UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

T E S I S

"DISEÑO DE UN SISTEMA DE CONTROL DE UNA MICROFABRICA BASADO EN UN DSP"

PRESENTADA POR:

OSCAR CUEVAS GUTIERREZ

PARA OBTENER EL GRADO DE

INGENIERO ELÉCTRICO Y ELECTRÓNICO
(AREA ELECTRÓNICA DIGITAL)

DIRECTORES:
DR. ALBERTO CABALLERO RUIZ
DR. LEOPOLDO RUIZ HUERTA

AGRADECIMIENTOS

A DIOS,
Gracias por haberme concedido
alcanzar la luz al final del túnel,
la que en algún momento me pareció muy lejana

A MIS PADRES,
A mi mamá Socorrito, y sabes estoy orgulloso
de que seas mi mamá, ya que todo lo que soy lo soy gracias a ti.
Gracias por haberme inculcado el gusto por la Escuela
siempre serás mi mama y mi primer maestra.
A mi papá Enrique, gracias por haberme tenido paciencia
y por apoyarme durante toda mi vida.
Gracias por no haberme fallado nunca como un buen padre.

A MIS HERMANOS
Juan Enrique, eres un Ejemplo muy difícil de seguir
Porque nunca te detienes, cada día cosechas un logro nuevo
Pero tus consejos son una motivación para seguir adelante.
Caty, gracias por toda esa vida
que hemos compartido, gracias por estar ahí en mis momentos tristes
y en mis momentos alegres.

A ADRIANA
Chaparrita, gracias por haber llegado a mi vida
Porque la llenaste de amor y ternura que solo un corazón amoroso puede dar.
Gracias por dejarme compartir contigo este logro tan importante.

A MIS TUTORES
Beto y Polo, gracias por haberme tenido tanta paciencia
Gracias por todo lo que aprendí de ustedes, ya que son
Profesionales muy exitosos y personas con un trato muy fino.

A MIS AMIGOS
Nora, Silvestre, Carlitos, José Luis, Luis, Libertad, Triny, Erika, Ricardo,
Fabiola, Gilberto y a todas aquellas grandes personas que conocí
Durante mi estancia en la UNAM y que hicieron que
ésta fuera una de las mejores etapas de mi vida.

A MIS COMPAÑEROS DEL LMM
A Héctor, Angélica, Bogar, Daniela, Guillermo, Oscar, Javier, Jorge
Gracias por ese ambiente de camaradería
y por hacer muy agradable el ambiente del laboratorio.
Sigan siendo las excelentes personas y profesionales que son hasta ahora.

A MIS SINODALES,
Gracias por sus comentarios, correcciones y aportaciones.

ÍNDICE

Introducción 1

1. Antecedentes
   - 1.1. Desarrollo de la micromecánica en el mundo 3
   - 1.2. Desarrollo de micromecánica en el CCADET-UNAM 10
   - 1.3. Introducción a los sistemas de control 13
     - 1.3.1. Control lazo abierto 14
     - 1.3.2. Control en lazo cerrado 15
     - 1.3.3. Algunos sistemas de control desarrollados en el LMM 16
     - 1.3.4. Sistema basado en PC 17
     - 1.3.5. Sistema basado en FPGA 18
     - 1.3.6. Sistema basado en PIC 20
     - 1.3.7. Sistema basado en Código G y M 21

2. Planteamiento del problema 24
   - 2.1. Necesidades 24
   - 2.2. Análisis de los sistemas de control para MMHs 24
     - 2.2.1. Sistema basado en PC 24
     - 2.2.2. Sistema basado en FPGA 25
     - 2.2.3. Sistema basado en PIC 25
     - 2.2.4. Sistema basado en Código G y M 25
   - 2.3. Nueva propuesta de sistema de control 26
   - 2.4. Objetivo 27
   - 2.5. Alcances 27

3. Desarrollo 28
   - 3.1. Actuadores en las MMH 29
   - 3.2. Selección de la etapa de potencia 32
   - 3.3. Selección del circuito secuenciador 35
   - 3.4. Etapa de potencia y secuenciador propuesta 40
   - 3.5. Selección del dispositivo controlador 44
     - 3.5.1. Microcontroladores y Microprocesadores 44
     - 3.5.2. Procesadores Digitales de Señales 47
     - 3.5.3. Introducción al DSP56F8357 49
   - 3.6. Diseño de la etapa de aislamiento 54
   - 3.7. Desarrollo del programa 59
     - 3.7.1. Declaración de registros variables generales 61
     - 3.7.2. Inicialización de registros y variables generales 62
     - 3.7.3. Inicialización de las señales de control 63
     - 3.7.4. Selección de modo 63
   - 3.8. descripción de la interfaz de usuario 77

4. Pruebas y Resultados 82
   - 4.1. Formas de onda 82
   - 4.2. Consumo de corriente en vacío 86
   - 4.3. Aumento de la temperatura vs tiempo 86
   - 4.4. Velocidad – consumo de corriente 88
   - 4.5. Maquinados 91
   - 4.6. Prueba de incremento de temperatura 92
   - 4.7. Consumo de corriente en vacío 93

5. Conclusiones 94
5.1. Trabajo a futuro 95

6. Referencias 96

INTRODUCCIÓN

En la actualidad, muchas áreas de la industria tienen fuertes tendencias a la miniaturización de productos [1]. Desde 1999, En el Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET), el Grupo de Micromecánica y Mecatrónica (GMM), ha estado trabajando en el desarrollo de dispositivos de bajo costo que mantengan una alta exactitud y precisión [2].

Como parte de esta línea de investigación, se ha desarrollado microequipo mecánico a través de generaciones secuenciales, que tiene como características generales, sencillez de manufactura y diseño, así como un bajo costo de producción y consumo energético [3]. Estos primeros prototipos están enfocados a crear un centro de maquinado y manipuladores diversos, teniendo como meta la automatización total de un proceso productivo.

Actualmente, se han desarrollado diversos sistemas de control para las micromáquinas herramienta desarrolladas en el GMM. Estos sistemas emplean dispositivos como la PC, FPGA y PICs. Buscando evaluar otra alternativa, se desarrolló el presente tema de tesis que se refiere a la implementación de un sistema de control basado en procesadores digitales de señales (DSPs), utilizando como caso de estudio, el primer prototipo desarrollado en el GMM.

Este trabajo se divide en capítulos, en el Capítulo 1 se mencionan las tecnologías existentes en el mundo para la fabricación de dispositivos compactos, a nivel mundial y en México; Se hace una descripción de la micromáquina que se utilizará como caso de estudio y de los sistemas de control previamente implementados en el GMM. Finalmente se hace una introducción a los sistemas de control.

En el Capítulo 2 se presenta el planteamiento del problema, se da una descripción de las necesidades que hay que cubrir para el control de la micromáquina, el objetivo y el alcance de este trabajo.

En el Capítulo 3 se encuentra el desarrollo de la solución al sistema de control. En él se presentan los dispositivos, circuitos y programas propuestos para cubrir el objetivo propuesto en el capítulo 2.

En el Capítulo 4 se presentan los resultados obtenidos en la realización de este trabajo y se hace una comparación con los objetivos planteados y se hace un par de pruebas para comparar este sistema con los desarrollados anteriormente.

En el Capítulo 5 se encuentran las conclusiones y el trabajo a futuro propuesto para la continuación y mejora de este proyecto.

Capítulo 1

ANTECEDENTES

1.1 DESARROLLO DE MICROMECANICA EN EL MUNDO

Actualmente, la tecnología tiende a compactar cada vez más los dispositivos y sistemas que produce, tal reducción se debe a que se necesitan sistemas o dispositivos muy pequeños para realizar tareas peligrosas o de difícil acceso para los humanos [3].

Una micromáquina es un sistema extremadamente pequeño (pocos milímetros), con elementos altamente sofisticados y funcionales que le permiten realizar tareas complicadas y delicadas (Dr. J. Lee, 1996) [4].

Las investigaciones y desarrollos relacionados con microdispositivos se dividen, dependiendo de las tecnologías empleadas para su fabricación y de la parte del mundo donde se desarrolla. Ejemplo de estas tecnologías son los *Micromechanical Systems* (MEMS), desarrollados principalmente en Estados Unidos, los *MicroSystem Technology* (MST) desarrollados principalmente en Europa y los *Micromachine Technology* (MMT) desarrollados en Japón, los cuales utilizan técnicas empleadas para la construcción de circuitos integrados de tecnología serie o por lotes para crear microelectrónica [4].

Los MEMS, son estructuras mecánicas con dimensiones de varias decenas de micras, que pueden construirse con el mismo proceso para la fabricación de circuitos integrados. Por lo cual, es posible colocar en un solo componente, dispositivos mecánicos con la electrónica necesaria para su funcionamiento.

Los dispositivos micromecánicos y MEMS son ampliamente empleados para diferentes aplicaciones. Las industrias automotriz y aeroespacial usan MEMS para producir diferentes sensores y actuadores. En óptica adaptativa y telecomunicaciones, los MEMS son usados para *interruptores* ópticos. Una

aplicación muy interesante de los MEMS es para *displays* basados en arreglos de microespejos que reflejan los rayos de luz externos para corresponder a los píxeles en una pantalla. Los MEMS permiten mejoramiento en las características de los intercambiadores de calor [1]. También son usados ampliamente en biotecnología, medicina, neurofisiología, etc.

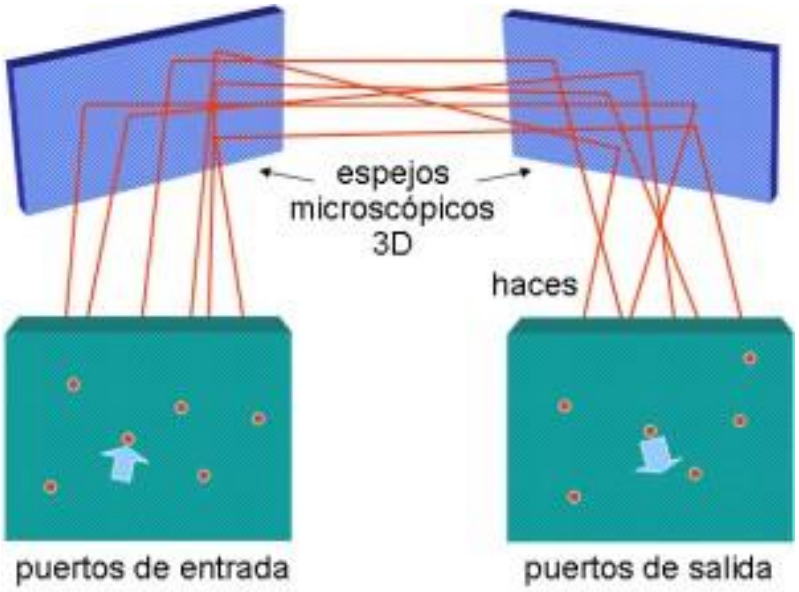
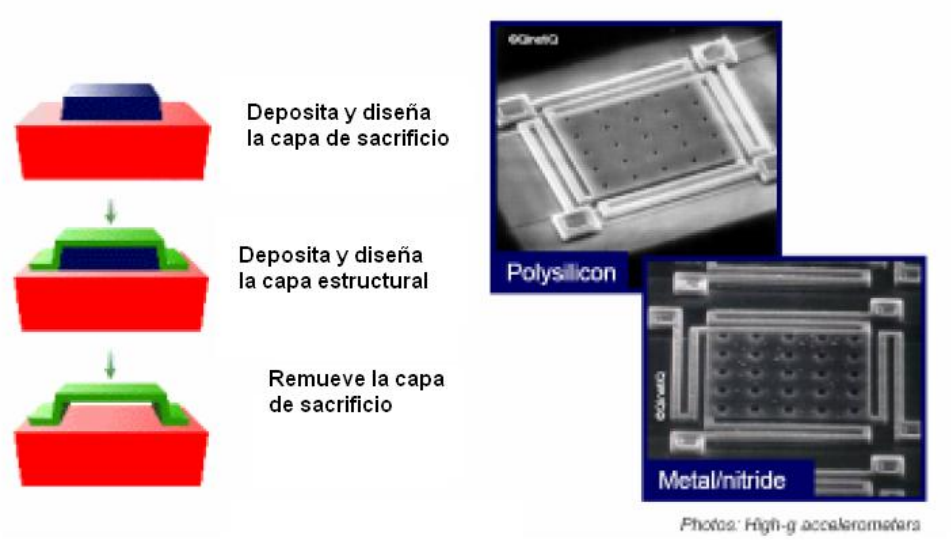


Figura 1.1 Conmutador óptico basado en MEMS

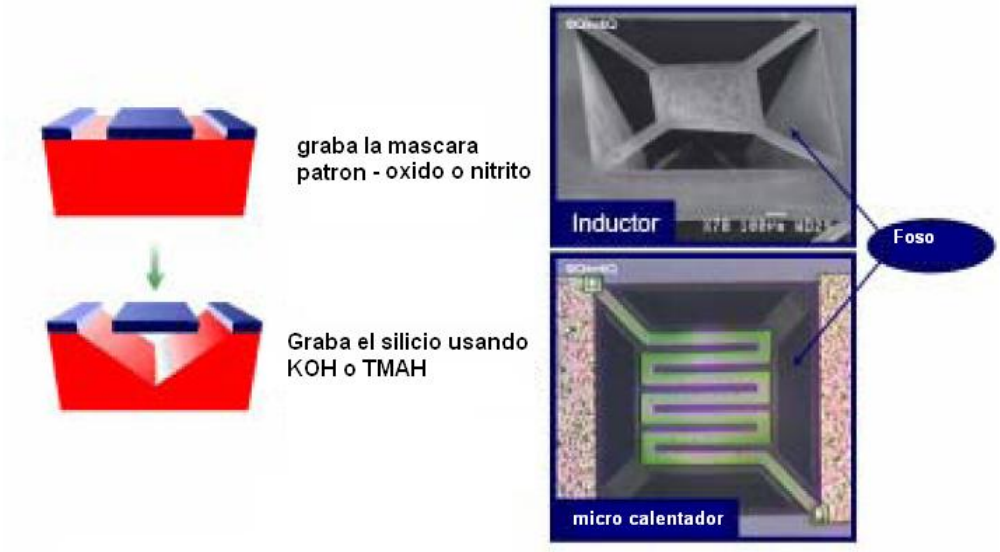
Los procesos para la fabricación de MEMS son: Micromaquinado superficial, en bulto y LIGA [5]. Con los dos primeros no se podía producir partes cónicas, tornillos u otros componentes con formas tridimensionales complejas, ya que, el micromaquinado superficial, es similar al que se usa para fabricar circuitos integrados. Para ello se usa un substrato de silicio y se empiezan a crecer las estructuras usando el proceso fotolitográfico, las estructuras son verticales. En el proceso de bulto, se usan químicos para ir desgastando el substrato, de tal manera, que se pueden generar secciones horizontales. En los dos procesos el substrato es un material semiconductor.

En el proceso LIGA, donde LIGA es el acrónimo del alemán Litografía, electrodepósito y modelado por rayos X, se crecen las estructuras y se les da un

acabado metálico. Se pueden generar piezas desde unas cuantas micras hasta varios centímetros. Al final, se ensamblan manualmente las piezas que conforman al componente. En la figura 1.3 se muestra el proceso de fabricación de por maquinado superficial y por bulto. En la figura 1.4 se muestra una propela hecha mediante el proceso LIGA.



(a)



(b)

Figura 1.3 (a) maquinado superficial, (b) maquinado por bulto [5]

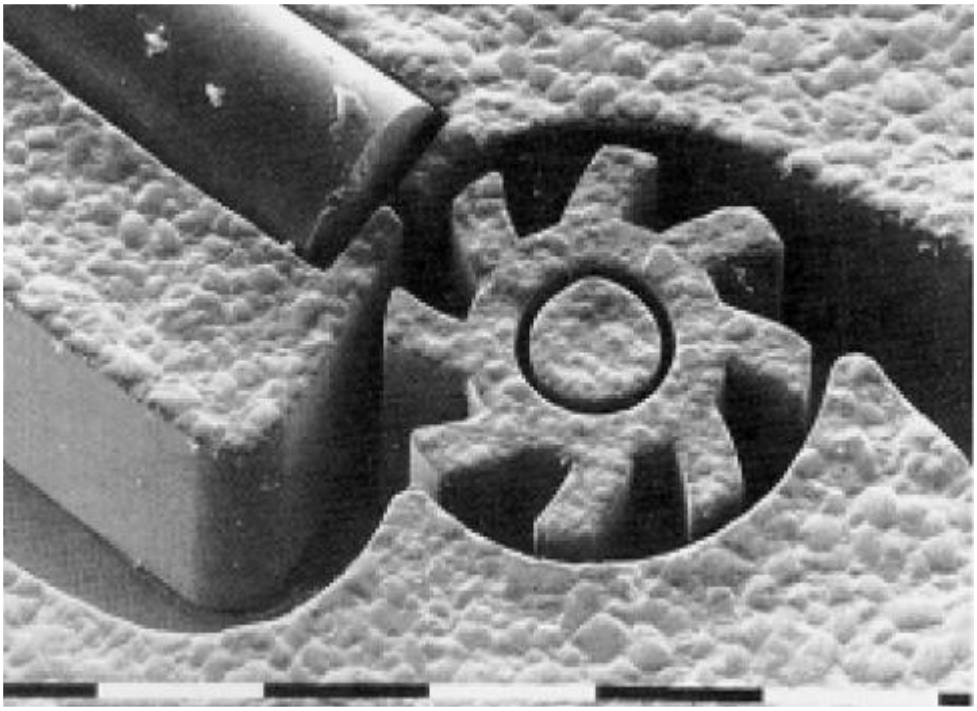


Figura 1.4 Propela hecha mediante el proceso LIGA [5]

Como una alternativa a las tecnologías antes mencionadas, en *el Mechanical Engineering Laboratory* (*MEL*) del *National Institute of Advanced Industrial Science and Technology* (*NIAIST*), se desarrolló una tecnología de fabricación de micromáquinas que tuvieran un tamaño comparable con las piezas que fabrican, esta relación entre el tamaño de la máquina herramienta y el tamaño de la pieza aseguraría la precisión requerida de detalles manufacturados, sin el uso de instrumentos y métodos superprecisos [6].

En 1996 el *MEL* desarrolló un microtorno [7]. Este microtorno, mostrado en la figura 1.5, mide 33 x 25 x 30.5 mm y su peso es alrededor de 100 g. El microtorno puede cortar aluminio con una precisión de 1.5 μm en la dirección de alimentación y 2.5 μm de redondez. El diámetro mínimo de trabajo que se ha logrado es de 60 μm. Posteriormente el microtorno fue equipado con un control numérico como las máquinas herramientas a escala humana, con lo que su precisión de maquinado se ha mejorado.

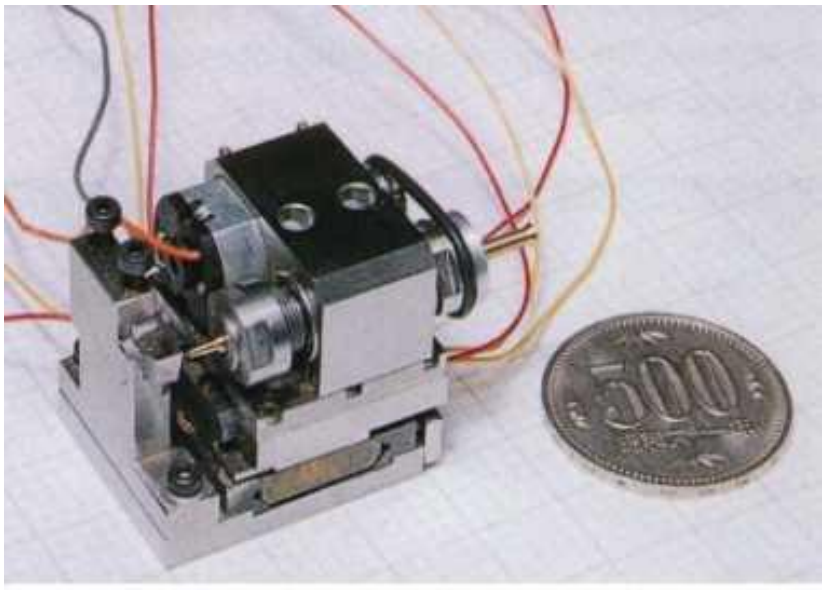


Figura 1.5 Microtorno Japonés

El microtorno forma parte de una microfábrica. Este término se refiere a un sistema de producción adecuado para fabricar dispositivos de tamaños inferiores al milímetro. Una microfábrica de escritorio, cuyo esquema se muestra en la figura 1.6, contiene máquinas herramienta para manufactura, dispositivos transportadores y de ensamble, robots para alimentar las máquinas herramienta, dispositivos para inspeccionar la calidad, sistemas de eliminación de desperdicio, sistemas de inspección de condiciones, sistemas de reemplazo de herramientas y todo esto en un área pequeña [6].

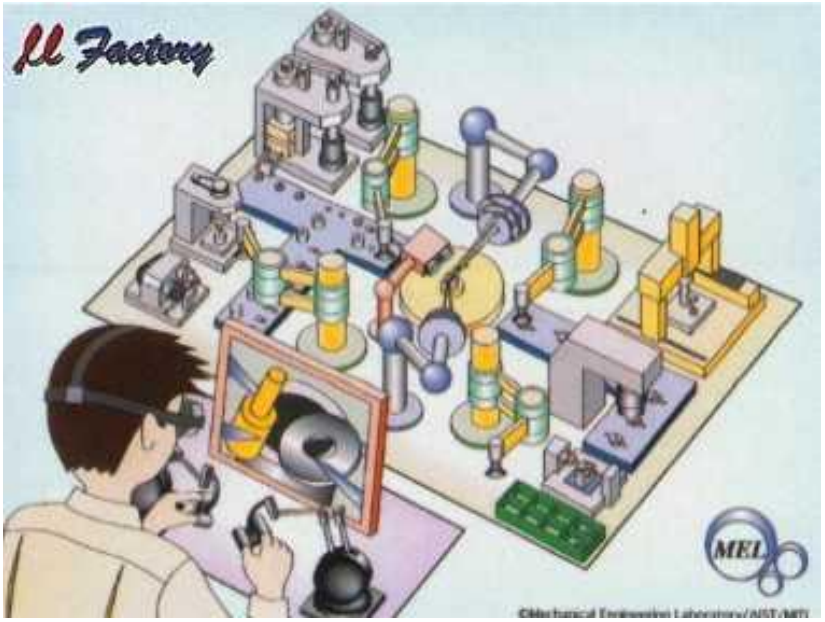


Figura 1.6 Esquema de una microfábrica

La microfábrica Japonesa, consiste en un microtorno, una microfresadora, una microprensa y máquinas de ensamblado como un brazo de transferencia y una mano con dos dedos mostradas en la figura 1.7. [6]

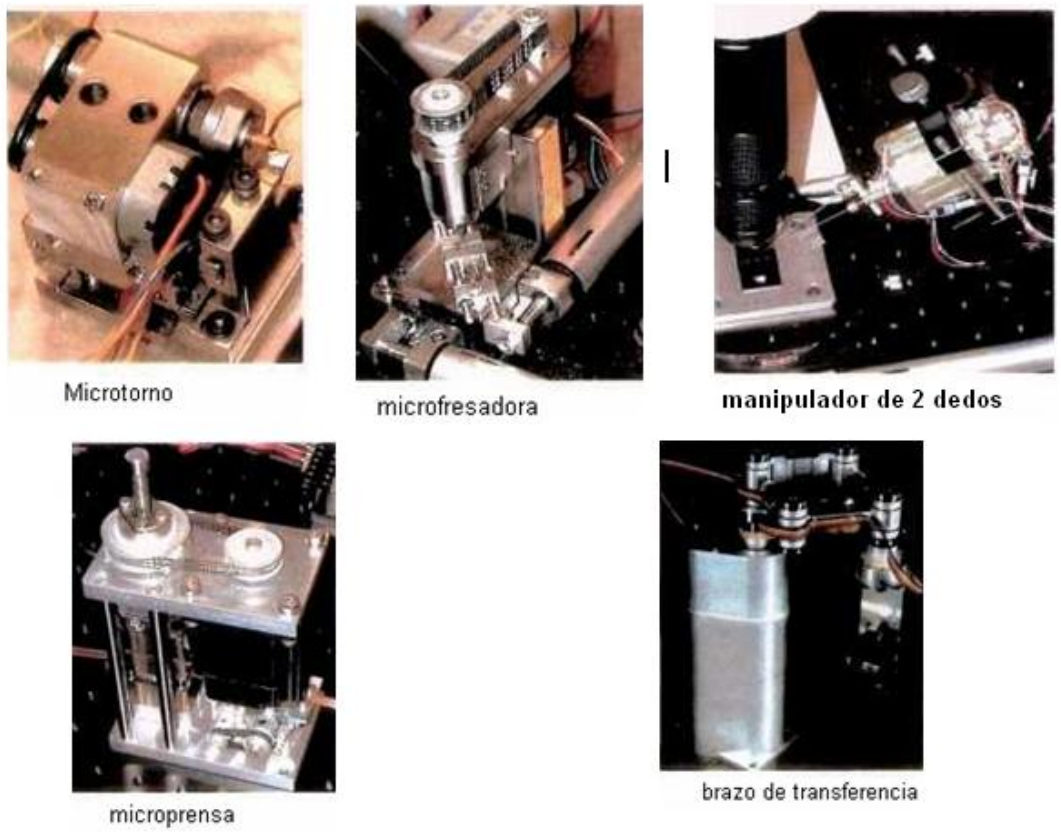


Figura 1.7 Micromáquinas de la Microfábrica Japonesa

Esta microfábrica puede producir y ensamblar las partes mecánicas miniaturizadas como las mostradas en la figura 1.8.

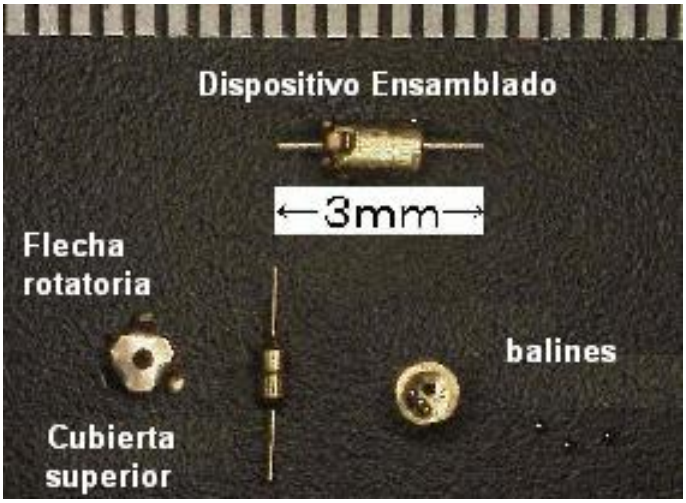


Figura 1.8 Partes diseñadas y ensambladas por la microfábrica japonesa

La microfábrica, mostrada en la figura 1.9, tiene dimensiones externas de 625 x 490 x 380 mm y su peso es de aproximadamente 34kg, su consumo de potencia es de 60W durante su operación. Utiliza Tres cámaras CCD miniatura que muestran la imagen de la sección que se encuentra maquinando en un monitor LCD de 5.8 pulgadas. Sólo se usan dos *palancas* y un *push button* para operar el equipo.



Figura 1.9 Microfábrica sobre mesa desarrollada en MEL

1.2 DESARROLLO DE MICROMECANICA EN EL CCADET - UNAM

En el Grupo de Micromecánica y Mecatrónica (GMM) del Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET) de la Universidad Nacional Autónoma de México (UNAM), se desarrollan micromáquinas bajo un principio similar al de la microfábrica japonesa, esto es, que el tamaño de las piezas fabricadas sea comparable con la de la máquina que las fabricó pero se busca reducir el costo al desarrollar generaciones secuenciales de microequipo [1].

La primera generación puede producirse por equipo de tamaño convencional, las máquinas herramienta de esta generación pueden tener tamaños completos de 200 a 100 mm. Usando microequipo de la primera generación, es posible producir micromáquinas herramienta, dispositivos microensamblados de una segunda generación. Cada nueva generación de microequipo puede ser producida usando el microequipo de las generaciones previas.

Se utilizan materiales como aluminio, acero y latón para mantener el bajo costo de las micromáquinas, además de reducir al mínimo el número de componentes comerciales, tales como motores, componentes mecánicos, programas, etc. Con esta idea en mente, actualmente se desarrollan y aplican algoritmos especiales para el control de tareas, reconocimiento de imágenes, caracterización, etc.

Como parte del trabajo desarrollado se han fabricado algunos prototipos de micromáquinas de primera generación. La figura 1.10 muestra uno de ellos. Las principales ventajas de este prototipo son la simplicidad de su diseño, la fácil fabricación de sus componentes y su sistema de control, pero entre sus inconvenientes están la baja rigidez y el modo dinámico del control de motores de pasos [1].

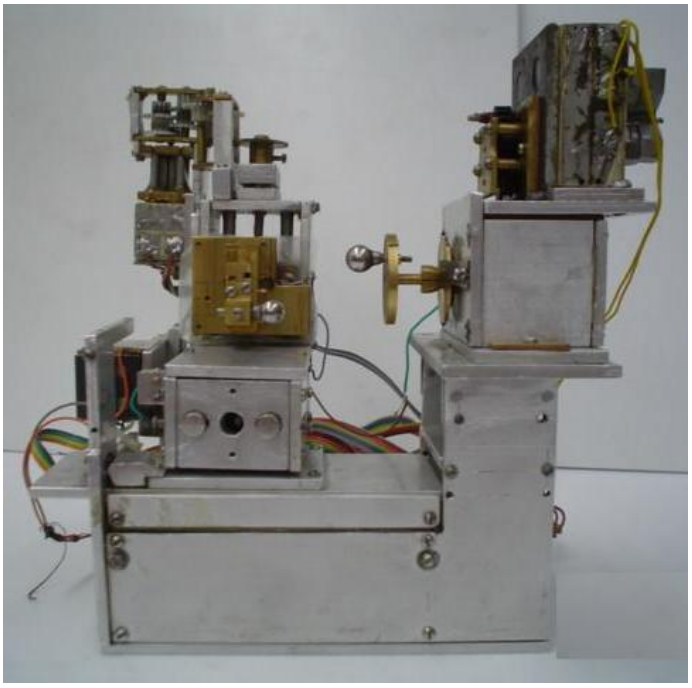


Figura 1.10 Micromáquina herramienta de 1ª generación fabricado en el GMM del CCADET

El primer prototipo tiene la capacidad de producir piezas dentro del rango de 5mm a 50µm, como las mostradas en la figura 1.11; cuenta con una resolución de 1.87µm por eje, tiene cuatro grados de libertad que son 3 ejes de translación y uno de rotación, además de un programa capaz de operar en un sistema Pentium con 32Mb en memoria RAM [1].

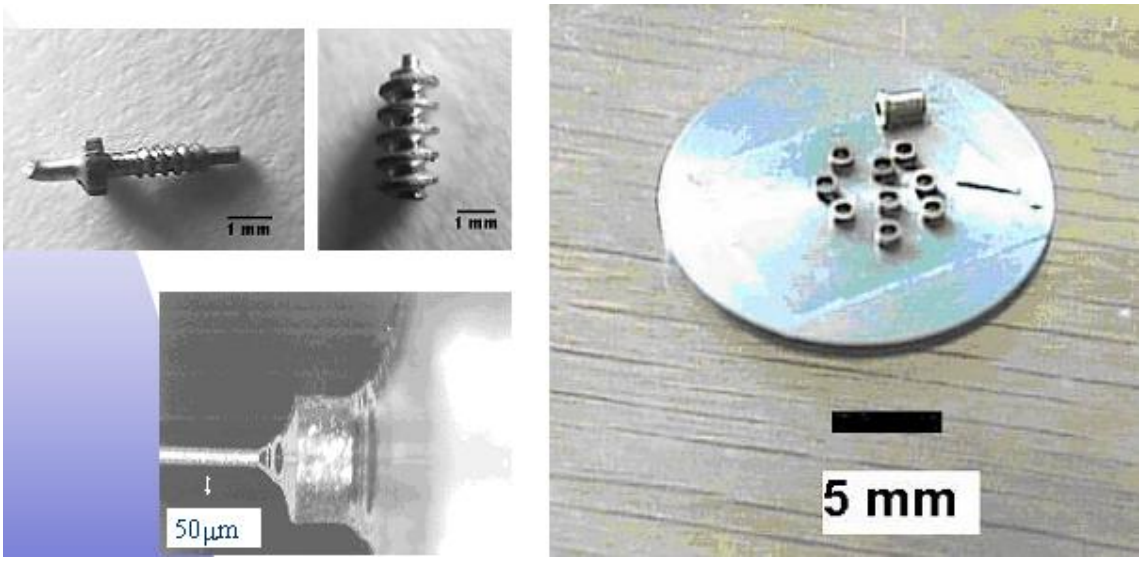


Figura 1.11 Micropiezas fabricadas con el primer prototipo

Este microcentro de maquinado de bajo costo tiene dimensiones aproximadas de 130x160x85 mm^3, y sus principales componentes se muestran en la figura 1.12.

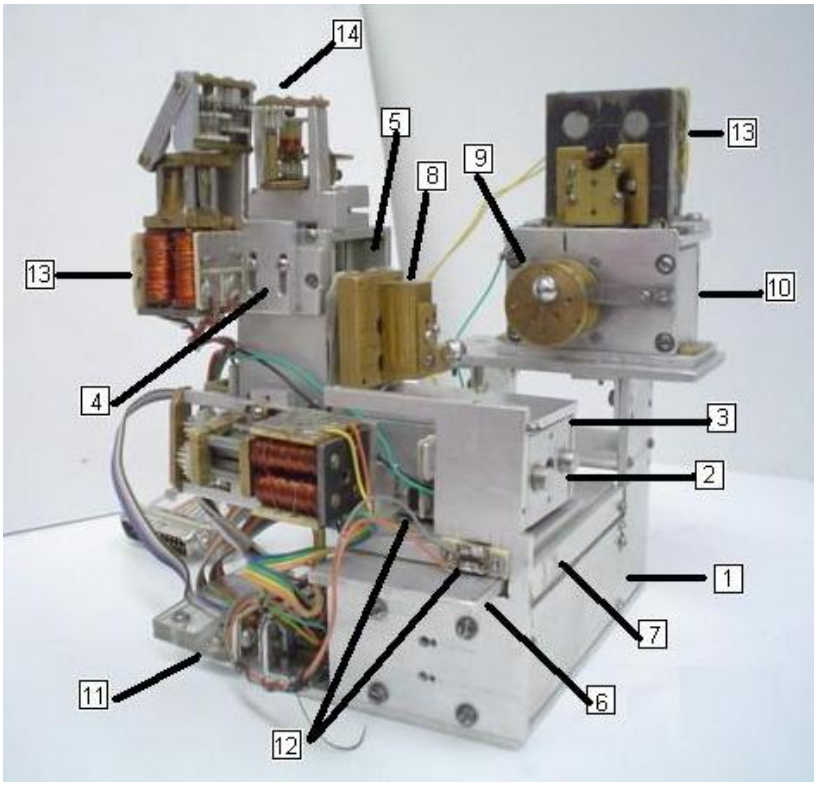


Figura 1.12 Primer prototipo de micromáquina herramienta de primera generación

En la base (1), las guías (2) (4) (6) para los tres carros (3) (5) (7) se instalan de manera secuencial, por ejemplo, cada guía subsecuente se instala sobre el carro previo para proveer translación a lo largo de los ejes X, Y y Z. Se cuenta con 3 *push buttons* (12) para detectar la posición de inicio en los ejes X, Y y Z. Para hacer tareas de torneado, el cortador se coloca en el sistema de sujeción de piezas (8) y la materia prima se sujeta al husillo (9). Si se quiere configurar como taladro o como fresadora el cortador se sujeta al husillo y la materia prima al sistema de sujeción de piezas. El sistema de sujeción de piezas está acoplado a una caja de reducción de engranes con una relación de 84.7:1. El husillo y la caja del husillo (10) también están instalados en la base. Los actuadores empleados para proporcionar el movimiento son motores de pasos bipolares (13) acoplados con cajas de engranes (14). Un motor de pasos se define como un motor que

convierte pulsos eléctricos en desplazamientos angulares discretos [8], en el capítulo 3 se profundizara acerca de los motores de pasos.

El microcentro de maquinado también cuenta con un puerto DB25 (11), que contiene las conexiones a los motores de pasos y a los sensores de contacto colocados en el inicio de carrera de cada eje.

1.3 INTRODUCCIÓN A LOS SISTEMAS DE CONTROL

Existen diferentes tipos de control, para nuestro caso nos interesan aquellos relacionados con el control de motores, como por ejemplo:

- Control numérico
- Control numérico asistido por Computadora
- Control Adaptable

El control numérico (CN) es la supervisión y regulación de determinadas tareas mecánicas de precisión, realizadas por una máquina herramienta. El control numérico asistido por computadora (CNC) y el control numérico directo (CND), que han dado a la industria manufacturera la capacidad de ejercitar un nuevo y mayor grado de libertad en el diseño y manufactura de productos. El código numérico estándar que utilizan las máquinas CNC es el código G y el código M que forman parte del estándar ANSI/EIA-RS-274-D. La estructura del programa-pieza contiene datos de cabecera, definiciones tecnológicas y geométricas, instrucciones de ejecución e instrucciones de fin de programa. El control adaptable, a veces denominado control adaptable automático (AAC), que es un tipo de sistema el cual automática y continuamente revisa el progreso en línea de una actividad (un proceso u operación), por ejemplo midiendo una o más variables del proceso, comparando las cantidades con otras cantidades medidas, o calculadas, o estableciendo valores o límites, y modificando las actividades ajustando automáticamente una o más variables para mejorar y optimizar los resultados [8].

Los sistemas de control se pueden clasificar en sistemas de control en lazo abierto y sistemas de control en lazo cerrado.

1.3.1 CONTROL EN LAZO ABIERTO

Los componentes básicos de un sistema de control se pueden describir mediante:

1. Objetivos de control
2. Componentes del sistema de control
3. Resultados o salidas [9]

La relación básica entre estos tres componentes se ilustra en la figura 1.13.

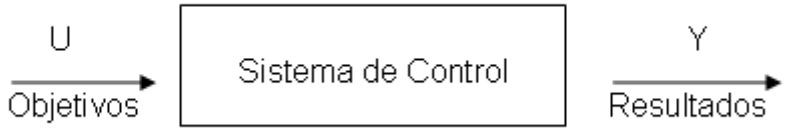


Figura 1.13 Componentes básicos de un sistema de control

Los objetivos se pueden identificar como entradas o señales actuantes "*u*", y los resultados se llaman salidas o variables controladas "*y*". En general el objetivo del sistema de control es controlar las salidas en alguna forma preescrita mediante las entradas a través de los sistemas de control.

Los elementos de un sistema de control en lazo abierto se pueden dividir en dos partes: el controlador y el proceso controlado, como se muestra en la Figura 1.14.

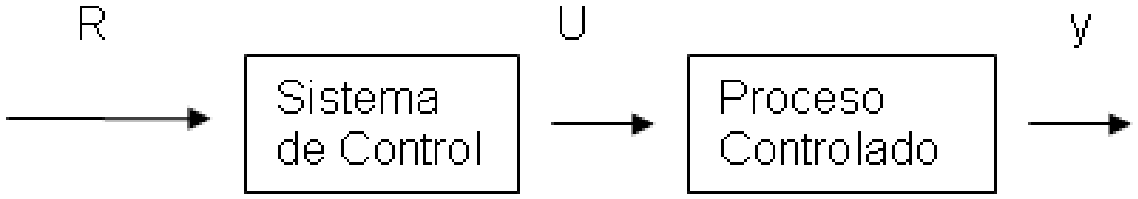


Figura 1.14 Elementos de un sistema de control en lazo abierto

Una señal de entrada o comando "*R"* se aplica al controlador, cuya salida actúa como señal actuante "*U"*; la señal actuante controla el proceso controlado de tal forma que la variable controlada "*y*" se desempeña de acuerdo con estándares preestablecidos. En los casos simples el control puede ser un amplificador, unión mecánica, filtro u otro elemento de control. En los casos más complejos el controlador puede ser una computadora o un microprocesador. Debido a la simplicidad y economía de los sistemas de control en lazo abierto se les encuentra en muchas aplicaciones no críticas.

1.3.2 CONTROL EN LAZO CERRADO

Lo que hace falta en el sistema de control en lazo abierto para que sea más exacto y más adaptable es una conexión o realimentación desde la salida hacia la entrada del sistema como se muestra en la figura 1.15. Para obtener un control más exacto, la señal controlada debe ser realimentada y comparada con la entrada de referencia para enviar una señal actuante proporcional a la diferencia de la entrada y la salida a través del sistema para corregir el error. Un sistema con una o más trayectorias de realimentación como el que se acaba de describir es denominado sistema en lazo cerrado [9].

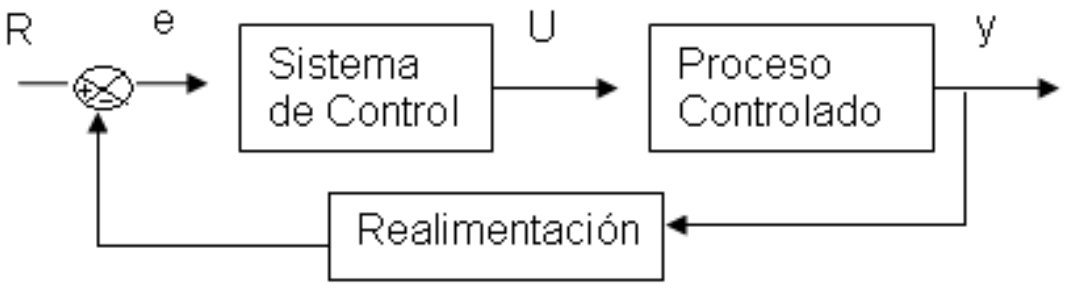


Figura 1.15 Elementos de un sistema de control en lazo cerrado

1.3.3 ALGUNOS SISTEMAS DE CONTROL IMPLEMENTADOS EN EL GMM

Para realizar el control de los movimientos de la micromáquina se necesita controlar cada uno de sus motores de pasos. Un típico circuito de control en lazo abierto de un motor de pasos se muestra en la figura 1.16 y consiste de un controlador de entrada, secuenciador lógico y una etapa de potencia. Aunque el control a lazo abierto es un método de manejo económicamente ventajoso, no está libre de algunas limitaciones, por ejemplo, las revoluciones del rotor llegan a ser inestables en ciertos rangos de velocidad.

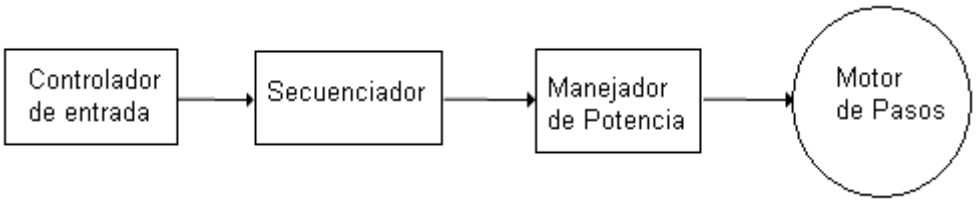


Figura 1.16 Circuito de control en lazo abierto de un motor de pasos

A continuación se da una descripción general del circuito mostrado en la figura 1.16. El controlador de entrada es el circuito generador de pulsos que genera el tren de pulsos requerido para acelerar o desacelerar el motor de pasos. El secuenciador lógico es un circuito que responde a comandos de pulso y controla la excitación de las bobinas secuencialmente, genera la secuencia de pulsos para que el motor gire en sentido horario o antihorario y evitando que pueda haber un corto circuito en la etapa de potencia, generalmente es ensamblado con circuitos integrados TTL o CMOS, las señales de salida del secuenciador lógico son transmitidas a las terminales de entrada de una etapa de potencia que enciende o apaga el motor de pasos. Cuando el potencial de un canal de salida del secuenciador está a nivel alto (H), el manejador de potencia excitará la fase correspondiente del embobinado. Similarmente, si la salida está a un nivel bajo (L), la fase correspondiente no se excitará o estará apagada. Como se muestra en la figura 1.17, si se sigue la secuencia 1→2→3→1... la dirección de rotación será horario y en cambio si se da la secuencia 1→3→2→1... la dirección de rotación será antihorario [10].

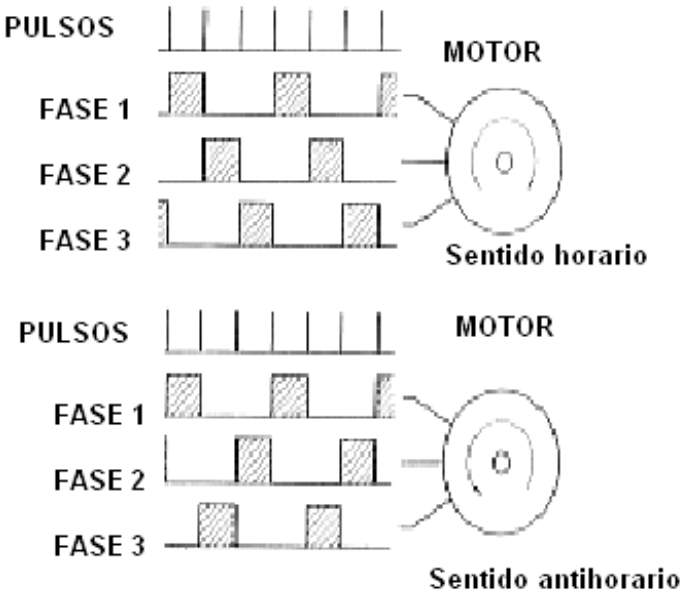


Figura 1.17 Series de pulsos de entrada y secuencia de excitación.

1.3.4 Sistema basado en PC

Este sistema de control se ilustra con la figura 1.18. Tiene la posibilidad de emplear una computadora personal (PC) como controlador de entrada utilizando dos puertos paralelos, a través de los cuales se mandan las señales para cada bobina, uno de los objetivos de este sistema fue eliminar en la medida de lo posible el hardware de control externo. Su etapa de potencia, está compuesta por un amplificador de DC de 16 canales y cuatro acondicionadores de señal para los sensores de contacto [4].

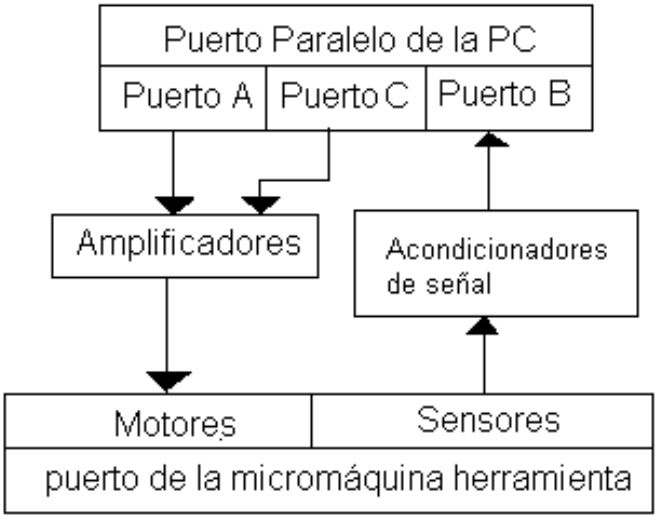


Figura 1.18 Diagrama de bloques del sistema de control basado en PC

El estado de cada uno de los motores es determinado por los datos que se encuentran en la sección A y C de cada uno de los puertos paralelos de la PC que se están utilizando. Para simplificar el sistema de control, los motores de pasos trabajan únicamente en modo estático.

La señales provenientes de los cuatro sensores de la MMH son registrados por la PC a través de la sección B del puerto paralelo; tres de estos sensores se emplean para determinar la posición inicial de los carros y uno de ellos es empleado para detectar el momento de contacto entre la herramienta y la pieza de trabajo con el fin de determinar posiciones relativas de diferentes instrumentos empleados en los procesos de manufactura. Todo el control es llevado a cabo por el procesador de la PC, y se hace por medio de una interfaz generada con el lenguaje de programación C++.

1.3.5 Sistema basado en FPGA

El objetivo de este sistema de control fue diseñar e implementar un sistema de control para los motores aplicados a un microequipo, con el uso de lógica programable [11].

El programa de control se desarrolló utilizando el paquete MAX+PLUS II cuyo diagrama de bloques se muestra en la figura 1.19.

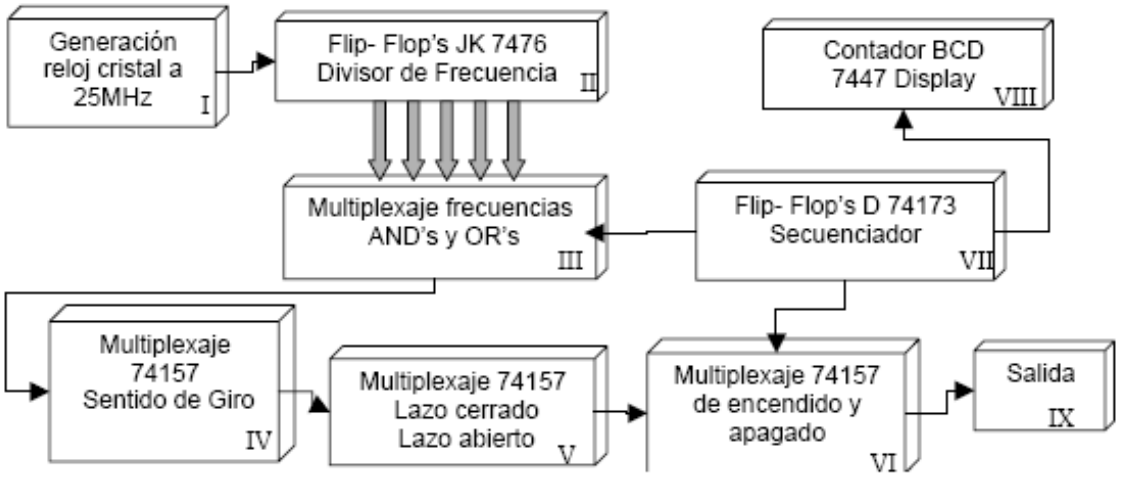


Figura 1.19 Diagrama de bloques del sistema de control del sistema basado en FPGA

El diagrama de bloques del sistema completo de control se muestra en la figura 1.20.

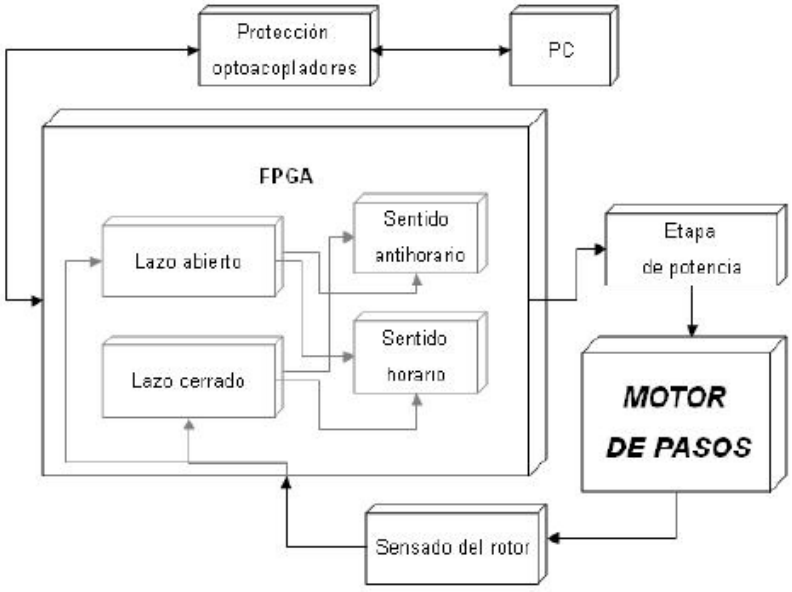


Figura 1.20 Diagrama completo del sistema de control basado en FPGA

Este sistema es un control de lazo cerrado, se utilizaron los 2 puertos paralelos de la PC para alimentar a la tarjeta del FPGA, una etapa de protección que estuvo basada en el optoacoplador PC817, la etapa de potencia de este circuito fue construida basada en el circuito integrado L293 que es la versión de baja potencia, hasta 1A, del circuito L298. El programa controlaba solamente 2 motores de pasos.

1.3.6 Sistema Basado en PIC

El objetivo de este sistema fue desarrollar un sistema de control para que con ayuda de una sola PC se puedan controlar dos MMH's [12]. El diagrama de bloques de este sistema se muestra en la figura 1.21

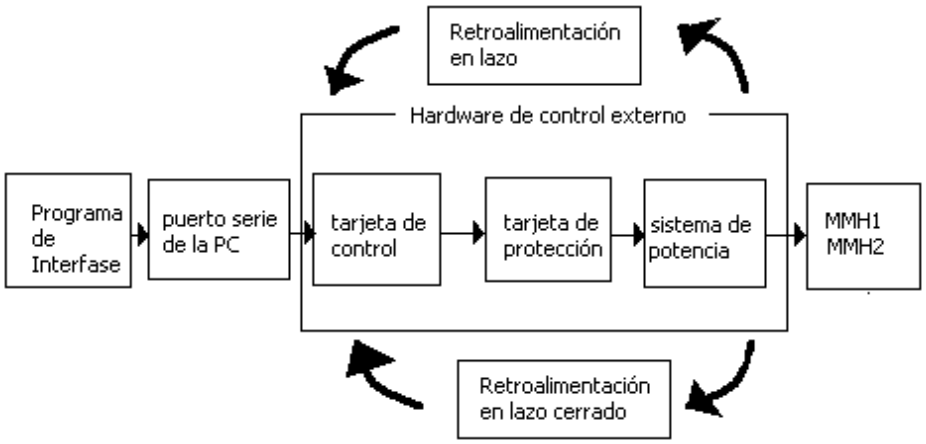
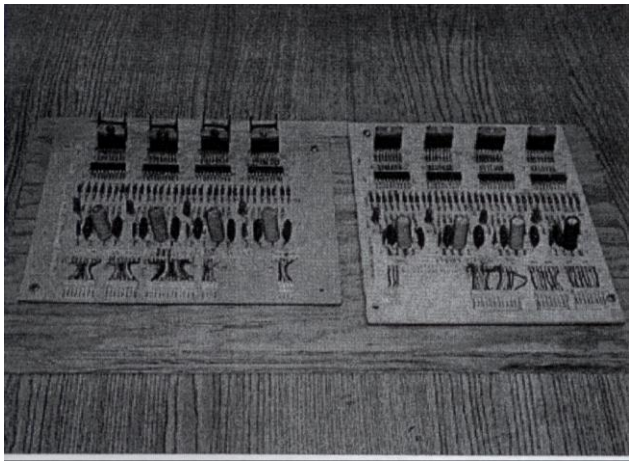


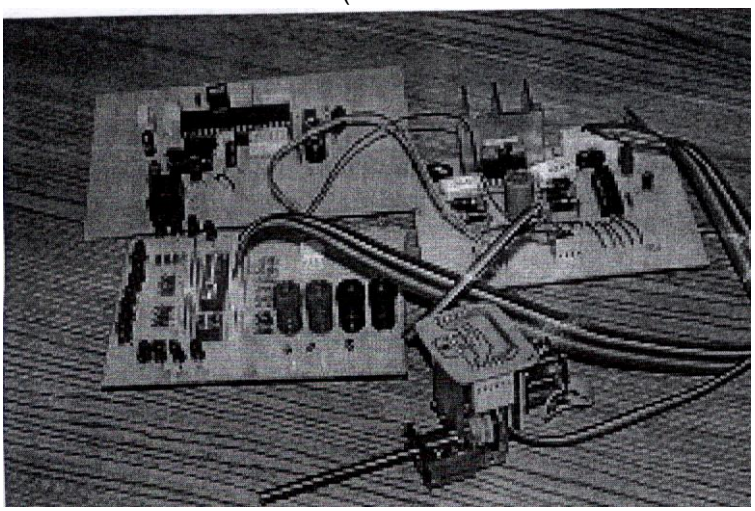
Figura 1.21 Diagrama de bloques del sistema de control basado en PIC

Como controlador de entrada se seleccionó el microcontrolador PIC 18F8720, como secuenciador se uso el L297, la etapa de potencia estuvo basada en el circuito L298 que fue diseñada con la técnica de tejido de alambre magneto, la tarjeta de aislamiento entre la etapa de potencia y la micromáquina se basó en el optoacoplador PC817, el programa de control se desarrolló en lenguaje ensamblador y el puerto de comunicación que se uso fue serial.

Para el control a lazo cerrado se diseñó una tarjeta que contenía dos optosensores y una fuente digital. El sistema completo se muestra en la figura 1.22.



(a



b)

Figura 1.22 Sistema de control basado en PIC a) etapas de potencia b) sistema a lazo cerrado

1.3.7 Sistema basado en Códigos G y M

El objetivo de este sistema fue desarrollar una aplicación que fuera capaz de interpretar un conjunto de instrucciones de código G y M y generar las señales de control necesarias para la manufactura de piezas de una máquina herramienta [13]. El código G y M es el código numérico estándar que utilizan las máquinas de control numérico (CNC), el programa-pieza es estructurado; contiene datos de

cabecera, definiciones técnicas y geométricas, además de instrucciones de ejecución e instrucciones de fin de programa.

El diagrama de bloques de este sistema de control está formado por 3 elementos esenciales que son: una PC que ejecuta el software de control, el puerto de comunicación y una etapa de potencia que va conectada a la micromáquina, como se muestra en la figura 1.23. La herramienta en que se desarrolló la aplicación es Visual Basic 6.0, el puerto de transmisión es el puerto paralelo, la etapa de potencia es la que se utilizó en el sistema basado en PC.

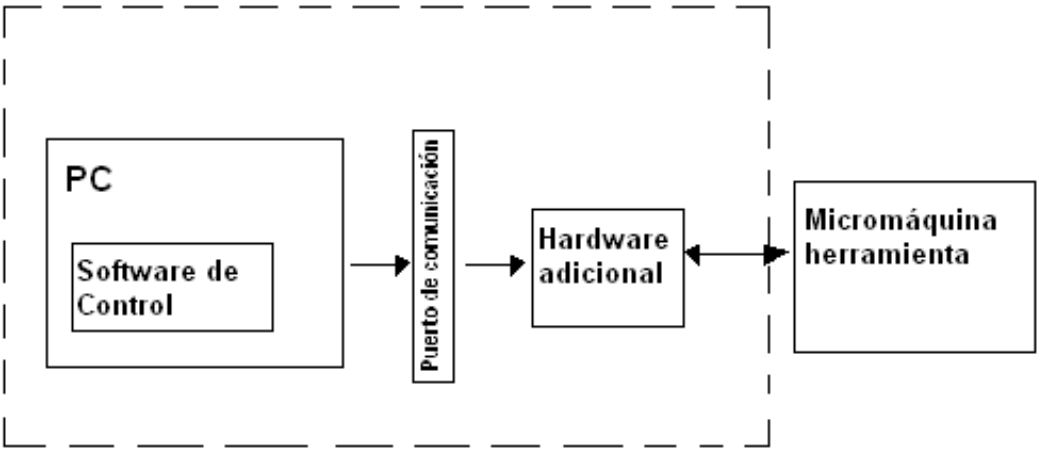


Figura 1.23 Esquema del sistema de control basado en códigos G y M

La aplicación ofrece dos modos de control, uno manual y control mediante código G y código M y una rutina de HOME. El sistema se muestra en la fig. 1.24.

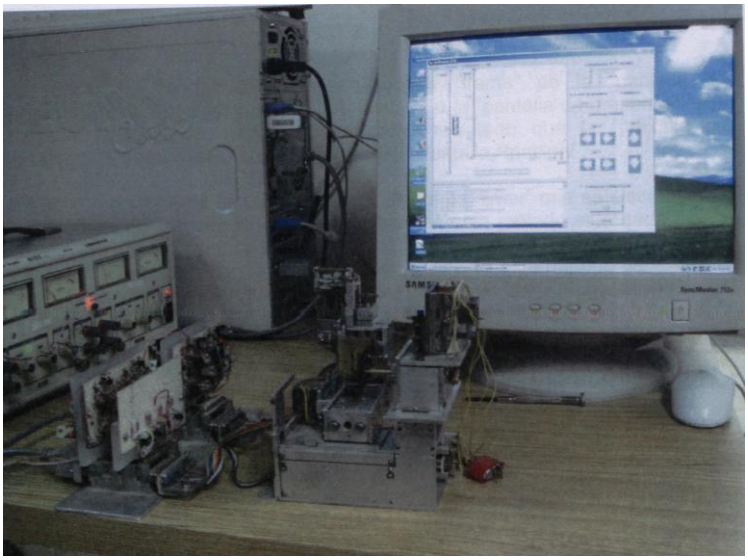


Figura 1.24 Sistema de control basado en Código G y M

El sistema recibe un archivo que contiene los códigos G y M que se procesan para obtener cada pieza, después se hace una interpretación gráfica y se generan las instrucciones para el control de la micromáquina y posteriormente son enviadas a la micromáquina herramienta. En el modo manual se pasa directamente a la interpretación gráfica y posteriormente pasa por la generación y envío de señales de control a la micromáquina. En ambos modos de control se puede desactivar el envío de las señales a la MHH.

Capítulo 2

PLANTEAMIENTO DEL PROBLEMA

2.1 NECESIDADES

Las necesidades básicas que debe cubrir el sistema de control para que opere de una manera eficiente son:

- El control de la micromáquina debe ser lo más independiente posible de la PC. Para que se pueda controlar al menos dos micromáquinas sin necesidad de aumentar el número de PCs.
- Alcanzar la mayor velocidad posible en los motores sin pérdida de pasos.
- Presentar el menor calentamiento posible durante la operación del sistema.

A continuación, se analizan algunos sistemas de control que se han desarrollado para el control de la micromáquina para revisar sus ventajas y desventajas. Finalmente, se realiza la nueva propuesta del sistema de control.

2.2 ANÁLISIS DE LOS SISTEMAS DE CONTROL PARA MMHs

2.2.1 SISTEMA BASADO EN PC

En este sistema la generación de los trenes de pulsos y la generación de las secuencias se hace completamente a través del programa, lo cual simplifica el sistema, se necesitan 20 bits para controlar los 4 motores de pasos del primer prototipo de máquina herramienta y recibir las 3 señales de fin carrera y de contacto de la herramienta con el material.

La desventaja que presenta es, que cada puerto paralelo provee 17 bits por lo que se necesitan 2 puertos paralelos para cubrir los bits necesarios, por lo que para controlar otra micromáquina se necesitaría o añadir otra PC o añadir dos puertos paralelos al sistema de control para cubrir las señales necesarias.

2.2.2 SISTEMA BASADO EN FPGA

Existen muchas ventajas en el uso de un FPGA, la más importante es, sin lugar a dudas, el "hardware reconfigurable", en segundo lugar éste es un sistema de control a lazo cerrado, con lo que se pueden controlar los motores a una velocidad por arriba de las 25000 r.p.m.

Las desventajas que presenta son que sólo puede controlar 2 motores de pasos y el alto costo de cada dispositivo FPGA.

2.2.3 SISTEMA DE CONTROL BASADO EN PIC

Para este sistema, el control se delegó a un dispositivo externo, en este caso a un microcontrolador PIC 18F8720, con este sistema se logró una mayor independencia de la PC, ya que se puede aumentar el número de máquinas herramienta a controlar, sin aumentar el número de PCs, ya que contiene hasta 68 puertos de entrada / salida. La generación de secuencias se delegó al CI L297 y la etapa de potencia estuvo basada en el CI L298.

Es un sistema de control que puede ser operado a lazo abierto, con lo cual se controlan los motores hasta los 390 Hz (5850 r.p.m.) y en lazo cerrado se obtuvieron hasta 20250 r.p.m. a 25V con un consumo de 1.2 A por motor.

Las desventajas de este sistema son que durante la operación a lazo cerrado la temperatura del motor se incremento a más de 40º C después de los 23V, por lo cual se decidió operar los motores un 20% por debajo de su frecuencia de operación máxima, la segunda desventaja importante es que durante la operación del sistema de control el circuito L298 presenta un calentamiento considerable.

2.2.4 SISTEMA DE CONTROL BASADO EN CÓDIGO G Y CÓDIGO M.

Este sistema proporciona un programa capaz de interpretar instrucciones de código numérico estándar y generar las señales de control hacia una

micromáquina herramienta con lo que se logra reducir el tiempo de manufactura de piezas.

Las desventaja de este sistema es que utiliza dos puertos paralelos de la PC para el control de una micromáquina y la etapa de potencia utilizada en el sistema de control basado en PC, por lo que presenta limitaciones similares a este sistema como son la necesidad de añadir otra PC o dos puertos paralelos más para poder controlar dos micromáquinas. En un futuro se pretende utilizar un dispositivo de control adicional (PIC, DSP, etc) con el fin de lograr independencia de la PC y que ésta pueda realizar otras tareas en paralelo a la manufactura de una pieza.

2.3 NUEVA PROPUESTA DE SISTEMA DE CONTROL.

- Encontrar un dispositivo de control que:
    - Contenga la mayor cantidad de puertos GPIO y módulos que sirvan para el control de motores de pasos, ya que se necesitan por lo menos 20bits para el primer prototipo y 24 para el segundo prototipo.
    - Proporcione Proporcionar las señales que controlan la operación de los motores, como son reloj, sentido de giro, paso completo, medio paso, etc.
    - Permita en un futuro, la implementación de los algoritmos de control, procesamiento y reconocimiento de imágenes que se han desarrollado en el GMM.

- Desarrollar una nueva etapa de potencia que presente un bajo calentamiento cuando el sistema se encuentre en operación.
- Los movimientos que debe realizar la micromáquina son:
    - Movimiento de traslación para los ejes X, Y y Z.
    - Movimiento de rotación para el husillo.
    - Movimiento de interpolación lineal entre los ejes X y Z.

- Desarrollar una interfaz, por medio de la cual, se pueda controlar a la micromáquina.

2.4 OBJETIVO

Desarrollar un sistema de control para una microfábrica de escritorio empleando como herramienta un DSP.

2.5 ALCANCES

- Proponer una nueva alternativa tecnológica para la implementación del sistema de control de MMH.
- Realizar un sistema de control que sea independiente de la PC
- Comparar la nueva alternativa desarrollada con las soluciones que ya se encuentran actualmente disponibles en el grupo.

Capítulo 3

DESARROLLO

Después de haber descrito el planteamiento del problema y haber dado una introducción a los sistemas de control de motores de pasos, podemos plantear la nueva alternativa de sistema de control, la cual se describe en forma general con el diagrama de bloques 3.1.

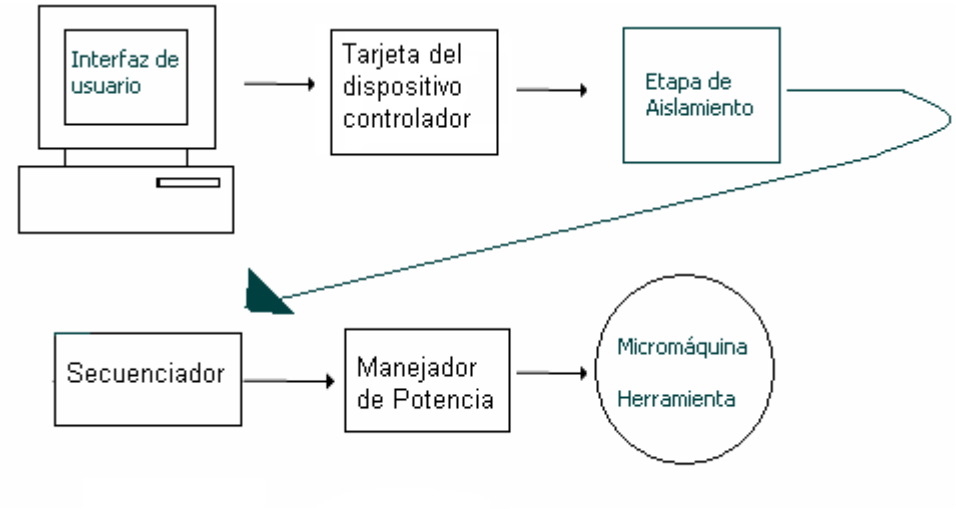


Figura 3.1 Diagrama de bloques del sistema de control propuesto

El primer punto de la figura 3.1 es la interfaz de usuario, por medio de esta interfaz el usuario podrá elegir el modo de operación y dar los comandos necesarios para controlar los movimientos de la micromáquina, la tarjeta que contiene al dispositivo de control será la encargada de generar las señales de control como son: reloj, sentido de giro, activación, paso, control y reinicio (*reset*). La etapa de aislamiento sirve para proteger la tarjeta del dispositivo de control de algún posible cortocircuito en las etapas posteriores, ya que no existe una conexión física entre la tarjeta de evaluación del dispositivo de control y los circuitos que manejan corrientes mas grandes.

3.1 ACTUADORES EN LAS MICROMÁQUINAS HERRAMIENTAS

Para realizar el control de los movimientos de la micromáquina se necesita controlar cada uno de sus motores de pasos, por lo que debemos conocer las características y su funcionamiento para poder llevarlo a cabo.

Un motor de pasos se define como un motor que convierte pulsos eléctricos en desplazamientos angulares discretos La señal de entrada (pulso) se convierte directamente en una posición requerida de la flecha, sin usar sensores de posición del rotor o realimentación [10].

Idealmente, un motor de pasos debería cubrir los siguientes requerimientos:

- Pasos muy pequeños
- Operación bidireccional
- Error de posicionamiento no acumulativo
- Pequeñas constantes de tiempo eléctricas y mecánicas
- Operación sin perder pasos

Los motores de pasos proveen muy alto torque a bajas velocidades, hasta 5 veces el torque continuo de un motor conmutado de CD del mismo tamaño o el doble de torque que el motor sin escobillas equivalente. Las ventajas más importantes de los motores de pasos son:

- Control en lazo abierto (excepto para aplicaciones especiales)
- Pasos muy pequeños
- Sincronización simple de un grupo de motores
- Alta confiabilidad

Por otro lado, los motores de pasos a veces muestran la tendencia de perder la sincronía, entran en resonancia con múltiplos de la frecuencia de entrada y oscilaciones al final de cada paso. La velocidad y eficiencia de los motores de pasos es menor que en los motores sin escobillas.

Los motores de pasos pueden clasificarse como:

- Con rotor activo (Magneto Permanente)
- Con motor reactivo (Reluctancia Variable)
- Motores híbridos

Los motores de magneto permanente se dividen en bipolares y unipolares. Como mencionamos anteriormente, los motores usados en las micromáquinas herramienta son motores de magneto permanente, bipolares y de 4 pasos por revolución. Un motor bipolar consiste de un rotor hecho de magneto permanente rodeado por polos en el estator formado por las bobinas, como se ilustra en la figura 3.2.

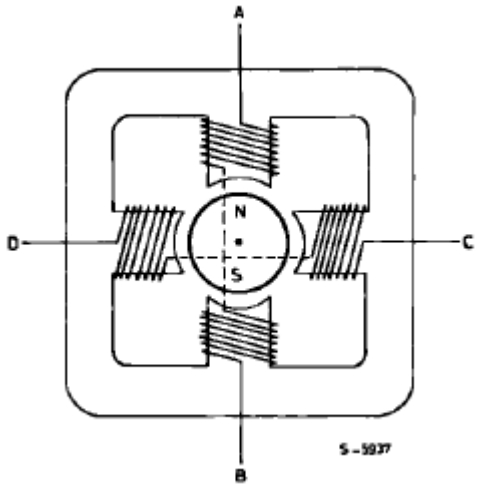


Figura 3.2 Diagrama simplificado de un motor de pasos de magneto permanente bipolar

En este tipo de motores la corriente puede fluir bidireccionalmente y los pasos del motor se consiguen polarizando las bobinas en secuencia. Para un motor de este tipo hay tres posibles secuencias de manejo. La primera es energizar las bobinas en la secuencia AB/CD/BA/DC (BA significa que el embobinado AB se energizó en sentido opuesto). Esta secuencia se conoce como una fase encendida (*"one phase on")* de paso completo. Como se ve en la figura 3.3 solo una fase se energiza en un momento dado.

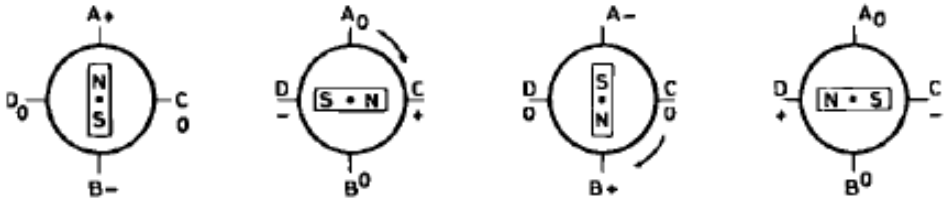


Figura 3.3 Una sola fase energizada (one phase on) paso completo

La segunda posibilidad es energizar ambas fases juntas, de tal manera que el rotor siempre se alinee entre dos polos. A este modo se le conoce como dos fases encendidas (*two phase on*) a paso completo, en este modo se da el más alto torque.

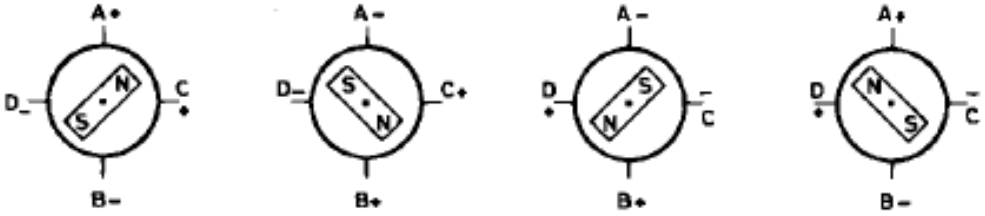


Figura 3.4 Dos fases encendidas (two phases on) paso completo

La tercera opción, mostrada en la figura 3.5, es energizar una bobina, por ejemplo AB, después dos bobinas AB y CD, de tal manera que el motor se mueva medio paso, después solo CD, a continuación CD y BA, posteriormente solo BA y así sucesivamente hasta obtener un giro completo. Esta secuencia se conoce como modo de medio paso tiene un torque menor que los modos anteriores.

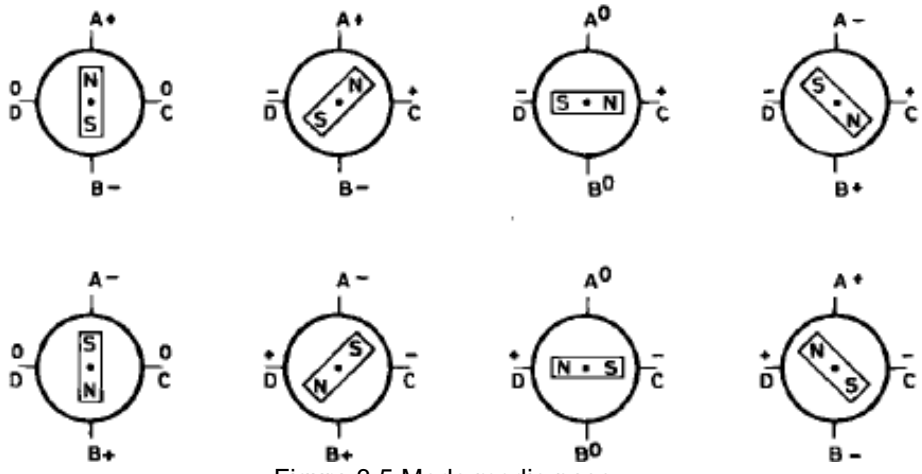


Figura 3.5 Modo medio paso

Para girar en sentido antihorario se utiliza la misma secuencia pero ahora en el orden inverso. Como se observa en los diagramas anteriores el motor tendría un ángulo de paso de 90º. Otros motores tienen múltiples polos para reducir el ángulo de paso.

3.2 SELECCIÓN DE LA ETAPA DE POTENCIA

Una vez que se conoce el tipo de motor con el que se va a trabajar, el diseñador debe seleccionar el tipo de circuito que se encargará de proveer la corriente necesaria para que funcione, a este circuito lo denominaremos la etapa de potencia.

En los proyectos anteriores se utilizó como dispositivo de manejador de potencia el circuito integrado L298, ya que este va emparejado con el secuenciador L297. La combinación del L297-L298 tiene las siguientes ventajas: se requieren pocos componentes, con lo que se obtienen costos reducidos, alta confiabilidad y se necesita poco espacio. Pero como se mencionó en 2.2.3, uno de sus inconvenientes es el calentamiento de más de 40º C de este circuito.

Se realizó una investigación para comparar si era más conveniente utilizar una etapa de potencia basada en MOSFETs y controlarla por medio de PWMs o mantener la configuración L297 – L298 modificando el circuito L298.

El transistor de efecto de campo FET, (*Field Effect Transistor*) es un dispositivo de tres terminales que se utiliza para aplicaciones diversas que se asemeja, en una gran proporción a las del transistor TBJ. La diferencia básica entre los dos tipos de transistores es el hecho de que el TBJ es un dispositivo controlado por corriente, mientras que el transistor JFET es un dispositivo controlado por tensión. El JFET es un dispositivo unipolar que depende únicamente de la conducción o bien de electrones, (canal n) o de huecos (canal p) [21].

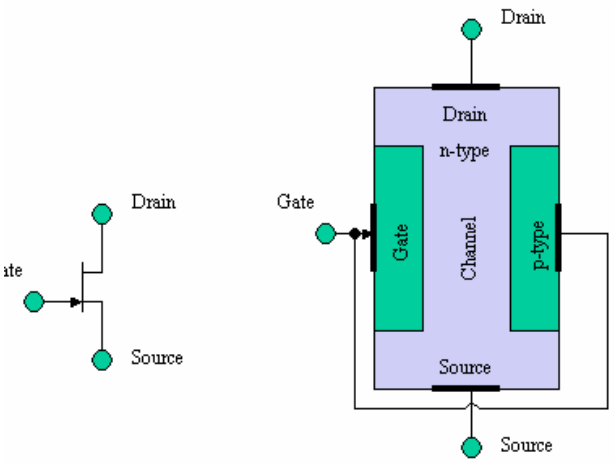


Figura 3.6 JFET canal n

Los transistores de efecto de campo metal óxido semiconductor (MOSFET) consisten en un transistor de efecto de campo basado en la estructura MOS. Las ventajas de los MOSFET son las siguientes:

- Alta impedancia de entrada
- Tienen gran estabilidad térmica
- Los MOSFET pueden disipar una mayor potencia y conmutar corrientes grandes.
- Los FET generan un nivel de ruido menor que los BJT.

Pero sus desventajas son:

- Sensibilidad a las cargas estáticas.
- Sensibilidad a los transitorios cuando los elementos son retirados o insertados.
- Más complejidad en el diseño debido a que se deben añadir circuitos de protección adicionales para proteger cada MOSFET, que a su vez reducen la impedancia de entrada original del MOSFET.
- Los MOSFET exhiben una respuesta en frecuencia pobre debido a la alta capacitancia de entrada.
- Algunos tipos de MOSFET presentan una linealidad muy pobre.

El motivo principal por el que descartamos los MOSFET es su alta sensibilidad, ya que la construcción y puesta a punto del prototipo requeriría muchos cambios y se podrían dañar muchos dispositivos debido a su sensibilidad. Otro motivo fue que las señales de control PWM (Modulación por ancho de pulso) eran más complejas de programar en el dispositivo de control que se seleccionó, que la generación secuencias de pulsos. Finalmente, no se contaba con mucha experiencia en el laboratorio en el campo de los MOSFETS.

Por lo tanto, se mantuvo el L297 para generar las secuencias y se mantendría la configuración del L298 implementándolo con componentes discretos para tener un comportamiento más eficiente. La descripción del circuito L298 se muestra en la figura 3.7.

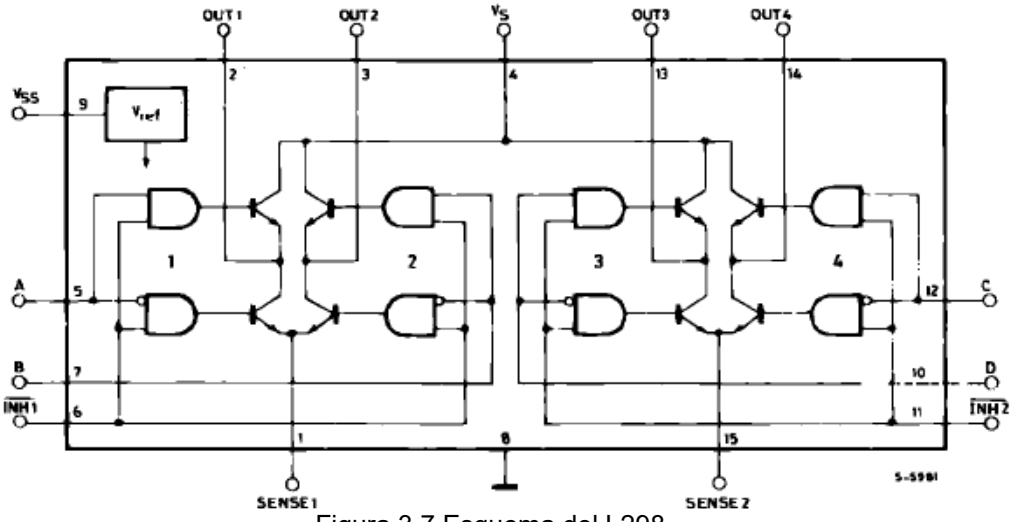


Figura 3.7 Esquema del L298

Es un circuito manejador con una configuración de dos puentes completos (*dual full- bridge*) con capacidad de manejo de altas corrientes y altos voltajes, diseñado para aceptar niveles lógicos TTL y manejar cargas inductivas como relevadores, solenoides, motores DC y de pasos. Tiene dos entradas de habilitación para activar o desactivar el dispositivo independientemente de las señales de entrada. Los emisores de los transistores inferiores de cada puente se conectan a un resistor eterno de sensado de corriente.

El L298 contiene dos etapas de potencia de salida (A;B). La etapa de potencia de salida es una configuración puente y sus salidas pueden manejar una carga inductiva. La corriente que fluye a través de la carga sale del puente al resistor de sensado (R_{SA};R_{SB}) permiten detectar la intensidad de la corriente. Cada puente se controla por medio de cuatro compuertas, cuyas entradas son: In1, In2, EnA y para el segundo puente In3, In4 y EnB. Las entradas In fijan el estado del puente, cuando la entrada "En" está en alto; cuando la señal En está en bajo se inhiben todas las entradas del puente [14].

3.3 SELECCIÓN DEL CIRCUITO SECUENCIADOR

El L297 integra todos los circuitos requeridos para el control de motores de pasos bipolares y unipolares. Se usa regularmente con un controlador de 2 puentes H, como el L298, para formar una interfaz completa microprocesador-motor de pasos bipolar.

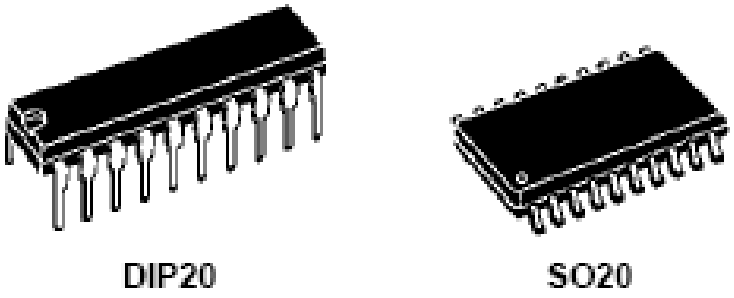


Figura 3.8 Encapsulado del L297

Está hecho con tecnología I^2L compatible SGS analógico/digital y es ensamblado en un encapsulado DIP de 20 pines como se muestra en la figura 3.8. Se alimenta a 5V y todas sus señales son compatibles con TTL/CMOS o con transistores de colector abierto. La alta densidad de integración es una de las características claves de su tecnología por lo que el L297 es un dispositivo muy compacto [15].

El L297 recibe las señales de control del dispositivo controlador de entrada y provee todas las señales necesarias a la etapa de potencia. Además incluye 2

circuitos “troceadotes” (*chopper*) PWM para regular el pico de corriente en las bobinas del motor. Con una etapa de potencia adecuada el L297 puede manejar dos fases de un motor bipolar de imán permanente, cuatro fases de motores unipolares de imán permanente o cuatro fases de motores de reluctancia variable. También es capaz de manejar paso completo y medio paso.

El corazón del diagrama de bloques del L297, mostrado en la figura 3.9, es un bloque llamado traductor, el cual genera las secuencias de fase adecuadas para operación a medio paso, paso completo a una fase y paso completo a dos fases. Este bloque es controlado por las entradas CW/ $\overline{\text{CCW}}$ con la que controlamos el sentido de giro CW horario o CCW antihorario, y HALF/ $\overline{\text{FULL}}$ con la que controlamos si queremos operación a medio paso HALF o paso completo FULL y por un reloj que avanza al traductor de un paso al siguiente.

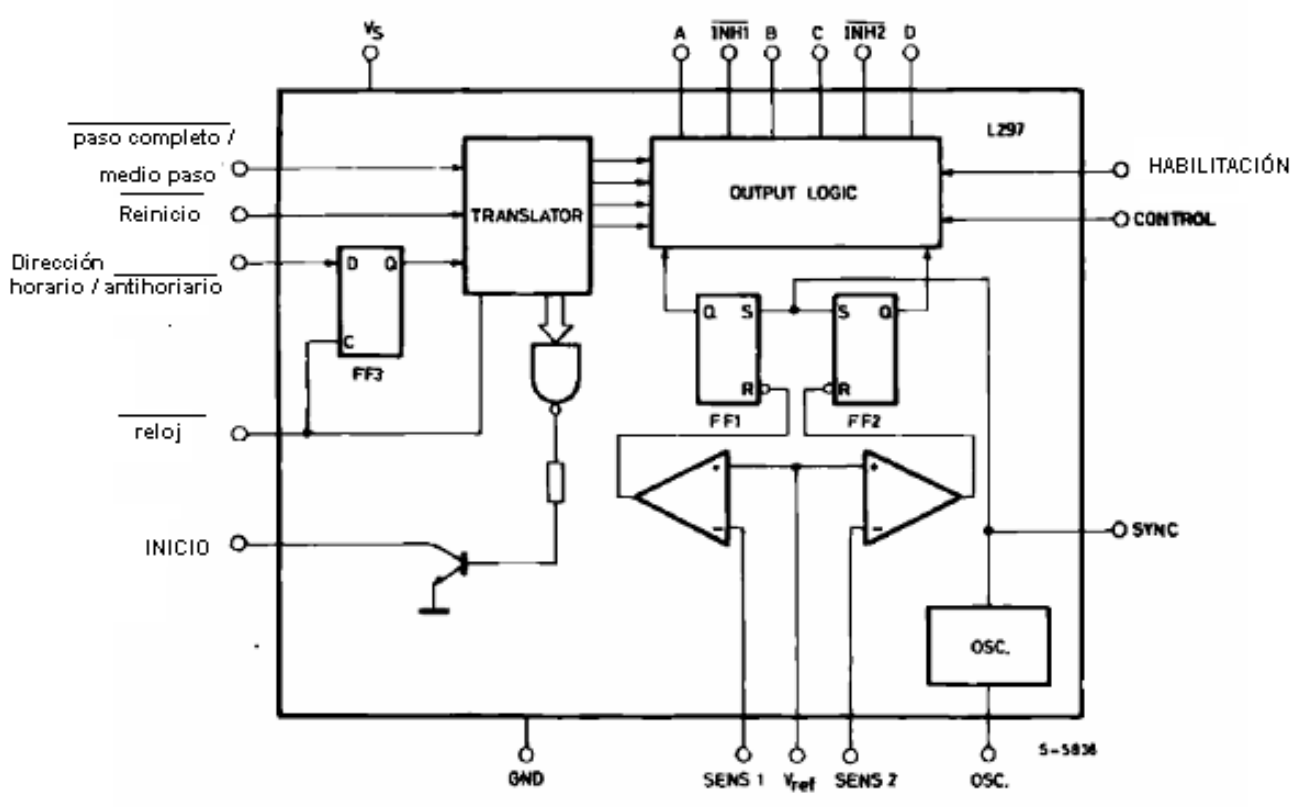


Figura 3.9 diagrama de bloques del L297

Cuatro salidas son provistas por el traductor para procesamiento del bloque de salidas lógicas que implanta las funciones de inhibición y de troceado. Internamente, el traductor consiste de un contador de 3 bits mas lógica combinacional que genera una secuencia básica de código *gray* de 8 pasos. Las tres secuencias de manejo pueden ser generadas fácilmente desde esta secuencia maestra. Si administramos un nivel ALTO en la entrada $\overline{\text{paso completo}}$ /medio paso Tendremos operación a medio paso, las formas de onda de las salidas para esta secuencia se muestran en la figura 3.10.

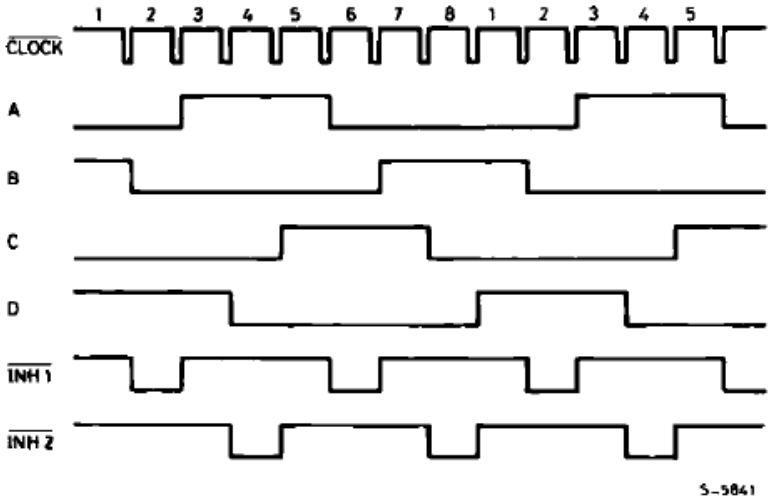


Figura 3.10 Formas de onda en operación a medio paso

Los modos de paso completo se obtienen saltando estados alternos de la secuencia de 8 pasos. Lo que pasa es que el reloj se salta el primer estado del contador de 3 bits en el traductor. El *bit* menos significativo de este contador no es afectado, por lo tanto la secuencia generada depende del estado del traductor cuando el modo de paso completo sea seleccionado.

Si el modo de paso completo es seleccionado cuando el traductor está en cualquier estado impar obtendremos la secuencia de paso completo a dos fases mostrada en la figura 3.11

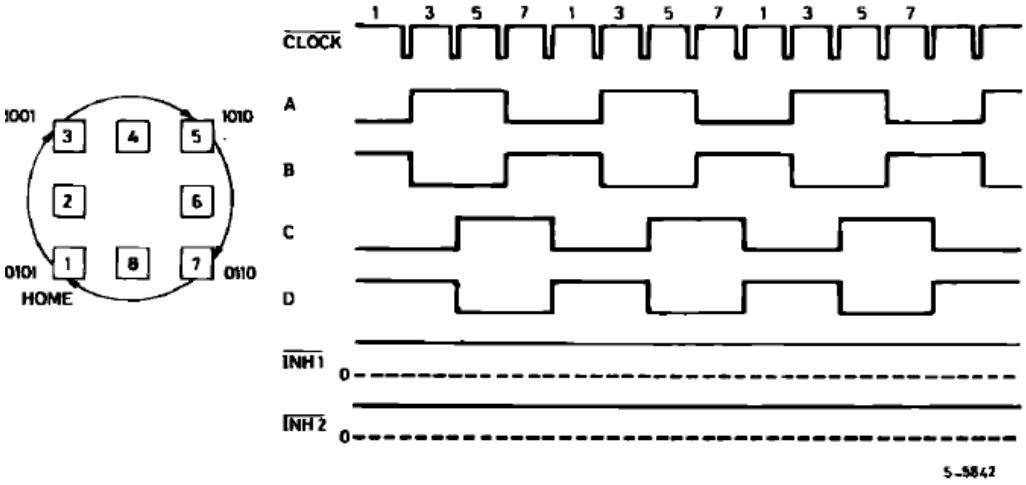


Figura 3.11 Operación a paso completo a dos fases (two phase on)

En contraste, el modo de paso completo con una sola fase energizada es obtenido seleccionando modo de paso completo cuando el traductor está en un estado par.

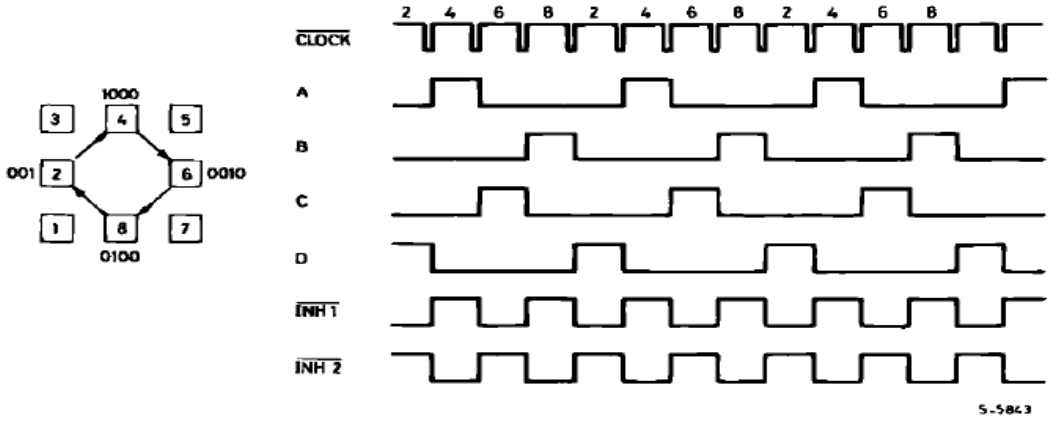


Figura 3.12 Operación de paso completo a una fase (one phase on)

Otras señales se conectan al bloque traductor: la señal de entrada RESET y la salida HOME. El RESET es una entrada de reset asíncrono que restaura el bloque de traductor a la posición de origen (estado 1, ABCD=0101). La salida HOME (colector abierto) señala esta condición y se diseñó para hacer una operación AND con la salida de un sensor de posición mecánico.

Finalmente, se encuentra la entrada ENABLE conectada al bloque de salidas lógicas, un nivel BAJO en esta entrada pondrá en BAJO a $\overline{INH1}$, $\overline{INH2}$, A, B, C, D.

Esta entrada es útil para desactivar el controlador del motor cuando el sistema se inicializa.

En los modos a medio paso y a paso completo a una fase se generan dos señales más $\overline{INH1}$ e $\overline{INH2}$ estas son señales de inhibición que se acoplan a las líneas de activación del L298 y sirven para apresurar el decremento de la corriente cuando la bobina se apaga. Como en el modo de paso completo a dos fases ambas bobinas son energizadas continuamente ninguna bobina se apaga y por lo tanto estas señales no se generan en ese modo.

Para ver como funcionan las señales $\overline{INH1}$ e $\overline{INH2}$ analicemos un puente H de la etapa de potencia conectado a la primer fase de un motor bipolar de dos fases mostrado en la figura 3.13, las entradas A y B determinan que pareja de transistores estará encendido, los demás transistores se encuentran apagados.

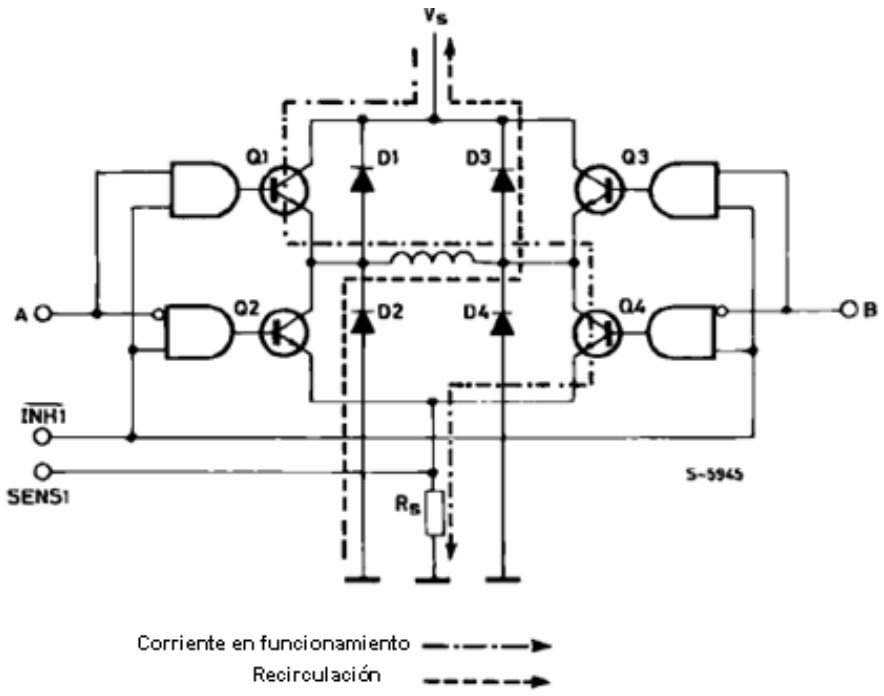


Figura 3.13 Funcionamiento de las señales $\overline{INH1}$ e $\overline{INH2}$

Si asumimos que A está en ALTO y B en bajo la corriente fluye a través de Q1, Q4 y la bobina del motor. Si A se manda a BAJO la corriente recircularía a través de D2, Q4 y Rs dando un decremento lento de la corriente e incrementando la disipación en Rs, por el otro lado, si A se manda a BAJO e $\overline{INH1}$ se activa, los cuatro transistores se apagan, en este caso la corriente recircula, de tierra a Vss vía D2 y D3, dando un decaimiento más rápido y así permitiendo una operación más rápida del motor. Además, ya que la recirculación de la corriente no fluye a través de Rs se puede utilizar un resistor que no disipe tanta potencia. Sucede exactamente la mismo con la segunda bobina, utilizando el otro puente H y las señales C, D e $\overline{INH2}$. Las señales $\overline{INH1}$ e $\overline{INH2}$ son generadas como funciones OR:

$$A + B = \overline{INH1} \qquad C + D = \overline{INH2}$$

Sin embargo, la salida lógica es más compleja porque las líneas de inhibición son también usadas por el troceador.

3.4 ETAPA DE POTENCIA Y SECUENCIADOR PROPUESTA

El circuito que se propone se muestra en la figura 3.14 consta de dos etapas, la etapa del secuenciador y la etapa de potencia, se elaboró un circuito por cada motor.

Los valores utilizados para las resistencias y capacitores en la etapa del secuenciador fueron los recomendados en la nota de aplicación del L297 [12] y se utilizaron compuertas AND 7408, y NOT 7404 para el manejo de las señales de fase ABCD.

Las señales de control llegan de la etapa de aislamiento a los puentes JP1 y JP2 mostrados en la figura 3.15, se utilizó una compuerta OR para tener un reinicio manual y por programa, la salida de las compuertas AND van directamente a las resistencias que activan los transistores del puente H.

El puente H, mostrado en la figura 3.16 fue construido con transistores NPN y PNP, TIP41 y TIP42. Se seleccionaron estos transistores porque soportan la corriente que consume cada motor. Como los TIP 42 van atados a Vss se les colocó en sus bases un transistor BC547 que serviría como interruptor para llevarlos de su estado activo a su estado de corte y así evitar que los dos se encuentren activos al mismo tiempo y produzcan un corto circuito.

Los diodos que se utilizaron para hacer circular las corrientes de marcha libre fueron los NTE 588 que son diodos de conmutación rápida y que soportan una corriente de 3A.

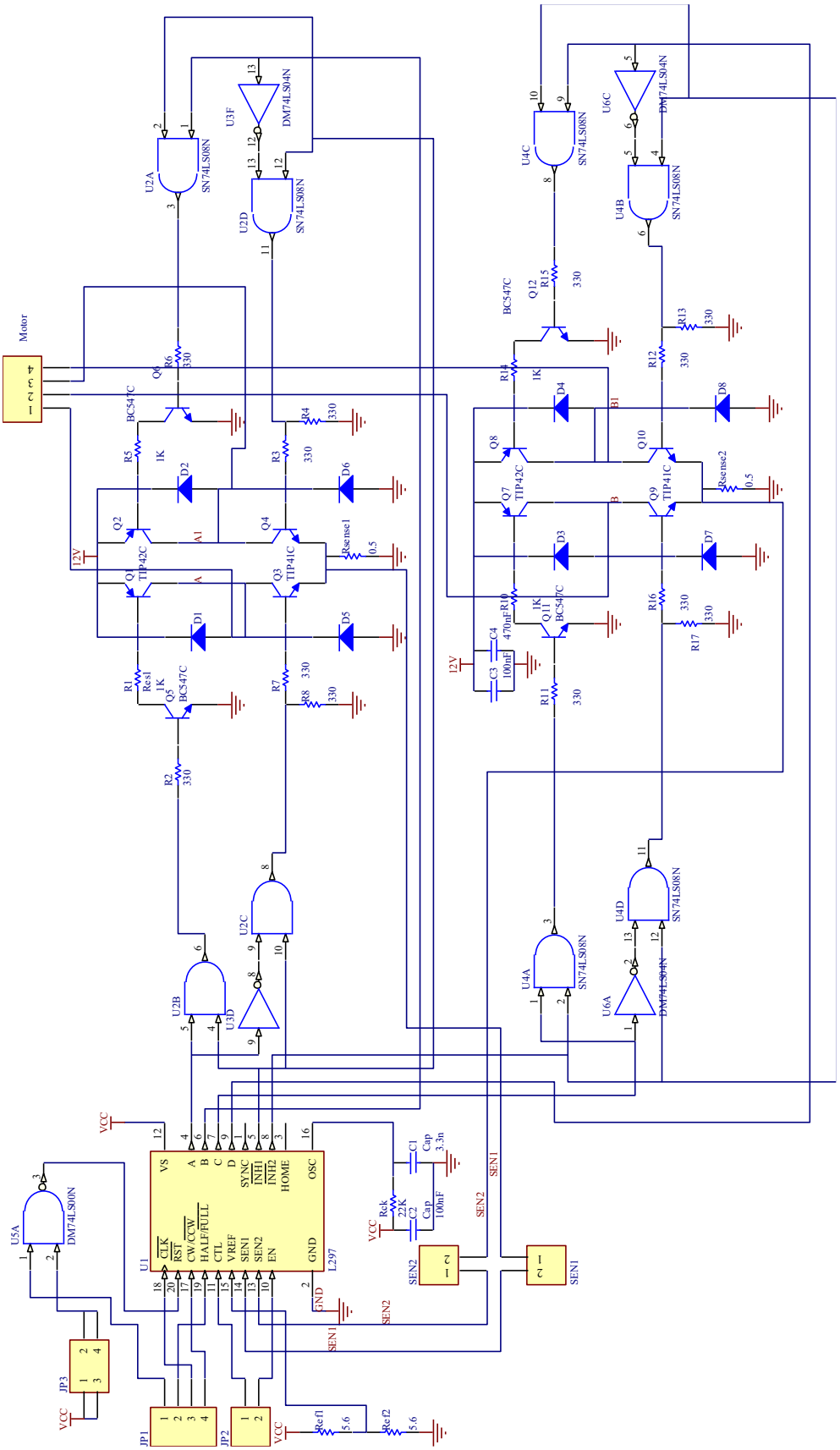


Figura 3.14 Etapa de potencia y secuenciador

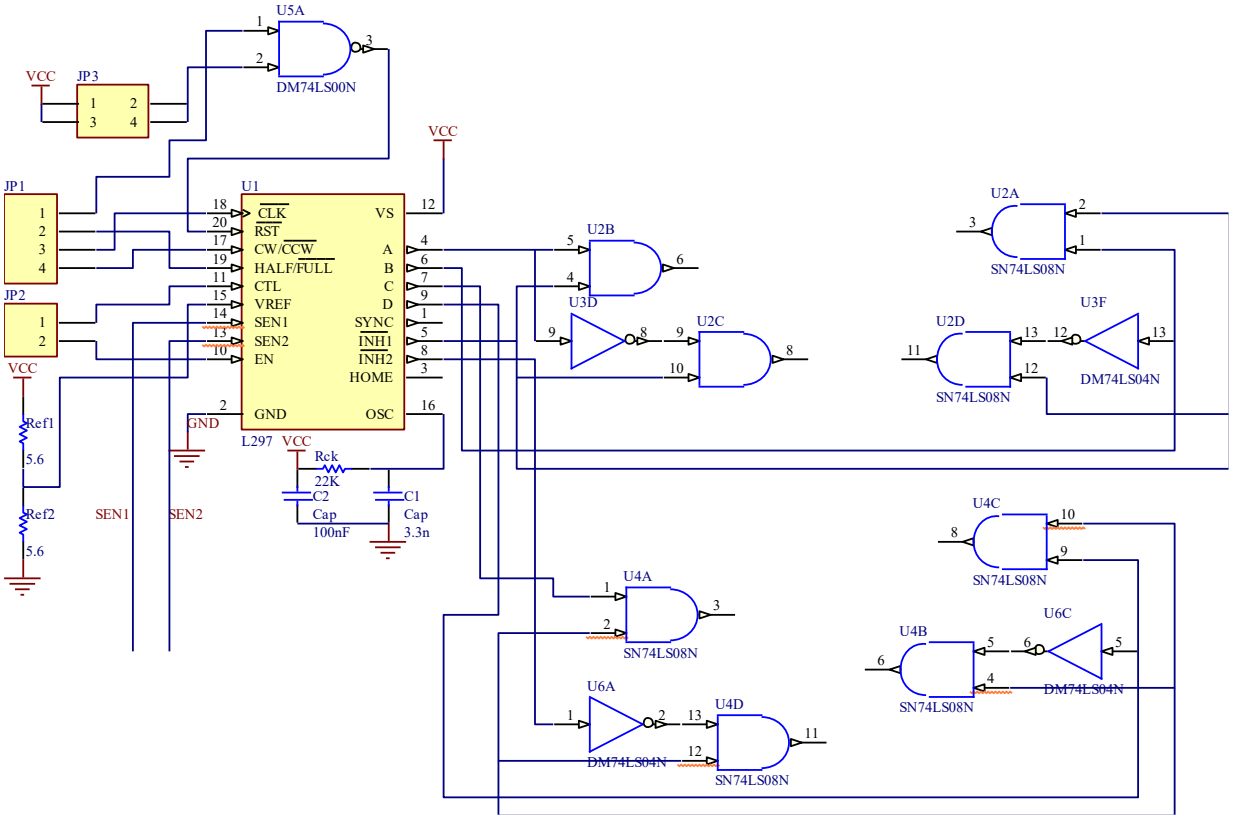


Figura 3.15 Etapa del secuenciador

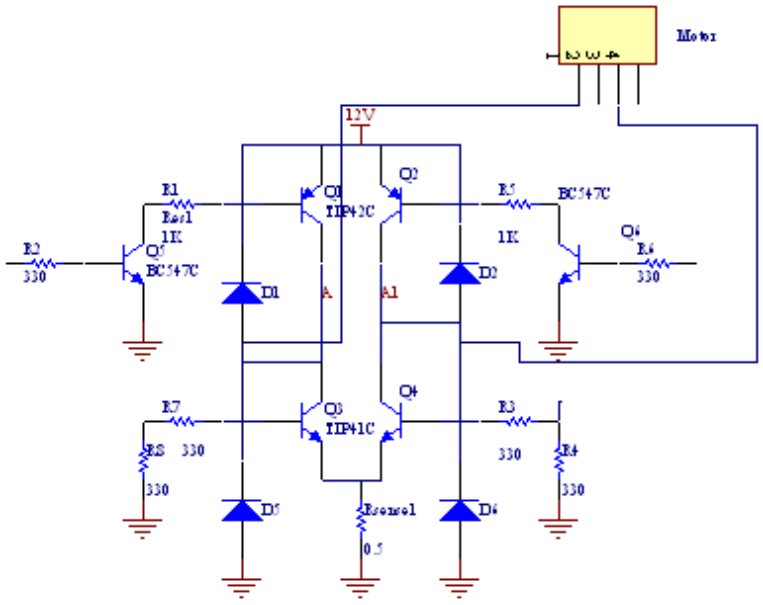


Figura 3.16 Etapa de potencia para una fase

La etapa de potencia y la etapa que contiene el secuenciador se construyeron en un solo módulo, cada módulo controla un motor de pasos. En la figura 3.17 se muestra una foto de un módulo que contiene a etapa de potencia y la de secuenciador.

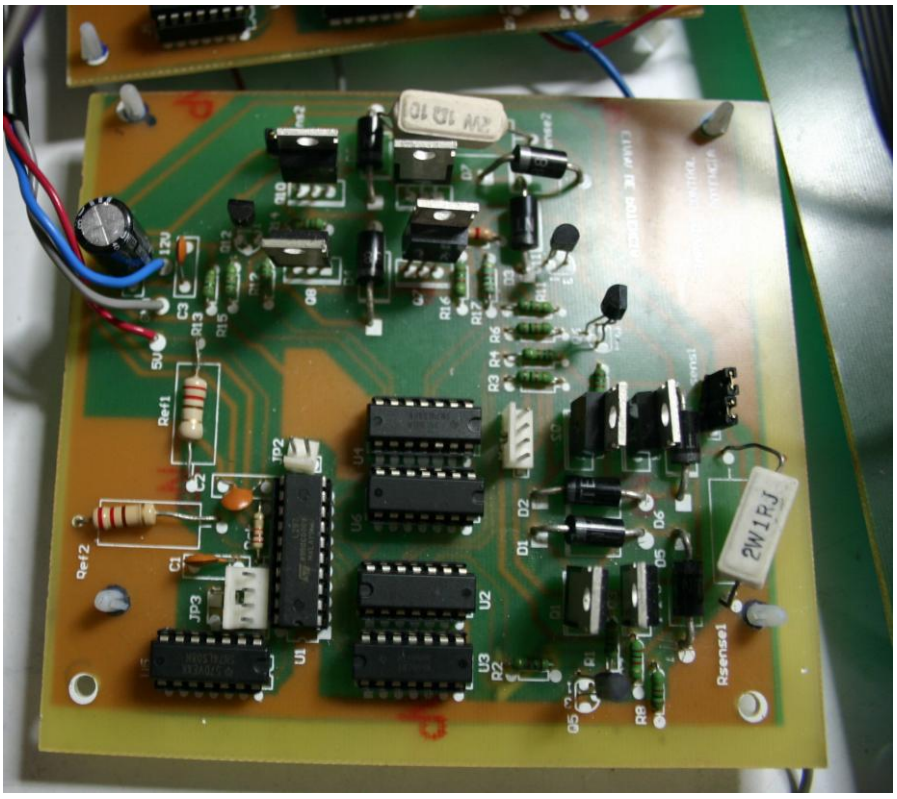


Figura 3.17 Etapa de potencia y de secuencia

3.5 SELECCIÓN DEL DISPOSITIVO DE CONTROL

Como se mencionó en 1.3.3, el controlador de entrada es el circuito que genera el tren de pulsos requerido para acelerar o desacelerar el motor de pasos, en este caso describiremos tres dispositivos que pueden servir como controladores de entrada: el microprocesador, el microcontrolador y el procesador digital de señales

3.5.1 Microprocesadores y Microcontroladores

Un microprocesador es un circuito digital de alta escala de integración que contiene la CPU completo en un sólo chip, es decir, contiene la unidad aritmética lógica (ALU), la lógica de control y Registros de memoria [16].

La ALU es el elemento clave del procesamiento del microprocesador, está gobernada por la unidad de control para realizar operaciones aritméticas y operaciones lógicas, los datos de la ALU los obtiene de los registros del microprocesador. Los microprocesadores contienen una colección de registros en los que durante la ejecución de un programa se almacenan temporalmente los datos y direcciones. Algunos registros se clasifican como registros de propósito general, otros registros tienen capacidades y funciones específicas y no se pueden usar como registros de propósito general, además existen registros denominados invisibles que son utilizados únicamente por el procesador y que no están disponibles para el programador.

La unidad de control se encarga de procesar las instrucciones internas y externas que utilizan los microprocesadores. Proporciona las señales de temporización y control para extraer e introducir los datos en el microprocesador y para sincronizar la ejecución de instrucciones.

Los buses son parte de las conexiones internas y externas que utilizan los microprocesadores para desplazar datos, direcciones y señales de control. El bus de direcciones tiene un solo sentido a través del cual el microprocesador envía un código de dirección a una memoria o a un dispositivo externo. El tamaño o anchura del bus de direcciones se especifica mediante el número de conductores. Cuantos más bits tenga el bus de direcciones a más posiciones de memoria puede acceder. El bus de datos es bidireccional, por lo que los datos o instrucciones se transfieren al microprocesador, o por lo que los resultados de una operación enviados por el microprocesador. El bus de control sirve para coordinar las operaciones y comunicarse con los dispositivos externos, este bus tiene señales que permiten leer y escribir datos en la memoria o puerto en el instante apropiado. Las líneas de control también se utilizan para insertar estados de espera, etc.

Un microcontrolador desde un punto de vista general es un circuito integrado programable que controla la operación de un sistema, desde un punto de vista técnico es un dispositivo que contiene un microprocesador, memoria para almacenar programas y datos que puede ser RAM, ROM, PROM, EPROM, EEPROM o Flash, líneas de entrada y salida a periféricos

Existen varios tipos de controladores como son los CISC que son los controladores que tienen un *set* de instrucciones complejo, esto es más de 80 instrucciones las cuales son sofisticadas y potentes que requieren muchos ciclos de instrucción para su ejecución. Los controladores RISC son controladores de set de instrucciones reducido, con pocas instrucciones que son más simples y que se ejecutan generalmente en un sólo ciclo de instrucción.

3.5.1.1 PROGRAMACIÓN DE MICROPROCESADORES Y MICROCONTROLADORES

Todos los microprocesadores trabajan con un conjunto de instrucciones básico, formulado por los diseñadores del procesador, hay básicamente tres tipos de instrucciones; aritméticas, lógicas y de transferencia de información. Cada instrucción consta de una cadena binaria que el microprocesador decodifica antes de ser ejecutada.

Las instrucciones en código binario se denominan lenguaje máquina y son las únicas que reconoce el microprocesador. Como programar en lenguaje de máquina es tan tedioso y propenso a errores se creó el lenguaje ensamblador, que es un lenguaje de bajo nivel el cual convierte las instrucciones escritas en una sintaxis similar al inglés y denominadas mnemónicos en lenguaje máquina. Otras instrucciones utilizadas en lenguaje ensamblador describen estructuras de datos y otras informaciones que necesita el programa ensamblador, estas instrucciones se denominan pseudo-operaciones, ya que no se traducen a instrucciones reales que emplee el microprocesador.

El lenguaje ensamblador y el correspondiente lenguaje máquina son específicos para un tipo de procesador o familia de procesadores. Los lenguajes de alto nivel son independientes del tipo de microprocesador, un programa llamado compilador o intérprete traduce las instrucciones de un programa escrito en un lenguaje de alto nivel a lenguaje máquina.

La ventaja del lenguaje ensamblador es que el programador tiene un control directo del proceso y normalmente, los programas escritos en dicho lenguaje se ejecutan más rápidamente. La ventaja de los lenguajes de alto nivel es que se puede diseñar un lenguaje de propósito general para aplicaciones específicas

3.5.2 Procesadores Digitales de Señales

Los procesadores digitales de señales (DSP) son microprocesadores especializados para efectuar operaciones numéricas en tiempo real, en aplicaciones donde se requieren numerosos cálculos aritméticos en tiempo limitado [17].

Por sus aplicaciones específicas, los DSPs tienen arquitecturas que son significativamente diferentes a los microprocesadores tradicionales. En su arquitectura los DSPs incorporan unidades de multiplicación y acumulación donde el tiempo de un ciclo de instrucción puede ser igual al tiempo de de un ciclo de la aritmética en hardware. En la actualidad existen DSPs que efectúan multiplicaciones de 32 bits en punto flotante en 60 u 80ns, esto es de 25 a 30 millones de operaciones de punto flotante por segundo (MFLOPS). Los actuales DSPs utilizan extensivamente operaciones en *pipeline*, memorias independientes y unidades de trabajo en paralelo.

Los DSPs disponen de una arquitectura y un conjunto de instrucciones orientadas al procesamiento digital de señales. La arquitectura de un DSP normalmente contiene buses separados para el direccionamiento simultáneo de dos operandos, un multiplicador, corrimientos, una unidad de direccionamiento, puertos seriales y

temporizadores. Entre las instrucciones de grandes posibilidades están el movimiento de bloques, multiplicaciones en un ciclo de instrucción, multiplicación acumulación para realizar operaciones de convolución.
Se eligió un DSP como controlador de entrada ya que es un dispositivo que permite realizar control y que en un futuro se puede programar para realizar reconocimientos de imágenes y otros algoritmos de control más complejos. Además, se busca evaluar una solución distinta a los sistemas de control desarrollados anteriormente y así poder comparar desempeños para en un futuro seleccionar la mejor alternativa para controlar las micromáquinas herramienta.

Uno de los requerimientos para seleccionar el DSP más adecuado fue el que presentara el mayor número de entradas y salidas. Se realizó una investigación entre dos familias de DSPs que se consideran como familias de controladores híbridos ya que combinan en un mismo chip el poder de procesamiento de un DSP y la funcionalidad de un microcontrolador, además de periféricos que facilitan el control de motores. Las familias que se revisaron fueron las TMS320CXX TMS320C2000 de *Texas Instruments*™ y la familia de controladores híbridos 56800/E de *Motorola*™ que consiste de dos subfamilias llamadas DSP56F80x/82x y DSP56F83xx/85x.

Las ventajas que presentan las familias de Motorota™ sobre las familias de Texas Instruments™ TMS320CXX y TMS320C2000 son los siguientes periféricos dedicados al control de motores de varios tipos :

- Módulos PWM,
- Módulos decodificadores de cuadratura (*Quadrature Decoders*), que nos permiten conocer la posición y velocidad de la flecha. Módulos
- *Quad Timers* que permiten generar cadenas de pulsos, hacer codificaciones de señales Quad, que nos permiten conocer los movimientos de las flechas.
- Los puertos de entrada y salida están integrados en la tarjeta sin necesidad de realizar interfaces adicionales.

Por lo anterior se decide por la familia DSP56F83xx y emplear en particular el DSP56F8357 mostrado en la figura 3.18, que a continuación se describe en detalle:

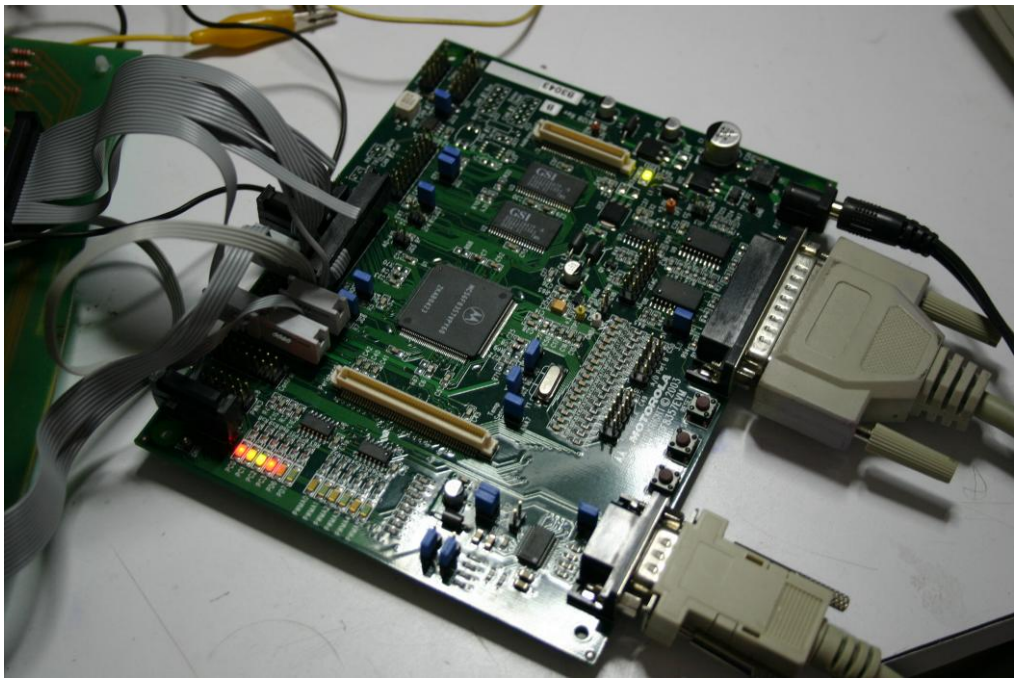


Figura 3.18 Tarjeta de evaluación 56F8357EVM

3.5.3 INTRODUCCIÓN AL DSP56F8357

Para manejar un nuevo microcontrolador o microprocesador se deben conocer los siguientes tópicos:

- Diagrama de bloques
- El modelo de programación
- Los mapas de memoria
- Los modos de direccionamiento
- El set de instrucciones

3.5.3.1 DIAGRAMA DE BLOQUES

En la figura 3.19 se muestra el diagrama de bloques del DSP56F8357, sus características más importantes:

- Funcionalidad de DSP y de MCU en una arquitectura unificada
- Espacio de memoria de dirección 24-bit *data memory address space*
- Espacio de direccionamiento de memoria de datos de 24bit
- Hasta 76 Puertos IO de propósito general (GPIO)
- 2 Módulos PWM de 6 canales cada uno
- 4 módulos ADC de 4 canales de 12bit
- 2 interfaces seriales (SCIs)
- Hasta 2 interfaces periféricas (SPIs)
- Hasta 4 *Quad timers* de propósito general [18]

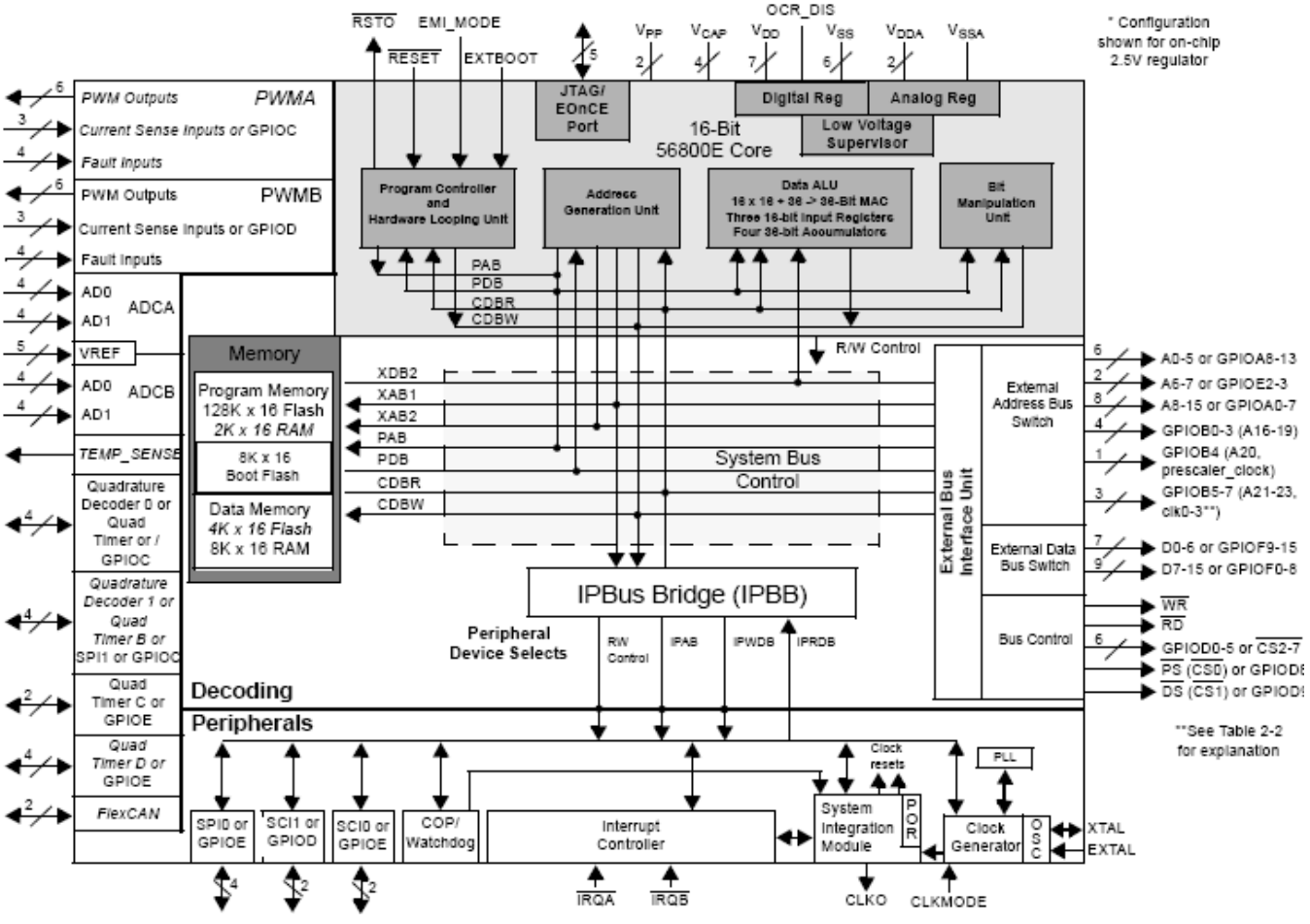


Figura 3.19 Diagrama de bloques DSP56F8357

3.5.3.2 MODELO DE PROGRAMACIÓN

El DSP56F8357 pertenece a la familia de procesadores DSP56800E, los registros que se consideran parte del modelo de programación para esta familia se muestran en la figura 3.20. Los registros de los periféricos de cada chip en particular se encuentran mapeados en un bloque de 64 localidades localizadas en la memoria.

El DSP56800E tiene una arquitectura Harvard que separa las memorias de programa y de datos. Esta arquitectura permite accesos simultáneos a las memorias de datos y de programa. La interfaz de memoria también soporta dos operaciones simultáneas de lectura y permite hasta tres accesos de memoria simultáneos de datos[19].

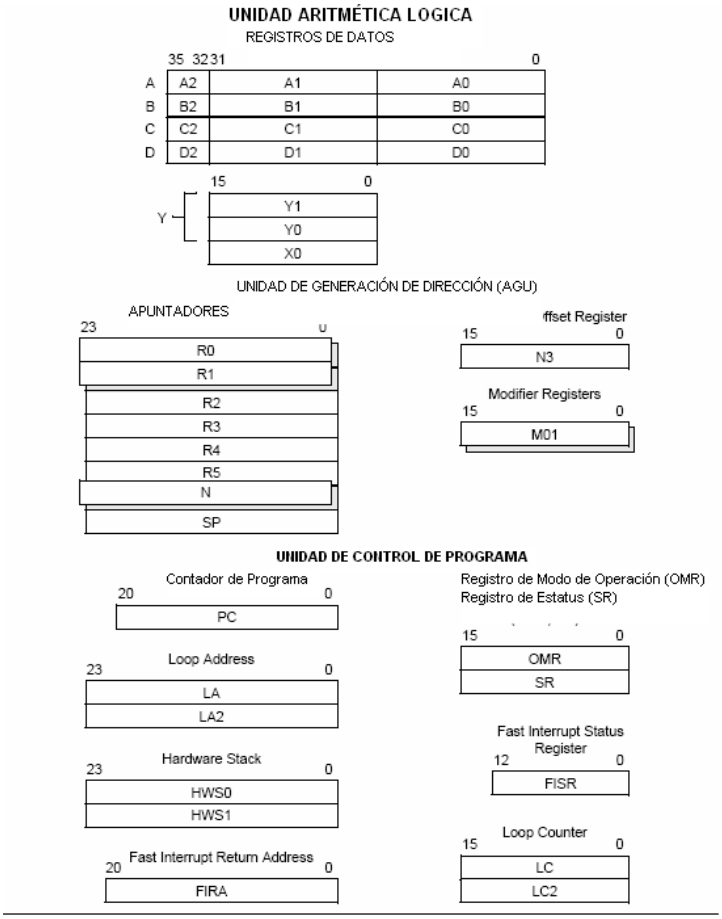


Figura 3.20 modelo de programación de la familia 56F800E

El DSP56800E está compuesto de varias unidades funcionales independientes. El controlador de programa, la unidad generadora de direcciones (AGU) y la unidad lógica-aritmética, además de los buses de dirección, de datos, la unidad de manipulación de bits y el modulo *Enhanced OnCE debugging module*

La unidad lógica aritmética de datos realiza todas las operaciones aritméticas, lógicas y de cambio sobre los operandos. Contiene entre otros componentes 3 registros de datos de 16bits. X0, Y0 y Y1, además de 4 registros acumuladores de 36bits A, B, C, D. Los operandos de la ALU pueden tener tamaños de 8,16, 32 o 36 bits y pueden ser localizados en memoria, en instrucciones inmediatas o en los registros de la ALU.

3..5.3.3 MODOS DE DIRECCIONAMIENTO

Los modos de direccionamiento especifican donde se pueden encontrar los operandos de una instrucción, en un valor inmediato, en un registro o en una localidad de memoria y provee la dirección exacta de los operandos. Los modos de operación de la familia DSP56800E se agrupan en cuatro categorías:

- Directo a registro: Directamente refiere a los registros en el chip como operandos.

- Indirecto a registro: Usa un registro de dirección como un apuntador para hacer referencia a la localidad en la memoria como un operando. El registro apuntador puede ser actualizado de alguna de las siguientes maneras: postincremento, postdecremento, etc

- Inmediato: El operando está contenido como un valor dentro de la misma instrucción.

- Absoluto: Usa la dirección contenida dentro la instrucción misma para hacer referencia a la localidad en la memoria como un operando

3.5.3.4 SET DE INSTRUCCIONES

El set de instrucciones puede dividirse en varias categorías generales que están basados en sus funciones: Multiplicación, aritméticas, *shifting*, lógicas, aritmética AGU, manipulación de bits, *looping*, movimiento, control de programa

3.5.3.5 MAPA DE MEMORIA

En las figuras 3.21 y 3.22 se muestran los mapas de memoria de programa y de datos del DSP56F8357

<table>
<tr><th rowspan="3">Begin/End Address</th><th>Mode 0 (MA = 0)</th><th colspan="2">Mode 1[1] (MA = 1)</th></tr>
<tr><th>Internal Boot</th><th colspan="2">External Boot</th></tr>
<tr><th>Internal Boot
16-Bit External Address Bus</th><th>EMI_MODE = 0[2,3]
16-Bit External Address Bus</th><th>EMI_MODE = 1[4]
20-Bit External Address Bus</th></tr>
<tr><td>P:$1F FFFF
P:$10 0000</td><td rowspan="2">External Program Memory[5]</td><td rowspan="2">External Program Memory[5]</td><td>External Program Memory[5]</td></tr>
<tr><td>P:$0F FFFF
P:$03 0000</td><td rowspan="6">External Program RAM
COP Reset Address = 02 0002[6]
Boot Location = 02 0000[6]</td></tr>
<tr><td>P:$02 FFFF
P:$02 F800</td><td colspan="2">On-Chip Program RAM
4KB</td></tr>
<tr><td>P:$02 F7FF
P:$02 2000</td><td colspan="2">Reserved
116KB</td></tr>
<tr><td>P:$02 1FFF
P:$02 0000</td><td>Boot Flash
16KB
COP Reset Address = 02 0002
Boot Location = 02 0000</td><td>Boot Flash
16KB
(Not Used for Boot in this Mode)</td></tr>
<tr><td>P:$01 FFFF
P:$01 0000</td><td rowspan="2">Internal Program Flash
256KB</td><td>Internal Program Flash[7]
128KB</td></tr>
<tr><td>P:$00 FFFF
P:$00 0000</td><td>External Program RAM
COP Reset Address = 00 0002
Boot Location = 00 0000</td></tr>
</table>

Figura 3.21 mapa de memoria de programa del DSP56F8357

Begin/End Address	EX = 0[2]	EX = 1
X:$FF FFFF X:$FF FF00	EOnCE 256 locations allocated	EOnCE 256 locations allocated
X:$FF FEFF X:$01 0000	External Memory	External Memory
X:$00 FFFF X:$00 F000	On-Chip Peripherals 4096 locations allocated	On-Chip Peripherals 4096 locations allocated
X:$00 EFFF X:$00 3000	External Memory	External Memory
X:$00 2FFF X:$00 2000	On-Chip Data Flash 8KB	
X:$00 1FFF X:$00 0000	On-Chip Data RAM 16KB[3]	

Figura 3.22 mapa de memoria de datos del DSP56F8357

3.6 DISEÑO DE ETAPA DE AISLAMIENTO

Una vez que fueron seleccionados los puertos por donde saldrían las señales se diseñó una etapa que aislara de manera óptica la tarjeta DSP56F8357EVM del las etapas que manejan una gran corriente, ya que, en el caso de que hubiera un cortocircuito en las etapas que manejan una corriente más grande éste no llegara a la tarjeta del DSP.

Se contaba con dos circuitos integrados que podían aislar la tarjeta del DSP con el resto de las tarjetas, uno de ellos es el buffer 74LS244, que es un buffer TRI-ESTADO, que se muestra en la figura 3.23.

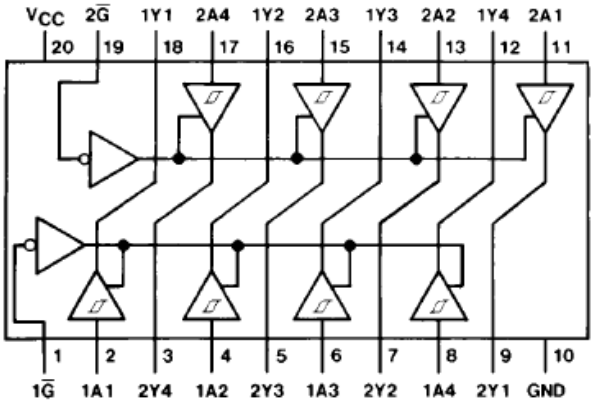


Figura 3.23 Buffer 74LS244

Nominalmente esta es su tabla de verdad

Inputs		Output
$\overline{G}$	A	Y
L	L	L
L	H	H
H	X	Z

L = Low Logic Level
H = High Logic Level
X = Either Low or High Logic Level
Z = High Impedance

Tabla 4.1 Tabla de verdad del buffer 74LS244

Pero este circuito presenta muchas inestabilidades, para el nivel bajo presenta un nivel de 1.2V y para el nivel alto presenta un nivel de 2.4V. Por lo que los circuitos no presentan un buen funcionamiento.

Se hicieron pruebas con el circuito PC847, que es un circuito que contiene 4 optoaisladores como se muestra en la figura 3.24.

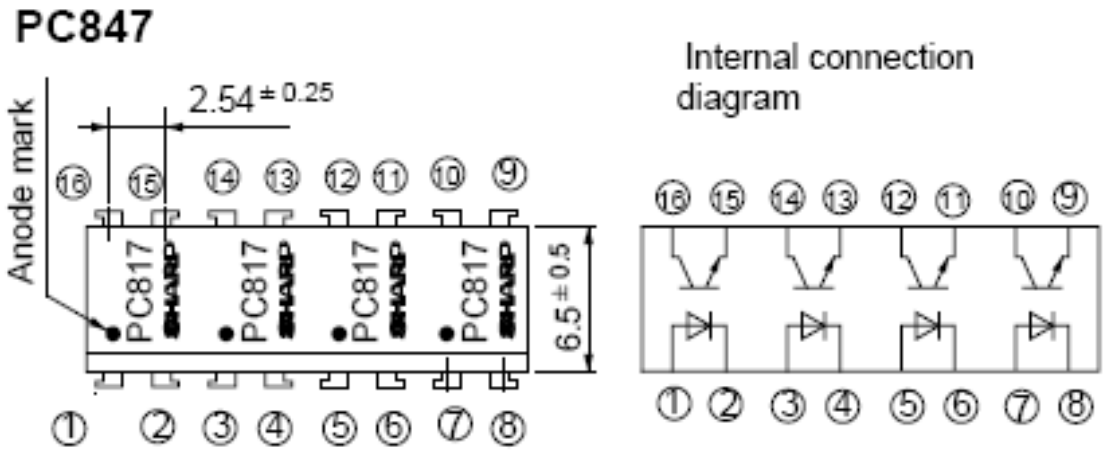


Figura 3.24 Optoaislador de 4 canales PC847

Un optoaislador es un dispositivo que contiene un LED infrarrojo y un fotodetector, como un diodo de silicio o un par de transistores Darlington o un SCR. La respuesta de longitud de onda de cada dispositivo se ajusta para ser lo más idéntica posible, a fin de permitir la más alta medida de acoplamiento posible.

El circuito de aplicación típica se muestra en la figura 3.25; cuando se aplica una tensión que genera una corriente que fluya a través del fotodiodo se establece un flujo luminoso que funciona como corriente de base para el fotodetector, con lo cual resulta una corriente de colector de salida que origina una caída de tensión en la resistencia RL. Se presenta un aislamiento óptico entre la red a la que pertenece el fotodiodo y la que tiene el fotodetector ya que no hay una conexión física entre ellos.

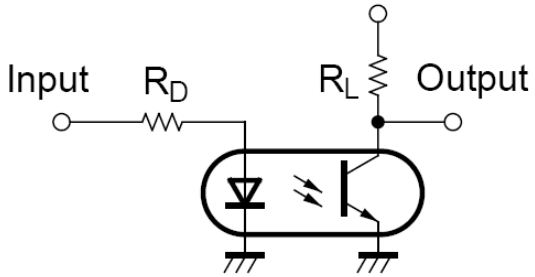


Figura 3.25 Aplicación Típica de un optoaislador

El nivel alto obtenido con este circuito fue de 3.4V y del nivel bajo fue de 0.2V por lo que se decidió utilizar este circuito integrado.

El circuito de aislamiento se muestra en la figura 3.26, está divido en dos partes:

La primera parte aísla las señales de control que son: Reset, Sentido de giro, activación, paso, control y reloj que vienen directamente de la tarjeta DSP56F8357EVM a través de un cable plano de 22 pines y las manda al circuito integrado PC847, que es un circuito que contiene 4 optoaisladores como se muestra en la figura 3.25, y las señales que salen de este circuito se mandan a la etapa del secuenciador. Se requirieron siete circuitos PC847 para aislar todas las señales de control, ya que se tienen 6 señales de control por motor y cuatro señales de sensado, lo que nos da un total de 28 señales a ser aisladas.

La segunda parte del circuito, mostrada en la figura 3.27 es la que aísla las 3 señales que indican que los ejes X, Y y Z de la micromáquina están al inicio de su carrera y la señal que indica que la herramienta del husillo ha hecho contacto con

el material. Estas señales son mandadas a una etapa de acondicionamiento de señal formada por transistores BC547 y una vez que tienen el nivel necesario son mandadas al circuito PC847 para posteriormente mandarlas al DSP. El circuito que se fabricó se muestra en la figura 3.28.

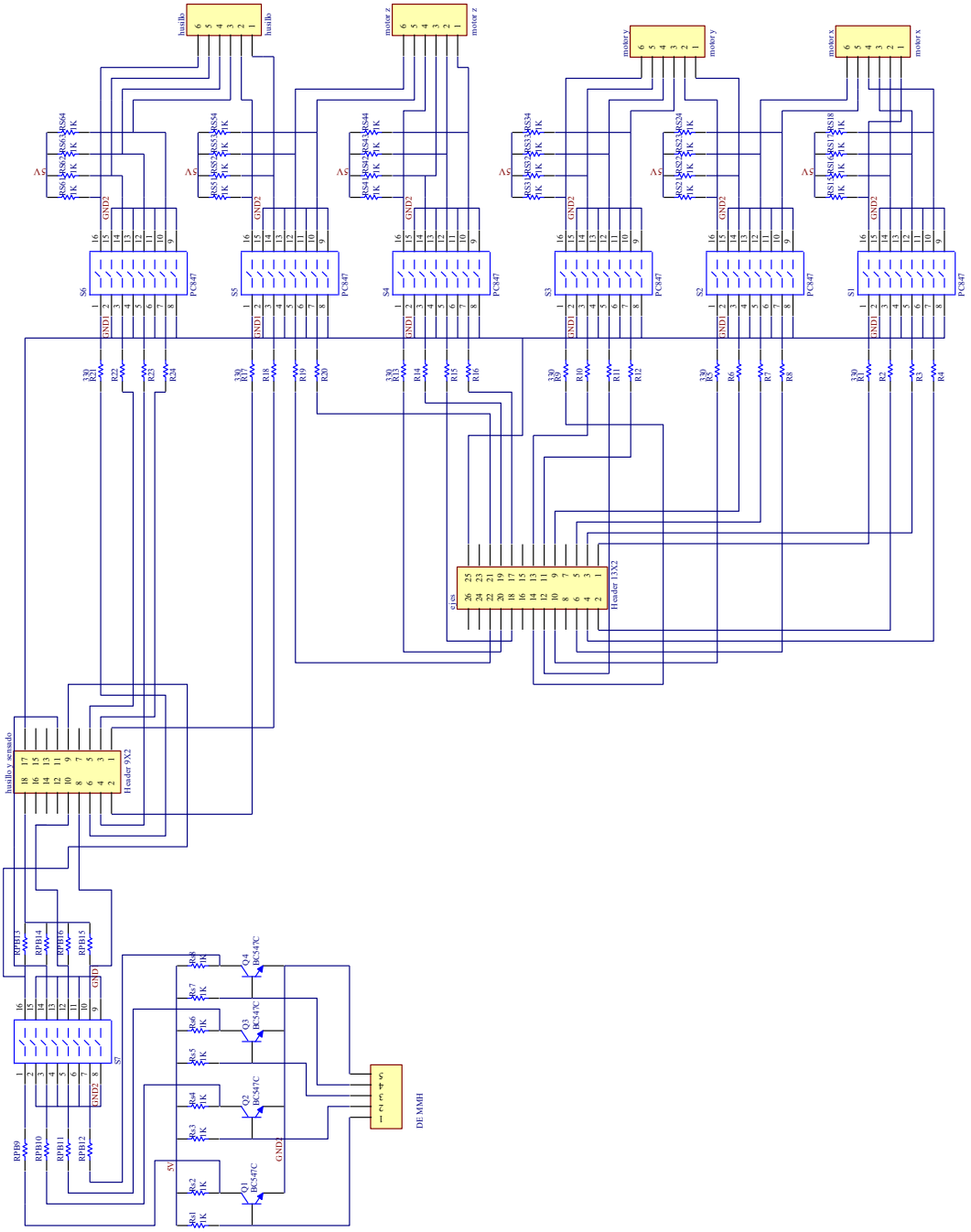


Figura 3.26 Etapa de aislamiento y sensado

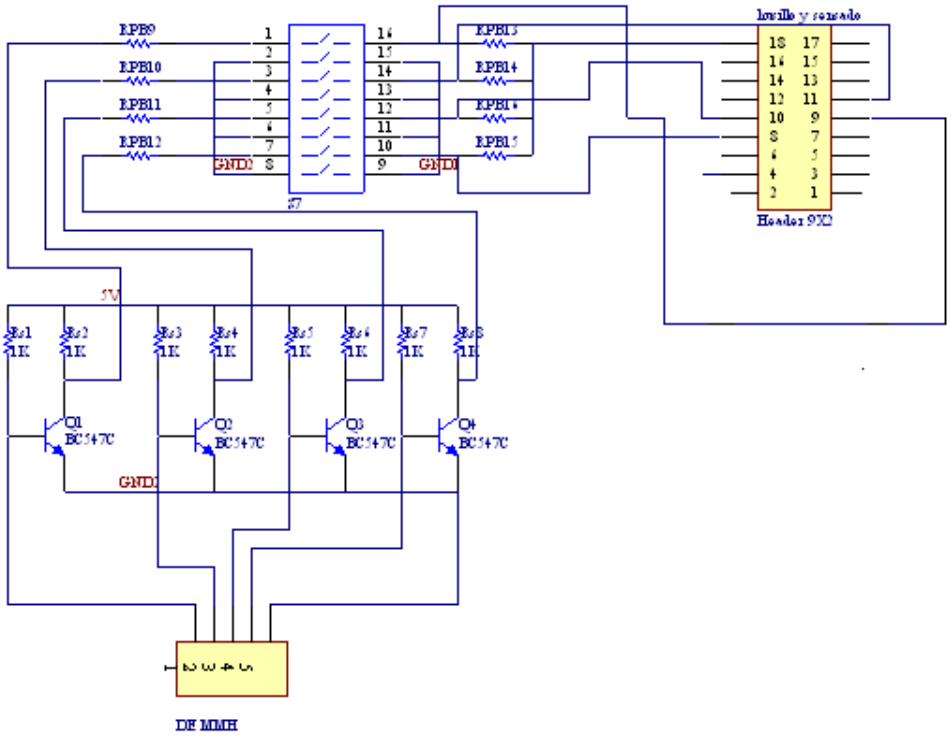


Figura 3.27 Circuito de sensado de señales de posición de inicio

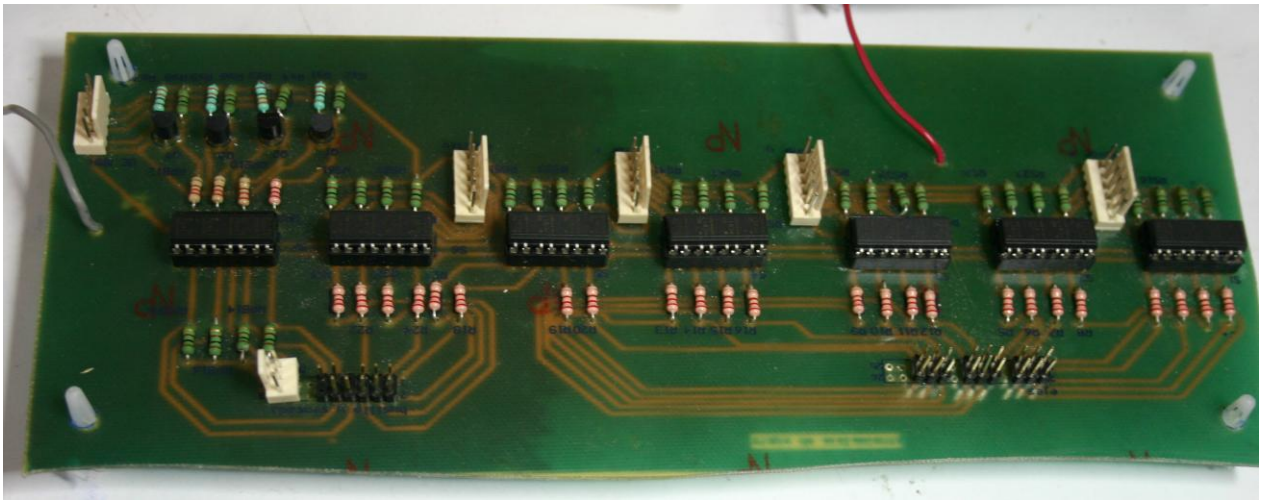


Figura 3.28 Etapa de aislamiento y sensado

3.7 DESARROLLO DEL PROGRAMA

Una vez que se completó el hardware que se requería, se procedió a realizar el programa para controlar la micromáquina. Las necesidades que debía cubrir el programa son las siguientes:

- Operación en Modo Manual (*Joystick*)
- Operación indicando el número de pasos.
- Operación Figura
- Posición HOME

Y los movimientos que se necesitaban llevar a acabo son

- Translación para los ejes X, Y y Z.
- Rotación para el husillo
- Interpolación lineal para los ejes X y Z

Aprovechando los conocimientos que se habían generado al realizar los sistemas de control anteriores, utilizamos el lenguaje de programación Visual Basic 6.0 para desarrollar la interfaz de usuario, como en ese lenguaje es sencillo controlar el puerto serial se decidió que la transmisión de datos seria a través de ese puerto. Las señales de control que se utilizaron fueron:

RELOJ	que consistía en un tren de pulsos, con el que cada pulso incrementaba un paso en el motor.
SENTIDO DE GIRO	Una señal en alto realizaría un giro en sentido horario y una señal en bajo daría un sentido de giro antihorario al motor de pasos
PASO	Una señal en alto daría un funcionamiento a medio paso y una señal en alto daría un funcionamiento a paso completo.
RESET	Una señal en cero resetearía el secuenciador

ENABLE	Una señal en cero desactiva el secuenciador y por lo tanto desactivaría el sistema de control.
CONTROL	Para definir la acción del Troceador.

El programa fue realizado en lenguaje ensamblador y se utilizó el programa CODEWARRIOR IDE que sirve para editar, ensamblar, depurar y bajar el programa a la tarjeta de evaluación. A continuación se explicará con detalle el funcionamiento de cada uno de los bloques que lo conforman.

El diagrama de bloques del programa en general se muestra en la figura 3.29

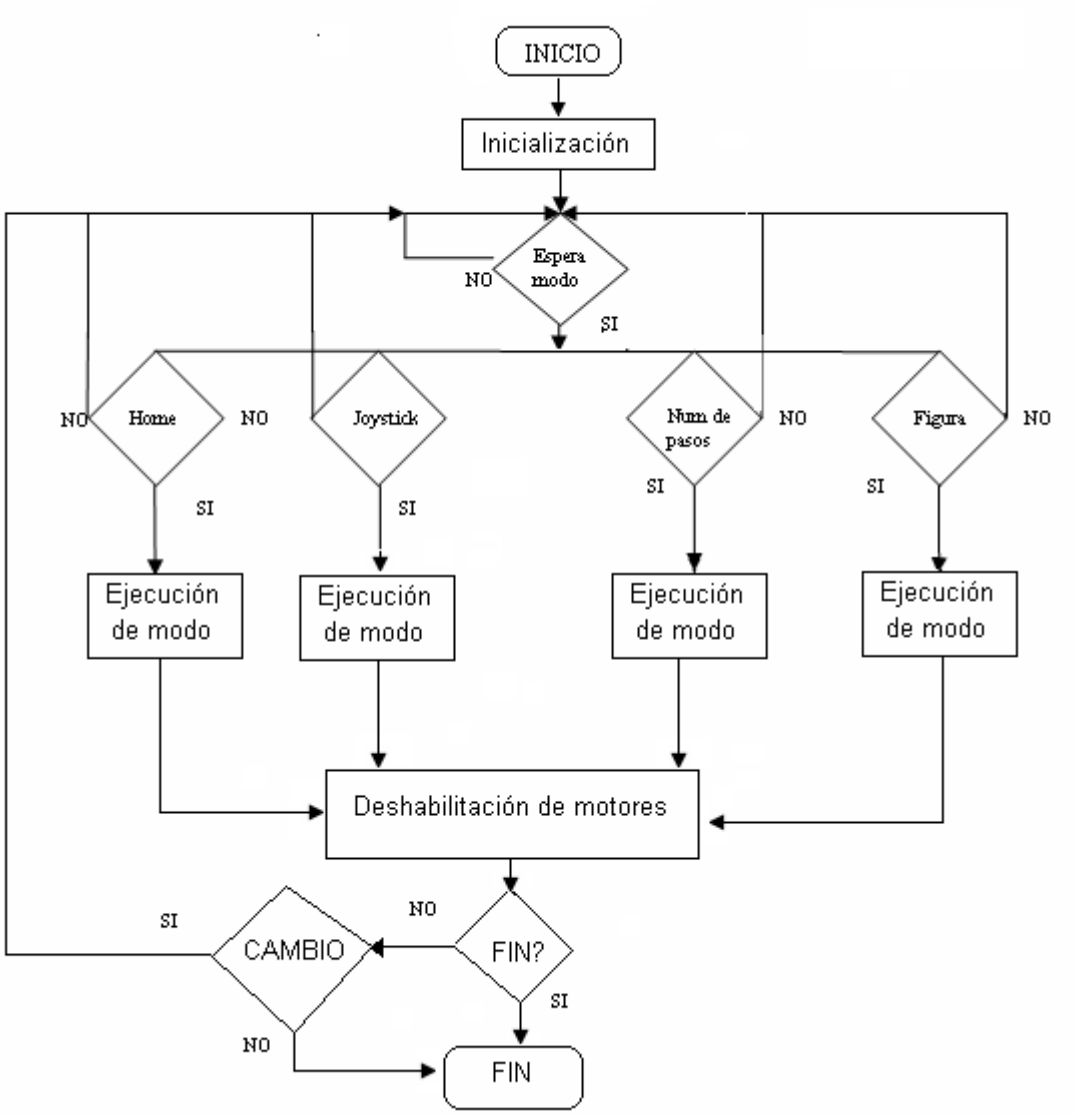


Figura 3.29 Diagrama de flujo del programa

3.7.1 Declaración de registros y variables generales

El programa inicia con la declaración y configuración de los registros y variables que se necesitan, (ver figura 3.30) y en esta parte se utiliza la instrucción EQU para utilizar los nombres de los registros y de las variables en lugar de sus direcciones, los registros que se usan son los de los puertos GPIOB, GPIOC, GPIOD, GPIOE y el módulo de transmisión serial SCI0. Las variables generales utilizadas se mencionan en la siguiente tabla.

Tabla 4.2 Variables Generales

Variable	Descripción
DATO	Contiene el valor que fue emitido por la interfaz
RELOJ	Contiene el valor del periodo de los trenes de pulsos de los motores X, Y y Z.
RETHUS	Contiene el valor del periodo del tren de pulsos del motor del husillo.
PASOSx	Contiene el número de pasos que dará el motor X.
PASOSy	Contiene el número de pasos que dará el motor Y.
PASOSz	Contiene el número de pasos que dará el motor Z.
RESET	Contiene el valor FFFF que proporciona un retardo, entre la desactivación y la activación del L297.

Las subrutinas que se usan para todos los modos son las siguientes:

Tabla 4.3 Subrutinas generales

Subrutina	Descripción
ESP_DATO	Rutina que espera que se reciba un dato a través del puerto serial SCI0
TIEMPO	Rutina que proporciona el retardo para las señales de reloj
RETARDO	Rutina que proporciona un retardo entre la desactivación y activación del L297

3.7.2 Inicialización de registros y variables

Se configuran los registros GPIO como salidas y 4 LSB del GPIOC se configuran como entradas para sensar las posiciones de inicio de los motores y el contacto del material con la herramienta, se configura el SCI0 para tener una transmisión a 9600 baudios, 1bit de inicio, 8 bits de datos y 1bit de parada. Todas las variables se borran.

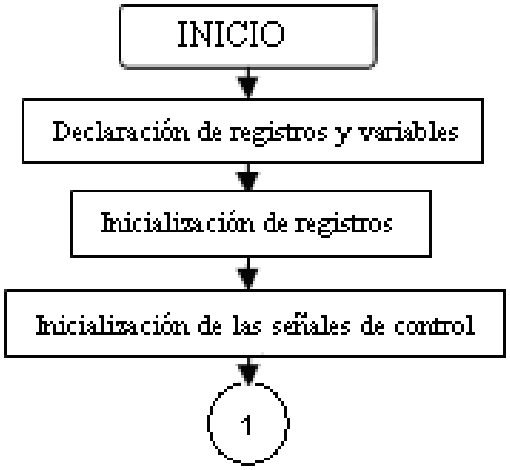


Figura 3.30 inicialización de los registros y variables

3.7.3 Inicialización de las señales de control

Los bits B7:B2 del puerto B controlan al motor X, los bits E13:E8 del puerto E controlan al motor Y, los bits E7:E4 y D1:D0 de los puertos E y D controlan al motor Z y los bits C9:C4 del puerto C controlan al husillo. A los bits C3:C0 llegan las señales de las posiciones de inicio.

Usando los registros anteriores los motores de los ejes X, Y y Z quedan configurados a paso completo, el troceador actuará sobre la líneas de activación, con reset en alto y desactivados para que no consuman corriente cuando el sistema no se esta utilizando, el sentido de giro y el reloj se le da según el tipo de movimiento seleccionado.

Antes de iniciar cualquier movimiento es recomendable reiniciar el L297 para asegurar que funcione correctamente, como se necesita una rutina de retardo para reiniciar al L297 se manda el Valor FFFFh a la variable RESET. Finalmente, el programa se queda esperando recibir el dato que le indica el modo en que va a trabajar, el programa brinca a la rutina de recepción de datos y se queda esperando hasta que la bandera RDRF del Registro de Estatus SCISR del SCI0 se ponga a uno, con lo que indica que se ha recibido un dato y está disponible en el registro de datos SCIDR. Para poder utilizar este dato se debe de leer primero el SCISR, después se manda el dato disponible a la variable DATO.

3.7.4 Selección de Modo

Al regresar de la subrutina de recepción de datos se compara el dato recibido con los valores preestablecidos para cada modo, así cuando el dato sea igual al valor del tipo modo el programa brincará al modo seleccionado.

3.7.4.1 MODO HOME

En la figura 3.31 se muestra el diagrama de flujo del modo HOME, este Modo hace que cada uno de los motores de los ejes X, Y y Z regresen a su posición de inicio. Las subrutinas utilizadas en este modo se muestran en la siguiente tabla.

Tabla 4.4 Subrutinas Modo HOME

Subrutina	Descripción
MotorxI	Mueve el motor X a la izquierda, en dirección hacia el sensor de fin de carrera.
MotoryI	Mueve el motor Y a la izquierda, en dirección hacia el sensor de fin de carrera.
MotorzAb	Mueve el motor Z hacia abajo, en dirección hacia el sensor de fin de carrera.

Se carga el valor BFFh a la variable RELOJ para tener una velocidad de 400Hz de reloj y 100Hz por paso para los tres motores X, Y y Z , a esta velocidad los motores no pierden pasos. Se carga el valor FFFFh a las variables PASOSx, PASOSy Y PASOSz, para garantizar que se alcance a hacer contacto con los sensores de posición de inicio.

Se llama a la subrutina MOTORxI que mueve el eje X en dirección al sensor, cuando el sensor se activa el motor se detiene, cambia de giro y avanza 200 pasos para superar el *juego mecánico* y pueda liberar el sensor posteriormente se desactiva el motor X y se llama a la subrutina MOTORyI que hace lo mismo, al igual que la subrutina MOTORzAb. Al terminar de colocar los ejes en su posición de inicio el sistema vuelve a la parte de selección de modo.

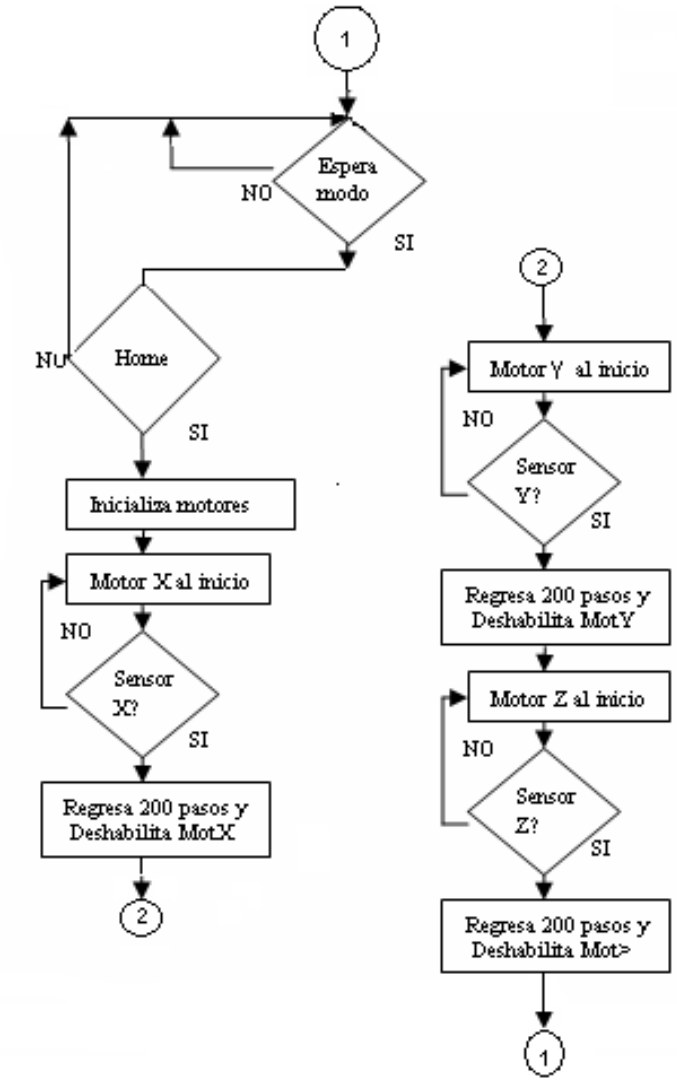


Figura 3.31 Modo HOME

3.7.4.2 MODO MANUAL

En este modo los motores se moverán según el botón seleccionado en la interfaz de usuario, los movimientos que se pueden realizar son: Eje X adelante, Eje X atrás, Eje Y adelante, Eje Y atrás, Eje Z arriba, Eje Z abajo, giro de husillo horario y giro de husillo antihorario. La primera parte de este modo se ilustra en el diagrama de flujo de la figura 3.32.

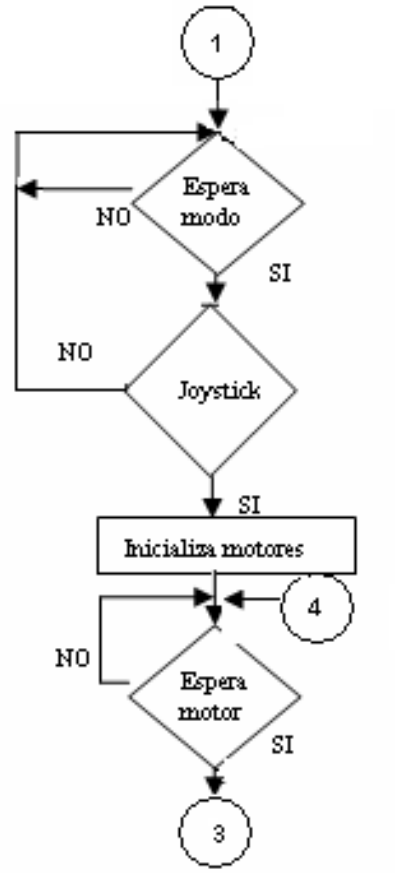


Figura 3.32 Inicialización Modo Manual

Las subrutinas empleadas en este modo se muestran en la siguiente tabla

Tabla 4.5 Subrutinas Modo Manual

Subrutina	Descripción
MotorxD	Mueve el motor X a la derecha.
MotorxI	Mueve el motor X a la izquierda, en dirección hacia el sensor de fin de carrera.
MotoryD	Mueve el motor Y a la derecha.
MotoryI	Mueve el motor Y a la izquierda, en dirección hacia el sensor de fin de carrera.
MotorzA	Mueve el motor Z hacia arriba.
MotorzAb	Mueve el motor Z hacia abajo, en dirección hacia el sensor de fin de carrera.
Husillo_D	Gira el motor del husillo en sentido horario.
Husillo_I	Gira el motor del husillo en sentido antihorario.

Al inicio, se borran las variables RELOJ, RETHUS, se carga el valor BFFh a la variable RELOJ para tener una velocidad de 400Hz de reloj y 100Hz por paso para los tres motores X, Y y Z, y se carga el valor 1300h a la variable RETHUS para tener una velocidad de 256 Hz de reloj y 64 HZ por paso para el motor del husillo, ya que este motor tiene bobinas con impedancias más grandes que las de los motores de los ejes. Se carga el valor FFFFh a las variables PASOSx, PASOSy y PASOSz. La parte del movimiento del motor, en este modo, se muestra en la figura 3.33, el programa espera al comando que le indica el motor y la dirección en la que se tiene que mover, si en el movimiento se pulsa el botón ALTO el movimiento se detiene y el motor se desactiva. Si no hay cambio de modo, espera el comando que le indica el siguiente motor a mover. Si ya no se desea mover ningún motor se puede terminar el programa.

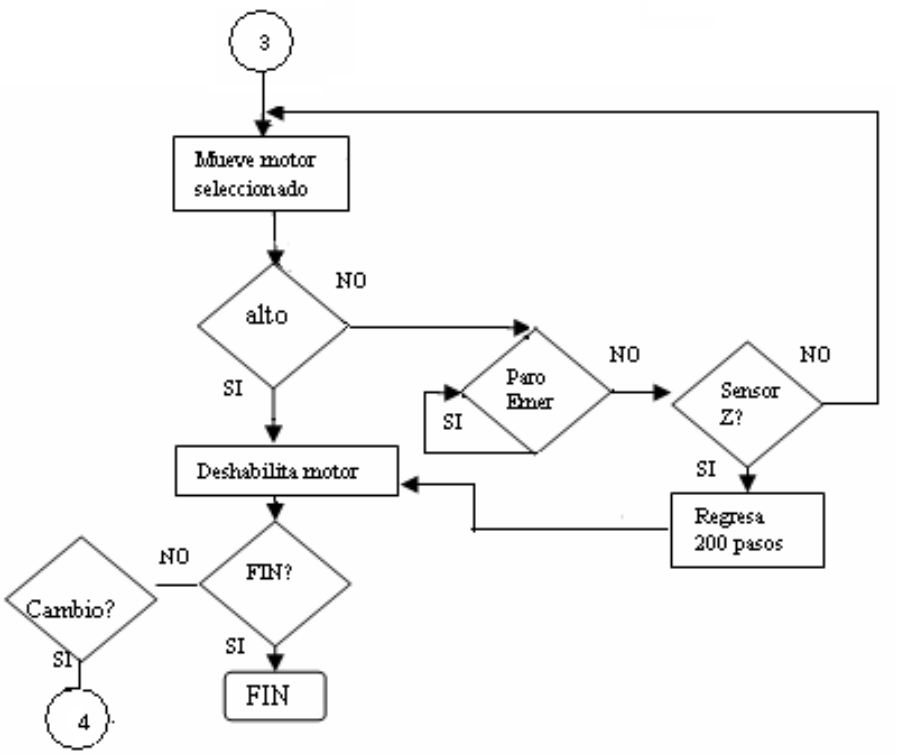


Figura 3.33 Diagrama del Modo Manual

Si en el movimiento se activa la señal de sensor del motor, se ejecuta la rutina de posición de inicio de ese motor, la cual consiste en detener al motor, cambiar de sentido de giro y regresar 200 pasos para liberar el botón. Si en algún momento del movimiento se oprime el botón de emergencia, como cuando ocurre un choque

o el eje se sigue moviendo cuando llega al final de la carrera, el programa se quedara detenido en ese lugar impidiendo el movimiento continúe, se debe reiniciar el DSP para que ejecutar de nuevo el programa.

3.7.4.3 MODO NÚMERO DE PASOS

En este modo, mostrado en la figura 3.34, se selecciona un motor y se le indica la cantidad de pasos a moverse, sólo se pueden dar valores de entre 1 y 255, ya que el tamaño de la cadena de datos a transmitir es de 8bits. Se carga el valor BFFh en la variable RELOJ y se borran las variables PASOS y DATO, la variable PASOS contiene el número de pasos a recorrer por el motor. El programa espera el comando que indica la selección del motor, después espera el comando que indica el número de pasos y después espera el comando que le indica el momento de iniciar el movimiento. Cuando alcanza el número de pasos indicado se desactiva el motor y el programa espera el comando que indica si se cambia de modo o se va a seleccionar nuevamente un motor.

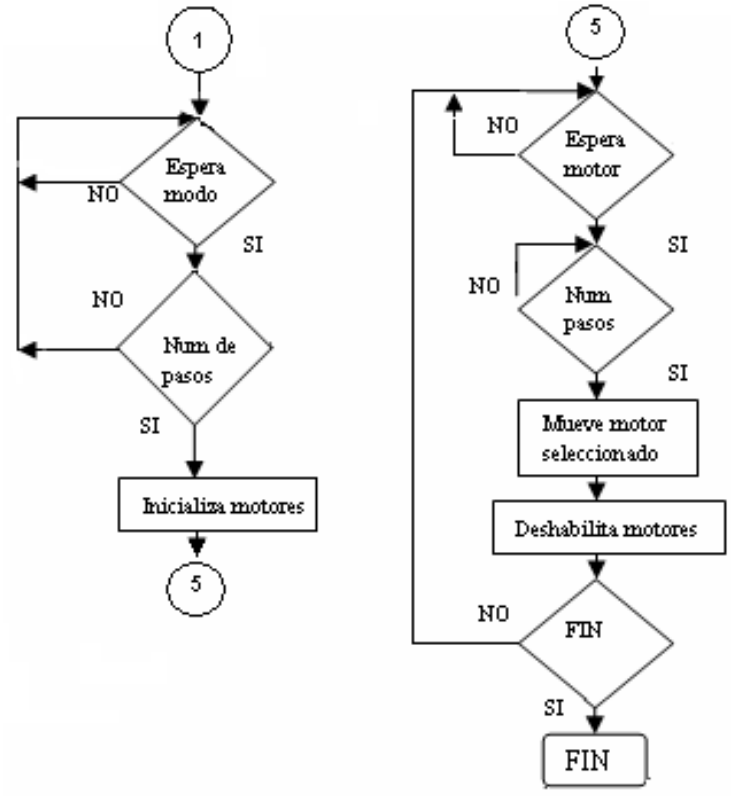


Figura 3.34 Modo Numero de Pasos

3.7.4.4 MODO FIGURA

En este modo se pueden programar una serie de movimientos horizontales, verticales o de interpolación lineal para crear diversas geometrías. Las variables que se utilizan en este modo se muestran en la siguiente tabla.

Tabla 4.6 Variables Modo Figura

Variable	Descripción
RELOJx	Contiene el valor del periodo del tren de pulsos para el motor X.
RELOJy	Contiene el valor del periodo del tren de pulsos para el motor Y.
RELOJz	Contiene el valor del periodo del tren de pulsos para el motor Z.
PASOSx	Contiene el número de pasos que dará el motor X.
PASOSx	Contiene el número de pasos que dará el motor Y.
PASOSy	Contiene el número de pasos que dará el motor Y.
PASOSz	Contiene el número de pasos que dará el motor Z.
RAZON	Contiene la relación que hay entre el número de pasos que tiene que dar el motor X y el motor Y durante un movimiento de interpolación lineal.

Las subrutinas que se utilizan en este modo se mencionan a continuación:

Tabla 4.7 Subrutinas Modo Figura

Subrutina	Descripción
Cálculos	Verifica si se va a mover solamente un motor o se van a mover los dos.
Zcero	Mueve solamente el motor X.
Xcero	Mueve solamente el motor Z.
ADELANTE	Mueve el motor X hacia delante.
ATRAS	Mueve el motor X hacia atrás.
ARRIBA	Mueve el motor Z hacia arriba.
ABAJO	Mueve el motor Z hacia abajo.

BACKLASH_X	Mueve el motor X 200 pasos para compensar el juego mecánico.
BACKLASH_Z	Mueve el motor z 200 pasos para compensar el juego mecánico.
DIVISION	Rutina que realiza una división entre el número de pasos que tiene dar el motor X y el motor Z para encontrar la relación entre los periodos de sus trenes de pulsos.
MOVER_Xch	Rutina de movimiento de interpolación lineal cuando el periodo del tren de pulsos para el motor X es mas pequeño que el del motor Z.
MOVER_Zch	Rutina de movimiento de interpolación lineal cuando el periodo del tren de pulsos para el motor Z es mas pequeño que el del motor X.

Al principio de este modo, se carga el valor BFFh en las variables RELOJx, RELOJy, y RELOJz se borran las variables PASOSx, PASOSy y PASOSz, se carga el valor FFFFh en PASOSy y se mueve el motor del eje Y hacia delante hasta que la herramienta hace contacto con el material en ese momento se activa el sensor H y manda una señal al DSP que ocasiona que se desactive el motorY.

En la figura 3.35 se muestra el diagrama de flujo correspondiente a esta parte del programa.

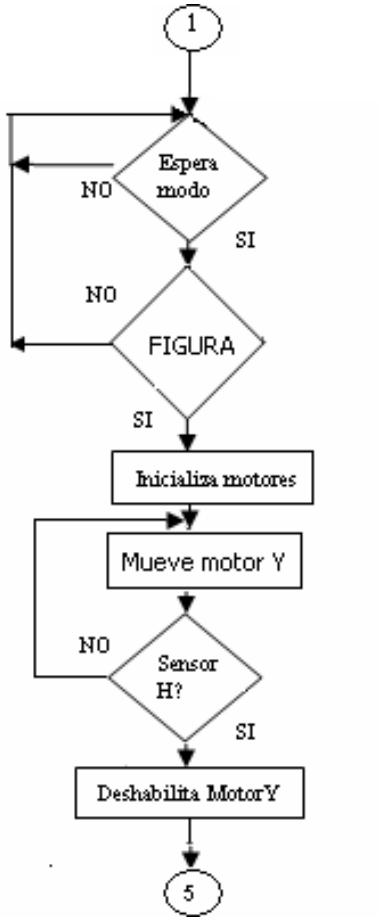


Figura 3.35 Inicialización modo Figura

En este modo se pueden programar movimientos de translación y de interpolación lineal. Para programar movimientos de traslación, esto es, donde solo se mueve un motor, se debe programar el número de pasos que debe recorrer el motor y el tipo de movimiento que debe realizar, esto es, hacia arriba, hacia abajo, hacia delante o hacia atrás.

Para programar movimientos de interpolación lineal se debe programar el número de pasos que debe recorrer cada motor y el sentido que debe recorrer cada motor; hacia delante o hacia atrás para el motor X y hacia arriba o hacia abajo para el motor Z. Al inicio de cada movimiento, el programa realizará el proceso mostrado en el diagrama de flujo de la figura 3.36. En este proceso, el programa discrimina si el movimiento que se va a realizar es de un solo motor o de dos motores.

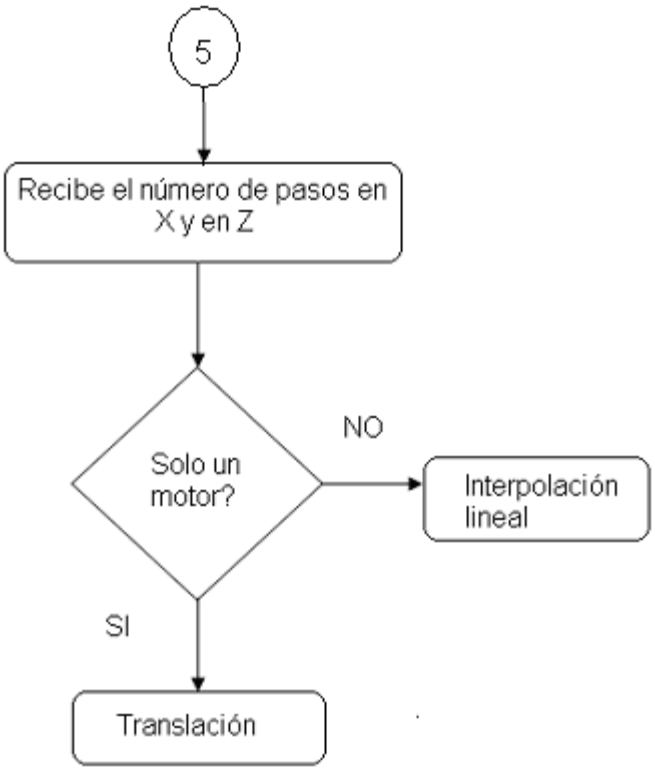


Figura 3.36 Selección de tipo de movimiento

Para explicar mejor el funcionamiento de este modo se usara el ejemplo de dibujar un triangulo que tenga 1000 pasos en la base y 500 de altura.

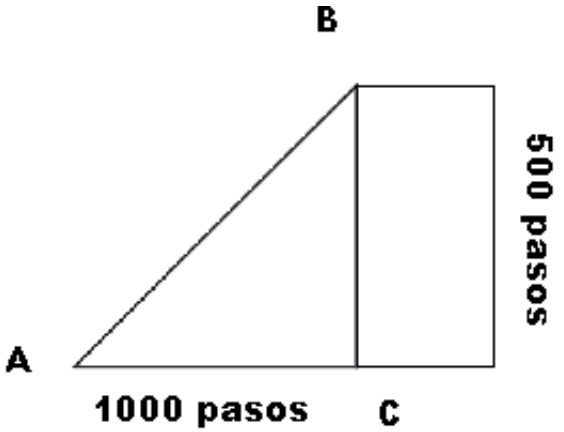


Figura 3.37 Ejemplo de Aplicación del Modo Figura

Iniciamos en el punto A, para llegar al punto B se debe realizar un movimiento de interpolación lineal en el cual, el motor X debe recorrer 1000 pasos y el motor Z

debe recorrer 500 pasos. Para realizar la trayectoria del punto B al C se debe realizar un movimiento de 500 pasos del motor Z hacia abajo. Finalmente se realiza un movimiento de 1000 pasos del motor X hacia atrás, para llegar del punto C al punto A y terminar nuestra figura.

Entonces programamos que el motor X recorra 1000 pasos hacia delante y que el motor Z recorra 500 pasos hacia arriba

Cuando ejecutamos el programa, se almacena el valor 1000 en la variable PASOSx y el valor 500 en la variable PASOSy. Posteriormente, se decrementa la variable PASOSx, si el resultado da cero el programa brinca a la parte que mueve sólo el motor Z, si no da cero se incrementa la variable para recuperar el valor original, posteriormente se realiza el mismo procedimiento para el motor Z, se decrementa la variable PASOSz y si da como resultado cero solamente se mueve el motor X. En este caso, ninguna de las dos restas dará como resultado cero, por lo que el programa ejecutara las subrutinas de interpolación lineal. Cuyo diagrama de flujo se muestra en la figura 3.38.

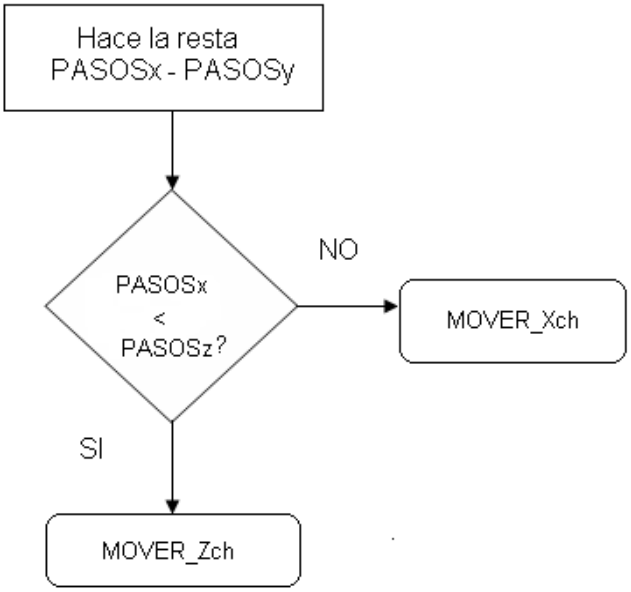


Figura 3.38 Diagrama Interpolación lineal

Se hace una resta para saber en cual eje se va a realizar la mayor cantidad de pasos y por lo tanto, el reloj de este motor tendrá el periodo más pequeño que el del otro motor. Se tiene una rutina para cuando el periodo del motorX es más pequeño que el del motor Z y una rutina para cuando se da el caso contrario, las dos rutinas son del mismo tamaño para que tengan la misma duración.

En este caso, el periodo del motor X es mas pequeño, por lo que se utiliza a rutina MOVER_Xch, en la que se realiza una división entre el número de pasos de X y el número de pasos de Z, el cociente se almacena en la variable RAZON. Si el número de pasos en X es el doble que el número de pasos en Z entonces RAZON será igual a 2, si es el triple entonces valdrá 3 y así sucesivamente. En este caso el resultado de 1000/ 500 nos dará como resultado 2, que es el valor que se almacenará en la variable RAZON.

Posteriormente, se borra el acumulador B, ambos motores se reinician y se activan, se manda el valor de RELOJx al Acumulador A y se ejecuta la rutina de TIEMPO, se empieza a decrementar el acumulador A, cuando alcanza el valor de cero la señal de control reloj para el motor X se pone a 1, se decrementa la variable PASOSx, se incrementa la variable B y se compara con el valor de RAZON, si no es igual el programa brinca a ejecutar la rutina de TIEMPO y realiza el mismo proceso sólo que ahora pone a cero la señal de control reloj del motor X, y así se va alternando para formar el tren de pulsos.

Cuando el valor de B es igual que el de RAZON, se pone a uno la señal de control reloj para el motor Z, borra el acumulador B y brinca para ejecutar la Rutina TIEMPO y ejecutar de nuevo el ciclo. En la figura 3.39 se ilustra un ejemplo con RAZON = 2.

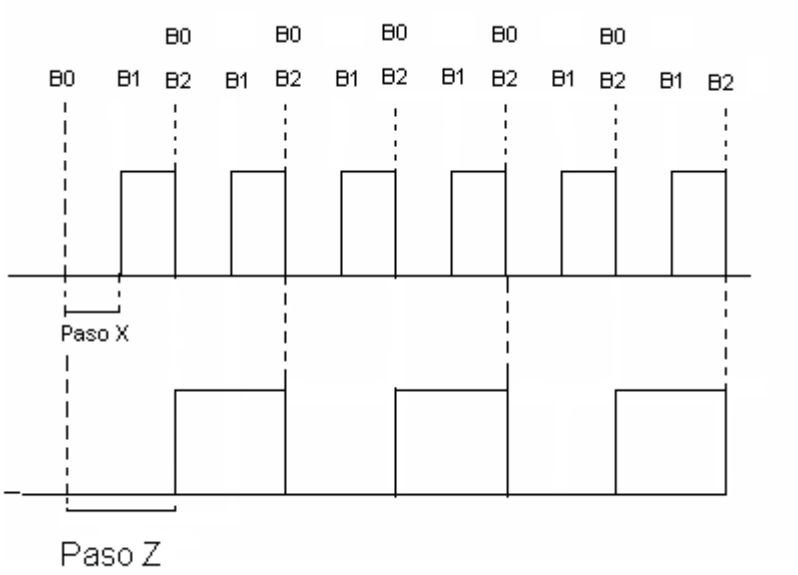


Figura 3.39 Interpolación lineal con una RAZON = 2

Cuando al decrementar la variable PASOSx se da el valor cero, esto indica que el programa ha dado el número de pasos que tenía que dar y sale de la rutina de interpolación lineal.

Para realizar la trayectoria del punto B al punto C, se debe hacer un movimiento de translación del motor Z hacia abajo. El diagrama de flujo que ilustra este tipo de movimiento se muestra en la figura 3.40

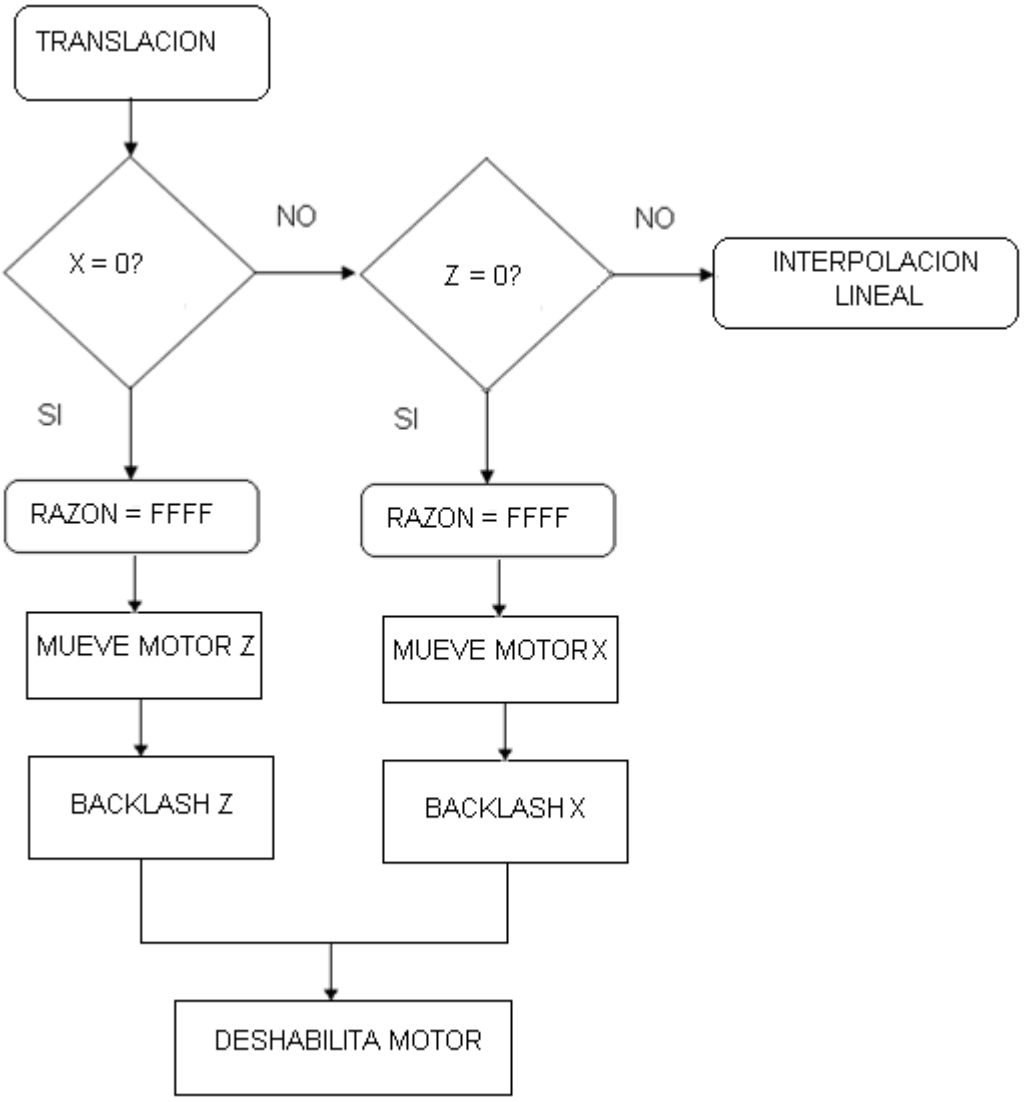


Figura 3.40 Movimiento de un solo motor

Para realizar el movimiento de traslación del motor Z hacia abajo se debe guardar el número de pasos, que en este caso son 500, en la variable PASOSz. Se debe guardar el valor 1 en la variable PASOSx, esto se hace para que el programa sepa que solo se va a mover el motor Z. El programa decrementa la variable PASOSx, como el resultado da cero, el programa brinca a la subrutina Xcero, que es la rutina que solo mueve al motor del eje Z. Se carga el valor FFFFh a la variable RAZON para que así jamás sea igual el valor del acumulador B con el valor de razón y por lo tanto no se mueva el motor X.

Posteriormente el programa ejecuta la instrucción que le indica hacia donde tiene que moverse, que en este caso es hacia abajo. En las subrutinas arriba o abajo, se manda llamar la subrutina BACKSLASH_Z, para compensar las pérdidas de pasos por el juego mecánico en el eje Z.

Con cada paso del motor se decrementa la variable PASOSz. Cuando esta variable llega a cero el motor se desactiva, entonces el programa carga el siguiente movimiento, que en este caso es el movimiento de traslación que cubre la trayectoria de C a A. En este caso se debe cargar el valor 1000 a PASOSx y el valor 1 a PASOSz. Al decrementar PASOSz el programa brincara a la rutina Zcero, que es la rutina que mueve solamente al motor X, carga el valor FFFF a la variable RAZON, para que jamás se iguale al valor de B y solo se mueva el motor X. Posteriormente, se ejecuta la instrucción que indica hacia donde se debe mover el eje X, que en este caso es hacia atrás. En las subrutinas adelante o atrás, se manda llamar la subrutina BACKLASH_X, para compensar las pérdidas de pasos por el juego mecánico en el eje X.

Con cada paso del motor se decrementa la variable PASOSz. Cuando esta variable llega a cero el motor se desactiva. Cuando ya no hay ningún movimiento programado el programa puede terminar en ese momento.

3.8 DESCRIPCION DE LA INTERFAZ DE USUARIO

Se requirió el desarrollo de una interfaz externa para que el usuario fuera capaz de tener el control del sistema. La interfaz de usuario fue hecha en Visual Basic 6.0 que es un lenguaje de programación visual, también llamado lenguaje de 4ª generación. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla [20].

Visual Basic 6.0 es también un programa *basado en objetos*, aunque no *orientado a objetos* como *C++* o *Java*. La diferencia está en que *Visual Basic 6.0* utiliza

objetos con *propiedades* y *métodos*, pero carece de los mecanismos de *herencia* y *polimorfismo* propios de los verdaderos lenguajes orientados a objetos como *Java* y *C++*.

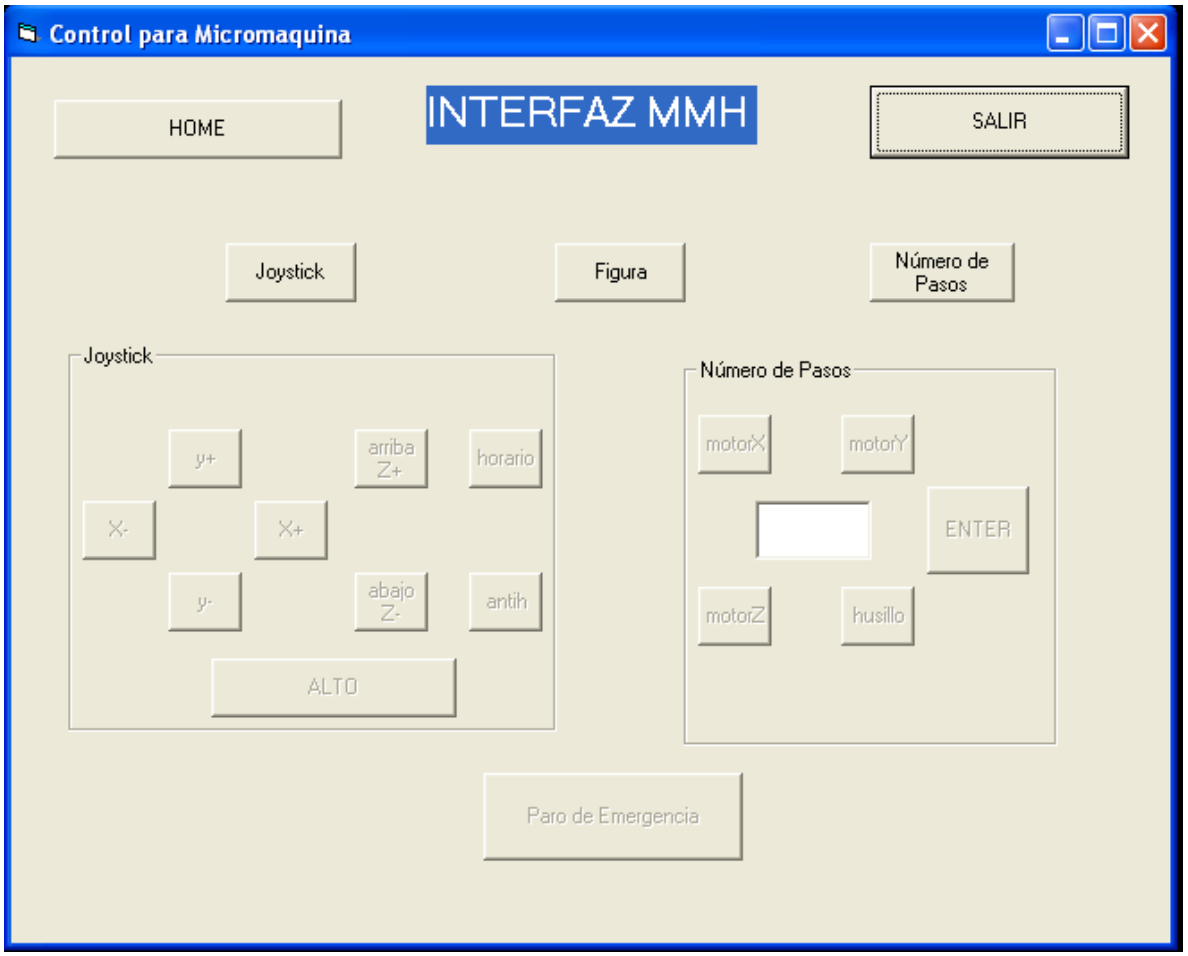


Figura 3.41 Interfaz de usuario

La interfaz desarrollada se muestra en la figura 3.41 y consta de un formulario en el cual se encuentran cuatro apartados para los diferentes modos.
Al principio sólo se encuentran activados los botones *JOYSTICK*, que corresponde al modo manual, NÚMERO DE PASOS, FIGURA, *HOME* Y SALIR.

Si se pulsa el botón JOYSTICK se manda un carácter a través del puerto serial para que sea recibido por el DSP, activan los botones para controlar el *Joystick*, mostrados en la figura 3.42, con estos botones se puede mover los ejes Y y X

hacia delante y hacia atrás. El eje Z hacia arriba y hacia abajo y el husillo en sentido horario y en sentido antihorario.

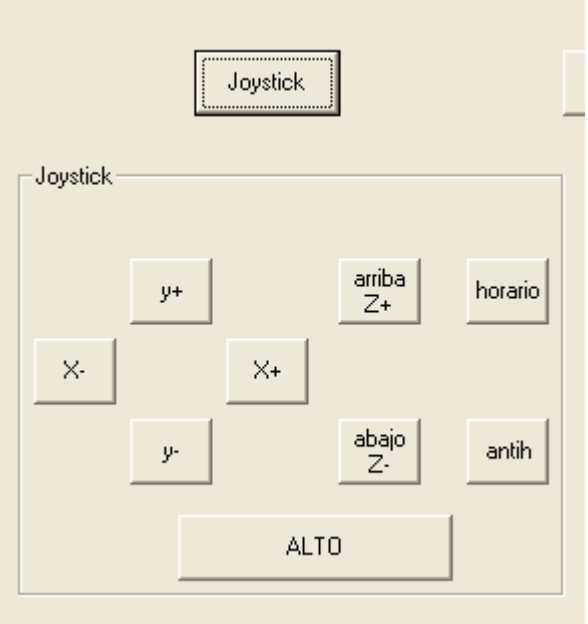


Figura 3.42 Botones del Modo Manual

Cuando se pulsa el botón del motor con el movimiento seleccionado se manda un carácter por el puerto serial para indicarle al DSP que motor y en que sentido se debe mover, los demás botones se inactivan, menos el botón ALTO que detiene el movimiento del motor y vuelve a activar el resto de los botones.
El botón paro de emergencia siempre está activo.

Después de pulsar el botón ALTO es posible cambiar de modo, por ejemplo si se pulsa el botón NUMERO DE PASOS, sólo permanecen activos los botones motorx, motory, motorz y husillo, (ver figura 3.43) con los cuales se selecciona el motor que se desea mover, una vez que se ha seleccionado el motor, se activa el cuadro de texto donde se puede ingresar el numero de pasos, se debe mencionar que sólo se puede mandar un carácter con valor entre 1 y 255, ya que el tamaño del dato que se puede transmitir es de 8 bits.

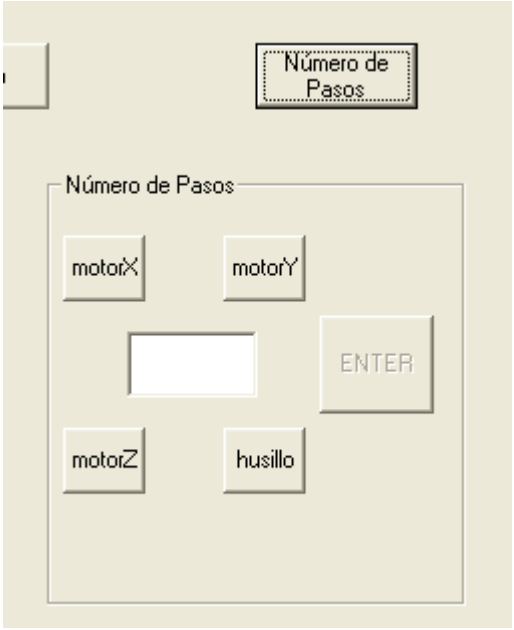


Figura 3.43 Botones del Modo Número de Pasos

Cuando se aprieta el botón ENTER se ejecuta el movimiento. Después de esto se puede volver a cambiar de modo si así se desea.

Cuando se pulsa el botón HOME la PC manda un comando al DSP que le indica que debe regresar los ejes a su posición de origen, cuando se ejecuta este comando primero se regresa el eje X, después el eje Y y regresar finalmente el eje Z.

Cuando se pulsa el botón FIGURA, el DSP realiza la figura que se programó previamente, la figura se forma con movimientos de translación y de interpolación lineal para realizar diagonales.

Finalmente el botón SALIR sirve para salir de la aplicación.

En la figura 3.44 se presenta una imagen del sistema completo.

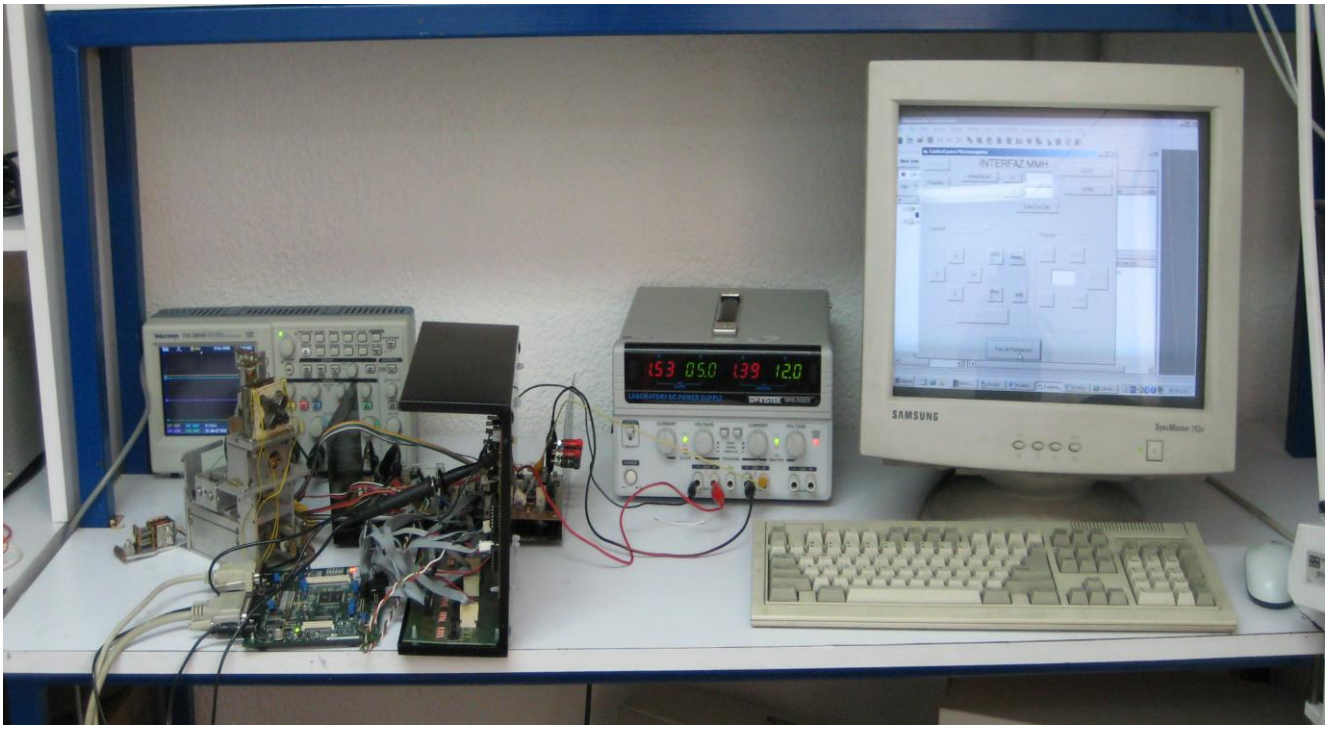


Figura 3.4 Sistema de control basado en DSP

Capítulo 4

PRUEBAS Y RESULTADOS

En esta etapa se realizaron pruebas para conocer el rendimiento del sistema desarrollado y se comparan con los objetivos planteados en 2.3.
Finalmente, se hace una prueba de incremento de temperatura y una prueba de consumo de corriente en vacío entre los sistemas de control basados en DSP, PIC, Códigos G y M y se comparan los resultados.

Las pruebas que se realizaron fueron las siguientes:

- Comprobación de las formas de onda de las señales de control en modo Manual.
- Comprobación de las formas de onda de las señales de control en modo figura.
- Consumo de corriente en vacío
- Aumento de temperatura vs. tiempo
- Corriente vs. velocidad
- Maquinados

4.1 FORMAS DE ONDA

Al probar el sistema en Modo Manual, se observaron las formas de onda obtenidas en las etapas de secuenciación y de potencia que fueron las señales de reloj, generadas por el DSP56F8357 y las señales A,B,C,D que salen del L297 y que son las responsables de polarizar directamente a las bobinas, La señal A, figura 4.1, polariza a la bobina para que la corriente fluya en un sentido y la señal B polariza a la bobina en sentido contrario, por lo tanto deben de estar defasadas 180º, como se observa en la figura 4.2, lo mismo sucede en las señales C y D que polarizan a la segunda bobina. El defasamiento que existe entre las señales A y C, al igual que entre las señales B y D es de 90º, como ya se mostró en el capítulo 3 en la figura 3.15. y como se puede observar en la figura 4.3.

En la figura 4.1 se muestra la señal de reloj suministrada al sistema, por cada 4 pulsos de reloj se produce un pulso que se traduce directamente en un paso, esto es debido a que el sistema se probó a paso completo.

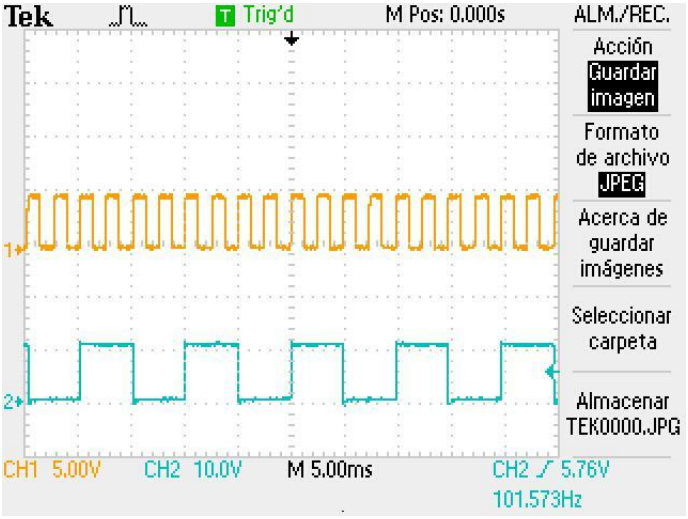


Figura 4.1 Señal de reloj con señal de A(relación 4/1)

En la figura 4.2 observamos el defasamiento de 180º que hay entre las señales A y B

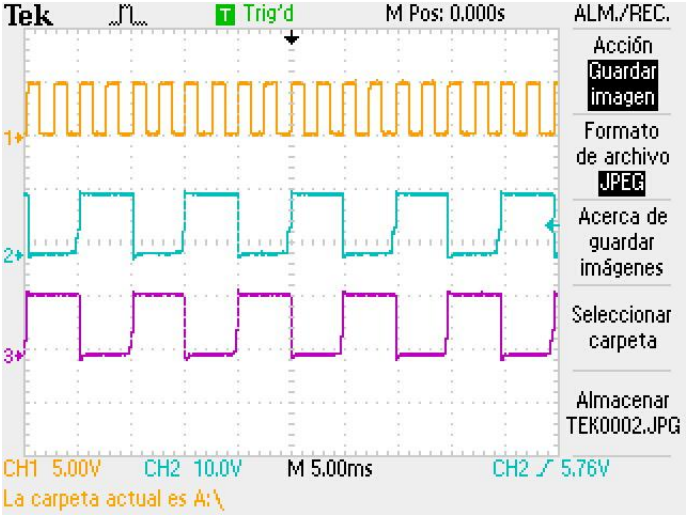


Figura 4.2 Señales RELOJ, A y B

En la figura 4.3 se muestran las 4 señales A, B, C y D que polarizan a las bobinas, las cuales corresponden correctamente a las señales mostradas en el manual del secuenciador L297 y que se muestran en la figura 4.4

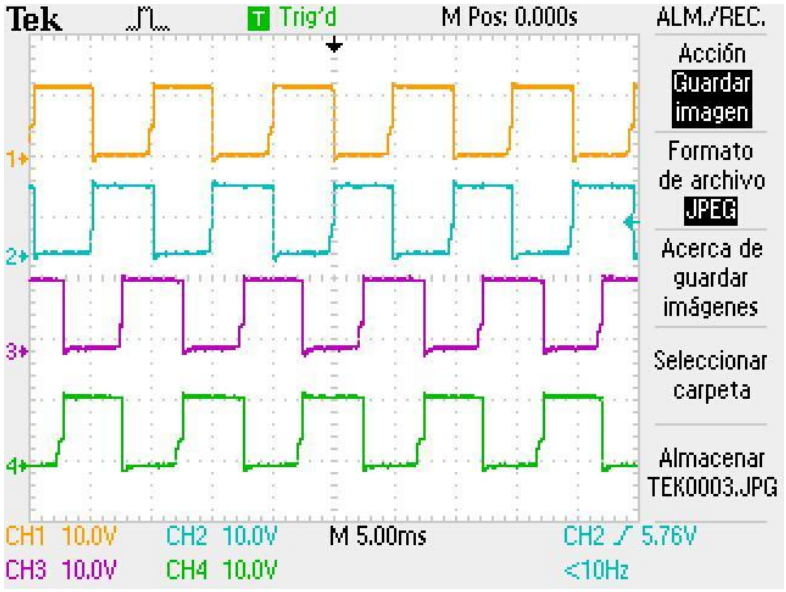


Figura 4.3 Señales A, B, C y D

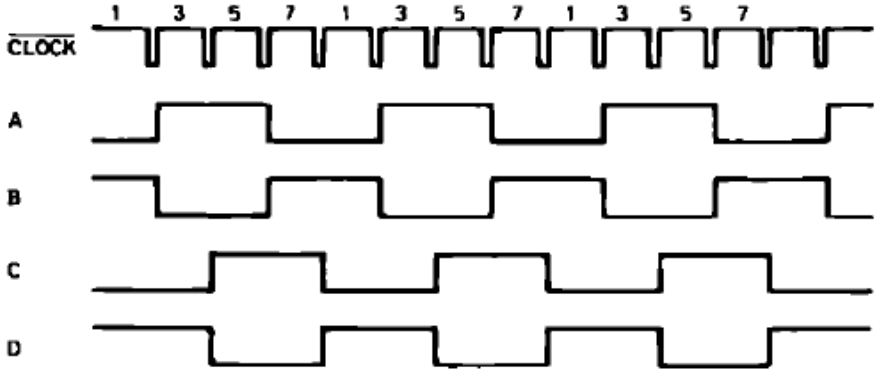


Figura 4.4 Operación a paso completo a dos fases (two phase on)

Al probar el sistema en Modo Figura se observaron las formas de onda y la magnitud de la frecuencia obtenida al realizar diagonales cuya relación entre las distancias recorridas en el eje X y el eje Z fueron de 2, 3 y 4 y se muestran en las

figuras 4.5, 4.6 y 4.7 respectivamente y cuyos resultados se presentan en la tabla 4.1.

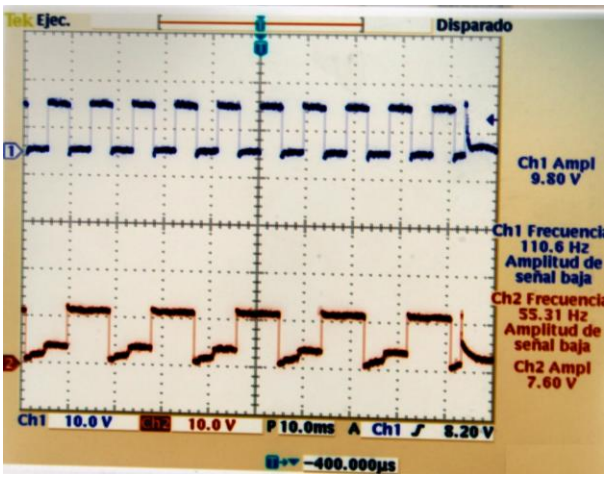


Figura 4.5 Interpolación lineal con razón 2

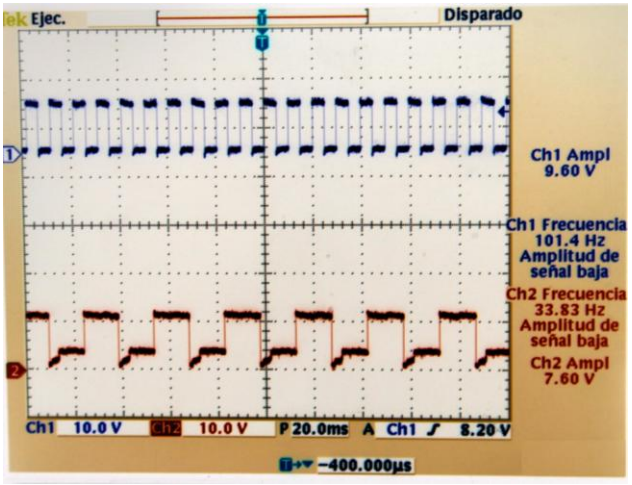


Figura 4.5 Interpolación lineal con razón igual a 3

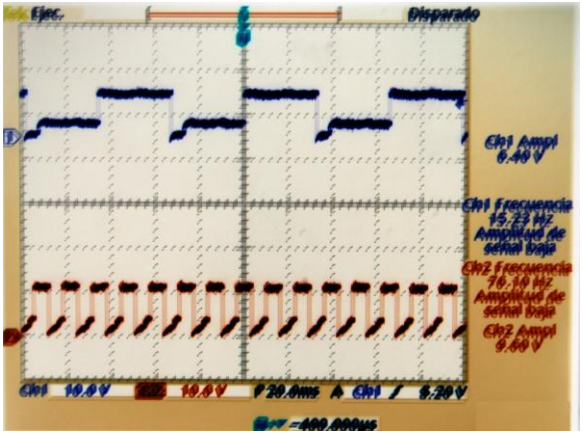


Figura 4.6 Interpolación lineal con razón 4

Tabla 4.1 Relación entre las frecuencias de los motores según sus distancias

Razón	Frecuencia Motor X [Hz]	Frecuencia Motor Z [Hz]
2	110.6	55.31
3	101.4	33.83
4	70.10	15.22

4.2 CONSUMO DE CORRIENTE EN VACÍO

Cuando no estaban conectados los motores el sistema consumía 0.50 A en la fuente de 5 V, esto se debe al consumo que realizan la electrónica de control como son: las 8 compuertas AND, las 8 compuertas NOR, los 4 controladores L297, etc.

4.3 AUMENTO DE TEMPERATURA vs. TIEMPO

Esta prueba ser realizó utilizando el motor X, se dejó trabajar el motor hasta que alcanzara una temperatura cercana a 50º C, ya que pasar esa temperatura podría dañar el motor. Las condiciones de la prueba fueron: un periodo de reloj de 3 ms (304 Hz), y una tensión de 12 V que alimentaba al puente H, cada 5 minutos se midió la temperatura utilizando un termómetro infrarrojo en la placa del motor donde se conectan las bobinas, la flecha, los TIPS de la etapa de potencia y en el L297 de la etapa de secuenciación. Los resultados se muestran en la tabla 4.2 y se ilustran en la gráfica mostrada en la figura 4.7.

Tabla 4.2 Relación entre las temperaturas de diversos elementos del sistema de control con respecto al tiempo

Tiempo	bobinas	flecha	TIPS	L297	corriente
[min]	[º C]]	[º C]	[º C]	[º C]	A
0	25.9	25.9	24.3	24.5	0.3
5	29,6	32,6	25.3	24.9	0.3
10	33,8	36,8	25.3	25.7	0.3
15	3,29	6,29	26.1	25.0	0.3
20	35,3	38,3	26.1	26.0	0,3
25	37,4	40,4	25.7	27.0	0.3
30	37,3	40,3	25.6	26.	0.3
35	37,3	40,3	26.8	28	0.3
40	37,2	40,2	26.4	27.5	0.3
45	38,6	41,6	27.0	28	0.3
50	38,4	43,4	27.5	27.1	0.3
55	38,4	43,4	27.0	27.3	0.3
60	38,7	43,7	26.7	26.5	0.3
65	39,9	44,9	26.9	26.3	0.3
70	39,7	44,7	27.3	27.5	0.3
75	39,4	44,4	27.0	27.1	0.3
80	39,4	45,4	27.0	27.0	0.3
85	39,5	45,5	27.0	27.0	0.3
90	39,6	45,6	27.0	27.0	0.3

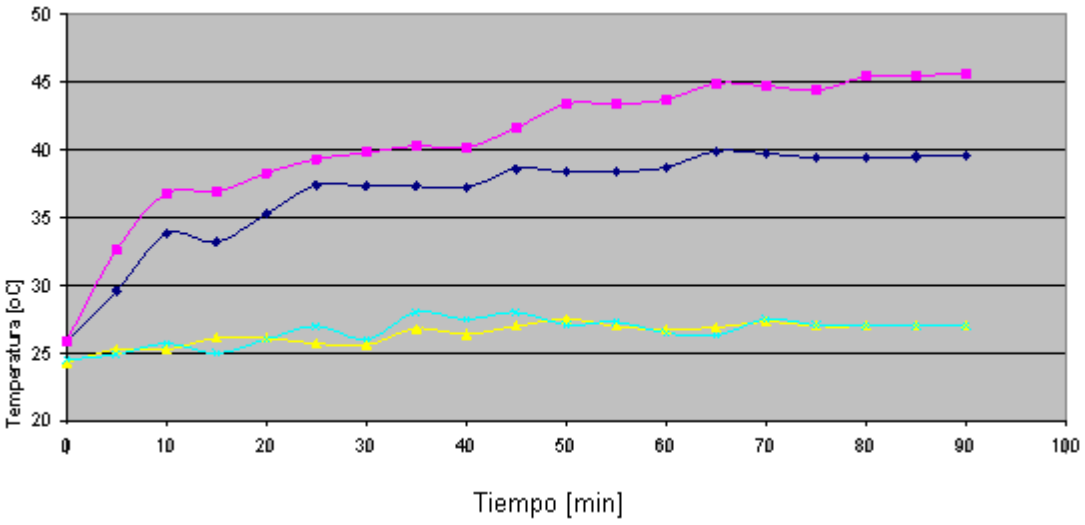


Figura 4.7 Variación de la temperatura del motor con respecto al tiempo temperatura en la flecha, temperatura en las bobinas, temperatura en los TIPS, temperatura L297

4.4 VELOCIDAD – CONSUMO DE CORRIENTE

Para realizar esta prueba se utilizó el motor del eje X y se realizó un aumento gradual de la frecuencia de reloj y el motor giró sin perder pasos hasta 524 Hz.

Los resultados se muestran en la tabla 4.3 , la figura 4.8 ilustra el incremento de velocidad y la figura 4.9 muestra la relación entre la velocidad y consumo de corriente.

Tabla 4.3 Relación entre velocidad y consumo de corriente

RELOJ HZ	PASO HZ	corriente A
40,4	10,1	0,95
42,84	10,71	0,95
50,84	12,71	0,85
61	15,25	0,83
76,28	19,07	0,81
87,2	21,8	0,8
101,68	25,42	0,77
122,04	30,51	0,78
135,6	33,9	0,76
152,4	38,1	0,73
157,36	39,34	0,72
181,28	45,32	0,69

203,24	50,81	0,67
221,72	55,43	0,64
243,92	60,98	0,61
271	67,75	0,59
286,96	71,74	0,57
304,88	76,22	0,54
325,2	81,3	0,52
348,4	87,1	0,49
375,24	93,81	0,47
406	101,5	0,43
423,6	105,9	0,41
434	108,5	0,41
437,6	109,4	0,41
465,6	116,4	0,41
472,8	118,2	0,4
488,4	122,1	0,39
512,8	128,2	0,39
524,8	131,20	0,39

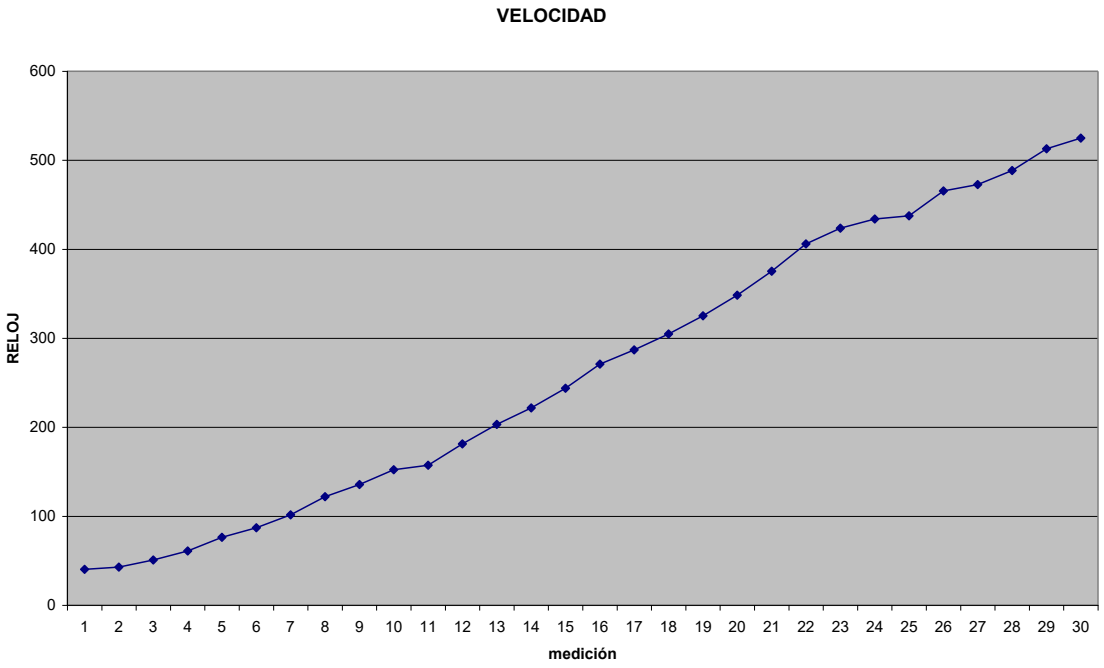


Figura 4.8 incremento de la frecuencia de reloj sin perder pasos

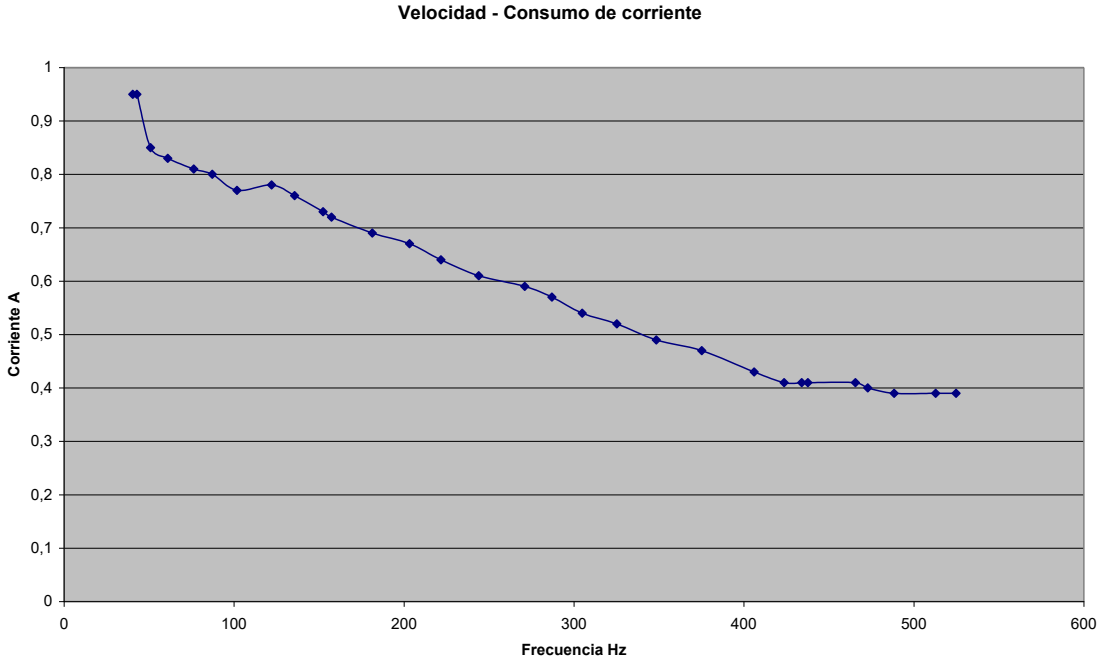
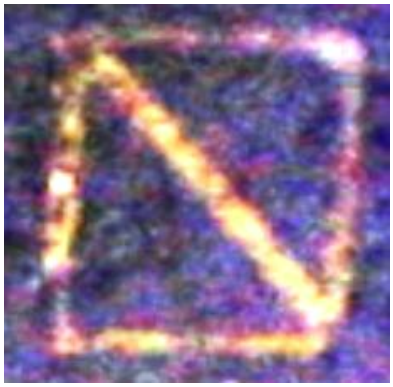


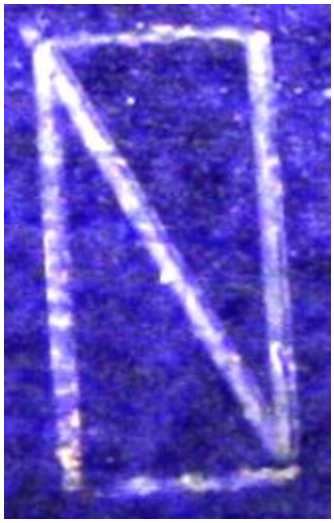
Figura 4.9 Relación entre la frecuencia y el consumo de corriente

4.5 MÁQUINADOS

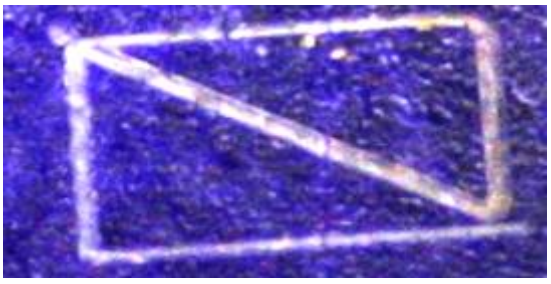
Se realizaron las siguientes geometrías, mostradas en la figura 4.10, que corresponden a un Cuadrado con una diagonal, un rectángulo vertical y uno horizontal ambos con una diagonal y un triángulo. En ellos se puede apreciar el juego mecánico, y ya no se pudo ajustar mejor esta variable debido a que la maquina ya presentaba deterioros, en algunos segmentos del recorrido de los ejes se atoraba y se realizaron pruebas hasta que uno de los ejes se atoró. Actualmente se esta desarrollando una nueva máquina para superar estos inconvenientes.



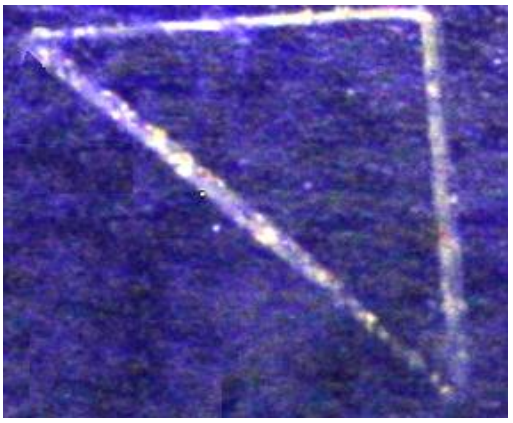
(a)



(b)



(c)



(d)

Figura 4.10 Maquinados hechos con el sistema basado en DSP a) cuadrado con diagonal b) rectángulo c) rectángulo horizontal con diagonal d) vertical con diagonal .

4.6 PRUEBA DE INCREMENTO DE TEMPERATURA

Se realizo una prueba de incremento de temperatura con respecto al tiempo con los sistemas basados en PIC, que utilizan una etapa de potencia basada en el arreglo L297-L298 y con el sistema basado en Códigos G y códigos M, cuya etapa de potencia era la misma que la del sistema basado en PC. Las condiciones de la prueba fueron: se dejo trabajar el motor hasta que alcanzara una temperatura cercana a 50º C, ya que pasar esa temperatura podría dañar el motor, se programaría un periodo de reloj de 3 ms (304 Hz), y una tensión de 12 V que alimentaba al puente H, cada 5 minutos se midió la temperatura utilizando un termómetro infrarrojo en la placa del motor donde se conectan las bobinas, la flecha, Como el sistema basado en PIC presentaba un calentamiento considerable se alimento el L298 a 10V. Los resultados se muestran en la tabla 4.4 y se ilustran en la gráfica de la figura 4.11

Tabla 4.4 Comparación de las temperaturas del motor con respecto del tiempo de las etapas de potencia de los distintos sistemas de control.

Tiempo	DSP	PIC	Códigos G y M
[min]	[º C]	[º C]	[º C]
0	25,9	24,5	25,4
5	29,6	28,1	27,4
10	33,8	43,3	33,4
15	33,2		39,4
20	35,3		
25	37,4		
30	37,3		
35	37,3		
40	37,2		
45	38,6		
50	38,4		
55	38,4		
60	38,7		
65	39,9		
70	39,7		
75	39,4		
80	39,4		
85	39,5		
90	39,6		

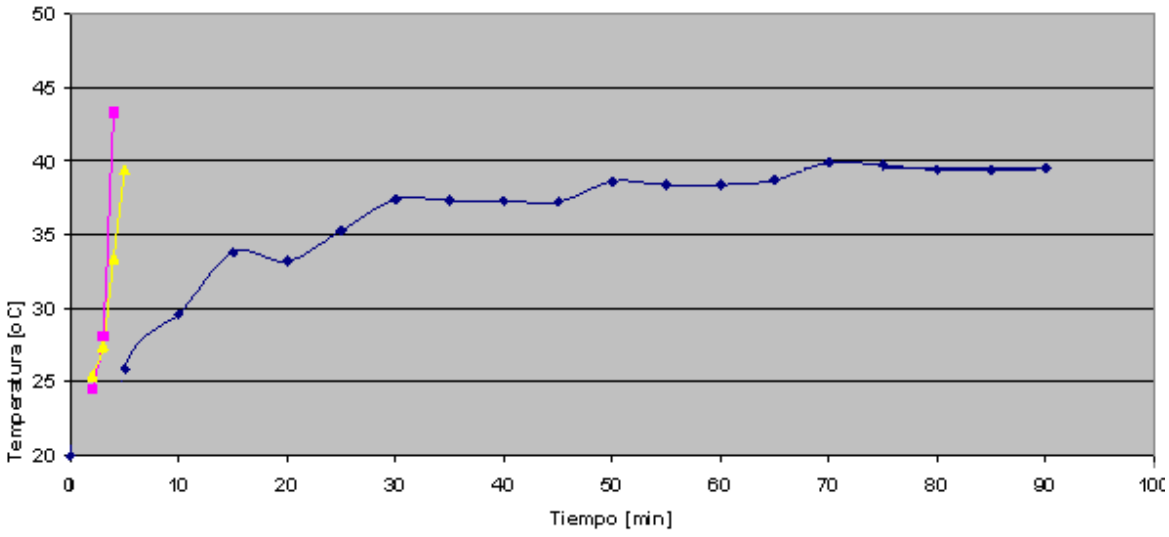


Figura 4.11 Comparación de la variación de la temperatura con respecto al tiempo de los sistemas basados en DSP, PIC y Códigos G y M

4.7 CONSUMO DE CORRIENTE EN VACÍO

En la tabla 4.5 se muestra la comparación entre el consumo de corriente al realizar esta prueba y el consumo de corriente en vacío.

Tabla 4.5 Comparación de consumo de corriente

Consumo	DSP	PIC	Códigos G y M
en vacío	0,5 A	0,4 A	0,24 A
Prueba T vs. t	0,8 A	0,84 A	0,67 A

CONCLUSIONES

Al revisar las metas propuestas podemos concluir que se cumple satisfactoriamente la implementación de una nueva alternativa para controlar la micromáquina herramienta basada en un DSP, el cual sólo necesita de la PC para desplegar la interfaz de usuario para mandar los comandos de control. Si se quiere ampliar el sistema ya sea añadiendo más motores a controlar o mas micromáquinas se pude realizar, ya que sobran suficientes puertos GPIO y 12 módulos PWM con lo que incluso se podrían desarrollar nuevas etapas de potencia basadas en tecnología MOSFET.

Actualmente, se encuentran ya implantadas varias alternativas para el control de motores, se seguirán desarrollando sistemas alternativos hasta que se encuentre el sistema más adecuado para el control de las micromáquinas herramienta.

El sistema implementado respondió satisfactoriamente al diseño propuesto con lo que se puede operar correctamente la micromáquina en sus movimientos de translación y de interpolación lineal.

Al comparar el sistema basado en un microcontrolador PIC con el sistema basado en un microcontrolador basado en DSP, los resultados muestran que la velocidad máxima obtenida en lazo abierto para el primer sistema fue de 397 Hz [12] y en el sistema basado en DSP fue de 524 Hz. Esto nos muestra un mejoramiento del 13%. Además de que se cumplió con el objetivo de desarrollar una nueva etapa de potencia más eficiente

Una de las desventajas que presenta el sistema basado en DSP fue que, al realizar la etapa de potencia utilizando elementos discretos se produce un incremento en el consumo de corriente que llega a se de 0.5 A solamente para la etapa de control además de que el tamaño del sistema crece considerablemente comparado con la etapa de potencia utilizando el circuito integrado L298, pero el calentamiento se que se producía fue exitosamente reducido al expandirlo a elementos discretos.

TRABAJO A FUTURO

Todos los sistemas son perfectibles, en cuanto al trabajo realizado se pueden integrar todas las etapas que ahora se realizaron por separado (aislamiento, secuenciación y potencia) en una sola para reducir el tamaño. Una vez que se encuentre en funcionamiento el nuevo prototipo que remplazará al prototipo en el que realizamos las pruebas se podrá ajustar la rutina de compensación de juego mecánico. Además, se pretende que utilizando un lenguaje de alto nivel como C, se pueda mejorar la rutina de interpolación lineal, ya que las operaciones utilizando punto flotante son más fáciles de implementar en un lenguaje de alto nivel que en lenguaje ensamblador.

En el futuro, también se puede ampliar el sistema para que realice control en lazo cerrado y así pueda controlar el segundo prototipo de primera generación que se esta desarrollando en el GMM. De las seis señales de control que se utilizaron se pueden reducir a cuatro ya que las señales CONTROL Y PASO permanecieron sin cambio con lo que se les puede asignar un valor constante y por lo tanto liberar más puertos GPIO.

Este sistema fue desarrollado para realizar tareas de control, no de procesamiento, pero el DSP56F8357 tiene la capacidad de realizar ambas funciones y ya que la parte de control se encuentra resuelta, ya se puede en un futuro implementar algoritmos de control, de reconocimiento de imágenes, etc.

Todavía en sistema se utilizó el puerto serial para la transmisión de datos, pero se esta buscando migrar al la transmisión de datos a través del puerto USB (Universal Serial Bus).

REFERENCIAS

1. E. Kussul, T. Baidyk, L. Ruiz Huerta, A Caballero Ruiz, G Velasco and L. Kasatkina. *DEVELOPMENT OF MMT FOR MICROFACTORIES*. 3 de octubre de 2002

2. L. Ruiz-Huerta, A. Caballero-Ruiz, G. Ascanio, T. Baidyk and E. Kussul. IN THE WAY TO THE SECOND GENERATION OF MICROEQUIPMENT. 9-10 octubre de 2004.

3. Munehisa Takeda, Group Manager, Mechanical System Department Advanced Technology. MULTIPLE DISTRIBUTED MICROMACHINE SYSTEM

4. Caballero R. A. *CARACTERIZACIÓN DE UN MICROCENTRO DE MAQUINADO DE BAJO COSTO*. Tesis de maestría, Área mecánica Septiembre 2000

5. Ramírez Leyva, Hugo. APLICACIONES Y TENDENCIAS DE LOS SISTEMAS MICROELECTROMECÁNICOS (MEMS) Instituto de Electrónica y Computación. Universidad Tecnológica Mixteca.

6. *MICROFACTORY*. *Desktop Machining Microfactory*, 2nd International Workshop on Microfactories, Fribourg, Switzerland, October 9-10,2000 Edited by FSRM-Swiss Foundation for research in Microtechnology, p13-16.

7. MICRO-LATHE EQUPPED WITH CLOSED-LOOP NUMERICAL CONTROL Yuichi Okazaki and Tokio Kitahara, Mechanical Engineering Laboratory / NIAIST/JAPAN International Workshop on Microfactories, Fribourg, Switzerland, October 9-10,2000 Edited by FSRM-Swiss Foundation for research in Microtechnology p87-90

8. Resendiz Sánchez, José. INVESTIGACIÓN DEL SISTEMA DE CONTROL DE UNA MICROMAQUINA HERRAMIENTA Y PROPUESTAS PARA LA OPTIMIZACIÓN DEL SISTEMA. Tesis de Licenciatura Diciembre 2005.

9. Kuo, *SISTEMAS DE CONTROL AUTOMÁTICO*, 7ª Ed. Prentice

10. Gieras, Jacek F; Wing, Mitchell. *PERMANENT MAGNET MOTOR TECHNOLOGY* Design and Applications. 1997

11. *Santos Carrasco Cesar,* EMPLEO DE FPGA's PARA EL SISTEMA DE CONTROL DE MOTORES DE PASOS BIMODALES EN MICROMECÁNICA *Tesis de licenciatura 2003*

12. Silva López, Héctor Hugo AUTOMATIZACIÓN DE UNA MICROMAQUINA HERRAMIENTA DE PRIMERA GENERACION. Tesis de licenciatura 2005

13. Yánez Soto Alfredo DESARROLLO DE UNA INTERFAZ COMPUTACIONAL PARA EL CONTROL DE MICROMAQUINAS-HERRAMIENTA. Tesis de Licenciatura Marzo 2007

14. L298 DUAL FULL BRIDGE Datasheet

15. THE L297 STEPPER MOTOR CONTROLLER, Application Note

16. Floyd, Thomas L. *FUNDAMENTOS DE SISTEMAS DIGITALES* 7ª edición

17. Escobar S, Larry. *ARQUITECTURAS DE DSP'S FAMILIA TMS320 Y EL TMS320C50*

18. Datasheet DSP56F8357 Rev. 10.0 09/2005

19. DSP56F800E 16bit DIGITAL SIGNAL PROCESSOR CORE Reference Manual Rev. 2.0, 12/2001

20. APRENDA VISUAL BASIC 6.0 Como si estuviera en primero. Escuela Superior de Ingenieros Industriales de San Sebastián. Universidad de Navarra. Agosto 1999.

21. Boylestad, Robert; Nashelsky, Louis. ELECTRÓNICA: TEORÍA DE CIRCUITOS. Pearson- Prentice Hall.6ª Edición.