo ALTO
            If Yp < Ydda Then GoTo ALTO
        ElseIf CUADRANTEFINAL = 2 Then
            If Xp = Xdda Then GoTo ALTO
            If Xp < Xdda Then GoTo ALTO
        ElseIf CUADRANTEFINAL = 3 Then
            If Yp = Ydda Then GoTo ALTO
            If Yp > Ydda Then GoTo ALTO
        Else
            If Xp = Xdda Then GoTo ALTO
            If Xp > Xdda Then GoTo ALTO
        EndIf
    EndIf

    If sentidox = 1 Then
        If PushBfinX = 0 Then GoTo ALTO
    Else
        If PushBiniX = 0 Then GoTo ALTO
    EndIf
    If sentidoy = 1 Then
        If PushBfinY = 0 Then GoTo ALTO
    Else
        If PushBiniY = 0 Then GoTo ALTO
    EndIf
    If CUADRANTE = 1 Then
        If Xo = Radio Then
            If Yo = 0 Then
                Inc CUADRANTE
            EndIf
        EndIf
    EndIf

```

```

        GoTo VERIFICACUADRANTE
    EndIf
EndIf
ElseIf CUADRANTE = 2 Then
    If Yo = Radio Then
        If Xo = 0 Then
            Inc CUADRANTE
            GoTo VERIFICACUADRANTE
        EndIf
    EndIf
ElseIf CUADRANTE = 3 Then
    If Xo = Radio Then
        If Yo = 0 Then
            Inc CUADRANTE
            GoTo VERIFICACUADRANTE
        EndIf
    EndIf
Else
    If Yo = Radio Then
        If Xo = 0 Then
            CUADRANTE = 1
            GoTo VERIFICACUADRANTE
        EndIf
    EndIf
EndIf
btfss    RC1IF
GoTo DSM
retfie fast
ElseIf NoCodigoG = 3 Then
    movf    DATOenviado,0
    movwf   TXREG1
    Clear    DATOenviado
    diopasoenx = 0
    diopasoeny = 0

    If CUADRANTE = CUADRANTEFINAL Then
        If CUADRANTEFINAL = 1 Then
            If Xp = Xdda Then GoTo ALTO
            If Xp < Xdda Then GoTo ALTO
        ElseIf CUADRANTEFINAL = 2 Then
            If Yp = Ydda Then GoTo ALTO
            If Yp > Ydda Then GoTo ALTO
        ElseIf CUADRANTEFINAL = 3 Then
            If Xp = Xdda Then GoTo ALTO
            If Xp > Xdda Then GoTo ALTO
        Else
            If Yp = Ydda Then GoTo ALTO
            If Yp < Ydda Then GoTo ALTO
        EndIf
    EndIf
EndIf

If sentidox = 1 Then
    If PushBfinX = 0 Then GoTo ALTO
Else
    If PushBiniX = 0 Then GoTo ALTO
EndIf
If sentidoy = 1 Then
    If PushBfinY = 0 Then GoTo ALTO
Else
    If PushBiniY = 0 Then GoTo ALTO
EndIf

```

```

    If CUADRANTE = 1 Then
        If Yo = Radio Then
            If Xo = 0 Then
                CUADRANTE = 4
                GoTo VERIFICACUADRANTE
            EndIf
        EndIf
    ElseIf CUADRANTE = 2 Then
        If Xo = Radio Then
            If Yo = 0 Then
                Dec CUADRANTE
                GoTo VERIFICACUADRANTE
            EndIf
        EndIf
    ElseIf CUADRANTE = 3 Then
        If Yo = Radio Then
            If Xo = 0 Then
                Dec CUADRANTE
                GoTo VERIFICACUADRANTE
            EndIf
        EndIf
    Else
        If Xo = Radio Then
            If Yo = 0 Then
                Dec CUADRANTE
                GoTo VERIFICACUADRANTE
            EndIf
        EndIf
    EndIf
    btfss    RC1IF
    GoTo DSM
    retfie fast
Else
    Dec PASOS
    If PASOS = 0 Then
        movf    DAT0enviado,0
        movwf   TXREG1
        Clear   DAT0enviado
        If Sangria = 1 Then
            If ejex = 1 Then
                Zm = 0
                Clear ejex
                Clear mascaraejex
                Clear diopasoenz
                sentidozant = sentidoz
                Set ejex
                Set mascaraejex
                Return
            ElseIf ejex = 1 Then
                Xm = 0
                Clear ejex
                Clear mascaraejex
                Clear diopasoex
                sentidoxant = sentidox
                Set ejex
                Set mascaraejex
                Return
            ElseIf ejey = 1 Then
                Ym = 0
                Clear ejey

```

```

        Clear mascaraeje
        Clear diopasoeny
        sentidoyant = sentidoy
        Return
    Else
        retfie fast
    EndIf
EndIf
If ejex = 1 Then sentidoxant = sentidox
If ejej = 1 Then sentidoyant = sentidoy
If ejez = 1 Then sentidozant = sentidoz
GoTo ALTO
Else
    movf    DAT0enviado,0
    movwf   TXREG1
    Clear   DAT0enviado
    btfss   RC1IF
    GoTo    VERIFICASECUENCIA
    retfie  fast
EndIf
EndIf

! ***** RUTINA PARA VERIFICAR LAS COORDENADAS EN EL
REGRESO A CERO DE MÁQUINA *****
VERIFICACEROMAQUINA:  If homez = 0 Then
                        GoTo CHECAEJEX
                        Else
                            GoSub CompensaJgoMecanicoZ
                            GoTo VERIFICASECUENCIA
                        EndIf
CHECAEJEX:            If homex = 0 Then
                        GoTo CHECAEJEY
                        Else
                            GoSub CompensaJgoMecanicoX
                            GoTo VERIFICASECUENCIA
                        EndIf
CHECAEJEY:            If homey = 0 Then
                        GoTo SANGRIAHOME
                        Else
                            GoSub CompensaJgoMecanicoY
                            GoTo VERIFICASECUENCIA
                        EndIf

! ***** RUTINA PARA EJECUTAR UNA SANGRÍA DESPUÉS DEL
REGRESO A HOME *****
SANGRIAHOME:          Set Sangria
                        Set mascaraSangria
                        Set iniciodecomando
                        Set sentidox
                        Set sentidoy
                        Clear sentidoz
                        GoSub CompensaJgoMecanicoZ
                        GoSub CompensaJgoMecanicoX
                        GoSub CompensaJgoMecanicoY
                        GoTo ALTO

! ***** RUTINA PARA CONFIGURAR LA INTERPOLACIÓN
LINEAL EN G28 *****
DDALINEALX:           If Xdda > Xp Then
                        sentidox = 1
                        ElseIf Xdda = Xp Then

```

```

        sentidox = sentidoxant
Else
        sentidox = 0
EndIf
PXdec = Abs(Xdda - Xp)
PX = PXdec / RESOL
Fcia = 170
Pdo = 1000000 / Fcia
PASOS = PX
GoSub CompensaJgoMecanicoX
GoTo VERIFICASECUENCIA

DDALINEALY: If Ydda > Yp Then
        sentidoy = 1
ElseIf Ydda = Yp Then
        sentidoy = sentidoyant
Else
        sentidoy = 0
EndIf
PYdec = Abs(Ydda - Yp)
PY = PYdec / RESOL
Fcia = 170
Pdo = 1000000 / Fcia
PASOS = PY
GoSub CompensaJgoMecanicoY
GoTo VERIFICASECUENCIA

DDALINEALZ: If Zdda > Zp Then
        sentidoz = 1
ElseIf Zdda = Zp Then
        sentidoz = sentidozant
Else
        sentidoz = 0
EndIf
PZdec = Abs(Zdda - Zp)
PZ = PZdec / RESOL
Fcia = 170
Pdo = 1000000 / Fcia
PASOS = PZ
GoSub CompensaJgoMecanicoZ
GoTo VERIFICASECUENCIA

/***** RUTINA PARA CONFIGURAR LA INTERPOLACIÓN *****/
LINEAL *****/
DDALINEAL: If Xdda > Xp Then
        sentidox = 1
ElseIf Xdda = Xp Then
        sentidox = sentidoxant
Else
        sentidox = 0
EndIf
If Ydda > Yp Then
        sentidoy = 1
ElseIf Ydda = Yp Then
        sentidoy = sentidoyant
Else
        sentidoy = 0
EndIf
PPR = 1000 / RESOL
PXdec = Abs(Xdda - Xp)
PYdec = Abs(Ydda - Yp)

```

```

PX = PXdec / RESOL
PY = PYdec / RESOL
HIPO = PX*PX
HIPO = HIPO + (PY*PY)
HIPO = Sqr HIPO
If NoCodigoG = 0 Then
    Fcia = 170
    Pdo = 1000000 / Fcia
    Set ejex
    Set mascaraejex
    Clear ejey
    Clear mascaraejey
    GoSub CompensaJgoMecanicoX
    Clear ejex
    Clear mascaraejex
    Set ejey
    Set mascaraejey
    GoSub CompensaJgoMecanicoY
    Clear ejey
    Clear mascaraejey
    Clear diopasoenx
    Clear diopasoeny
    GoTo INTERLINEALX
ElseIf NoCodigoG = 1 Then
    Fcia = (PPR * Avance) / 60
    Pdo = 1000000 / Fcia
    Set ejex
    Set mascaraejex
    Clear ejey
    Clear mascaraejey
    GoSub CompensaJgoMecanicoX
    Clear ejex
    Clear mascaraejex
    Set ejey
    Set mascaraejey
    GoSub CompensaJgoMecanicoY
    Clear ejey
    Clear mascaraejey
    Clear diopasoenx
    Clear diopasoeny
    GoTo INTERLINEALX
ElseIf NoCodigoG = 28 Then
    Fcia = 170
    Pdo = 1000000 / Fcia
    Set ejex
    Set mascaraejex
    Clear ejey
    Clear mascaraejey
    GoSub CompensaJgoMecanicoX
    Clear ejex
    Clear mascaraejex
    Set ejey
    Set mascaraejey
    GoSub CompensaJgoMecanicoY
    Clear ejey
    Clear mascaraejey
    Clear diopasoenx
    Clear diopasoeny
    GoTo INTERLINEALX
Else
    Clear n

```

```

retfie fast
EndIf

```

```

***** RUTINA DE INTERPOLACIÓN LINEAL CON DDA *****
*****
INTERLINEALX:    QX = QX + PX
                  GoTo PRIMERACONDx
INTERLINEALY:    QY = QY + PY
                  GoTo PRIMERACONDy
PRIMERACONDx:    If ContadorX > PX Then GoTo INTERLINEALY
                  If ContadorX = PX Then GoTo INTERLINEALY
                  If QX > HIPO Then QX = QX - HIPO : Inc ContadorX :
diopasoenx = 1 : GoTo SECUENCIADDALINEAL
                  GoTo INTERLINEALY
PRIMERACONDy:    If ContadorY > PY Then GoTo VECONTEOX
                  If ContadorY = PY Then GoTo VECONTEOX
                  If QY > HIPO Then QY = QY - HIPO : Inc ContadorY :
diopasoeny = 1 : GoTo CHECAEJEDDALINEALY
                  If diopasoenx = 1 Then GoTo AVANCEPORJOYSTICK
                  GoTo INTERLINEALX

VECONTEOX:       If diopasoenx = 1 Then GoTo AVANCEPORJOYSTICK
                  If ContadorX > PX Then
                      GoTo ALTO
                  ElseIf ContadorX = PX Then
                      GoTo ALTO
                  Else
                      GoTo INTERLINEALX
                  EndIf

```

```

***** ESTABLECIENDO LA SECUENCIA DE PASO PARA LA
INTERPOLACIÓN LINEAL *****

```

```

SECUENCIADDALINEAL:  If sentidox = 0 Then GoTo
ACTUALIZAEDODDALINEALX
                      If ESTADox = 1 Then
                          ESTADox = 4
                          GoTo COMBINACIONESDDALINEAL
                      Else
                          Dec ESTADox
                          GoTo COMBINACIONESDDALINEAL
                      EndIf
ACTUALIZAEDODDALINEALX: If ESTADox = 4 Then
                          ESTADox = 1
                          GoTo COMBINACIONESDDALINEAL
                      Else
                          Inc ESTADox
                          GoTo COMBINACIONESDDALINEAL
                      EndIf
CHECAEJEDDALINEALY: If sentidoy = 1 Then GoTo
ACTUALIZAEDODDALINEALY
                      If ESTADoy = 1 Then
                          ESTADoy = 4
                          GoTo OTROEJEDDALINEAL
                      Else
                          Dec ESTADoy
                          GoTo OTROEJEDDALINEAL
                      EndIf
ACTUALIZAEDODDALINEALY: If ESTADoy = 4 Then
                          ESTADoy = 1
                          GoTo OTROEJEDDALINEAL
                      Else

```

```

        Inc ESTADOy
        GoTo OTROEJEDDALINEAL
    EndIf

COMBINACIONESDDALINEAL: Select ESTADOx
    Case 1
        Set COMBINACION.0
        Clear COMBINACION.1
        GoTo INTERLINEALY
    Case 2
        Set COMBINACION.0
        Set COMBINACION.1
        GoTo INTERLINEALY
    Case 3
        Clear COMBINACION.0
        Set COMBINACION.1
        GoTo INTERLINEALY
    Case Else
        Clear COMBINACION.0
        Clear COMBINACION.1
        GoTo INTERLINEALY
    EndSelect

OTROEJEDDALINEAL:      Select ESTADOy
    Case 1
        Set COMBINACION.2
        Clear COMBINACION.3
        GoTo AVANCEPORJOYSTICK
    Case 2
        Set COMBINACION.2
        Set COMBINACION.3
        GoTo AVANCEPORJOYSTICK
    Case 3
        Clear COMBINACION.2
        Set COMBINACION.3
        GoTo AVANCEPORJOYSTICK
    Case Else
        Clear COMBINACION.2
        Clear COMBINACION.3
        GoTo AVANCEPORJOYSTICK
    EndSelect

'***** RUTINA PARA CONFIGURAR LA INTERPOLACIÓN *****
CIRCULAR *****
DSMCIRCULAR:      If NoCodigoG = 2 Then                                'Determinando
el cuadrante actual para G02
    If Xp > Idda Then
        If Yp > Jdda Then
            CUADRANTE = 1
            sentidox = 1
            sentidoy = 0
        ElseIf Yp = Jdda Then
            CUADRANTE = 2
            sentidox = 0
            sentidoy = 0
        Else
            CUADRANTE = 2
            sentidox = 0
            sentidoy = 0
        EndIf
    ElseIf Xp = Idda Then

```

```

    If Yp > Jdda Then
        CUADRANTE = 1
        sentidox = 1
        sentidoy = 0
    Else
        CUADRANTE = 3
        sentidox = 0
        sentidoy = 1
    EndIf
Else
    If Yp > Jdda Then
        CUADRANTE = 4
        sentidox = 1
        sentidoy = 1
    ElseIf Yp = Jdda Then
        CUADRANTE = 4
        sentidox = 1
        sentidoy = 1
    Else
        CUADRANTE = 3
        sentidox = 0
        sentidoy = 1
    EndIf
EndIf
Else
    'Determinando el
    cuadrante actual para G03
    If Xp > Idda Then
        If Yp > Jdda Then
            CUADRANTE = 1
            sentidox = 0
            sentidoy = 1
        ElseIf Yp = Jdda Then
            CUADRANTE = 1
            sentidox = 0
            sentidoy = 1
        Else
            CUADRANTE = 2
            sentidox = 1
            sentidoy = 1
        EndIf
    ElseIf Xp = Idda Then
        If Yp > Jdda Then
            CUADRANTE = 4
            sentidox = 0
            sentidoy = 0
        Else
            CUADRANTE = 2
            sentidox = 1
            sentidoy = 1
        EndIf
    Else
        If Yp > Jdda Then
            CUADRANTE = 4
            sentidox = 0
            sentidoy = 0
        ElseIf Yp = Jdda Then
            CUADRANTE = 3
            sentidox = 1
            sentidoy = 0
        Else
            CUADRANTE = 3

```

```

        sentidox = 1
        sentidoy = 0
    EndIf
EndIf
EndIf

    If NoCodigoG = 2 Then 'Determinando
el cuadrante final para G02
        If Xdda > Idda Then
            If Ydda > Jdda Then
                CUADRANTEFINAL = 1
            ElseIf Ydda = Jdda Then
                CUADRANTEFINAL = 1
            Else
                CUADRANTEFINAL = 2
            EndIf
        ElseIf Xdda = Idda Then
            If Ydda > Jdda Then
                CUADRANTEFINAL = 4
            Else
                CUADRANTEFINAL = 2
            EndIf
        Else
            If Ydda > Jdda Then
                CUADRANTEFINAL = 4
            ElseIf Ydda = Jdda Then
                CUADRANTEFINAL = 3
            Else
                CUADRANTEFINAL = 3
            EndIf
        EndIf
    Else 'Determinando
el cuadrante final para G03
        If Xdda > Idda Then
            If Ydda > Jdda Then
                CUADRANTEFINAL = 1
            ElseIf Ydda = Jdda Then
                CUADRANTEFINAL = 2
            Else
                CUADRANTEFINAL = 2
            EndIf
        ElseIf Xdda = Idda Then
            If Ydda > Jdda Then
                CUADRANTEFINAL = 1
            Else
                CUADRANTEFINAL = 3
            EndIf
        Else
            If Ydda > Jdda Then
                CUADRANTEFINAL = 4
            ElseIf Ydda = Jdda Then
                CUADRANTEFINAL = 4
            Else
                CUADRANTEFINAL = 3
            EndIf
        EndIf
    EndIf

    PPR = 1000 / RESOL
    PXdec = Abs(Xdda - Idda)

```

```

PYdec = Abs(Ydda - Jdda)
PX = PXdec / RESOL
PY = PYdec / RESOL
Radio = PX*PX
Radio = Radio + (PY*PY)
Radio = Sqr Radio

Xodec = Abs(Xp - Idda)
Yodec = Abs(Yp - Jdda)
Xo = Xodec / RESOL
Yo = Yodec / RESOL

D = Xo * Xo
D = D + (Yo * Yo)
D = D - (Radio * Radio)
Fcia = (PPR * Avance) / 60
Pdo = 1000000 / Fcia

```

VERIFICACUADRANTE:

```

Select CUADRANTE
Case 1
    If NoCodigoG = 2 Then
        sentidox = 1
        sentidoy = 0
    Else
        sentidox = 0
        sentidoy = 1
    EndIf
    Set ejex
    Set mascaraejex
    Clear ejey
    Clear mascaraejey
    GoSub CompensaJgoMecanicoX
    Clear ejex
    Clear mascaraejex
    Set ejey
    Set mascaraejey
    GoSub CompensaJgoMecanicoY
    Clear ejey
    Clear mascaraejey
    Clear diopasoenx
    Clear diopasoeny
    If NoCodigoG = 2 Then
        GoTo DSMCIRCULARCUAD1
    Else
        GoTo DSMCIRCULARCUAD2
    EndIf
Case 2
    If NoCodigoG = 2 Then
        sentidox = 0
        sentidoy = 0
    Else
        sentidox = 1
        sentidoy = 1
    EndIf
    Set ejex
    Set mascaraejex
    Clear ejey
    Clear mascaraejey
    GoSub CompensaJgoMecanicoX
    Clear ejex

```

```

Clear mascaraejex
Set ejey
Set mascaraejey
GoSub CompensaJgoMecanicoY
Clear ejey
Clear mascaraejey
Clear diopasoenx
Clear diopasoeny
If NoCodigoG = 2 Then
    GoTo DSMCIRCULARCUAD2
Else
    GoTo DSMCIRCULARCUAD1
EndIf
Case 3
If NoCodigoG = 2 Then
    sentidox = 0
    sentidoy = 1
Else
    sentidox = 1
    sentidoy = 0
EndIf
Set ejex
Set mascaraejex
Clear ejey
Clear mascaraejey
GoSub CompensaJgoMecanicoX
Clear ejex
Clear mascaraejex
Set ejey
Set mascaraejey
GoSub CompensaJgoMecanicoY
Clear ejey
Clear mascaraejey
Clear diopasoenx
Clear diopasoeny
If NoCodigoG = 2 Then
    GoTo DSMCIRCULARCUAD1
Else
    GoTo DSMCIRCULARCUAD2
EndIf
Case Else
If NoCodigoG = 2 Then
    sentidox = 1
    sentidoy = 1
Else
    sentidox = 0
    sentidoy = 0
EndIf
Set ejex
Set mascaraejex
Clear ejey
Clear mascaraejey
GoSub CompensaJgoMecanicoX
Clear ejex
Clear mascaraejex
Set ejey
Set mascaraejey
GoSub CompensaJgoMecanicoY
Clear ejey
Clear mascaraejey
Clear diopasoenx

```

```

Clear diopasoeny
If NoCodigoG = 2 Then
    GoTo DSMCIRCULARCUAD2
Else
    GoTo DSMCIRCULARCUAD1
EndIf
End Select

```

***** RUTINA PARA EJECUTAR LA INETRPOLACIÓN *****

```

CIRCULAR CON DSM *****
DSMCIRCULARCUAD1: DeltaX = (2 * Xo) + 1
                  DeltaY = (-2 * Yo) + 1
                  GoTo DSM
DSMCIRCULARCUAD2: DeltaX = (-2 * Xo) + 1
                  DeltaY = (2 * Yo) + 1
DSM:              D1 = Abs(D + DeltaX)
                  D2 = Abs(D + DeltaY)
                  D3 = Abs(D + DeltaX + DeltaY)
                  If D1 > D2 Then GoTo COMPARAD2COND3
                  If D1 > D3 Then
                      D = D + DeltaX + DeltaY
                      DeltaX = DeltaX + 2
                      DeltaY = DeltaY + 2
                      diopasoenx = 1
                      diopasoeny = 1
                      GoTo ACTUALIZACONTADORES
                  Else
                      D = D + DeltaX
                      DeltaX = DeltaX + 2
                      diopasoenx = 1
                      diopasoeny = 0
                      GoTo ACTUALIZACONTADORES
                  EndIf
COMPARAD2COND3:  If D2 > D3 Then
                  D = D + DeltaX + DeltaY
                  DeltaX = DeltaX + 2
                  DeltaY = DeltaY + 2
                  diopasoenx = 1
                  diopasoeny = 1
                  GoTo ACTUALIZACONTADORES
                  Else
                      D = D + DeltaY
                      DeltaY = DeltaY + 2
                      diopasoenx = 0
                      diopasoeny = 1
                      GoTo ACTUALIZACONTADORES
                  EndIf
ACTUALIZACONTADORES: Select CUADRANTE
Case 1
    If NoCodigoG = 2 Then
        If diopasoenx = 1 Then Inc Xo
        If diopasoeny = 1 Then Dec Yo
    Else
        If diopasoenx = 1 Then Dec Xo
        If diopasoeny = 1 Then Inc Yo
    EndIf
Case 2
    If NoCodigoG = 2 Then

```

```

                If diopasoenx = 1 Then Dec Xo
                If diopasoeny = 1 Then Inc Yo
            Else
                If diopasoenx = 1 Then Inc Xo
                If diopasoeny = 1 Then Dec Yo
            EndIf
        Case 3
            If NoCodigoG = 2 Then
                If diopasoenx = 1 Then Inc Xo
                If diopasoeny = 1 Then Dec Yo
            Else
                If diopasoenx = 1 Then Dec Xo
                If diopasoeny = 1 Then Inc Yo
            EndIf
        Case Else
            If NoCodigoG = 2 Then
                If diopasoenx = 1 Then Dec Xo
                If diopasoeny = 1 Then Inc Yo
            Else
                If diopasoenx = 1 Then Inc Xo
                If diopasoeny = 1 Then Dec Yo
            EndIf
        EndSelect
        If diopasoenx = 1 Then GoTo
SECUENCIADSMCIRCULAR
        If diopasoeny = 1 Then GoTo
CHECAEJEDSMCIRCULARY
        GoTo DSM

```

***** ESTABLECIENDO LA SECUENCIA DE PASO PARA LA
INTERPOLACIÓN CIRCULAR *****

```

SECUENCIADSMCIRCULAR:      If sentidox = 0 Then GoTo
ACTUALIZAEDODSMCIRCULARX   'sentidox = 1
                            If ESTADox = 1 Then
                                ESTADox = 4
                                GoTo COMBINACIONESDSMCIRCULAR
                            Else
                                Dec ESTADox
                                GoTo COMBINACIONESDSMCIRCULAR
                            EndIf
ACTUALIZAEDODSMCIRCULARX:  If ESTADox = 4 Then
                                ESTADox = 1
                                GoTo COMBINACIONESDSMCIRCULAR
                            Else
                                Inc ESTADox
                                GoTo COMBINACIONESDSMCIRCULAR
                            EndIf
CHECAEJEDSMCIRCULARY:      If sentidoy = 1 Then GoTo
ACTUALIZAEDODSMCIRCULARY   If ESTADoy = 1 Then
                                ESTADoy = 4
                                GoTo OTROEJEDSMCIRCULAR
                            Else
                                Dec ESTADoy
                                GoTo OTROEJEDSMCIRCULAR
                            EndIf
ACTUALIZAEDODSMCIRCULARY:  If ESTADoy = 4 Then
                                ESTADoy = 1
                                GoTo OTROEJEDSMCIRCULAR
                            Else

```

```

Inc ESTADOy
GoTo OTROEJEDSMCIRCULAR
EndIf

COMBINACIONESDSMCIRCULAR: Select ESTADOx
Case 1
Set COMBINACION.0
Clear COMBINACION.1
If diopasoeny = 1 Then GoTo

CHECAEJEDSMCIRCULAR
GoTo AVANCEPORJOYSTICK
Case 2
Set COMBINACION.0
Set COMBINACION.1
If diopasoeny = 1 Then GoTo

CHECAEJEDSMCIRCULAR
GoTo AVANCEPORJOYSTICK
Case 3
Clear COMBINACION.0
Set COMBINACION.1
If diopasoeny = 1 Then GoTo

CHECAEJEDSMCIRCULAR
GoTo AVANCEPORJOYSTICK
Case Else
Clear COMBINACION.0
Clear COMBINACION.1
If diopasoeny = 1 Then GoTo

CHECAEJEDSMCIRCULAR
GoTo AVANCEPORJOYSTICK
EndSelect

OTROEJEDSMCIRCULAR: Select ESTADOy
Case 1
Set COMBINACION.2
Clear COMBINACION.3
GoTo AVANCEPORJOYSTICK
Case 2
Set COMBINACION.2
Set COMBINACION.3
GoTo AVANCEPORJOYSTICK
Case 3
Clear COMBINACION.2
Set COMBINACION.3
GoTo AVANCEPORJOYSTICK
Case Else
Clear COMBINACION.2
Clear COMBINACION.3
GoTo AVANCEPORJOYSTICK
EndSelect

! ***** RUTINA PARA EJECUTAR UNA PAUSA DURANTE EL
MAQUINADO *****
EjecutaPausa: If Pausa > 60 Then Pausa = 60
Pausa = Pausa * 1000
DelayMS Pausa
GoTo ALTO

! ***** RUTINA PARA VERIFICAR DÍGITO DECIMAL
*****
VERIFICADIGDECIMAL: While cnt2<2
Select CadenadeControl[n]

```

'0 decimal

Case 48

```
If ejex = 1 Then
    'Inc n
    'Clear ejex
    'GoTo OBTENEJE
    GoTo Ignora
ElseIf ejey = 1 Then
    GoTo Ignora
ElseIf ejez = 1 Then
    'Inc n
    'GoTo FINDELINEA
    GoTo Ignora
ElseIf coordenadai = 1 Then
    'Inc n
    'Clear coordenadai
    'GoTo OBTENEJE
    GoTo Ignora
ElseIf coordenadaj = 1 Then
    'Inc n
    'Clear coordenadaj
    'GoTo PARAMETROS
    GoTo Ignora
Else
    Clear n
    retfie fast
EndIf
```

Case 49

'1 decimal

```
If ejex = 1 Then
    'Inc n
    'Clear ejex
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Xdda = Xdda + 0.1
        Else
            Xdda = Xdda - 0.1
        EndIf
    Else
        If Positivo = 1 Then
            Xdda = Xdda + 0.01
        Else
            Xdda = Xdda - 0.01
        EndIf
    EndIf
    'GoTo OBTENEJE
ElseIf ejey = 1 Then
    'Inc n
    'Clear ejey
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Ydda = Ydda + 0.1
        Else
            Ydda = Ydda - 0.1
        EndIf
    Else
        If Positivo = 1 Then
            Ydda = Ydda + 0.01
        Else
            Ydda = Ydda - 0.01
        EndIf
    EndIf
```

```

EndIf
GoTo Ignora
ElseIf ejez = 1 Then
    'Inc n
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Zdda = Zdda + 0.1
        Else
            Zdda = Zdda - 0.1
        EndIf
    Else
        If Positivo = 1 Then
            Zdda = Zdda + 0.01
        Else
            Zdda = Zdda - 0.01
        EndIf
    EndIf
    'GoTo FINDELINEA
    GoTo Ignora
ElseIf coordenadai = 1 Then
    'Inc n
    'Clear coordenadai
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Idda = Idda + 0.1
        Else
            Idda = Idda - 0.1
        EndIf
    Else
        If Positivo = 1 Then
            Idda = Idda + 0.01
        Else
            Idda = Idda - 0.01
        EndIf
    EndIf
    'GoTo OBTENEJE
    GoTo Ignora
ElseIf coordenadaj = 1 Then
    'Inc n
    'Clear coordenadaj
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Jdda = Jdda + 0.1
        Else
            Jdda = Jdda - 0.1
        EndIf
    Else
        If Positivo = 1 Then
            Jdda = Jdda + 0.01
        Else
            Jdda = Jdda - 0.01
        EndIf
    EndIf
    'GoTo PARAMETROS
    GoTo Ignora
Else
    Clear n
    retfie fast
EndIf
Case 50

```

'2 decimal

```

If ejex = 1 Then
    'Inc n
    'Clear ejex
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Xdda = Xdda + 0.2
        Else
            Xdda = Xdda - 0.2
        EndIf
    Else
        If Positivo = 1 Then
            Xdda = Xdda + 0.02
        Else
            Xdda = Xdda - 0.02
        EndIf
    EndIf
    'GoTo OBTENEJE
ElseIf ejey = 1 Then
    'Inc n
    'Clear ejey
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Ydda = Ydda + 0.2
        Else
            Ydda = Ydda - 0.2
        EndIf
    Else
        If Positivo = 1 Then
            Ydda = Ydda + 0.02
        Else
            Ydda = Ydda - 0.02
        EndIf
    EndIf
    GoTo Ignora
ElseIf ejez = 1 Then
    'Inc n
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Zdda = Zdda + 0.2
        Else
            Zdda = Zdda - 0.2
        EndIf
    Else
        If Positivo = 1 Then
            Zdda = Zdda + 0.02
        Else
            Zdda = Zdda - 0.02
        EndIf
    EndIf
    'GoTo FINDELINEA
    GoTo Ignora
ElseIf coordenadai = 1 Then
    'Inc n
    'Clear coordenadai
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Idda = Idda + 0.2
        Else
            Idda = Idda - 0.2
        EndIf
    Else

```

```

        If Positivo = 1 Then
            Idda = Idda + 0.02
        Else
            Idda = Idda - 0.02
        EndIf
    EndIf
    'GoTo OBTENEJE
    GoTo Ignora
ElseIf coordenadaj = 1 Then
    'Inc n
    'Clear coordenadaj
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Jdda = Jdda + 0.2
        Else
            Jdda = Jdda - 0.2
        EndIf
    Else
        If Positivo = 1 Then
            Jdda = Jdda + 0.02
        Else
            Jdda = Jdda - 0.02
        EndIf
    EndIf
    'GoTo PARAMETROS
    GoTo Ignora
Else
    Clear n
    retfie fast
EndIf
Case 51
'3 decimal

If ejex = 1 Then
    'Inc n
    'Clear ejex
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Xdda = Xdda + 0.3
        Else
            Xdda = Xdda - 0.3
        EndIf
    Else
        If Positivo = 1 Then
            Xdda = Xdda + 0.03
        Else
            Xdda = Xdda - 0.03
        EndIf
    EndIf
    'GoTo OBTENEJE
ElseIf ejey = 1 Then
    'Inc n
    'Clear ejey
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Ydda = Ydda + 0.3
        Else
            Ydda = Ydda - 0.3
        EndIf
    Else
        If Positivo = 1 Then
            Ydda = Ydda + 0.03

```

```

Else
    Ydda = Ydda - 0.03
EndIf
EndIf
GoTo Ignora
ElseIf ejez = 1 Then
    'Inc n
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Zdda = Zdda + 0.3
        Else
            Zdda = Zdda - 0.3
        EndIf
    Else
        If Positivo = 1 Then
            Zdda = Zdda + 0.03
        Else
            Zdda = Zdda - 0.03
        EndIf
    EndIf
    'GoTo FINDELINEA
    GoTo Ignora
ElseIf coordenadai = 1 Then
    'Inc n
    'Clear coordenadai
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Idda = Idda + 0.3
        Else
            Idda = Idda - 0.3
        EndIf
    Else
        If Positivo = 1 Then
            Idda = Idda + 0.03
        Else
            Idda = Idda - 0.03
        EndIf
    EndIf
    'GoTo OBTENEJE
    GoTo Ignora
ElseIf coordenadaj = 1 Then
    'Inc n
    'Clear coordenadaj
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Jdda = Jdda + 0.3
        Else
            Jdda = Jdda - 0.3
        EndIf
    Else
        If Positivo = 1 Then
            Jdda = Jdda + 0.03
        Else
            Jdda = Jdda - 0.03
        EndIf
    EndIf
    'GoTo PARAMETROS
    GoTo Ignora
Else
    Clear n
    retfie fast

```

'4 decimal

```
EndIf
Case 52
    If ejex = 1 Then
        'Inc n
        'Clear ejex
        If cnt2 = 0 Then
            If Positivo = 1 Then
                Xdda = Xdda + 0.4
            Else
                Xdda = Xdda - 0.4
            EndIf
        Else
            If Positivo = 1 Then
                Xdda = Xdda + 0.04
            Else
                Xdda = Xdda - 0.04
            EndIf
        EndIf
        'GoTo OBTENEJE
    ElseIf ejey = 1 Then
        'Inc n
        'Clear ejey
        If cnt2 = 0 Then
            If Positivo = 1 Then
                Ydda = Ydda + 0.4
            Else
                Ydda = Ydda - 0.4
            EndIf
        Else
            If Positivo = 1 Then
                Ydda = Ydda + 0.04
            Else
                Ydda = Ydda - 0.04
            EndIf
        EndIf
        GoTo Ignora
    ElseIf ejez = 1 Then
        'Inc n
        If cnt2 = 0 Then
            If Positivo = 1 Then
                Zdda = Zdda + 0.4
            Else
                Zdda = Zdda - 0.4
            EndIf
        Else
            If Positivo = 1 Then
                Zdda = Zdda + 0.04
            Else
                Zdda = Zdda - 0.04
            EndIf
        EndIf
        'GoTo FINDELINEA
        GoTo Ignora
    ElseIf coordenadai = 1 Then
        'Inc n
        'Clear coordenadai
        If cnt2 = 0 Then
            If Positivo = 1 Then
                Idda = Idda + 0.4
            Else
```

```

        Idda = Idda - 0.4
    EndIf
Else
    If Positivo = 1 Then
        Idda = Idda + 0.04
    Else
        Idda = Idda - 0.04
    EndIf
EndIf
'GoTo OBTENEJE
GoTo Ignora
ElseIf coordenadaj = 1 Then
    'Inc n
    'Clear coordenadaj
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Jdda = Jdda + 0.4
        Else
            Jdda = Jdda - 0.4
        EndIf
    Else
        If Positivo = 1 Then
            Jdda = Jdda + 0.04
        Else
            Jdda = Jdda - 0.04
        EndIf
    EndIf
    'GoTo PARAMETROS
    GoTo Ignora
Else
    Clear n
    retfie fast
EndIf
Case 53
'5 decimal
If ejex = 1 Then
    'Inc n
    'Clear ejex
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Xdda = Xdda + 0.5
        Else
            Xdda = Xdda - 0.5
        EndIf
    Else
        If Positivo = 1 Then
            Xdda = Xdda + 0.05
        Else
            Xdda = Xdda - 0.05
        EndIf
    EndIf
    'GoTo OBTENEJE
ElseIf ejey = 1 Then
    'Inc n
    'Clear ejey
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Ydda = Ydda + 0.5
        Else
            Ydda = Ydda - 0.5
        EndIf
    EndIf

```

```

Else
    If Positivo = 1 Then
        Ydda = Ydda + 0.05
    Else
        Ydda = Ydda - 0.05
    EndIf
EndIf
GoTo Ignora
ElseIf ejez = 1 Then
    'Inc n
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Zdda = Zdda + 0.5
        Else
            Zdda = Zdda - 0.5
        EndIf
    Else
        If Positivo = 1 Then
            Zdda = Zdda + 0.05
        Else
            Zdda = Zdda - 0.05
        EndIf
    EndIf
    'GoTo FINDELINEA
    GoTo Ignora
ElseIf coordenadai = 1 Then
    'Inc n
    'Clear coordenadai
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Idda = Idda + 0.5
        Else
            Idda = Idda - 0.5
        EndIf
    Else
        If Positivo = 1 Then
            Idda = Idda + 0.05
        Else
            Idda = Idda - 0.05
        EndIf
    EndIf
    'GoTo OBTENEJE
    GoTo Ignora
ElseIf coordenadaj = 1 Then
    'Inc n
    'Clear coordenadaj
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Jdda = Jdda + 0.5
        Else
            Jdda = Jdda - 0.5
        EndIf
    Else
        If Positivo = 1 Then
            Jdda = Jdda + 0.05
        Else
            Jdda = Jdda - 0.05
        EndIf
    EndIf
    'GoTo PARAMETROS
    GoTo Ignora

```

```

Else
    Clear n
    retfie fast
EndIf
Case 54
'6 decimal
If ejex = 1 Then
    'Inc n
    'Clear ejex
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Xdda = Xdda + 0.6
        Else
            Xdda = Xdda - 0.6
        EndIf
    Else
        If Positivo = 1 Then
            Xdda = Xdda + 0.06
        Else
            Xdda = Xdda - 0.06
        EndIf
    EndIf
    'GoTo OBTENEJE
ElseIf ejey = 1 Then
    'Inc n
    'Clear ejey
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Ydda = Ydda + 0.6
        Else
            Ydda = Ydda - 0.6
        EndIf
    Else
        If Positivo = 1 Then
            Ydda = Ydda + 0.06
        Else
            Ydda = Ydda - 0.06
        EndIf
    EndIf
    GoTo Ignora
ElseIf ejez = 1 Then
    'Inc n
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Zdda = Zdda + 0.6
        Else
            Zdda = Zdda - 0.6
        EndIf
    Else
        If Positivo = 1 Then
            Zdda = Zdda + 0.06
        Else
            Zdda = Zdda - 0.06
        EndIf
    EndIf
    'GoTo FINDELINEA
    GoTo Ignora
ElseIf coordenadai = 1 Then
    'Inc n
    'Clear coordenadai
    If cnt2 = 0 Then

```

```

        If Positivo = 1 Then
            Idda = Idda + 0.6
        Else
            Idda = Idda - 0.6
        EndIf
    Else
        If Positivo = 1 Then
            Idda = Idda + 0.06
        Else
            Idda = Idda - 0.06
        EndIf
    EndIf
    'GoTo OBTENEJE
    GoTo Ignora
ElseIf coordenadaj = 1 Then
    'Inc n
    'Clear coordenadaj
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Jdda = Jdda + 0.6
        Else
            Jdda = Jdda - 0.6
        EndIf
    Else
        If Positivo = 1 Then
            Jdda = Jdda + 0.06
        Else
            Jdda = Jdda - 0.06
        EndIf
    EndIf
    'GoTo PARAMETROS
    GoTo Ignora
Else
    Clear n
    retfie fast
EndIf
Case 55
'7 decimal
    If ejex = 1 Then
        'Inc n
        'Clear ejex
        If cnt2 = 0 Then
            If Positivo = 1 Then
                Xdda = Xdda + 0.7
            Else
                Xdda = Xdda - 0.7
            EndIf
        Else
            If Positivo = 1 Then
                Xdda = Xdda + 0.07
            Else
                Xdda = Xdda - 0.07
            EndIf
        EndIf
        'GoTo OBTENEJE
    ElseIf ejey = 1 Then
        'Inc n
        'Clear ejey
        If cnt2 = 0 Then
            If Positivo = 1 Then
                Ydda = Ydda + 0.7
            EndIf
        EndIf
    EndIf

```

```

        Else
            Ydda = Ydda - 0.7
        EndIf
    Else
        If Positivo = 1 Then
            Ydda = Ydda + 0.07
        Else
            Ydda = Ydda - 0.07
        EndIf
    EndIf
    GoTo Ignora
ElseIf ejez = 1 Then
    'Inc n
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Zdda = Zdda + 0.7
        Else
            Zdda = Zdda - 0.7
        EndIf
    Else
        If Positivo = 1 Then
            Zdda = Zdda + 0.07
        Else
            Zdda = Zdda - 0.07
        EndIf
    EndIf
    'GoTo FINDELINEA
    GoTo Ignora
ElseIf coordenadai = 1 Then
    'Inc n
    'Clear coordenadai
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Idda = Idda + 0.7
        Else
            Idda = Idda - 0.7
        EndIf
    Else
        If Positivo = 1 Then
            Idda = Idda + 0.07
        Else
            Idda = Idda - 0.07
        EndIf
    EndIf
    'GoTo OBTENEJE
    GoTo Ignora
ElseIf coordenadaj = 1 Then
    'Inc n
    'Clear coordenadaj
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Jdda = Jdda + 0.7
        Else
            Jdda = Jdda - 0.7
        EndIf
    Else
        If Positivo = 1 Then
            Jdda = Jdda + 0.07
        Else
            Jdda = Jdda - 0.07
        EndIf
    EndIf

```

```

EndIf
'GoTo PARAMETROS
GoTo Ignora
Else
Clear n
retfie fast
EndIf
Case 56
'8 decimal
If ejex = 1 Then
'Inc n
'Clear ejex
If cnt2 = 0 Then
If Positivo = 1 Then
Xdda = Xdda + 0.8
Else
Xdda = Xdda - 0.8
EndIf
Else
If Positivo = 1 Then
Xdda = Xdda + 0.08
Else
Xdda = Xdda - 0.08
EndIf
EndIf
'GoTo OBTENEJE
ElseIf ejey = 1 Then
'Inc n
'Clear ejey
If cnt2 = 0 Then
If Positivo = 1 Then
Ydda = Ydda + 0.8
Else
Ydda = Ydda - 0.8
EndIf
Else
If Positivo = 1 Then
Ydda = Ydda + 0.08
Else
Ydda = Ydda - 0.08
EndIf
EndIf
GoTo Ignora
ElseIf ejez = 1 Then
'Inc n
If cnt2 = 0 Then
If Positivo = 1 Then
Zdda = Zdda + 0.8
Else
Zdda = Zdda - 0.8
EndIf
Else
If Positivo = 1 Then
Zdda = Zdda + 0.08
Else
Zdda = Zdda - 0.08
EndIf
EndIf
'GoTo FINDELINEA
GoTo Ignora
ElseIf coordenadai = 1 Then

```

```

'Inc n
'Clear coordenadai
If cnt2 = 0 Then
    If Positivo = 1 Then
        Idda = Idda + 0.8
    Else
        Idda = Idda - 0.8
    EndIf
Else
    If Positivo = 1 Then
        Idda = Idda + 0.08
    Else
        Idda = Idda - 0.08
    EndIf
EndIf
'GoTo OBTENEJE
GoTo Ignora
ElseIf coordenadaj = 1 Then
'Inc n
'Clear coordenadaj
If cnt2 = 0 Then
    If Positivo = 1 Then
        Jdda = Jdda + 0.8
    Else
        Jdda = Jdda - 0.8
    EndIf
Else
    If Positivo = 1 Then
        Jdda = Jdda + 0.08
    Else
        Jdda = Jdda - 0.08
    EndIf
EndIf
'GoTo PARAMETROS
GoTo Ignora
Else
    Clear n
    retfie fast
EndIf
Case 57
'9 decimal
If ejex = 1 Then
'Inc n
'Clear ejex
If cnt2 = 0 Then
    If Positivo = 1 Then
        Xdda = Xdda + 0.9
    Else
        Xdda = Xdda - 0.9
    EndIf
Else
    If Positivo = 1 Then
        Xdda = Xdda + 0.09
    Else
        Xdda = Xdda - 0.09
    EndIf
EndIf
'GoTo OBTENEJE
ElseIf ejey = 1 Then
'Inc n
'Clear ejey

```

```

If cnt2 = 0 Then
    If Positivo = 1 Then
        Ydda = Ydda + 0.9
    Else
        Ydda = Ydda - 0.9
    EndIf
Else
    If Positivo = 1 Then
        Ydda = Ydda + 0.09
    Else
        Ydda = Ydda - 0.09
    EndIf
EndIf
GoTo Ignora
ElseIf ejez = 1 Then
    'Inc n
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Zdda = Zdda + 0.9
        Else
            Zdda = Zdda - 0.9
        EndIf
    Else
        If Positivo = 1 Then
            Zdda = Zdda + 0.09
        Else
            Zdda = Zdda - 0.09
        EndIf
    EndIf
    'GoTo FINDELINEA
    GoTo Ignora
ElseIf coordenadai = 1 Then
    'Inc n
    'Clear coordenadai
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Idda = Idda + 0.9
        Else
            Idda = Idda - 0.9
        EndIf
    Else
        If Positivo = 1 Then
            Idda = Idda + 0.09
        Else
            Idda = Idda - 0.09
        EndIf
    EndIf
    'GoTo OBTENEJE
    GoTo Ignora
ElseIf coordenadaj = 1 Then
    'Inc n
    'Clear coordenadaj
    If cnt2 = 0 Then
        If Positivo = 1 Then
            Jdda = Jdda + 0.9
        Else
            Jdda = Jdda - 0.9
        EndIf
    Else
        If Positivo = 1 Then
            Jdda = Jdda + 0.09

```

```

Else
    Jdda = Jdda - 0.09
EndIf
EndIf
'GoTo PARAMETROS
GoTo Ignora
Else
    Clear n
    retfie fast
EndIf
Case Else
    Clear n
    retfie fast
EndSelect
Inc n
Inc cnt2
Wend
Clear cnt2

Ignora:
If ejex = 1 Then
    Clear ejex
    GoTo OBTENEJE
ElseIf ejey = 1 Then
    Clear ejey
    If NoCodigoG = 1 Then
        GoTo PARAMETROS
    ElseIf NoCodigoG = 2 Then
        GoTo OBTENEJE
    ElseIf NoCodigoG = 3 Then
        GoTo OBTENEJE
    Else
        GoTo FINDELINIA
    EndIf
ElseIf ejez = 1 Then
    GoTo FINDELINIA
ElseIf coordenadai = 1 Then
    Clear coordenadai
    GoTo OBTENEJE
ElseIf coordenadaj = 1 Then
    Clear coordenadaj
    GoTo PARAMETROS
Else
    Clear n
    retfie fast
EndIf

'***** ROUTINA PARA OBTENER UN VALOR DE 2
DÍGITOS *****
OBTEN2DIG: While cnt2<2
Select CadenadeControl[n]
Case 48
'CERO "0"
If cnt2 = 0 Then
    CadenadeControl[n] = 0
    POSICION = CadenadeControl[n]
    ValorDos = POSICION * 10
Else
    CadenadeControl[n] = 0
    POSICION = CadenadeControl[n]
    ValorDos = ValorDos + POSICION
EndIf

```

'UNO "1"	Case 49 If cnt2 = 0 Then CadenadeControl[n] = 1 POSICION = CadenadeControl[n] ValorDos = POSICION * 10 Else CadenadeControl[n] = 1 POSICION = CadenadeControl[n] ValorDos = ValorDos + POSICION EndIf
'DOS "2"	Case 50 If cnt2 = 0 Then CadenadeControl[n] = 2 POSICION = CadenadeControl[n] ValorDos = POSICION * 10 Else CadenadeControl[n] = 2 POSICION = CadenadeControl[n] ValorDos = ValorDos + POSICION EndIf
'TRES "3"	Case 51 If cnt2 = 0 Then CadenadeControl[n] = 3 POSICION = CadenadeControl[n] ValorDos = POSICION * 10 Else CadenadeControl[n] = 3 POSICION = CadenadeControl[n] ValorDos = ValorDos + POSICION EndIf
'CUATRO "4"	Case 52 If cnt2 = 0 Then CadenadeControl[n] = 4 POSICION = CadenadeControl[n] ValorDos = POSICION * 10 Else CadenadeControl[n] = 4 POSICION = CadenadeControl[n] ValorDos = ValorDos + POSICION EndIf
'CINCO "5"	Case 53 If cnt2 = 0 Then CadenadeControl[n] = 5 POSICION = CadenadeControl[n] ValorDos = POSICION * 10 Else CadenadeControl[n] = 5 POSICION = CadenadeControl[n] ValorDos = ValorDos + POSICION EndIf
'SEIS "6"	Case 54 If cnt2 = 0 Then CadenadeControl[n] = 6 POSICION = CadenadeControl[n] ValorDos = POSICION * 10

```

Else
    CadenadeControl[n] = 6
    POSICION = CadenadeControl[n]
    ValorDos = ValorDos + POSICION
EndIf
Case 55
'SIETE "7"
    If cnt2 = 0 Then
        CadenadeControl[n] = 7
        POSICION = CadenadeControl[n]
        ValorDos = POSICION * 10
    Else
        CadenadeControl[n] = 7
        POSICION = CadenadeControl[n]
        ValorDos = ValorDos + POSICION
    EndIf
Case 56
'OCHO "8"
    If cnt2 = 0 Then
        CadenadeControl[n] = 8
        POSICION = CadenadeControl[n]
        ValorDos = POSICION * 10
    Else
        CadenadeControl[n] = 8
        POSICION = CadenadeControl[n]
        ValorDos = ValorDos + POSICION
    EndIf
Case 57
'NUEVE "9"
    If cnt2 = 0 Then
        CadenadeControl[n] = 9
        POSICION = CadenadeControl[n]
        ValorDos = POSICION * 10
    Else
        CadenadeControl[n] = 9
        POSICION = CadenadeControl[n]
        ValorDos = ValorDos + POSICION
    EndIf
Case Else
    Clear n
    retfie fast
EndSelect
Inc n
Inc cnt2
Wend
Clear cnt2

If lineacodigog = 1 Then
    If numerocodigo = 1 Then
        NoCodigoG = ValorDos.BYTE0
        Clear numerocodigo
        GoTo OBTENEJE
    ElseIf noavance = 1 Then
        Avance = ValorDos * (0.1)
        Clear noavance
        GoTo FINDELINEA
    ElseIf NoPausa = 1 Then
        Pausa = ValorDos.BYTE0
        Clear NoPausa
        GoTo FINDELINEA
    Else

```

```

        Clear n
        retfie fast
    EndIf
ElseIf lineacodigom = 1 Then
    If numerocodigo = 1 Then
        NoCodigoM = ValorDos.BYTE0
        Clear numerocodigo
        GoTo PARAMETROS
    Else
        Clear n
        retfie fast
    EndIf
Else
    Clear n
    retfie fast
EndIf

'***** ROUTINA PARA OBTENER UN VALOR DE 3 DÍGITOS *****'
OBTEN3DIG:    While cnt3<3
                Select CadenadeControl[n]
                Case 48
                    'CERO "0"
                    If cnt3 = 0 Then
                        CadenadeControl[n] = 0
                        POSICION = CadenadeControl[n]
                        ValorTres = POSICION * 100
                    ElseIf cnt3 = 1 Then
                        CadenadeControl[n] = 0
                        POSICION = CadenadeControl[n]
                        ValorTres = ValorTres + (POSICION * 10)
                    Else
                        CadenadeControl[n] = 0
                        POSICION = CadenadeControl[n]
                        ValorTres = ValorTres + POSICION
                    EndIf
                Case 49
                    'UNO "1"
                    If cnt3 = 0 Then
                        CadenadeControl[n] = 1
                        POSICION = CadenadeControl[n]
                        ValorTres = POSICION * 100
                    ElseIf cnt3 = 1 Then
                        CadenadeControl[n] = 1
                        POSICION = CadenadeControl[n]
                        ValorTres = ValorTres + (POSICION * 10)
                    Else
                        CadenadeControl[n] = 1
                        POSICION = CadenadeControl[n]
                        ValorTres = ValorTres + POSICION
                    EndIf
                Case 50
                    'DOS "2"
                    If cnt3 = 0 Then
                        CadenadeControl[n] = 2
                        POSICION = CadenadeControl[n]
                        ValorTres = POSICION * 100
                    ElseIf cnt3 = 1 Then
                        CadenadeControl[n] = 2
                        POSICION = CadenadeControl[n]
                        ValorTres = ValorTres + (POSICION * 10)

```

```

Else
    CadenadeControl[n] = 2
    POSICION = CadenadeControl[n]
    ValorTres = ValorTres + POSICION
EndIf
Case 51
'TRES "3"
    If cnt3 = 0 Then
        CadenadeControl[n] = 3
        POSICION = CadenadeControl[n]
        ValorTres = POSICION * 100
    ElseIf cnt3 = 1 Then
        CadenadeControl[n] = 3
        POSICION = CadenadeControl[n]
        ValorTres = ValorTres + (POSICION * 10)
    Else
        CadenadeControl[n] = 3
        POSICION = CadenadeControl[n]
        ValorTres = ValorTres + POSICION
    EndIf
Case 52
'CUATRO "4"
    If cnt3 = 0 Then
        CadenadeControl[n] = 4
        POSICION = CadenadeControl[n]
        ValorTres = POSICION * 100
    ElseIf cnt3 = 1 Then
        CadenadeControl[n] = 4
        POSICION = CadenadeControl[n]
        ValorTres = ValorTres + (POSICION * 10)
    Else
        CadenadeControl[n] = 4
        POSICION = CadenadeControl[n]
        ValorTres = ValorTres + POSICION
    EndIf
Case 53
'CINCO "5"
    If cnt3 = 0 Then
        CadenadeControl[n] = 5
        POSICION = CadenadeControl[n]
        ValorTres = POSICION * 100
    ElseIf cnt3 = 1 Then
        CadenadeControl[n] = 5
        POSICION = CadenadeControl[n]
        ValorTres = ValorTres + (POSICION * 10)
    Else
        CadenadeControl[n] = 5
        POSICION = CadenadeControl[n]
        ValorTres = ValorTres + POSICION
    EndIf
Case 54
'SEIS "6"
    If cnt3 = 0 Then
        CadenadeControl[n] = 6
        POSICION = CadenadeControl[n]
        ValorTres = POSICION * 100
    ElseIf cnt3 = 1 Then
        CadenadeControl[n] = 6
        POSICION = CadenadeControl[n]
        ValorTres = ValorTres + (POSICION * 10)
    Else

```

```

        CadenadeControl[n] = 6
        POSICION = CadenadeControl[n]
        ValorTres = ValorTres + POSICION
    EndIf
    Case 55
        'SIETE "7"
        If cnt3 = 0 Then
            CadenadeControl[n] = 7
            POSICION = CadenadeControl[n]
            ValorTres = POSICION * 100
        ElseIf cnt3 = 1 Then
            CadenadeControl[n] = 7
            POSICION = CadenadeControl[n]
            ValorTres = ValorTres + (POSICION * 10)
        Else
            CadenadeControl[n] = 7
            POSICION = CadenadeControl[n]
            ValorTres = ValorTres + POSICION
        EndIf
    Case 56
        'OCHO "8"
        If cnt3 = 0 Then
            CadenadeControl[n] = 8
            POSICION = CadenadeControl[n]
            ValorTres = POSICION * 100
        ElseIf cnt3 = 1 Then
            CadenadeControl[n] = 8
            POSICION = CadenadeControl[n]
            ValorTres = ValorTres + (POSICION * 10)
        Else
            CadenadeControl[n] = 8
            POSICION = CadenadeControl[n]
            ValorTres = ValorTres + POSICION
        EndIf
    Case 57
        'NUEVE "9"
        If cnt3 = 0 Then
            CadenadeControl[n] = 9
            POSICION = CadenadeControl[n]
            ValorTres = POSICION * 100
        ElseIf cnt3 = 1 Then
            CadenadeControl[n] = 9
            POSICION = CadenadeControl[n]
            ValorTres = ValorTres + (POSICION * 10)
        Else
            CadenadeControl[n] = 9
            POSICION = CadenadeControl[n]
            ValorTres = ValorTres + POSICION
        EndIf
    Case Else
        Clear n
        retfie fast
    EndSelect
    Inc n
    Inc cnt3
Wend
Clear cnt3

If backlash = 1 Then
    If ejex = 1 Then
        JgoMecanicoX = ValorTres / RESOL
    
```

```

        Clear ejex
        GoTo FINDELINEA
    ElseIf ejey = 1 Then
        JgoMecanicoY = ValorTres / RESOL
        Clear ejey
        GoTo FINDELINEA
    ElseIf ejez = 1 Then
        JgoMecanicoZ = ValorTres / RESOL
        Clear ejez
        GoTo FINDELINEA
    Else
        Clear n
        retfie fast
    EndIf
    ElseIf porfrecuencia = 1 Then
        Fcia = ValorTres
        GoTo FINDELINEA
    Else
        Clear n
        retfie fast
    EndIf

    ***** RUTINA PARA OBTENER UN VALOR DE 4 DÍGITOS *****
OBTEN4DIG:    While cnt4<4
                Select CadenadeControl[n]
                Case 48

                    'CERO "0"

                    If cnt4 = 0 Then
                        CadenadeControl[n] = 0
                        POSICION = CadenadeControl[n]
                        ValorCuatro = POSICION * 1000
                    ElseIf cnt4 = 1 Then
                        CadenadeControl[n] = 0
                        POSICION = CadenadeControl[n]
                        ValorCuatro = ValorCuatro + (POSICION * 100)
                    ElseIf cnt4 = 2 Then
                        CadenadeControl[n] = 0
                        POSICION = CadenadeControl[n]
                        ValorCuatro = ValorCuatro + (POSICION * 10)
                    Else
                        CadenadeControl[n] = 0
                        POSICION = CadenadeControl[n]
                        ValorCuatro = ValorCuatro + POSICION
                    EndIf
                Case 49

                    'UNO "1"

                    If cnt4 = 0 Then
                        CadenadeControl[n] = 1
                        POSICION = CadenadeControl[n]
                        ValorCuatro = POSICION * 1000
                    ElseIf cnt4 = 1 Then
                        CadenadeControl[n] = 1
                        POSICION = CadenadeControl[n]
                        ValorCuatro = ValorCuatro + (POSICION * 100)
                    ElseIf cnt4 = 2 Then
                        CadenadeControl[n] = 1
                        POSICION = CadenadeControl[n]
                        ValorCuatro = ValorCuatro + (POSICION * 10)
                    Else
                        CadenadeControl[n] = 1

```

```

        POSICION = CadenadeControl[n]
        ValorCuatro = ValorCuatro + POSICION
    EndIf
    Case 50
        If cnt4 = 0 Then
            CadenadeControl[n] = 2
            POSICION = CadenadeControl[n]
            ValorCuatro = POSICION * 1000
        ElseIf cnt4 = 1 Then
            CadenadeControl[n] = 2
            POSICION = CadenadeControl[n]
            ValorCuatro = ValorCuatro + (POSICION * 100)
        ElseIf cnt4 = 2 Then
            CadenadeControl[n] = 2
            POSICION = CadenadeControl[n]
            ValorCuatro = ValorCuatro + (POSICION * 10)
        Else
            CadenadeControl[n] = 2
            POSICION = CadenadeControl[n]
            ValorCuatro = ValorCuatro + POSICION
        EndIf
    Case 51
        If cnt4 = 0 Then
            CadenadeControl[n] = 3
            POSICION = CadenadeControl[n]
            ValorCuatro = POSICION * 1000
        ElseIf cnt4 = 1 Then
            CadenadeControl[n] = 3
            POSICION = CadenadeControl[n]
            ValorCuatro = ValorCuatro + (POSICION * 100)
        ElseIf cnt4 = 2 Then
            CadenadeControl[n] = 3
            POSICION = CadenadeControl[n]
            ValorCuatro = ValorCuatro + (POSICION * 10)
        Else
            CadenadeControl[n] = 3
            POSICION = CadenadeControl[n]
            ValorCuatro = ValorCuatro + POSICION
        EndIf
    Case 52
        If cnt4 = 0 Then
            CadenadeControl[n] = 4
            POSICION = CadenadeControl[n]
            ValorCuatro = POSICION * 1000
        ElseIf cnt4 = 1 Then
            CadenadeControl[n] = 4
            POSICION = CadenadeControl[n]
            ValorCuatro = ValorCuatro + (POSICION * 100)
        ElseIf cnt4 = 2 Then
            CadenadeControl[n] = 4
            POSICION = CadenadeControl[n]
            ValorCuatro = ValorCuatro + (POSICION * 10)
        Else
            CadenadeControl[n] = 4
            POSICION = CadenadeControl[n]
            ValorCuatro = ValorCuatro + POSICION
        EndIf

```

'DOS "2"

'TRES "3"

'CUATRO "4"

'CINCO "5"

Case 53

```
If cnt4 = 0 Then
    CadenadeControl[n] = 5
    POSICION = CadenadeControl[n]
    ValorCuatro = POSICION * 1000
ElseIf cnt4 = 1 Then
    CadenadeControl[n] = 5
    POSICION = CadenadeControl[n]
    ValorCuatro = ValorCuatro + (POSICION * 100)
ElseIf cnt4 = 2 Then
    CadenadeControl[n] = 5
    POSICION = CadenadeControl[n]
    ValorCuatro = ValorCuatro + (POSICION * 10)
Else
    CadenadeControl[n] = 5
    POSICION = CadenadeControl[n]
    ValorCuatro = ValorCuatro + POSICION
EndIf
```

'SEIS "6"

Case 54

```
If cnt4 = 0 Then
    CadenadeControl[n] = 6
    POSICION = CadenadeControl[n]
    ValorCuatro = POSICION * 1000
ElseIf cnt4 = 1 Then
    CadenadeControl[n] = 6
    POSICION = CadenadeControl[n]
    ValorCuatro = ValorCuatro + (POSICION * 100)
ElseIf cnt4 = 2 Then
    CadenadeControl[n] = 6
    POSICION = CadenadeControl[n]
    ValorCuatro = ValorCuatro + (POSICION * 10)
Else
    CadenadeControl[n] = 6
    POSICION = CadenadeControl[n]
    ValorCuatro = ValorCuatro + POSICION
EndIf
```

'SIETE "7"

Case 55

```
If cnt4 = 0 Then
    CadenadeControl[n] = 7
    POSICION = CadenadeControl[n]
    ValorCuatro = POSICION * 1000
ElseIf cnt4 = 1 Then
    CadenadeControl[n] = 7
    POSICION = CadenadeControl[n]
    ValorCuatro = ValorCuatro + (POSICION * 100)
ElseIf cnt4 = 2 Then
    CadenadeControl[n] = 7
    POSICION = CadenadeControl[n]
    ValorCuatro = ValorCuatro + (POSICION * 10)
Else
    CadenadeControl[n] = 7
    POSICION = CadenadeControl[n]
    ValorCuatro = ValorCuatro + POSICION
EndIf
```

'OCHO "8"

Case 56

```
If cnt4 = 0 Then
    CadenadeControl[n] = 8
```

```

        POSICION = CadenadeControl[n]
        ValorCuatro = POSICION * 1000
    ElseIf cnt4 = 1 Then
        CadenadeControl[n] = 8
        POSICION = CadenadeControl[n]
        ValorCuatro = ValorCuatro + (POSICION * 100)
    ElseIf cnt4 = 2 Then
        CadenadeControl[n] = 8
        POSICION = CadenadeControl[n]
        ValorCuatro = ValorCuatro + (POSICION * 10)
    Else
        CadenadeControl[n] = 8
        POSICION = CadenadeControl[n]
        ValorCuatro = ValorCuatro + POSICION
    EndIf
    Case 57

'NUEVE "9"

    If cnt4 = 0 Then
        CadenadeControl[n] = 9
        POSICION = CadenadeControl[n]
        ValorCuatro = POSICION * 1000
    ElseIf cnt4 = 1 Then
        CadenadeControl[n] = 9
        POSICION = CadenadeControl[n]
        ValorCuatro = ValorCuatro + (POSICION * 100)
    ElseIf cnt4 = 2 Then
        CadenadeControl[n] = 9
        POSICION = CadenadeControl[n]
        ValorCuatro = ValorCuatro + (POSICION * 10)
    Else
        CadenadeControl[n] = 9
        POSICION = CadenadeControl[n]
        ValorCuatro = ValorCuatro + POSICION
    EndIf
    Case Else
        retfie fast
    EndSelect
    Inc n
    Inc cnt4
    Wend
    Clear cnt4

    If porpasos = 1 Then
        PASOS = ValorCuatro
        GoTo FINDELINEA
    Else
        Clear n
        retfie fast
    EndIf
    retfie fast

'***** RUTINA PARA OBTENER UN VALOR DE 5
DÍGITOS *****
OBTEN5DIG:    While cnt5<5
                Select CadenadeControl[n]
                Case 48

'CERO "0"

                If cnt5 = 0 Then
                    CadenadeControl[n] = 0
                    POSICION = CadenadeControl[n]
                    ValorCinco = POSICION * 10000

```

```

ElseIf cnt5 = 1 Then
    CadenadeControl[n] = 0
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 1000)
ElseIf cnt5 = 2 Then
    CadenadeControl[n] = 0
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 100)
ElseIf cnt5 = 3 Then
    CadenadeControl[n] = 0
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 10)
Else
    CadenadeControl[n] = 0
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + POSICION
EndIf
Case 49

'UNO "1"

If cnt5 = 0 Then
    CadenadeControl[n] = 1
    POSICION = CadenadeControl[n]
    ValorCinco = POSICION * 10000
ElseIf cnt5 = 1 Then
    CadenadeControl[n] = 1
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 1000)
ElseIf cnt5 = 2 Then
    CadenadeControl[n] = 1
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 100)
ElseIf cnt5 = 3 Then
    CadenadeControl[n] = 1
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 10)
Else
    CadenadeControl[n] = 1
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + POSICION
EndIf
Case 50

'DOS "2"

If cnt5 = 0 Then
    CadenadeControl[n] = 2
    POSICION = CadenadeControl[n]
    ValorCinco = POSICION * 10000
ElseIf cnt5 = 1 Then
    CadenadeControl[n] = 2
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 1000)
ElseIf cnt5 = 2 Then
    CadenadeControl[n] = 2
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 100)
ElseIf cnt5 = 3 Then
    CadenadeControl[n] = 2
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 10)
Else
    CadenadeControl[n] = 2
    POSICION = CadenadeControl[n]

```

```

        ValorCinco = ValorCinco + POSICION
    EndIf
    Case 51
        'TRES "3"
        If cnt5 = 0 Then
            CadenadeControl[n] = 3
            POSICION = CadenadeControl[n]
            ValorCinco = POSICION * 10000
        ElseIf cnt5 = 1 Then
            CadenadeControl[n] = 3
            POSICION = CadenadeControl[n]
            ValorCinco = ValorCinco + (POSICION * 1000)
        ElseIf cnt5 = 2 Then
            CadenadeControl[n] = 3
            POSICION = CadenadeControl[n]
            ValorCinco = ValorCinco + (POSICION * 100)
        ElseIf cnt5 = 3 Then
            CadenadeControl[n] = 3
            POSICION = CadenadeControl[n]
            ValorCinco = ValorCinco + (POSICION * 10)
        Else
            CadenadeControl[n] = 3
            POSICION = CadenadeControl[n]
            ValorCinco = ValorCinco + POSICION
        EndIf
        Case 52
            'CUATRO "4"
            If cnt5 = 0 Then
                CadenadeControl[n] = 4
                POSICION = CadenadeControl[n]
                ValorCinco = POSICION * 10000
            ElseIf cnt5 = 1 Then
                CadenadeControl[n] = 4
                POSICION = CadenadeControl[n]
                ValorCinco = ValorCinco + (POSICION * 1000)
            ElseIf cnt5 = 2 Then
                CadenadeControl[n] = 4
                POSICION = CadenadeControl[n]
                ValorCinco = ValorCinco + (POSICION * 100)
            ElseIf cnt5 = 3 Then
                CadenadeControl[n] = 4
                POSICION = CadenadeControl[n]
                ValorCinco = ValorCinco + (POSICION * 10)
            Else
                CadenadeControl[n] = 4
                POSICION = CadenadeControl[n]
                ValorCinco = ValorCinco + POSICION
            EndIf
            Case 53
                'CINCO "5"
                If cnt5 = 0 Then
                    CadenadeControl[n] = 5
                    POSICION = CadenadeControl[n]
                    ValorCinco = POSICION * 10000
                ElseIf cnt5 = 1 Then
                    CadenadeControl[n] = 5
                    POSICION = CadenadeControl[n]
                    ValorCinco = ValorCinco + (POSICION * 1000)
                ElseIf cnt5 = 2 Then
                    CadenadeControl[n] = 5
                    POSICION = CadenadeControl[n]

```

```

        ValorCinco = ValorCinco + (POSICION * 100)
ElseIf cnt5 = 3 Then
    CadenadeControl[n] = 5
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 10)
Else
    CadenadeControl[n] = 5
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + POSICION
EndIf
Case 54

'SEIS "6"

If cnt5 = 0 Then
    CadenadeControl[n] = 6
    POSICION = CadenadeControl[n]
    ValorCinco = POSICION * 10000
ElseIf cnt5 = 1 Then
    CadenadeControl[n] = 6
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 1000)
ElseIf cnt5 = 2 Then
    CadenadeControl[n] = 6
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 100)
ElseIf cnt5 = 3 Then
    CadenadeControl[n] = 6
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 10)
Else
    CadenadeControl[n] = 6
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + POSICION
EndIf
Case 55

'SIETE "7"

If cnt5 = 0 Then
    CadenadeControl[n] = 7
    POSICION = CadenadeControl[n]
    ValorCinco = POSICION * 10000
ElseIf cnt5 = 1 Then
    CadenadeControl[n] = 7
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 1000)
ElseIf cnt5 = 2 Then
    CadenadeControl[n] = 7
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 100)
ElseIf cnt5 = 3 Then
    CadenadeControl[n] = 7
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 10)
Else
    CadenadeControl[n] = 7
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + POSICION
EndIf
Case 56

'OCHO "8"

If cnt5 = 0 Then
    CadenadeControl[n] = 8
    POSICION = CadenadeControl[n]

```

'NUEVE "9"

```
ValorCinco = POSICION * 10000
ElseIf cnt5 = 1 Then
    CadenadeControl[n] = 8
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 1000)
ElseIf cnt5 = 2 Then
    CadenadeControl[n] = 8
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 100)
ElseIf cnt5 = 3 Then
    CadenadeControl[n] = 8
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 10)
Else
    CadenadeControl[n] = 8
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + POSICION
EndIf
Case 57

If cnt5 = 0 Then
    CadenadeControl[n] = 9
    POSICION = CadenadeControl[n]
    ValorCinco = POSICION * 10000
ElseIf cnt5 = 1 Then
    CadenadeControl[n] = 9
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 1000)
ElseIf cnt5 = 2 Then
    CadenadeControl[n] = 9
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 100)
ElseIf cnt5 = 3 Then
    CadenadeControl[n] = 9
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + (POSICION * 10)
Else
    CadenadeControl[n] = 9
    POSICION = CadenadeControl[n]
    ValorCinco = ValorCinco + POSICION
EndIf
Case Else
    retfie fast
EndSelect
Inc n
Inc cnt5
Wend
Clear cnt5

If NoCodigoG = 0 Then
    If ejex = 1 Then
        If Positivo = 1 Then
            Xdda = ValorCinco * (1.0)
            GoTo PARAMETROS
        Else
            Xdda = ValorCinco * (-1.0)
            GoTo PARAMETROS
        EndIf
    ElseIf ejey = 1 Then
        If Positivo = 1 Then
            Ydda = ValorCinco * (1.0)
```

```

        GoTo PARAMETROS
    Else
        Ydda = ValorCinco * (-1.0)
        GoTo PARAMETROS
    EndIf
ElseIf ejez = 1 Then
    If Positivo = 1 Then
        Zdda = ValorCinco * (1.0)
        GoTo PARAMETROS
    Else
        Zdda = ValorCinco * (-1.0)
        GoTo PARAMETROS
    EndIf
Else
    Clear n
    retfie fast
EndIf
ElseIf NoCodigoG = 1 Then
    If ejex = 1 Then
        If Positivo = 1 Then
            Xdda = ValorCinco * (1.0)
            GoTo PARAMETROS
        Else
            Xdda = ValorCinco * (-1.0)
            GoTo PARAMETROS
        EndIf
    ElseIf ejey = 1 Then
        If Positivo = 1 Then
            Ydda = ValorCinco * (1.0)
            GoTo PARAMETROS
        Else
            Ydda = ValorCinco * (-1.0)
            GoTo PARAMETROS
        EndIf
    Else
        Clear n
        retfie fast
    EndIf
ElseIf NoCodigoG = 28 Then
    If ejex = 1 Then
        If Positivo = 1 Then
            Xdda = ValorCinco * (1.0)
            GoTo PARAMETROS
        Else
            Xdda = ValorCinco * (-1.0)
            GoTo PARAMETROS
        EndIf
    ElseIf ejey = 1 Then
        If Positivo = 1 Then
            Ydda = ValorCinco * (1.0)
            GoTo PARAMETROS
        Else
            Ydda = ValorCinco * (-1.0)
            GoTo PARAMETROS
        EndIf
    ElseIf ejez = 1 Then
        If Positivo = 1 Then
            Zdda = ValorCinco * (1.0)
            GoTo PARAMETROS
        Else
            Zdda = ValorCinco * (-1.0)

```

```

        GoTo PARAMETROS
    EndIf
Else
    Clear n
    retfie fast
EndIf
ElseIf NoCodigoG = 2 Then
    If ejex = 1 Then
        If Positivo = 1 Then
            Xdda = ValorCinco * (1.0)
            GoTo PARAMETROS
        Else
            Xdda = ValorCinco * (-1.0)
            GoTo PARAMETROS
        EndIf
    ElseIf ejey = 1 Then
        If Positivo = 1 Then
            Ydda = ValorCinco * (1.0)
            GoTo PARAMETROS
        Else
            Ydda = ValorCinco * (-1.0)
            GoTo PARAMETROS
        EndIf
    ElseIf coordenadai = 1 Then
        If Positivo = 1 Then
            Idda = ValorCinco * (1.0)
            GoTo PARAMETROS
        Else
            Idda = ValorCinco * (-1.0)
            GoTo PARAMETROS
        EndIf
    ElseIf coordenadaj = 1 Then
        If Positivo = 1 Then
            Jdda = ValorCinco * (1.0)
            GoTo PARAMETROS
        Else
            Jdda = ValorCinco * (-1.0)
            GoTo PARAMETROS
        EndIf
    Else
        Clear n
        retfie fast
    EndIf
ElseIf NoCodigoG = 3 Then
    If ejex = 1 Then
        If Positivo = 1 Then
            Xdda = ValorCinco * (1.0)
            GoTo PARAMETROS
        Else
            Xdda = ValorCinco * (-1.0)
            GoTo PARAMETROS
        EndIf
    ElseIf ejey = 1 Then
        If Positivo = 1 Then
            Ydda = ValorCinco * (1.0)
            GoTo PARAMETROS
        Else
            Ydda = ValorCinco * (-1.0)
            GoTo PARAMETROS
        EndIf
    ElseIf coordenadai = 1 Then

```

```

        If Positivo = 1 Then
            Idda = ValorCinco * (1.0)
            GoTo PARAMETROS
        Else
            Idda = ValorCinco * (-1.0)
            GoTo PARAMETROS
        EndIf
    ElseIf coordenadaj = 1 Then
        If Positivo = 1 Then
            Jdda = ValorCinco * (1.0)
            GoTo PARAMETROS
        Else
            Jdda = ValorCinco * (-1.0)
            GoTo PARAMETROS
        EndIf
    Else
        Clear n
        retfie fast
    EndIf
Else
    Clear n
    retfie fast
EndIf

! ***** RUTINA DE ALTO *****
*****
ALTO:      Clear n
           Clear Positivo
           Clear joystick
           Clear porpasos
           Clear porfrecuencia
           Clear partedecimal
           Clear home
           Clear homex
           Clear mascarahomex
           Clear homey
           Clear mascarahomey
           Clear homez
           Clear mascarahomez
           Clear ejex
           Clear mascaraejex
           Clear ejey
           Clear mascaraejey
           Clear ejez
           Clear mascaraejez
           Clear ejeh
           Clear coordenadai
           Clear mascaracoordi
           Clear coordenadaj
           Clear mascaracoordj
           Clear lineacodigog
           Clear lineacodigom
           Clear diopasoenx
           Clear diopasoeny
           Clear diopasoenz
           Clear diopasoenh
           Clear numerocodigo
           Clear solouneje
           Clear masdeuneje
           Clear iniciodecomando
           Clear DAT0enviado

```

Clear DAT0entrada
Clear cnt5
Clear cnt4
Clear cnt3
Clear cnt2
Fcia = 170
Clear Pdo
Clear POSICION
Clear PASOS
Clear ValorCinco
Clear ValorCuatro
Clear ValorTres
Clear ValorDos
Clear Avance
Clear CUADRANTE
Clear CUADRANTEFINAL
Clear noavance
Clear Xdda
Clear PXdec
Clear Ydda
Clear PYdec
Clear Zdda
Clear PZdec
Clear Idda
Clear Xodec
Clear Jdda
Clear Yodec
Clear PPR
Clear PX
Clear PY
Clear PZ
Clear QX
Clear QY
Clear ContadorX
Clear ContadorY
Clear HIPO
Clear Radio
Clear Xo
Clear Yo
Clear D
Clear D1
Clear D2
Clear D3
Clear DeltaX
Clear DeltaY
Clear Pausa
Clear NoPausa
Clear Sangria
Clear mascaraSangria
Clear backlash
Clear JgoMecanico
Clear dummy1
Clear dummy2
Clear dummy3
Clear dummy4
Clear dummy5
Clear dummy6
Clear dummy7
Clear dummy8
Clear dummy9
Clear dummy10

```

Clear dummy11
Clear dummy12
Clear dummy13
Clear dummy14
Clear dummy15
Clear dummy16
NoCodigoM = 0
NoCodigoG = 99
retfie fast

/***** RUTINA DE CONFIGURACIÓN PARA
INICIAR EL PROGRA *****/
INICIO: Clear CadenadeControl
Clear n
Clear Positivo
Clear joystick
Clear porpasos
Clear porfrecuencia
Clear partedecimal
Clear home
Clear homex
Clear mascarahomex
Clear homey
Clear mascarahomey
Clear homez
Clear mascarahomez
Clear ejex
Clear mascaraejex
Clear ejey
Clear mascaraejey
Clear ejez
Clear mascaraejez
Clear ejeh
Clear coordenadai
Clear mascaracoordi
Clear coordenadaj
Clear mascaracoordj
Clear sentidox
Clear sentidoy
Set sentidoz
Clear sentidoxant
Clear sentidoyant
Set sentidozant
Clear sentidoh
Clear sentidoi
Clear sentidoj
Clear cerodepieza
Clear lineacodigog
Clear lineacodigom
Clear diopasoenx
Clear diopasoeny
Clear diopasoenz
Clear diopasoenh
Clear numerocodigo
Clear solouneje
Clear masdeuneje
Clear iniciodecomando
Clear sistemaabsoluto
Clear NoPausa
Clear Sangria
Clear mascaraSangria

```

Clear backlash
Clear JgoMecanicoX
Clear JgoMecanicoY
Clear JgoMecanicoZ
Clear COMBINACION
Clear CUADRANTE
Clear CUADRANTEFINAL
Clear DATOenviado
Clear DATOentrada
Clear Avance
Clear noavance
Clear cnt5
Clear cnt4
Clear cnt3
Clear cnt2
Clear Fcia
Clear Pdo
Clear Xm
Clear Ym
Clear Zm
Clear Xp
Clear Yp
Clear Zp
Clear Xcp
Clear Ycp
Clear Zcp
Clear POSICION
Clear PASOS
Clear ValorCinco
Clear ValorCuatro
Clear ValorTres
Clear ValorDos
Clear Xdda
Clear PXdec
Clear Ydda
Clear PYdec
Clear Zdda
Clear PZdec
Clear Idda
Clear Xodec
Clear Jdda
Clear Yodec
Clear PPR
Clear PX
Clear PY
Clear PZ
Clear HIPO
Clear Radio
Clear Xo
Clear Yo
Clear D
Clear D1
Clear D2
Clear D3
Clear DeltaX
Clear DeltaY
Clear QX
Clear QY
Clear ContadorX
Clear ContadorY
Clear Pausa

```

Clear JgoMecanico
Clear dummy1
Clear dummy2
Clear dummy3
Clear dummy4
Clear dummy5
Clear dummy6
Clear dummy7
Clear dummy8
Clear dummy9
Clear dummy10
Clear dummy11
Clear dummy12
Clear dummy13
Clear dummy14
Clear dummy15
Clear dummy16
JgoMecanicoX = 1436      '(204um)
JgoMecanicoY = 1264     '(178um)
JgoMecanicoZ = 1        '(140nm)
NoCodigoM = 0
NoCodigoG = 99
RESOL = 0.14            '0.1
ESTADOx = 4
ESTADOy = 4
ESTADOz = 4
ESTADOh = 4

Output PORTD
Clear LATD
Clear TRISD
Output PORTC.6
Input PORTC.7

Input PORTE.7
Input PORTE.6
Input PORTE.5
Input PORTE.4
Input PORTE.3
Input PORTE.2
Set LATE
Set TRISE
Clear ADCON0.0
Set ADCON1

Input PORTA.0
Input PORTA.1
Set LATA
Set TRISA
Clear ADCON0.0
Set ADCON1
PORTD = %00000000
PORTA = %00000000

PORTE = %00000000

Input PORTJ.1
Input PORTJ.0
Set LATJ
Set TRISJ

```

SPBRG = 31

Set TXEN
Clear SYNC
Clear BRGH
Set SPEN
Set CREN
Clear RC1IF
Set RC1IE
Set PEIE
Set GIE

ESPERA: **GoTo** ESPERA