



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

"AUTOMATIZACIÓN DE UNA MÁQUINA
HERRAMIENTA CONVENCIONAL A MÁQUINA DE
CONTROL NUMÉRICO MEDIANTE
MICROCONTROLADORES"

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO MECÁNICO
P R E S E N T A N:
MARTÍN ORTEGA BREÑA
SHAIR MENDOZA FLORES

DIRECTOR DE TESIS: ING. ULISES M. PEÑUELAS RIVAS



México, D.F.

OCTUBRE DE 2004



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**ESTA TESIS NO SALE
DE LA BIBLIOTECA**

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: Martín Ortega

Breña

FECHA: 26 oct 2004

FIRMA: [Firma]

Agradecimientos

A Dios:

Por haberme dejado concluir mis estudios, a pesar de no ser lo que el desea de mi, siempre esta conmigo y no me olvida.

A Mamá:

Porque sin ella y sin su esfuerzo de todos estos años no lo hubiera logrado, gracias por tu esfuerzo mamá y por haber dedicado gran parte de tu vida a ayudarme y siempre apoyarme en las buenas y en las malas.

A Papá:

Porque a pesar de tus problemas personales, me apoyaste económicamente y con lo mucho o poco de tu apoyo logre concluir, gracias papá.

A Yidel:

Gracias hermano por haber compartido conmigo estos años de estudio y espero que sea un ejemplo a tu vida, que a pesar de todas las dificultades que hemos vivido, podemos soñar en que lograremos cumplir nuestros objetivos.

A Claudia:

A ti Claudia, porque estuviste conmigo desde el inicio de nuestra carrera y siempre me apoyaste. Aun teniendo problemas entre los dos, siempre estuviste presente, gracias mimocha.

A mis amigos:

Ulises Peñuelas: por haberme siempre dado confianza en que podíamos resolver los problemas de esta tesis.

Germán López: Por haber compartido conmigo tus conocimientos y siempre sonreír a los problemas y mostrarme que la amistad va mas allá de esta vida. Descansa en paz.

Martín: Por apoyarme en este proyecto de tesis y saber que si es posible trabajar en equipo.

Rafael, Sergio, Gabriel, Eduardo, Fernando, Rodrigo, Luis, Orlando, Reyna, Samuel y Juan Carlos por haber compartido un momento en esta odisea de la universidad y ser mis amigos.

Automatización de una máquina – herramienta convencional a máquina de control numérico mediante microcontroladores

Introducción
Antecedentes

Capítulo 1. Definición de CNC y su entorno

- 1.1 Tipos de automatización
- 1.2 Control numérico computarizado
 - 1.2.1 Definición de puntos de referencia de la máquina
- 1.3 Características del CNC

Capítulo 2. Diseño conceptual de la transmisión

- 2.1 Selección de componentes
 - 2.1.1 Engranajes
 - 2.1.2 Sistemas de transmisión flexible
 - 2.1.3 Tornillo sinfín – corona
 - 2.1.4 Tornillo - tuerca
- 2.2 Diseño de detalle de la transmisión

Capítulo 3. Control electrónico

- 3.1 Sistema motriz
 - 3.1.1 Opciones disponibles
 - 3.1.2 Selección del sistema motriz
- 3.2 Control del sistema motriz
 - 3.2.1 Microcontroladores PIC16F7X y PIC16F87X
 - 3.2.2 Expansión de memoria externa
 - 3.2.3 Diseño de tarjeta de control de motores
- 3.3 Sensores
 - 3.3.1 Sensores ópticos
 - 3.3.2 Sensores mecánicos
 - 3.3.3 Selección de sensores

Capítulo 4. Interfase PC – Microcontroladores.

- 4.1 Interprete de código Protel – CNC
- 4.2 Desarrollo de software
 - 4.2.1 Interfase con el usuario
 - 4.2.2 Microcontroladores
- 4.3 Protocolos de Comunicación
 - 4.3.1 USART
 - 4.3.2 I²C
 - 4.3.3 Paralelo

Conclusiones

Anexo A. Código fuente de programa TORITO

Anexo B. Programas de ensamblador para Microcontroladores y diagramas de flujo

Anexo C. Diagramas electrónicos

Bibliografía

Objetivo

El siguiente proyecto de tesis tiene como objetivo la automatización de un taladro convencional para perforar tarjetas de circuitos impresos generados a través del software Protel.

Protel es un software para diseñar circuitos electrónicos y es capaz de generar el código de control numérico de las perforaciones en la tarjeta diseñada.

En este proyecto nos dimos a la tarea de la reparación y mejoramiento de una máquina ya existente. Con esto se busca obtener una máquina mas práctica con mayores características de resolución y de movimiento.

Justificación del tema

En el laboratorio de diseño mecatrónico es común desarrollar tarjetas electrónicas por lo cual se necesita un dispositivo capaz de realizar la perforación de tarjetas fenólicas, con una resolución adecuada que permita tener perforaciones muy cercanas una de otra (0.1 mm).

Un aspecto importante es aumentar la capacidad de almacenamiento de información, por lo cual es necesario dejar expandible los dispositivos de memoria.

Tambien es necesario contar con elementos de medición que nos garanticen la posición exacta de las bancadas, lo cual permitirá tener certeza que llegan a la posición deseada.

Introducción

En los siguientes capítulos se presenta el detalle de cada dispositivo, tanto electrónico como mecánico, empleado en el desarrollo de este proyecto.

En el capítulo I se da la descripción del control numérico por computadora, el entorno de trabajo, así como el uso de la automatización de la industria, en procesos rutinarios. Se mencionan las ventajas y desventajas existentes del uso del CNC contra las máquinas convencionales.

Dentro del capítulo II, se describen algunos elementos mecánicos para transmitir movimiento y potencia, engranes, poleas lisas, dentadas y bandas. Se mencionan sus ventajas y desventajas, tomando la mejor alternativa de transmitir movimiento acorde a las necesidades que se requieren, utilizando los parámetros adecuados. Se busca tomar en cuenta aquellas que nos favorezcan para evitar el backlash, facilitar la construcción y disminuir el costo.

En el capítulo III se explican los actuadores eléctricos. Tal es el caso de motores eléctricos de corriente alterna, de corriente directa, servomotores y motores a pasos. Por otro lado se incluye el diseño de las tarjetas controladoras de los motores seleccionados, así como el diseño de las tarjetas de los microcontroladores (μ Cs) empleados y la selección de memorias externas como almacenamiento de información. El uso de encoders y sensores es parte de este capítulo. Se toma en cuenta el uso mas adecuado de estos dispositivos para la máquina de control numérico desarrollada.

En el capítulo IV se describe la parte de desarrollo de software de interfase PC - usuario, microcontrolador, la codificación del software que genera Protel para la creación del código CNC y la decodificación de este mismo para el entendimiento a nivel microcontrolador de este código.

En las conclusiones, se mencionan los resultados obtenidos del desarrollo de la máquina perforada de circuitos impresos, dando sus ventajas y desventajas contra máquinas desarrolladas anteriormente dentro de esta facultad.

En los anexos mostramos los diagramas de flujo, códigos fuentes de los programas creados para la interfase PC -Usuario, PC - μ Cs, los diagramas electrónicos de las tarjetas diseñadas para la manipulación de la máquina.

Antecedentes

Como antecedente de este proyecto se tenía este taladro automatizado, sin embargo por falta de mantenimiento se deterioró, a tal grado que dejó de funcionar por completo. El anterior proyecto tiene las siguientes características:

Parte mecánica: Mesas de trabajo, mesa longitudinal, mesa transversal, transmisión de movimiento rotacional a movimientos lineales a través de un tornillo sin fin de cuerda fina, sistema de reducción de par por medio de engranes en sus actuadores, así como en su husillo.

Parte electrónica: Motores a pasos en sus dos direcciones lineales (transversal y longitudinal) con una resolución de 1.8° por paso, el motor del husillo de AC y con variador de velocidad manual, un puente H de transistores, así como con una fuente de alimentación de 127[AC] , ± 8.5 , ± 5 [V], un microcontrolador HC11 y con una comunicación entre dispositivos electrónicos vía serial y como sistema de almacenamiento una memoria eeprom y una memoria RAM de 64 kbytes cada una.

La integración de interfase PC – Máquina CNC contaba con un programa que permitía cargar programas de trabajo (ciclos de trabajo) en código CNC.

Sin embargo este proyecto tenía las siguientes desventajas:

Para posicionar las bancadas y realizar la perforación, no se tenía certeza de que llegaba a la posición indicada, dado que el movimiento era a través de un posicionamiento de conteo de pasos del motor y no se contaba con un dispositivo que pudiera cuantificar la posición actual de las mesas, tanto transversal como longitudinal.

Otra gran desventaja de este proyecto fue que solo tiene una capacidad de almacenamiento de 64 kbytes, y sin posibilidad de expansión de memoria.

El tamaño de palabra para este proyecto es de 8 bits, lo que significa que tiene un rango de movimiento demasiado pequeño.

Capítulo I

Definición de Control Numérico por Computadora (CNC)

Puede definirse como un dispositivo capaz de controlar el movimiento exacto de uno o varios órganos de la máquina herramienta de forma automática a partir de una serie de datos numéricos programados que hacen funcionar los controladores y motores eléctricos de la máquina herramienta.¹

1.1 Tipos de Automatización

Existen cinco formas de automatizar en la industria moderna, de modo que se deberá analizar cada situación a fin de decidir correctamente el esquema más adecuado.

Los tipos de automatización son:

- Control automático de procesos
- Procesamiento electrónico de datos
- La automatización fija
- El control numérico por computadora
- La automatización flexible.

El Control Automático de procesos. El Control Automático de procesos, se refiere usualmente al manejo de procesos caracterizados por diversos tipos de cambios (generalmente químicos y físicos); un ejemplo de esto lo podría ser el proceso de refinación de petróleo.

El Proceso Electrónico de datos. Este proceso electrónico de datos frecuentemente es relacionado con los sistemas de información, centros de cómputo etc. Sin embargo en la actualidad también se considera dentro de esto la obtención, análisis y registros de datos a través de interfaces y computadoras.

La Automatización fija. La Automatización fija, es aquella asociada al empleo de sistemas lógicos tales como: los sistemas de relevadores y compuertas lógicas; sin embargo estos sistemas se han ido flexibilizando al introducir algunos elementos de programación como en el caso de los Controladores Lógicos Programables (PLC'S).

Control Numérico por computadora. Un mayor nivel de flexibilidad lo poseen las máquinas de control numérico por computadora. Este tipo de control se ha aplicado con éxito a máquinas herramienta. Entre ellas podemos mencionar:

- Fresadoras CNC.
- Tornos CNC.
- Máquinas de electroerosionado
- Máquinas de corte por hilo.

Automatización flexible. El mayor grado de flexibilidad en cuanto a automatización se refiere es el uso de los robots industriales que en conjunto con máquinas CNC se les denomina como "Celdas de Manufactura Flexible".

También se emplean sistemas CAD / CAM que generan el programa de maquinado de forma automática. En el sistema CAD (diseño asistido por computadora) la pieza que se desea maquinar se diseña en la computadora con herramientas de dibujo y modelado sólido.

Posteriormente, el sistema CAM (manufactura asistida por computadora) toma la información del diseño y genera la ruta de corte que tiene que seguir la herramienta para fabricar la pieza deseada; a partir de esta ruta de corte se crea automáticamente el programa de maquinado, el cual puede ser introducido a la máquina mediante un disco de información o enviado electrónicamente.

Hoy día los equipos CNC con la ayuda de los lenguajes conversacionales y los sistemas CAD / CAM, permiten a las empresas producir con mucha mayor rapidez y calidad sin necesidad de tener personal altamente especializado.^{1,2}

1.2 Control Numérico por Computadora

CNC se refiere al control numérico de máquinas, generalmente máquinas herramientas. Normalmente este tipo de control se ejerce a través de una computadora y la máquina está diseñada a fin de obedecer las instrucciones de un programa dado.

Esto se ejerce a través del siguiente proceso:

- Interfase del operador
- Interfase de la máquina
- Control

Por lo tanto los elementos básicos de un sistema de control numérico por computadora son: La interfase del operador, la interfase de la máquina y el control que une a los anteriores. El control es el corazón del sistema. Procesa la información recibida de ambas interfaces, interpretándola y manipulándola con lógica de hardware y de programas (software). La memoria provee el medio para guardar los programas y manejar los datos entrantes. Basado en la información recibida, el control devuelve datos a las distintas interfaces.

La interfase del operador consiste en dispositivos que mandan, reciben e interpretan información. Considerando que las operaciones realizadas por sistemas CNC están definidas por el software, los dispositivos de interfase se necesitan para ingresar los programas de la memoria. La estación del operador es el otro gran elemento de interfase. Ésta contiene todos los interruptores, botones, displays, etc. requeridos para operar y monitorear las actividades de la máquina.

Los dispositivos de mecanizado son manejados por el control. Basados en la información entregada por la interfase del operador y realimentados por varios dispositivos de la máquina, el control conecta y desconecta las acciones y movimientos de ella.

El control

El control realiza decisiones de tiempo real en un proceso. Existen distintos tipos de sistemas de control, sin embargo, cada uno puede dividirse en las mismas unidades funcionales. Cada unidad realiza funciones específicas, y todas las unidades funcionan juntas para ejecutar las instrucciones programadas. Estas unidades son:

- Unidad de entrada: Todas las instrucciones y datos son ingresados a través de ella, pudiendo ser software o señales análogas, las cuales son convertidas a digitales.
- Unidad de memoria: La memoria guarda lo recibido por la anterior. También guarda los resultados de las operaciones aritméticas y provee información a la salida. Se divide en memoria RAM y ROM, siendo la principal diferencia entre éstas que la primera puede ser leída y reescrita, mientras la segunda sólo puede ser leída.
- Unidad aritmética: Realiza cálculos y toma decisiones.
- Unidad de control: Toma las instrucciones de la memoria y las interpreta una por una. Luego manda las instrucciones apropiadas a otras unidades para ejecutar las instrucciones.
- Unidad de salida: Toma datos y los envía, por ejemplo para encender y apagar dispositivos, mostrar información; convierte las señales digitales a análogas.

Los tipos de control más conocidos son los de lazo cerrado y de lazo abierto. La diferencia entre ambos es la presencia o ausencia respectivamente de realimentación de la posición de los ejes.

Interfase del operador

La interfase del operador consiste en todos los dispositivos, exclusivos de la máquina, que mandan y reciben información de control. Los más comunes se detallan a continuación:

- Estaciones de operación: Contienen todos los interruptores, botones y displays necesarios para operar la máquina. El propósito principal de ellas es iniciar la operación automática, ingresar los datos y monitorear las actividades
- Computadora. El uso de ellos es ventajoso para ciertos usos particulares, pero por su elevado costo no vale la pena tenerlos inoperantes durante tanto tiempo. Sin embargo, hoy en día se comienzan a usar modelos más baratos, los que permiten mejorar aún más el control de calidad
- MODEM (Modulator demodulator): Convierte los datos del control en una forma compatible con líneas telefónicas. Su uso principal se relaciona con diagnósticos, y permite a los fabricantes recibir datos de problemas de sus clientes directamente

Interfase de la máquina

La interfase de la máquina contiene todos los dispositivos usados para monitorear y controlar la máquina herramienta. Puede monitorear las posiciones, la presión de aire o hidráulica, controlar los motores, etc. Sus principales componentes son:

- Interruptores de proximidad y de límites: Los primeros son usados para determinar la ubicación de un miembro de la máquina. Se ubican cada cierto tramo del campo de acción de la máquina, sin embargo, hoy en día ya no se usan por su inexactitud. Los interruptores de límites se ubican al final de los ejes y evitan accidentes desconectando la energía cuando algún miembro de la máquina sale del campo de acción
- Interruptores de presión y temperatura: Determinan las condiciones de operación de la máquina, y permiten monitorearlas
- Válvulas de control: Muchos miembros de las máquinas herramientas son controlados por cilindros u otros dispositivos hidráulicos o con aire, por ejemplo, los cambiadores automáticos de herramientas o los controladores de avance de algún eje
- Servomecanismos: Los servomecanismos (también llamados “servos”) son grupos de elementos que convierten la entrada CNC en desplazamientos mecánicos de precisión. Estos elementos incluyen motores (hidráulicos o eléctricos), juegos de engranajes y reductores. Los servos pueden ser de loop cerrado o abierto, dependiendo de si el control recibe la confirmación de que el movimiento se hizo o no.

El objeto del control numérico es lograr la automatización en la fabricación de piezas. Deberá tenerse en consideración que la automatización no implica producción masiva, la automatización debe observarse como la manufactura de piezas que cumplen con especificaciones rigurosas y en las que para su fabricación intervino poco la mano del hombre.

El objeto de las máquinas automáticas es poder reproducir las piezas diseñadas el número de veces que sea necesario y disminuir al máximo la intervención del hombre en la operación de la máquina.

La automatización implica autocorrección, esto significa que para lograr la automatización de una máquina herramienta no sólo es necesario la coordinación de las partes de la máquina, sino que también deberá incluirse que la máquina debe inspeccionar y con los servomecanismos adecuados, corregir las deficiencias o variaciones detectadas.

Las máquinas herramienta automáticas logran su objetivo de operación por medio de motores especiales que manejan a las piezas o a las herramientas de corte de acuerdo a las necesidades de la manufactura. Los motores ejecutan los movimientos que los operadores harían para producir las piezas. Las operaciones de inspección y auto corrección se llevan a cabo por medio de sensores, los que pueden ser mecánicos, eléctricos, electrónicos, sonoros, magnéticos, térmicos o de detección de luz.^{1,2,3}

1.2.1 Definición de puntos de referencia de la Máquina

La máquina posee varios puntos características que determinan el buen funcionamiento de esta, estos puntos son los siguientes:

Home

Esta es una referencia de la máquina que permite establecer un origen de partida, para el movimiento de los ejes. Sin este punto de referencia no hay un correcto posicionamiento de las coordenadas finales de movimiento de los ejes.

Cero Pieza

Con esta referencia la máquina toma un nuevo punto de partida, en el cual son la referencia de inicio para el programa almacenado que contiene las instrucciones de movimientos y procesos

Posición de referencia de Herramienta

En el ciclo automático, se necesita si así lo requiere cambiar herramientas, debido a que no se cuenta con una torreta de herramientas o carrusel para cambiar la herramienta, se emplea un punto de referencia el cual permite al usuario poder hacer este cambio de herramienta.

1.3 Características del CNC

El CNC posee las características mostradas en la tabla 1.1.

Ventajas	Desventajas
Mayor precisión y mejor calidad de productos.	Alto costo de la maquinaria
Mayor uniformidad en los productos producidos.	Falta de opciones o alternativas en caso de fallas.
Un operario puede operar varias máquinas a la vez.	Es necesario mantener un gran volumen de producción a fin de lograr una mayor eficiencia de la capacidad instalada
Fácil procesamiento de productos de apariencia complicada.	
Flexibilidad para el cambio en el diseño y en modelos en un tiempo corto.	
Fácil control de calidad.	
Reducción en costos de inventario, traslado y de fabricación en los modelos y abrazaderas.	
Es posible satisfacer pedidos urgentes.	
No se requieren operadores con experiencia.	Es necesario programar en forma correcta la selección de las herramientas de corte y la secuencia de operación para un eficiente funcionamiento.
Se reduce la fatiga del operador.	
Mayor seguridad en las labores.	
Aumento del tiempo de trabajo en corte por maquinaria.	Los costos de mantenimiento aumenta, ya que el sistema de control es más complicado y surge la necesidad de entrenar al personal de servicio y operación.
Fácil control de acuerdo con el programa de producción lo cual facilita la competencia en el mercado.	
Fácil administración de la producción e inventario lo cual permite la determinación de objetivos o políticas de la empresa.	
Permite simular el proceso de corte a fin de verificar que este sea correcto.	

Tabla 1.1 Características del CNC

Capítulo II

Diseño conceptual de la transmisión

Una frecuente tarea la cual surge dentro del diseño es la transmisión de potencia de una fuente conductora (usualmente fuente rotacional) a una fuente conducida. Esto es llamado transmisión de potencia, es importante notar que algo relacionado con la transmisión de potencia es la sincronización.

En sistemas modernos, los sensores usualmente determinan la información de sincronización, además, algunas veces estos sensores deben ser mecánicamente acoplados para manejar fuentes de movimiento y normalmente a través de mecanismos que anteriormente se manejaban.

Las diferencias entre la transmisión de potencia y sincronización son pequeñas. En la transmisión de potencia, las fuerzas son más duraderas que en la sincronización y los métodos para acoplar requieren técnicas de diseño más robustas.

Sincronizar requiere gran precisión. Esto generalmente se traduce teniendo menos “backlash” y componentes mecánicos más precisos lo que significa que son más costosos.

Las formas para conseguir la transmisión de potencia son las siguientes:

- Engranajes
- Sistemas de transmisión flexible
- Poleas y bandas
- Cadenas y ruedas dentadas
- Poleas sincrónicas y engranes
- Flechas

Para designar que tipo de transmisión será la más adecuada para el movimiento de los ejes principales de la máquina, se tomarán en cuenta varios aspectos:

- Aspectos económicos
- Mayor precisión
- Mayor velocidad
- Mayor disponibilidad de elementos mecánicos en el mercado

Pero antes de ir a la selección se mencionarán aspectos de los diferentes tipos de configuraciones de transmisión que se tienen.

2.1 Selección de componentes

2.1.1 Engranés

En casi todas las máquinas hay transmisión de movimiento de rotación de un eje a otro. Los engranes (o ruedas dentadas) constituyen uno de los mejores medios disponibles para hacerlo.

Los engranes se utilizan para recibir o transmitir movimiento o potencia a otro elemento de características similares de otro engrane o tornillo sinfín.

Existen diferentes tipos de engranes siendo los más comunes los engranes cilíndricos rectos, los cuales se caracterizan en tener los dientes paralelos al eje de rotación, tienen gran utilidad en la industria de las máquinas y herramientas, pero presentan problemas cuando trabajan a elevadas velocidades de rotación, debido al ruido que se genera, por el choque o contacto entre los dientes.

Los engranes rectos se emplean generalmente para transmitir movimiento de rotación entre ejes paralelos.

2.1.2 Sistemas de transmisión flexible

Existen muchas formas de transmitir entre ejes, pero una de las más utilizadas consiste en la utilización de elementos flexibles, como bandas y cadenas.

Las ventajas de la utilización de transmisiones flexibles son:

- Permite transmitir potencia entre ejes muy separados, lo que proporciona flexibilidad en el diseño de la máquina y permitiendo muy diversas configuraciones en la transmisión del movimiento.
- En el caso de bandas, la transmisión es silenciosa.
- Las bandas absorben vibraciones e impactos, mejorando el funcionamiento y la vida de la máquina.

En cuanto a las desventajas de su utilización frente a otro tipo de elementos se tiene:

- En bandas no dentadas, al realizarse la transmisión por fricción, las relaciones de velocidad son poco precisas (por el deslizamiento), dependiendo de la potencia que transmiten.
- Las cadenas no transmiten movimiento con relación de transmisión constante.
- Las cadenas en general son ruidosas, especialmente a alta velocidad.
- La potencia que consiguen transmitir es inferior a engranajes.^{10,11}

Bandas de transmisión

Los mecanismos de correas y poleas son aquellos encargados de transmitir la rotación entre dos ejes (paralelos o no), por medio de la fuerza de rozamiento generada entre la polea y la banda (excepto en las bandas dentadas en que la transmisión se asegura por empuje).

Este mecanismo se emplea siempre que se quiera transmitir el movimiento entre dos ejes, en posiciones relativas cualesquiera, cuando la distancia sea excesiva para el empleo económico de las ruedas dentadas, o cuando aun pudiendo emplear estas, se desee una cierta flexibilidad en la conexión.

Existe una gran variedad de tipos y formas de bandas, siendo los mas comunes los presentados en la tabla 2.1.





TIPO DE BANDA	FIGURA	JUNTA	INTERVALO DE TAMAÑOS	DISTANCIA ENTRE CENTROS
Plana		Sí	$t = \begin{cases} 0.03 \text{ a } 0.20 \text{ in} \\ 0.75 \text{ a } 5 \text{ mm} \end{cases}$	No hay límite superior
Redonda		Sí	$d = \frac{1}{4} \text{ a } \frac{7}{8} \text{ in}$	No hay límite superior
Trapezoidal o en V		Ninguna	$b = \begin{cases} 0.31 \text{ a } 0.91 \text{ in} \\ 8 \text{ a } 19 \text{ mm} \end{cases}$	Limitada
Reguladora		Ninguna	$p = 2 \text{ mm y mayor}$	Limitada

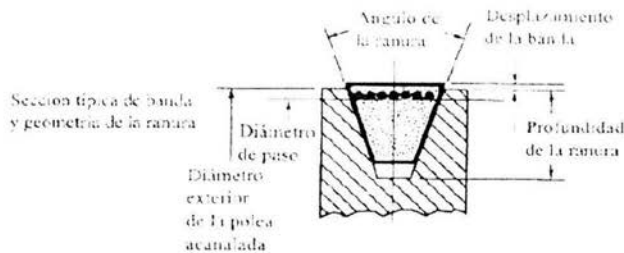
Tabla 2.1 Tipos de banda

Las bandas suelen estar hechas de un material compuesto, en el cual se incluye uretano y también tela impregnada de caucho, y viene reforzada con alambres de acero o cuerda de nylon (para aumentar su resistencia a la tracción). En ocasiones también se impregnan las superficies con un material antifricción.

Las poleas planas se utilizan sobre poleas lisas, sin acanaladuras. A veces se utilizan varias unidas para transmitir mayor potencia. Son las más utilizadas para configuraciones cruzadas.

Las correas en V se utilizan sobre poleas ranuradas en V. Son ligeramente menos eficientes que las planas, pudiéndose utilizar también varias en poleas múltiples. No

pueden deslizar lateralmente sobre la polea, por lo que no hay riesgo de que se salgan, a diferencia de las planas (Fig. 2.1).



Sección transversal de una banda en V y ranura de polea acanalada

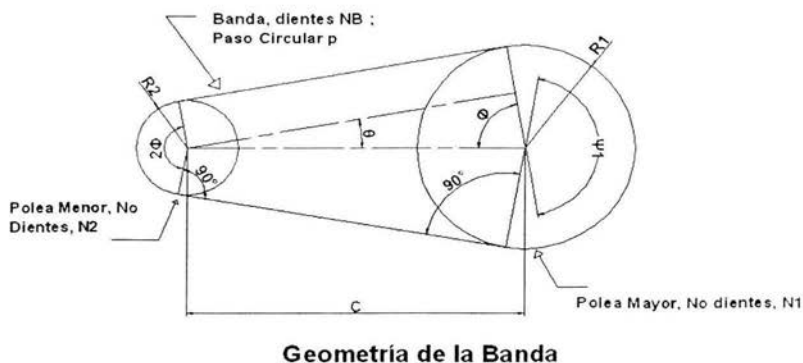
Fig. 2.1 Bnada en V

Las bandas dentadas permiten una relación de velocidad constante y no requieren pre - tensión para su funcionamiento. Necesitan de poleas dentadas especiales, y son mas caras por lo regular.

Como se ha comentado, la potencia transmisible depende del coeficiente de fricción entre polea y banda. Para aumentar la fricción y con ella la potencia transmisible, se utilizan bandas trapezoidales sobre poleas con acanaladuras. El utilizar bandas es aplicable cuando se quiere incrementar el torque entre el elemento conductor y el elemento conducido.

Considere lo siguiente:

Dos poleas impulsadas por una banda, la primera polea tendrá un radio R_1 y la segunda polea tiene un diferente radio R_2 (Fig 2.2).



Geometría de la Banda

Fig. 2.2 Características Geométricas de un sistema de transmisión por bandas y poleas síncronas

- C = Distancia entre centros [in]
 L_B = Longitud de la banda (in) = ρN_B
 ρ = Paso de la banda [in]
 N_B = Número de dientes en la banda = ρL_B
 N_1 = Número de dientes de la polea mayor
 N_2 = Numero de dientes en la polea menor
 Φ = Ángulo de la envolvente sobre la polea menor [radianes]
 θ = Ángulo entre la porción recta de la banda y la línea de centros [radianes]
 R_1 = Radio de la polea mayor [in] = $\frac{N_1 \rho}{2\pi}$
 R_2 = Radio de la polea menor [in] = $\frac{N_2 \rho}{2\pi}$
 π = 3.14159

De acuerdo a las siguientes ecuaciones podemos determinar la distancia entre centros óptima para un buen funcionamiento:

$$C = \left(\frac{1}{2}\right)\rho[(N_B - N_1) + k(N_1 N_2)]$$

$$\text{Donde } k = \left(\frac{1}{\pi}\right)\left[\text{Tan}\left(\frac{\pi}{4} - \frac{\phi}{2}\right) + \phi\right]$$

Y ϕ se determina de

$$\left(\frac{1}{\pi}\right)\left[\text{Tan}(\phi) - \phi\right] = \frac{(N_B - N_1)}{(N_1 - N_2)}$$

En el siguiente esquema (Fig. 2.3) completo de un sistema de transmisión flexible (banda , poleas).¹⁰

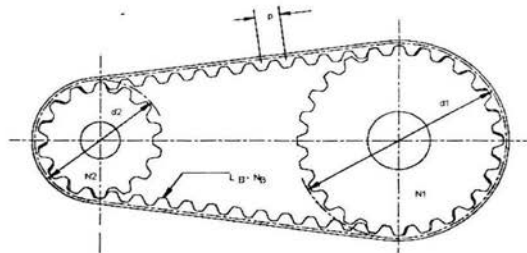


Fig. 2.3 Sistema de transmisión flexible

2.1.3 Tornillo sinfin - corona

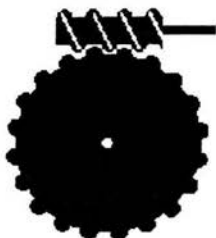


Fig. 2.4 Transmisión sinfin corona

Este mecanismo permite transmitir el movimiento entre árboles que se cruzan (Fig. 2.4). El árbol motor coincide siempre con el tornillo sin fin, que comunica el movimiento de giro a la rueda dentada que engrana con él, llamada *corona*. Una vuelta completa del tornillo provoca el avance de un diente de la corona. En ningún caso puede usarse la corona como rueda motriz. Puede observarse un tornillo sin fin en el interior de muchos contadores mecánicos.⁶

Por cada vuelta de la corona, el tornillo completa tantas vueltas como número de dientes tenga la corona. Por lo tanto, la relación de transmisión del mecanismo es simplemente

$$i = 1 / d \text{ corona}$$

donde :

i : relación de transmisión
 d corona : número de dientes de la corona

La relación de transmisión resulta ser inferior a la unidad, de manera que el mecanismo actúa siempre como reductor. Alternativamente, puede calcularse en función del diámetro primitivo de la corona y del paso de rosca del tornillo, definido como la distancia entre dos *surcos* consecutivos de la *hélice*. La expresión adecuada resulta ser:

$$i = p / (\pi D \text{ corona})$$

donde:

p : paso de rosca del tornillo
 D corona : diámetro primitivo de la corona

2.1.4 Tornillo - tuerca



Fig 2.5 Transmisión Tornillo - cuerda

El giro de un tornillo alrededor de su eje produce un movimiento rectilíneo de avance, que lo acerca o lo separa de la *tuerca*, fija (Fig. 2.5). Alternativamente, una tuerca móvil puede desplazarse de la misma manera a lo largo de un tornillo o *husillo*. El mecanismo es capaz de ejercer grandes presiones en el sentido de avance del tornillo. Por eso se usa, por ejemplo, para construir tornillos de banco.

Hay diferentes tipos de tornillos y tuercas. Un parámetro característico es el número de entradas o *surcos* (hélices independientes) del tornillo. En tornillos de una sola entrada, el *paso* de rosca del tornillo coincide con el avance del tornillo producido al girar 360° alrededor de su eje.⁶

Por lo tanto, las velocidades cumplen la relación:

$$V = \omega p / (2 \pi)$$

Donde:

V : velocidad de avance del tornillo

ω : velocidad de giro del tornillo

p : paso de rosca

2.2 Diseño de detalle de la transmisión

De acuerdo a las descripciones anteriores de medios de transmisión de potencia, hemos optado por seleccionar poleas y bandas dentadas, de acuerdo a sus ventajas sobre los demás medios de transmisión.

Los siguientes cálculos muestran la selección de parámetros a tomar para tener las características deseadas.

Paso tornillo sin fin (eje transversal y longitudinal) = 0.635 mm

$$\text{Avance} = \frac{0.635 \text{ mm}}{2\pi \text{ rad}} = 0.10106 \left[\frac{\text{mm}}{\text{rad}} \right] \quad \text{Avance} = 0.0017638 \left[\frac{\text{mm}}{\circ} \right]$$

Al estar empleando motores de paso, se hizo el análisis de acuerdo al número de pasos deseados a la salida de la transmisión, esto es:

Relación de transmisión en las poleas dentadas es de;

$$R_p = \frac{44}{14}$$

La relación de transmisión de 3.1428.

Ahora el avance final del tornillo sin fin es:

$$\text{Avance}_T = R_p \times \text{Avance}_{\text{Sin fin}}$$

$$\text{Avance}_T = 0.0017638 \left[\frac{\text{mm}}{\circ} \right] \times 3.1428 = 0.31763 \frac{\text{mm}}{\circ}$$

Si tomamos en cuenta que la resolución del motor de pasos es de $1.8 \left[\frac{\circ}{\text{paso}} \right]$

Se tiene:

$$\text{Avance}_T = 0.31763 \left[\frac{\text{mm}}{\circ} \right] \times 1.8 \left[\frac{\circ}{\text{paso}} \right]$$

$$= 0.00998 \left[\frac{\text{mm}}{\text{paso}} \right]$$

Al obtener el inverso de la operación tenemos:

$$\text{Avance}_T = 100.222 \left[\frac{\text{paso}}{\text{mm}} \right]$$

Esta última cantidad nos dice que por cada π rad que se mueva la entrada de la transmisión de las poleas dentadas, tendremos 2π rad de movimiento en el tornillo sinfin.

Estas operaciones son realizadas en función de que los motores de pasos se mueven a una sincronización de señales, por tal motivo se hizo este análisis.

Los siguientes datos corresponden a las poleas y bandas seleccionadas (Tabla 2.2 y 2.3 respectivamente):

	POLEA A	POLEA B
Paso	2.03 mm [MXL]	2.03 mm [MXL]
Número Dientes	14	44
Material	Aleación Aluminio	Aleación Aluminio
Ancho de banda	6 mm	6 mm
Paso Diametral	9.1 mm	28.5 mm
Diámetro barreno	8 mm	8 mm
Config. Flanco	2 flancos/ con barreno	2 flancos/ con barreno
Diámetro Exterior	8.6 mm	28 mm

Tabla 2.2 Datos de las poleas seleccionadas

	BANDA A	BANDA B
Tipo	MXL	MXL
Paso	2.03 mm	2.03 mm
Ancho banda	6 mm	6 mm
Numero dientes	70mm	80m
Longitud de banda	142 mm	163mm

Tabla 2.3 Datos de las bandas seleccionadas

Para el diseño de la transmisión para bajar y subir el motor del husillo se usó un tren de engranes compuestos, de tal modo que la relación de transmisión de este arreglo es la siguiente:

$$R_T = \frac{\text{No. Dientes de engranes impulsores}}{\text{No. Dientes de engranes impulsados}}$$

Se muestra en el siguiente diagrama (Fig. 2.6) el tren de engranes usado en la transmisión:

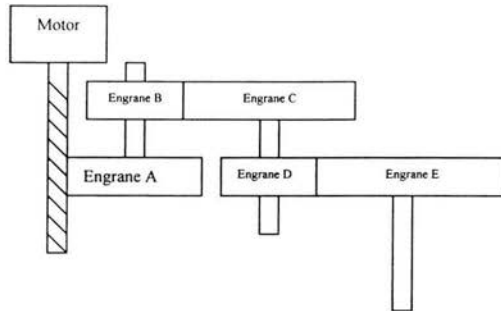


Fig. 2.6 Tren de Engranes Compuesto para el sistema de bajar y subir

Engranes Impulsores	Engranes Impulsados
Engrane B con 20 dientes	Engrane A con 40 dientes
Engrane D con 15 dientes	Engrane C con 36 dientes
	Engrane E con 45 dientes

Entonces la relación de transmisión final es:

$$R_T = \frac{1 \times 20 \times 15}{40 \times 36 \times 45} \qquad R_T = 0.004629$$

Es decir que por cada 216 vueltas que dé el motor (sinfin), el engrane E que sube y baja el motor del husillo, solo se mueve una sola vuelta.

Esto muestra que un sistema de transmisión sinfin - corona sirve para reducir el movimiento.

Capítulo III

Control electrónico

3.1 Sistema motriz

3.1.1 Sistemas disponibles

Los sistemas motrices básicos para controlar movimiento, que incluya control de velocidad, torque o posición son:

- Motores de aire
- Motores hidráulicos
- Clutch/Freno
- Motores a pasos
- Motores de inducción AC
- Servomotores

Motores de aire: se utiliza aire comprimido para obtener movimiento. La presión y el flujo determinan la velocidad y el par. La posición generalmente no es un requerimiento.

Ventajas

Bajo costo
Componentes disponibles
Fácil de aplicar
Fácil mantenimiento
Fácil de comprender
Fuente de poder centralizada

Desventajas

Ruido del compresor
Difícil de regular la velocidad
Susceptible a contaminación
Ineficiencia energética

Motores hidráulicos: utilizan aceite a presión para mover un pistón. Mayor presión da como resultado mayor par.

Ventajas

Fácil de aplicar
Alto par disponible
Fuente de poder centralizada
Fácil de comprender

Desventajas

Ruido considerable
Difícil de controlar la velocidad
Posicionamiento lento
Susceptibles a fugas
Ineficiencia energética
Riesgo de fuego
Se requiere mucho mantenimiento

Clutch/Freno: Un dispositivo que acopla un eje continuamente en movimiento y una carga. El desacoplamiento de la carga provoca el paro. La variación de los tiempos encendido/apagado varía el desplazamiento.

Ventajas

Fácil de aplicar
Bajo costo
Buenos para inicio/paro con cargas ligeras
Provee una manera sencilla de unificar velocidad

Desventajas

Aceleración sin control
Inexactos
Susceptibles al desgaste
Desempeño con baja repetitibilidad

Motor a pasos: Dispositivo electromecánico que convierte un pulso digital a un movimiento de rotación o desplazamiento específico. Un tren de impulso da como resultado una velocidad de rotación.

Ventajas

Control simple
Costo moderado
Bueno para cargas constantes
Buena precisión de posicionamiento

Desventajas

Susceptibles a perder pasos
Malo para cargas que varían
Ineficiencia energética
Tamaño grande
Problemas de resonancia

Motores de inducción AC: ampliamente utilizados para requerimientos de velocidad constante, Arrancadores eléctricos proveen una protección a conexiones y sobrecarga. Tecnología actual permite capacidad de velocidad variable.

Ventajas

Motor simple
Bajo costo
Tecnología bien desarrollada
Control de encendido/apagado directo
Posibilidad de control de velocidad burdo
Cableado sencillo
Amplia variedad de productos
Muchos vendedores disponibles

Desventajas

Control de posición limitado
Tamaño relativamente grande

Servomotores: un motor con algún mecanismo de retroalimentación. Circuitos electrónicos controlan la velocidad y posición.

Ventajas

Alto desempeño
Tamaño reducido
Gran variedad de componentes

Se logran altas velocidades con controles especializados

Desventajas

Mayor costo
Alto desempeño limitado por los controles
Par a alta velocidad limitado por el conmutador o los electrónicos

3.1.2 Selección del sistema motriz

Debido a la necesidad de tener un control sobre la posición de los motores, las alternativas existentes en el proyecto son motores de pasos, de corriente directa o servomotores. Existen diferencias de costo entre estas opciones. Los motores de pasos y los de corriente directa son considerablemente más económicos.

En cuanto al posicionamiento, los motores de corriente directa necesitan de elementos auxiliares para saber la posición en la que se encuentra el husillo. Por otro lado, los motores de pasos tienen una buena precisión en el posicionamiento.

La ventaja más clara de los de corriente directa sobre los motores de pasos es la velocidad. Sin embargo, para el par requerido en la máquina, el costo del motor de corriente directa resulta ser mayor a los motores de pasos.

Es por estas razones que se optó por la elección de motores de pasos. Se sacrificó velocidad a cambio de obtener par y precisión. Como el control de los motores es mediante encoders lineales (más adelante en este capítulo) no es necesario que los motores tengan por sí mismos un control de posición. Se pudieron haber utilizado motores de corriente directa si se hubiera tenido la posibilidad de adquirirlos.

3.2 Control sistema motriz

3.2.1 Microcontroladores PIC16F7X y PIC16F87X

El diseño del controlador está formado básicamente por tres microcontroladores (μ Cs), uno de ellos teniendo la tarea de ser el μ C maestro y dos ellos siendo esclavos, y circuitos controladores de motores a pasos y de corriente directa. El diseño de estos últimos se mencionara en los subcapítulos siguientes.

Los microcontroladores utilizados son el PIC16F877 como maestro y los PIC16F73 como esclavos. Ahora bien, se busca no tener un μ C tan robusto para el manejo de los μ Cs esclavos, debido a su manejo de información.

Considerando que se necesitaba tener en el μ C esclavo comunicación USART, un puerto de ocho bits, convertidores analógicos digitales y contadores de 16 bits, se tomó la decisión de emplear un μ C de la misma familia pero con menor capacidad. La mejor selección fue el μ C PIC16F73 ya que cumplía con dichas especificaciones.^{8,9}

Las características de ambos microcontroladores se muestran en la tabla 3.1.

	PIC16F877	PIC16F73
Pines	40	28
Palabra	8 bits	8 bits
Puertos	5 con 33 entradas / salidas digitales	3 con 19 entradas / salidas digitales
Velocidad	Hasta 20 MHz	Hasta 20 MHz
Ciclos de instrucción	200 ns	200 ns
Memoria de programa	8k x 14 bits (Flash)	4k x 14 bits (Flash)
Memoria RAM	368 x 8 bytes	192 x 8 bytes
Memoria ROM	256 x 8 bytes	
Fuentes de interrupción	14	11
Niveles de desbordamiento	8	8
Direccionamiento	Directo e indirecto	Directo e indirecto
Código de protección	Programable	Programable
Temporizadores / Contadores	8 bits con preescala 16 bits con preescala 8 bits con preescala, postescala y periodo de registro	8 bits con preescala 16 bits con preescala 8 bits con preescala, postescala y periodo de registro
Módulos de captura	2	2
Módulos de comparación	2	2
Modulación de ancho de pulso (PWM)	2	2

Captura	16 bits con máxima resolución de 12.5 ns	16 bits con máxima resolución de 12.5 ns
Comparación	16 bits con máxima resolución de 200 ns	16 bits con máxima resolución de 200 ns
PWM	Máxima resolución de 10 bits	Máxima resolución de 10 bits
Canales analógico / digital	10 multicanales convertidores	5 multicanales convertidores
Modos de comunicación	SPI (Interfase periférica serie) UART (Unidad de transmisión recepción asíncrona) USART (Unidad de transmisión recepción síncrona / asíncrona) SCI (Interfase de comunicación serie) I2C (Circuito integrado interno) SSP (Puertos seriales síncronos) PSP (Puerto paralelo esclavo) de 8 bits	SPI (Interfase periférica serie) USART (Unidad de transmisión recepción síncrona / asíncrona) SCI (Interfase de comunicación serie) I2C (Circuito integrado interno)
Funcionamiento para leer / grabar memorias externas	Si	No

Tabla 3.1 Características de Microcontroladores empleados

La figura 3.1 muestra el diagrama de pines del μC maestro :

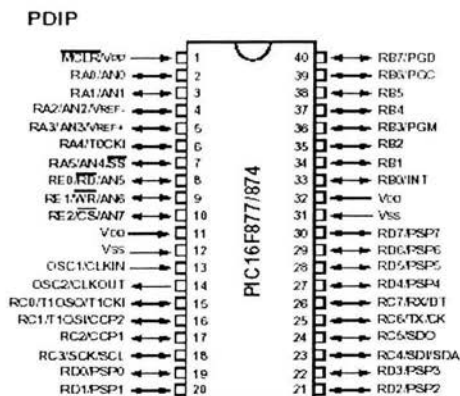


Fig. 3.1 Microcontrolador principal PIC16F877

La figura 3.2 muestra el diagrama de pines del μ C esclavo:

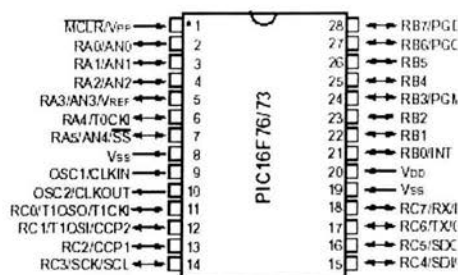


Fig. 3.2 Diagrama del Microcontrolador esclavo PIC16F73

Otro parámetro de selección fue la facilidad de programación, ya que solo cuentan con 35 instrucciones para la programación, de ambos controladores mencionados.

3.2.2 Expansión de Memoria EEPROM

En la descripción de los μ Cs y sus características, tanto el PIC16F877 como el PIC16F3 cuentan con una capacidad de memoria (memoria flash) de 8 y 4 k respectivamente. Esto puede no ser suficiente para la longitud del programa que se vaya almacenar. Por tal razón se busca ampliar la capacidad de memoria.

Debido a la poca memoria EEPROM del μ C maestro, se buscó tener facilidad de manejo y almacenamiento de información. Tal motivo nos hace seleccionar un tipo de memoria que cumpla con ciertos requisitos:

- Memoria programable y borrrable eléctricamente
- Bajo costo

Estas dos características hacen que seleccionemos este tipo de memoria. Otro tipo de memoria (Flash) podría convenir, pero su costo es mayor que la EEPROM. También se busca tener almacenada la información del programa durante cierto tiempo y una memoria RAM no facilita esta tarea.

Por lo tanto la memoria que se empleó fue una EEPROM 2864, donde se tiene la capacidad de 8k x 8 bits de memoria, con 13 bits de direccionamiento.²¹ Esto permite almacenar hasta 32768 líneas de código CNC.

La figura siguiente muestra el diagrama de pines de la memoria empleada (Fig 3.3).

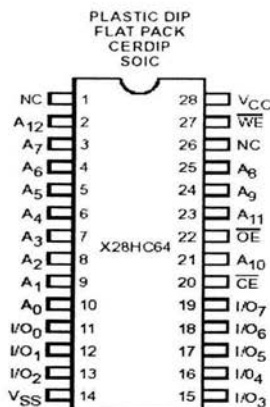


Fig 3.3 Memoria Empleada

Se dispusieron dos memorias en paralelo para garantizar que se almacenaran hasta 16 bits de información de datos, con el objetivo de tener mayor rango de capacidad de datos de posicionamiento (ver figura 4.16)

3.2.3 Diseño de tarjeta de control de motores

Para el control de los motores fue necesario diseñar una tarjeta que fuera el vínculo entre una parte de baja potencia (microcontroladores) y una de alta potencia (motores). Cuando pretendemos controlar cargas inductivas con un microcontrolador, nos encontramos que no podemos realizarlo directamente. Debemos usar un "driver" o exitador para separar la señal emitida por el microcontrolador, del circuito inductivo o de potencia. Para esto se necesita un Puente H que puede armarse de distintas formas.

Los puentes-H (llamados "H BRIDGES" en inglés) son circuitos que permiten controlar motores eléctricos de corriente directa en dos direcciones desde un circuito digital (TTL, CMOS, el puerto de una computadora, desde un microcontrolador, etc...).

Un motor nunca debe ser conectado directamente a la salida digital de un circuito, por dos razones que se mencionan a continuación:

Razón 1: Un circuito digital tradicional generalmente no tiene la capacidad de corriente necesaria para hacer que un motor eléctrico gire. Si se conecta directamente un motor, un foco incandescente o algún otro elemento que consuma mucha corriente, lo más probable es

que el circuito se sobrecaliente y se queme en unos segundos. La manera más sencilla de manejar un elemento electromecánico pequeño con un circuito digital es utilizando un transistor como interruptor. Así el circuito digital solo prende y apaga el transistor y el transistor es el que prende y apaga el motor.

Razón 2: Casi todos los dispositivos electromecánicos (aunque sean pequeños) son muy inductivos. No permiten ser apagados de golpe. Es decir, cuando se desconecta un motor eléctrico que está funcionando, el motor (debido a que es un dispositivo inductivo) trata todavía de mantener por una fracción de segundo la corriente circulando a través de él. Y durante este pequeñísimo tiempo puede generarse una chispa en la parte del circuito que realizó la desconexión. Esta chispa puede muy fácilmente dañar circuitos electrónicos. Según el tamaño del motor y según la corriente que esté utilizando, esta chispa puede o no ser visible, pero siempre existe a menos que se coloque en paralelo con el motor un diodo de protección.

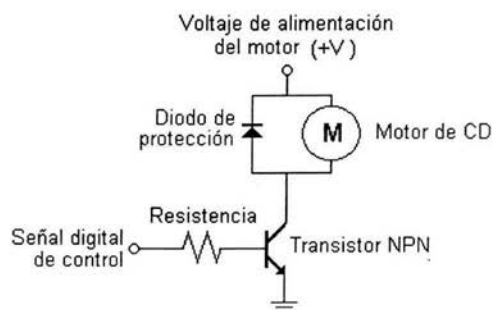


Fig. 3.4 Activación de un motor mediante un transistor

Un puente H es básicamente un arreglo de CUATRO interruptores acomodados de la siguiente manera:

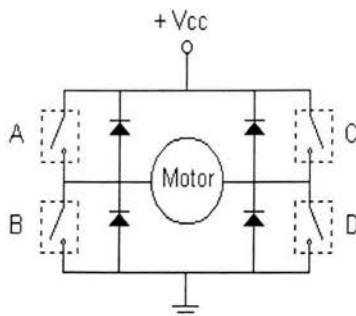


Fig. 3.5 Puente H

Estos interruptores (A,B,C y D) pueden ser de transistores bipolares (como el de arriba), de mosfets, de jfets, de relevadores o de cualquier combinación de elementos. El punto central es: los puentes H se utilizan para que un motor eléctrico de corriente directa funcione en dos sentidos sin tener que manejar voltajes negativos.

Si se cierran solamente los contactos **A y D** la corriente circulará en un sentido a través del motor (o del relevador o de cualquier sistema que esté conectado ahí en medio), y si se cierran solamente los contactos **B y C** la corriente circulará en sentido contrario. El puente H necesita de cuatro diodos de protección para el motor.

Elegimos este circuito integrado por ser de fácil adquisición y económico, además se evita el trabajo de armar un Puente H con componentes discretos.

Este circuito permite realizar un Puente H para hacer girar motores de CC en ambas direcciones, del tipo de los que se utilizan en juguetes, radio grabadores, lectoras de CDs, etc. Con dos circuitos como este, (4 integrados LM 386) se pueden controlar motores paso a paso de cuatro terminales.²⁰

3.3 Sensores

Los sensores son dispositivos utilizados para proveer de señales representativas de condiciones en procesos con máquinas. En muchas situaciones, estos dispositivos proveen de señales analógicas que representan un rango de valores.

Sensores de tipo discreto proveen información que representa la presencia o ausencia de un objeto.

Tipos de sensores

Estos se encuentran en muchas configuraciones para acoplarse a cualquier necesidad en aplicaciones comerciales e industriales. Se necesitan de muchos tipos de sensores dado las características operacionales de cada uno.

Existen interruptores de contacto, inductivos, capacitivos, ultrasónicos y fotoeléctricos

3.3.1 Sensores ópticos

Encoders

Toda máquina que se mueve necesita alguna manera de medir el movimiento. Dado que las máquinas herramienta, máquinas de inspección, equipo de manejo de materiales han evolucionado y han pasado de máquinas rudimentarias manuales a piezas automatizadas altamente sofisticadas, tienen mecanismos de medición internos. El tipo más común de componente usado hoy en día es el encoder.

Los encoders pueden ser caracterizados en ópticos (fotoeléctricos), magnéticos y de contacto mecánico. Los encoders fotoeléctricos en particular, debido a su gran exactitud, alta confiabilidad y costo relativamente bajo, juegan un papel importante en la tecnología de máquinas herramienta.

Hay dos tipos básicos de encoders: rotatorios y lineales. Mientras que los principios técnicos detrás de ellos son similares, sus aplicaciones específicas no lo son.

El principio básico

La mayoría de los encoders rotativos y lineales de hoy en día operan bajo el mismo principio de escaneo fotoeléctrico de rejillas muy finas. Estos también se llaman sensores fotoeléctricos.

Los sensores fotoeléctricos son un dispositivo para sensor posición. Usan un haz de luz modelado que se obstruye o se refleja con el objeto a sensor.

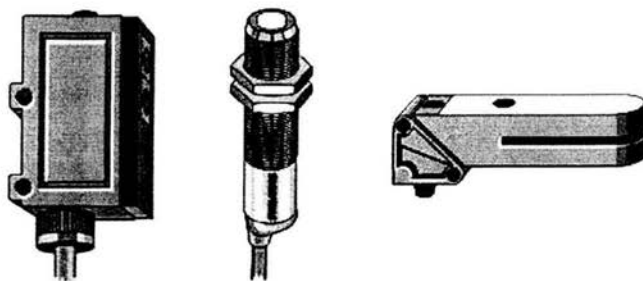


Fig. 3.6 Tipos de sensores ópticos

Un sensor fotoeléctrico incluye los siguientes elementos: un emisor (fuente de luz), un receptor para detectar la luz emitida y electrónicos que evalúan y amplifican la señal detectada causando que la salida del sensor cambie de estado.

Luz modulada

Usar luz modulada aumenta el rango de sensado disminuyendo los efectos de la luz ambiental. La luz modulada esta en una frecuencia entre 5 y 30 KHz. Esto permite que el sensor fotoeléctrico distinga la luz modulada de la ambiental. Las fuentes de luz utilizadas por estos sensores están en un rango del espectro del verde visible a el infrarrojo. Los diodos emisores de luz (LED) son típicamente usados.

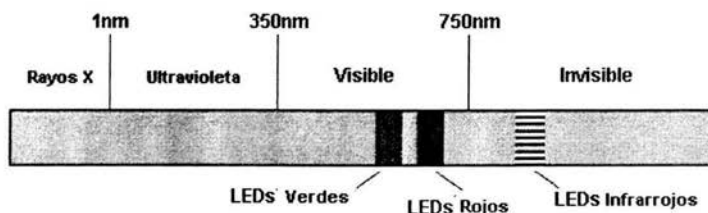


Fig 3.7 Región del espectro electromagnético utilizada en sensores ópticos

Exceso de ganancia

En algunos ambientes, especialmente en aplicaciones industriales, el aire contiene contaminantes como polvo, suciedad, humo y humedad. Un sensor operando en este tipo de entorno con contaminantes requiere más luz para operar apropiadamente. El exceso de ganancia es la cantidad de luz emitida por el transmisor en exceso de la requerida para operar el receptor. Con aire limpio, un exceso de ganancia mayor que 1 es generalmente suficiente. Por otra parte, si un ambiente tiene contaminantes capaces de absorber el 50% de la luz emitida, se necesitaría un mínimo de exceso de ganancia de 2 para operar el sensor.

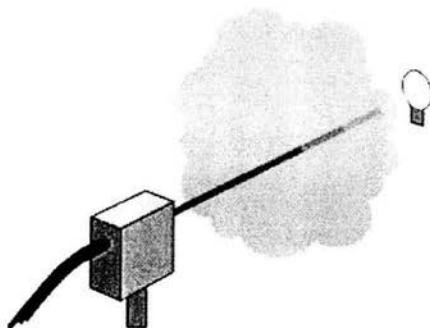


Fig. 3.8 Exceso de ganancia para ambientes contaminados

Técnicas de escaneo de los sensores fotoeléctricos

La mejor técnica depende de el objetivo. Algunos objetivos son opacos y otros ligeramente reflectivos. En algunos casos es necesario detectar cambios en el color. La distancia también es un factor en la selección de una técnica. Las técnicas más comunes son: escaneo de rayo pasado, escaneo reflectivo o retroreflectivo y escaneo difuso.

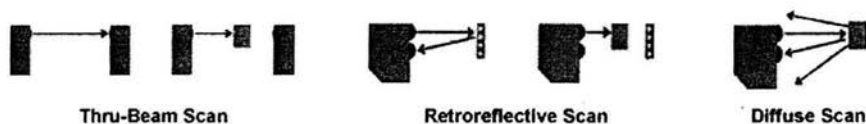


Fig. 3.9 Técnicas de escaneo de sensores fotoeléctricos

Escaneo de rayo pasado

Esta técnica requiere separar las unidades de emisión y recepción. Con ésta técnica, un objeto puesto en el camino del haz de luz bloquea la luz al receptor. Esto causa que la salida del receptor cambie de estado. Cuando el objetivo deja de bloquear el haz la salida del receptor vuelve a su estado normal.

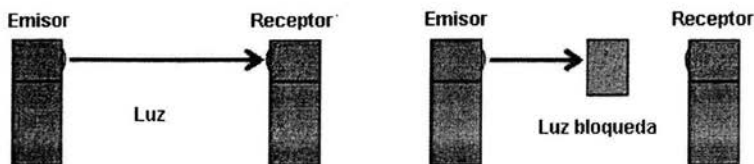


Fig. 3.10 Escaneo de rayo pasado

Escaneo retroreflectivo

Esta técnica tiene la unidad de emisión y la recepción del mismo lado. En el lado opuesto existe una placa reflectiva que tiene la capacidad de reflejar el haz del luz del emisor en dirección del receptor. Cuando un objetivo interfiere el paso de la luz, el receptor deja de detectar luz.

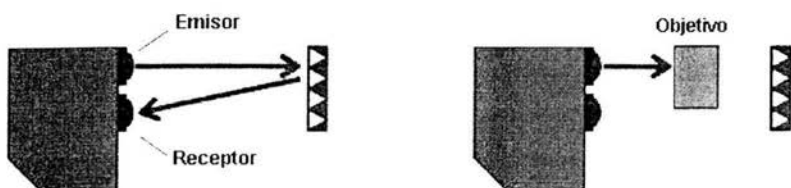


Fig. 3.11 Escaneo retroreflectivo

Escaneo difuso

A diferencia de las técnicas anteriores, ésta activa el receptor cuando el objetivo se encuentra presente. Para esto es necesario que el objetivo tenga la capacidad de reflejar la luz del emisor.

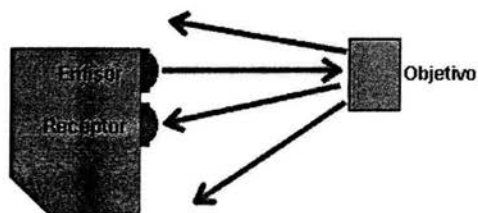


Fig. 3.12 Escaneo difuso

Operación con luz ausente

Es el modo de operación en el cual la carga es energizada cuando la luz del emisor está ausente en el receptor.

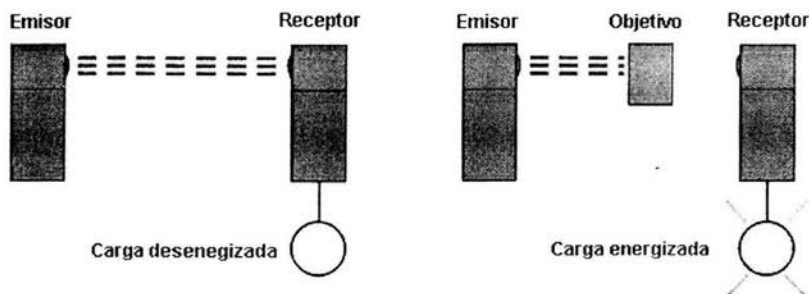


Fig. 3.13 Operación con luz ausente

Operación con luz presente

Es el modo de operación en el cual la carga es energizada cuando la luz del emisor está presente en el receptor.

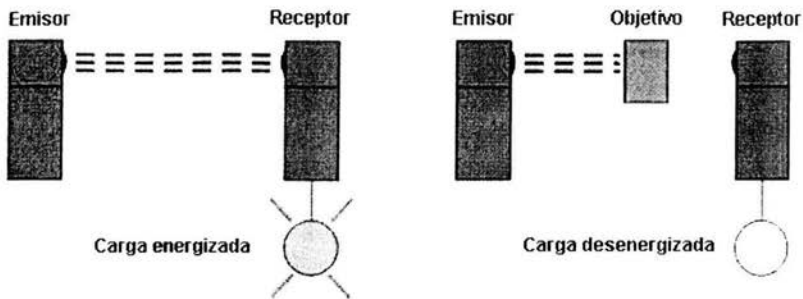


Fig. 3.14 Operación con luz presente

Sensor de ranura (Slot BERO)

El objetivo se pone adentro de la ranura del sensor. La luz emitida pasa a través del objeto. Cambios de contraste, arrugas o agujeros en el objetivo variarán la cantidad de luz que llega al receptor. Este tipo de sensor es el utilizado como unidad lectora para los encoders.

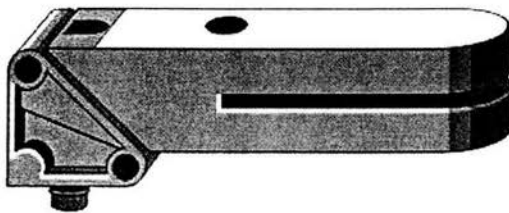


Fig. 3.15 Sensor de ranura

La unidad lectora de un encoder consiste de una fuente de luz, un lente condensador para enfocar el haz de luz, la retícula de escaneo con una rejilla y celdas fotovoltaicas de silicio. Cuando la rejilla se mueve relativamente a la unidad de lectura, las líneas de la rejilla coinciden alternadamente con los espacios en la unidad de lectura. La fluctuación periódica de intensidad de luz se convierte en señales eléctricas mediante celdas fotovoltaicas. Las señales de salida son dos señales sinusoidales que son interpoladas o digitalizadas según sea necesario.¹⁷

Encoders rotatorios

En varios sectores de la tecnología de máquinas es necesario traducir las posiciones angulares y movimientos angulares en señales eléctricas, ya sea para despliegue, automatización o control numérico. Los encoders rotativos son usados para este propósito de medir la rotación en motores. También son utilizados para medir movimientos lineales, por ejemplo, cuando se utilizan husillos especialmente con bolas recirculantes.

Categorías de encoders rotatorios			
Encoders rotatorios incrementales		Encoders rotatorios absolutos	
Encoders rotatorios para medición de ángulos	Encoders rotatorios para medición de posición lineal (utilizados en conjunto con tornillos de bolas o piñón cremallera)	Encoders rotatorios absolutos para una vuelta (para medición de ángulos)	Encoders absolutos para varias vueltas (encoders multivuelta, utilizados principalmente con tornillos de bolas)

Tabla 3.2 Categorías de encoders rotatorios

Incrementales

Las señales de salida de los encoders rotativos incrementales son evaluadas por un contador electrónico en el cual el valor medido está determinado por el conteo de los “incrementos”. Estos encoders conforman la mayoría de los encoders rotativo utilizados hoy en día.

Cuando se trabaja con encoders rotativos incrementales para medir longitud, generalmente se está trabajando en máquinas que utilizan husillos. Los encoders se fijan generalmente en la parte trasera del servomotor o al husillo el en lado opuesto del motor para tener una precisión libre de “backlash”.

Los encoders rotatorios estándar para aplicaciones de medición de longitud – y en particular para medición de desplazamientos en tornillos de bolas recirculantes. Se representan por un eje de encoder que incorpora electrónicos digitalizadores. El número de ciclos de onda cuadrada de las señales de salida por cada rotación del eje es idéntico a el número de líneas en la graduación del disco.

Los encoders rotatorios incrementales con acoplamientos integrales utilizados para medir longitudes están también disponibles en el mercado. El diseño tiene algunas características favorables. Especialmente en la versión en la cual el acoplamiento no está montado en un lado del rotor, esto es, entre el eje del motor y el eje del encoder, pero esta fijo permanentemente al estator. En esos casos, el eje del motor está conectado directamente al eje del encoder rotacional. La unidad de lectura está conectada al encoder por rodamientos, sin embargo si una conexión rígida a la carcasa. En lugar de eso, se coloca un acoplamiento entre estos componentes y compensa errores de alineamiento entre ambos ejes.

Todas estas versiones de encoders rotacionales son, en principio, sistemas de medición angular y son, entregando requerimientos de precisión, utilizados en muchos casos para este propósito. La resolución de dichos encoders puede ser incrementa con la utilización de interpolación electrónica. También existen, por supuesto, encoders rotacionales de precisión específicamente diseñados para medición de ángulos.

Absolutos

Los encoders rotatorios absolutos proveen un valor de posición angular que se deriva de un patrón en el disco codificado. La señal del código es procesada en una computadora o un control numérico. Después de encender la máquina, suponiendo que hubiera una interrupción de energía, el valor de la posición está disponible inmediatamente. Dado que este tipo de encoders requieren una óptica y electrónica más sofisticada que los incrementales, su precio es mayor.

El codificador más utilizado es el codificador de Grey, donde solo un bit cambia con la transición de un paso a otro. Un codificador binario también es comúnmente utilizado para resoluciones muy altas. En este código, más de una señal puede cambiar al pasar de un paso al siguiente. De ahí que se deben tomar precauciones para evitar ambigüedades. Además de éstos dos códigos, existen variedad de códigos que pueden ser utilizados, aunque están perdiendo su importancia dado que los programas de computadora modernos están basados en sistema binario para aumentar la velocidad.

Encoders lineales

Dado que la tendencia de las máquinas es incrementar la precisión y resolución, mayor confiabilidad, velocidad y eficiencia, los sistemas de retroalimentación lo deben hacer también. Actualmente, los sistemas de retroalimentación lineales están disponibles con resoluciones en un rango por debajo de micras.

Resoluciones sub-micras son utilizados en la industria de semiconductores y en maquinado de ultra precisión. Lograr esta resolución es posible con el uso de reglas graduadas lineales que transmiten la información de desplazamiento directamente a un lector digital, controlador numérico o algún otro dispositivo periférico para evaluación.

Como en los rotativos, las graduaciones lineales operan bajo el mismo principio fotoeléctrico para lectura, pero las graduaciones lineales están construidos de manera recta y sus señales de salida son interpoladas o digitalizadas de una manera directa. Una de estas señales es siempre utilizada por el lector digital para determinar y establecer la posición de "home" en el eje lineal de la máquina en caso de una interrupción de energía o para referenciar la pieza de trabajo.

Existen dos versiones de graduación lineal: expuesta o cubierta.

Con una graduación cubierta o sellada, la unidad de lectura se monta en un carro guiado por tornillo de bolas a lo largo de la graduación de cristal; el carro está conectado a la máquina mediante un acoplamiento libre de "backlash" que compensa errores de alineación entre la graduación y las guías de la máquina. Una serie de labios selladores protegen al cristal de la contaminación. Las aplicaciones típicas de los encoders lineales cubiertos son principalmente máquinas herramienta de corte o aquellas máquinas que estén ubicadas en un ambiente agresivo.

Los encoders lineales también consisten en una graduación de cristal y una unidad de lectura, pero los dos componentes están separados. La ventaja de no tener contacto es que son más fáciles de montar y se pueden lograr mayores velocidades de traslación dado que no hay fricción entre la unidad de lectura y la placa graduada. Los encoders lineales expuestos se pueden encontrar en máquinas de medición por coordenadas y máquinas de manejo de material.

Otra versión del arreglo entre placa graduada y unidad de lectura es aquella que utiliza una placa a base de metal en lugar de cristal. Con una placa de metal, las líneas se dibujan con material de alta reflectividad como oro que refleja la luz a la unidad de lectura y a las celdas fotovoltaicas. La ventaja de este tipo de placas graduadas es que pueden ser fabricadas en longitudes muy grandes, hasta 30 metros, para máquinas grandes. Las graduaciones de cristal están limitadas a unos 3 metros.

Se tienen que tomar en cuenta muchas consideraciones mecánicas para los encoders lineales. No es tan simple seleccionar un encoder basado en la longitud e instalarlo en una máquina. Se debe considerar velocidades de traslación, precisión y resolución requeridas, comportamiento térmico y guías para el montaje.¹⁹

3.3.2 Sensores mecánicos

Interruptores de contacto

El interruptor de contacto es un dispositivo mecánico que utiliza el contacto físico para detectar la presencia de un objeto. Un interruptor de contacto consiste en un cuerpo del interruptor y un cabezal operativo. El cuerpo del interruptor incluye los contactos eléctricos para energizar y desenergizar el circuito. El cabezal operativo incorpora algún tipo de brazo de palanca, conocido como actuador. El actuador gira cuando el objetivo aplica una fuerza en él. Este movimiento cambia el estado de los contactos en el cuerpo del interruptor.



Fig. 3.16 Interruptores de contacto

Actuadores

Hay muchos tipos de actuadores. Se utiliza un actuador de rueda en la mayoría de las aplicaciones con un brazo rotativo. Para movimiento transversal es mejor la utilización de un actuador de tipo trinche.

Cuando el objetivo entra en contacto con el actuador, hace girar al actuador de su posición libre hasta la posición de operación. En este punto los contactos eléctricos en el cuerpo del interruptor cambian de estado. Un resorte regresa el brazo del actuador y los contactos eléctricos a su posición libre cuando el actuador ya no está en contacto con el objetivo.

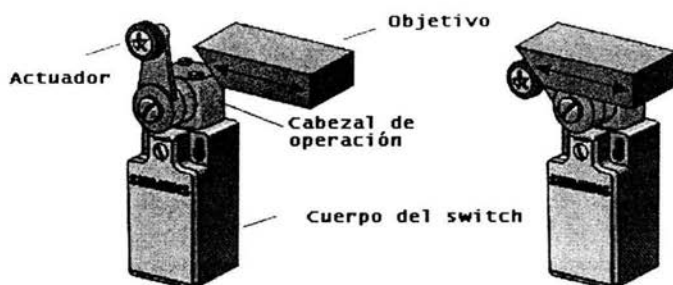


Fig. 3.17 Operación del actuador mecánico

En algunas aplicaciones es deseable que el actuador y los contactos se mantengan en su estado de operación después de que el actuador ya no esté sometido a contacto con el objetivo. El actuador y los contactos eléctricos regresan a la posición original sólo cuando de le aplica una fuerza al actuador en dirección opuesta. Esto se llama interruptor enclavado.

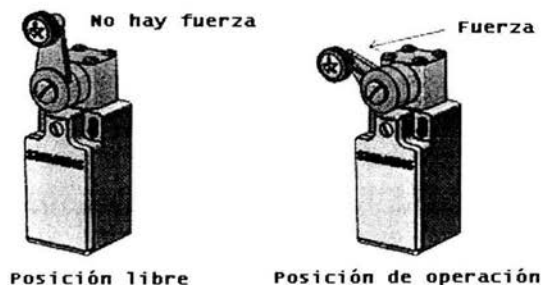


Fig. 3.18 Posición libre y de operación del interruptor

Las cargas no deben estar conectadas en el lado positivo de un interruptor. Esto podría significar riesgo de choque eléctrico.¹⁵

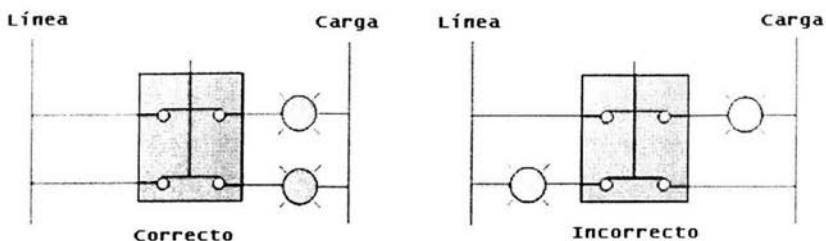


Fig. 3.19 Conexión correcta de una carga en un interruptor mecánico

3.3.3 Selección de sensores

Para la máquina son necesarios dos tipos de sensores: de fin de carrera y lectores de los encoders.

En el caso de los sensores para fin de carrera, se pudo haber optado por sensores ópticos, sin embargo, es más complicado realizar las conexión además de que son susceptibles a contaminación. Es por eso que se optó por sensores mecánicos de contacto en cada uno de los extremos de los ejes X, Y y Z. Estos sensores son normalmente abiertos, por lo que dejan pasar la señal en cuanto alguno de los carros de la máquina llega a su límite. Las señales de estos sensores están conectadas directamente a cada uno de los microcontroladores. En la figura 3.20 se muestra la localización de dichos sensores.

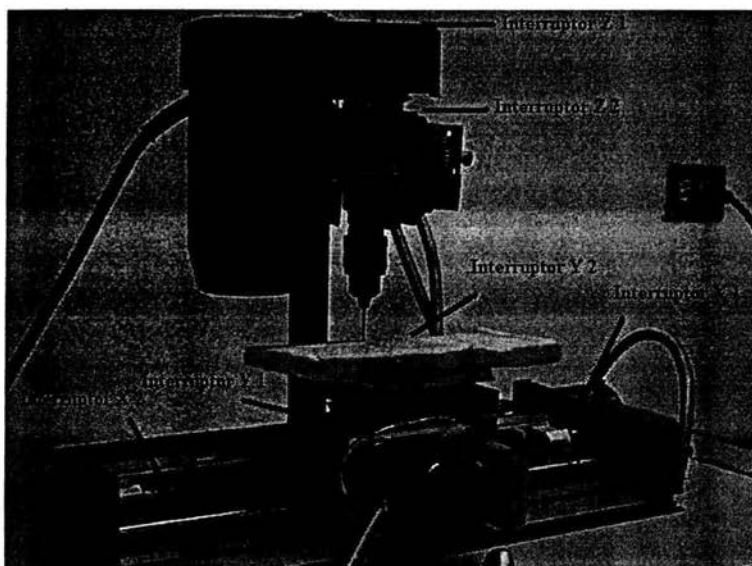


Fig. 3.20 Localización de interruptores mecánicos

En el caso de los sensores para los encoders lineales, éstos son necesariamente ópticos. Son sensores de ranura que cuentan por un lado con un emisor y en el otro un

receptor. Lo que éste sensor detecta son los cambios de luz en la tira graduada del encoder. Cada vez que la luz es bloqueada en el sensor por una de las líneas de la tira, se envía un pulso a los microcontroladores. En la figura 3.21 se muestra la localización de dichos sensores.

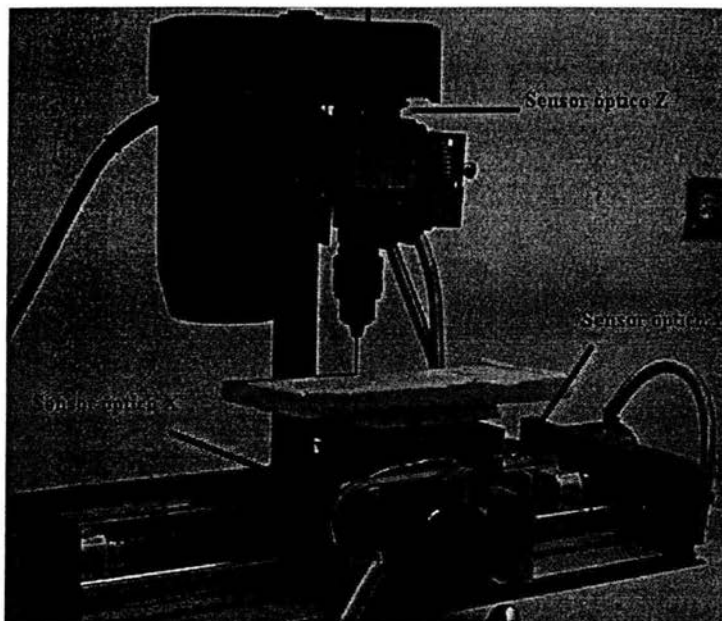


Fig. 3.21 Localización de sensores ópticos

Capítulo IV

Interfaz PC – μ Cs.

En el presente capítulo se explica el desarrollo del software “TORITO”, el cual presenta una interfase gráfica al usuario, así como el software desarrollado para controlar a los microcontroladores PIC16F877 y PIC16F73 y su comunicación entre dispositivos PC- μ Cs.

4.1 Interprete del Código Protel – CNC

Como se menciona dentro de los objetivos de esta tesis, uno de ellos es el automatizar un proceso de perforación de tarjetas de circuitos electrónicos. Se tomó el software Protel como herramienta principal para diseñar el circuito adecuado que permitiera generar el código correspondiente de control numérico. Este código permite una resolución máxima de milésimas de milímetro.

Este código presenta una extensión txt, que permite su fácil acceso por medio de cualquier bloc de notas.

El código generado presenta las características que se muestran a continuación:

```
M71
M48
T1F00S00C0.81
T2F00S00C3.18
T3F00S00C0.89
T4F00S00C0.70
T5F00S00C1.14
T6F00S00C0.76
T7F00S00C0.71
T8F00S00C1.25
T9F00S00C0.75
T10F00S00C3.00
%
T01
X0069Y010151
X007154
X007408
X007662
X007916
Y011675
X007662
X007408
X007154
X0145Y012725
```

X01425

T10
X01245Y013625
X01345
X01395
X01495
M30

Las dos primeras líneas de este código representan ordenes a la máquina, como coordenadas en milímetros (M71) y comienzo de programa (M48)

M71
M48

A continuación se muestra la información de número de herramienta, avance, velocidad de giro del husillo y diámetro de la herramienta correspondiente, T, F, S y C respectivamente.

Como ejemplo se tiene:

T1F00S00C0.81

T – Herramienta No. [1]
F – Sin Avance [00]
S – Velocidad del husillo [00]
C- Diámetro de herramienta [0.81] en [mm]

Los valores de F y S son cero, dado que se pueden emplear velocidades de avance y del husillo variables acordes a la rapidez que se desee en el proceso, cuidando que la velocidad del husillo no sea mayor a la de desboque.

Al termino de estas líneas viene un símbolo “ % “ que nos representa ya sea el final de herramientas o inicio de programa.

Como en toda máquina de control numérico se deben tomar en cuenta las direcciones de movimiento y posición. Es así como este código también presenta las siguientes características:

T01
X0069Y010151
X007154
X007408

T01 – Se utilizara la herramienta No. 1
X0069 – Movimiento en coordenada X
Y010151 – Movimiento en coordenada Y

Al momento de generar el código anterior en Protel, se requiere dar el número de dígitos a la izquierda del punto decimal y hacia la derecha de éste, por tanto, se tomo como base dar cuatro cifras a la izquierda del punto decimal y solo dos a la derecha del mismo. Por tal motivo la coordenada:

Y010151

Se escribiría correctamente como:

Y0101.51 [MM]

De igual modo corresponde a la coordenada X.

Estas líneas de coordenadas representan un registro o línea del programa, a diferencia de otra nomenclatura de programa de control numérico que antecede a cada registro una N, que indica el número de línea dentro del programa. El código utilizado en este proyecto omite dicha literal N, por lo cual solo coloca en cada registro o línea una palabra. Dicha palabra consta como, se ha visto, de una letra (dirección) y una combinación de cifras que indican el posicionamiento que debe realizarse. A su vez, el control lee los registros en orden ascendente.

La máxima capacidad que posee el programa para almacenar líneas o palabras es de 32,768. Gracias a la capacidad de memoria externa que se añadió, se pudo ampliar hasta 262,144 líneas de programa.

Al final del programa se cuenta con una instrucción que indica fin del programa. Esta instrucción es : M30.

4.2 Desarrollo de Software

4.2.1 Interfase Usuario

Explicado el código que genera Protel damos paso a la interpretación y codificación de este último.

El programa seleccionado para el desarrollo de la interfase de usuario fue el programa de aplicación Visual Basic 6.0, por su facilidad de programación y su excelente presentación gráfica.

Dentro del software se pretende dejar el código antes explicado en una forma usual, de manera que el usuario final comprenda totalmente el significado de cada línea del registro o línea dentro del programa. Se dio el formato adecuado dejando solo dos cifras después del punto decimal y colocando la información de las herramientas visibles al usuario como se muestra a continuación (Fig 4.1):



Fig 4.1 Codificación de Código Numérico

En los anexos presentados en este proyecto se muestra el código fuente que se desarrolló para la interpretación del código (ver anexo a)

Después de la interpretación correcta del código CNC, se facilita una herramienta de simulación, la cual permite observar los perforaciones a realizar. La figura muestra la simulación de una tarjeta (Fig 4.2).

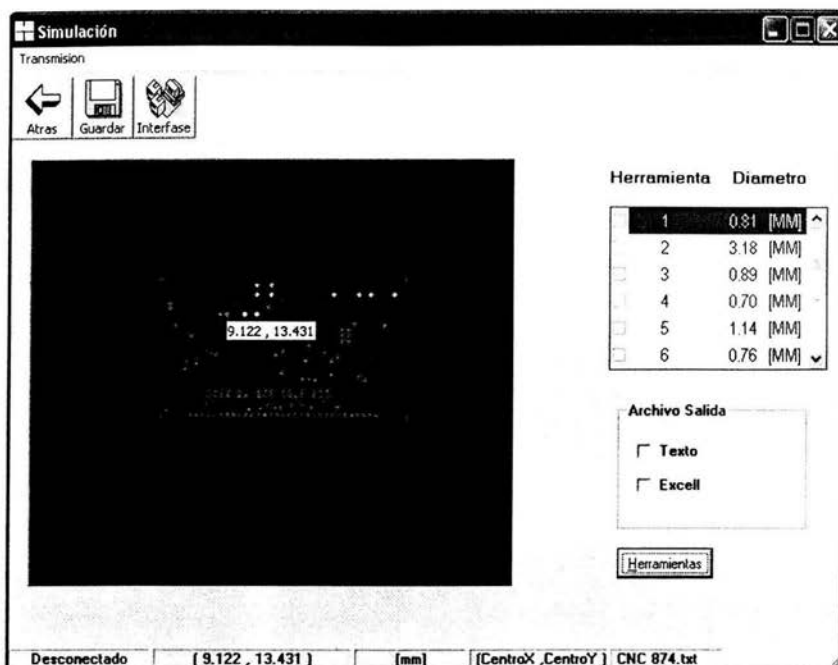


Fig 4.2 Simulación de tarjeta a perforar

Se asignó un color para cada herramienta, lo que indica que ese color corresponde a un proceso de perforado, y que otro color es un cambio de herramienta. Esto facilita el entendimiento de cómo opera el sistema de perforación de tarjetas.

En la pantalla en negro el cursor indica la posición exacta en donde se hará el barreno correspondiente a cada herramienta.

En la parte derecha de la figura se muestran las características de cada herramienta como número de herramienta y diámetro.

Ahora es necesario traducir el código codificado a un lenguaje apropiado, de manera que los microcontroladores puedan entender. Este lenguaje es convertido a lenguaje hexadecimal de la manera siguiente:

Teniendo en cuenta que el código esta en función de posición, tan solo se hará una división entre la resolución del encoder incremental, que para nuestro caso dicha resolución es de 0.15 [mm]. Esto tiene como limitante que nuestro mínimo movimiento puede ser de 0.15 [mm], es decir decimas de milímetro.

Como ejemplo se tiene la siguiente coordenada:

X8.13 Coordenada de posición geométrica

$$\frac{8.13}{0.15} = 54.2$$

Dado que el resultado de la división no fue un número sin decimales se cierra el número al valor entero próximo, por lo que queda:

X54 Número decimal

Al hacer la conversión a hexadecimal obtenemos

X36 Número hexadecimal

De este modo hemos transformado una coordenada de posición geométrica a una coordenada hexadecimal, la cual es enviada al microcontrolador principal, tema que se discutirá en los siguientes subcapítulos.

Los archivos que se generan al guardar el archivo codificado y transformarlo son respectivamente:

Codificado.txt
HexTnPic.txt

Un fragmento de los archivos se muestra a continuación:

Tarjeta11.txt			N10	X 59.0376	Y 48.7868
			N11	X 61.5376	Y 48.7868
N0	T1		N12	X 64.321	Y 46.2372
N1	X 84.8	Y 63.2542	N13	X 64.321	Y 52.2372
N2	X 103.7016	Y 62.0348	N14	X 61.7636	Y 55.8718
N3	X 103.7017	Y 64.5348	N15	X 51.1144	Y 74.0292
N4	X 78.8	Y 63.2542	N16	X 47.0596	Y 76.9432
N5	X 61.7636	Y 64.5078	N17	X 44.5596	Y 76.9432
N6	X 55.7636	Y 64.5078	N18	X 41.2085	Y 76.9812
N7	X 55.7636	Y 55.8718	N19	X 41.2085	Y 70.9812
N8	X 55.685	Y 52.2372	N20	M30	
N9	X 55.685	Y 46.2372			

HexTnPic.txt	x0000y0004
	x1002y0002
T1	x1007y000C
x0039y002A	x1003y0002
x000Dy1001	x1002y0000
x0000y0002	x1002y0000
x1011y1001	x1000y1004
x100By0001	x1000y0007
x1004y0000	x1000y0002
x1000y1006	x1000y0003
x1000y1002	x0002y0000
x1000y1004	M30
x0002y0002	
x0002y0000	
x0002y1002	

El archivo HexTnPic.txt contiene la información correspondiente a los movimientos en X y Y pero en lenguaje fácil de entender por el microcontrolador.

Un tercer archivo con nombre LonWord.txt fue generado. Dicho archivo contiene la información de longitud de palabra necesario para cubrir el desplazamiento de acuerdo a la transformación de la coordenada geométrica. En este último observamos que la mayor longitud de palabra para cada instrucción era de 3 cifras; es decir:

x0FFF

Por tal motivo se tuvo que concatenar dos memorias de ocho bits por palabra dado que el alcance de hardware del μ C Maestro es de solo ocho pines por puerto, direccionar una palabra de 12 bits es complicado y no se cuenta con los suficientes pines para lograr esta tarea.

La máquina perforadora es capaz de horadar circuitos impresos hasta de 300 mm x 50 mm, teniendo como limitante la geometría de la máquina en el eje Y. Tal razón impide que se extienda aún más este límite.

La nueva sintaxis tiene la siguiente información:

x a b c d	y a b c d
x	dirección
primera cifra	a
	a = 0 – sentido positivo
	a = 1 – sentido negativo

Se debe tener en cuenta que son datos hexadecimales. Es decir podemos tener 16 posibles banderas de señalización, lo cual permitiría realizar hasta 16 posibilidades de

instrucciones diferentes. Tan solo una de ellas está siendo utilizada y nos indica el sentido de giro del motor.

Las siguientes tres cifras:

segunda	b
tercera	c
cuarta	d

nos indican solo las cantidades de movimiento. Esto implica que si hacemos ahora la conversión de datos hexadecimales a decimales y multiplicamos por la resolución del encoder, para una palabra de 3 datos tendríamos:

x0FFF	Número hexadecimal
x04095	Número decimal
x0614.25	Coordenada de posición Geométrica

Es decir 614.25 [mm] como máximo desplazamiento en cualquiera de los dos ejes X , Y. Sin embargo por la restricción geométrica no es posible.

Estos archivos son almacenados en la PC, en el directorio C:\>, para su fácil localización.

Realizada la manipulación de los códigos tanto en forma usual como a nivel hexadecimal, se procede al proceso de manufactura o perforado. Continuando con la interfase usuario se tiene la pantalla que se muestra a continuación (Fig 4.3):



Fig 4.3 Conexión a la máquina herramienta

A través de esta pantalla y sus opciones podemos comunicarnos con el entorno de la máquina. Esto mediante una comunicación serial, la cual será tratada en el siguiente subcapítulo. Aquí tenemos la opción de manipular la máquina en forma manual (Fig 4.4),



Fig 4.4 Modo Manual de operación

transmitir el programa realizado (Fig 4.5)



Fig 4.5 Modo de transmisión de datos

o comenzar de forma automática el proceso (Fig 4.6)



Fig 4.6 Modo Automático

Para manipular la máquina en forma manual se accede a una pantalla donde vemos los controles de los tres ejes con los que cuenta la máquina y el encendido y apagado del motor del husillo (Fig 4.7).

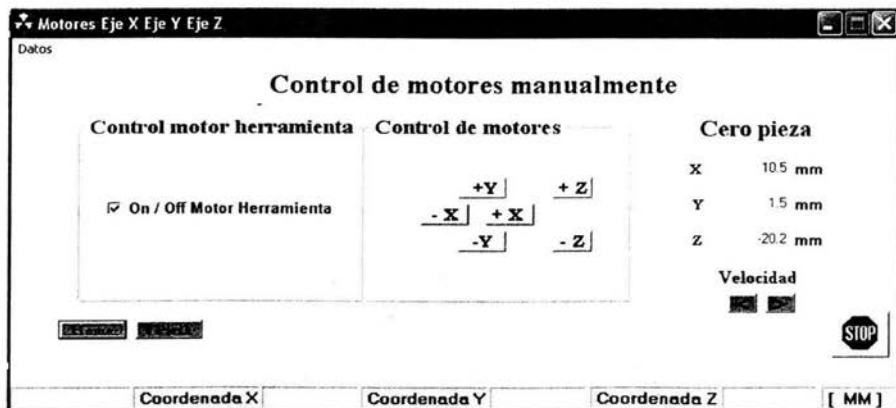


Fig 4.7 Pantalla de control manual

También se tiene un regulador de velocidad por software, dos referencias (Home y Cero pieza) y un botón de paro general.

En modo transmisión cargamos el programa realizado en Protel (Fig 4.8).

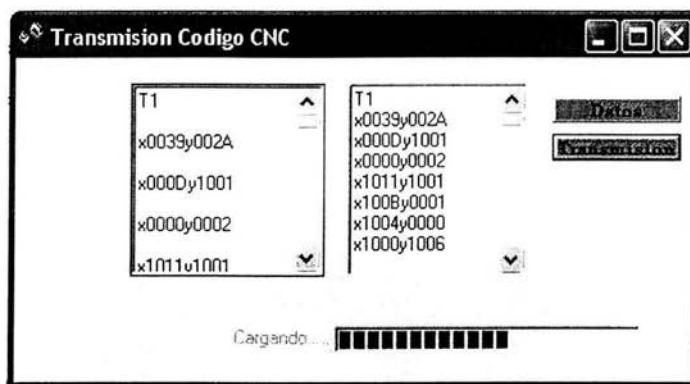


Fig 4.8. Transmitiendo programa

Y por ultimo modo automático se comienza el proceso de manera automática y solo se detiene cuando debe hacerse un cambio de herramienta (Fig 4.9)

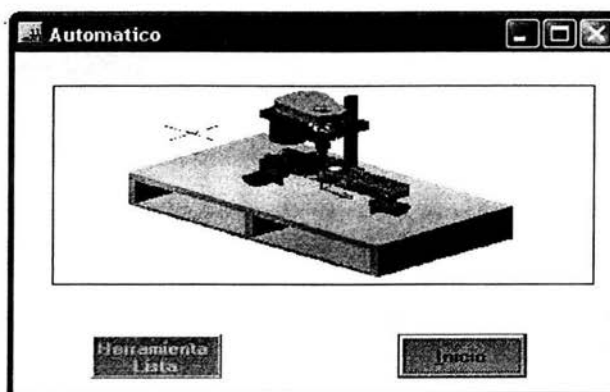


Fig 4.9. Modo automático de operación

Cuando hay cambios de herramienta el propio software manda un mensaje donde se alerta de la petición de cambio de herramienta. Una vez realizado esto se continua el proceso oprimiendo el botón de "Herramienta Lista".

Los códigos fuentes correspondientes a esta parte de la interfase visual son presentados en los anexos (ver anexo a).

4.2.2 Software μ Cs

Como se explicó en el capítulo anterior se usaron dos diferentes tipos de μ Cs, PIC16F877 y PIC1673. Esto como consecuencia de usar encoders incrementales de posición para cada eje.

El software que se utilizó para la programación de los microcontroladores es MPLAB 6.50, debido a que incorpora simuladores. Esto permitió saber si los programas realizados funcionaban correctamente. La figura siguiente (Fig 4.10) muestra el panel frontal de dicho software.

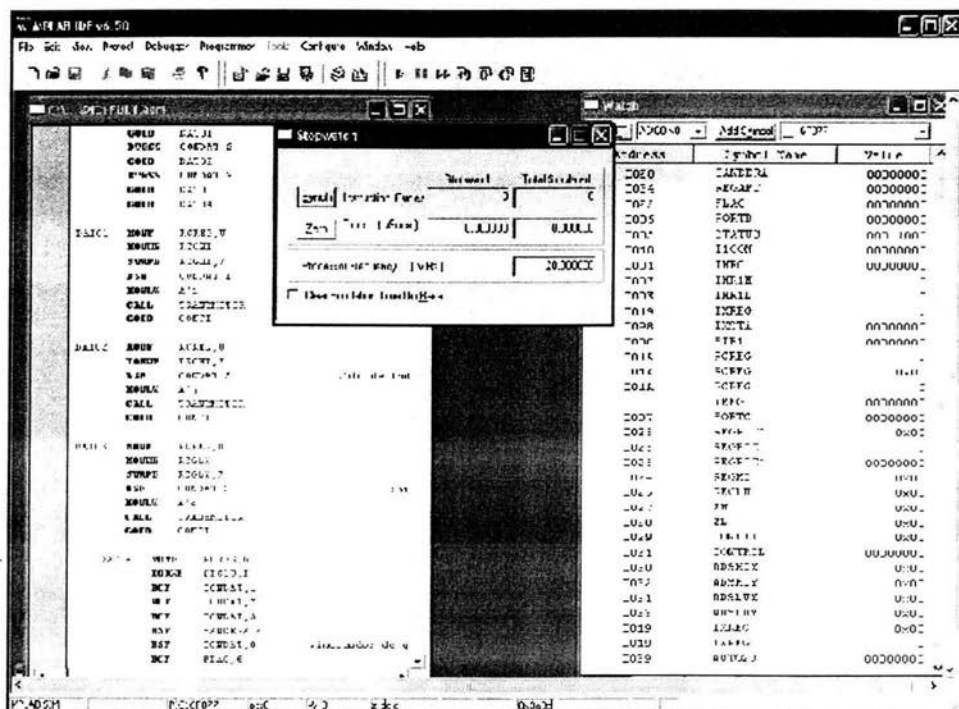


Fig 4.10 Software utilizado para crear los programas de los microcontroladores

La programación se desarrollo a nivel código ensamblador. Estos códigos se muestran en el anexo (b) así como sus diagrama de flujo.

Se utilizó una topología para la comunicación PC - μ Cs, como se muestra (Fig 4.11):

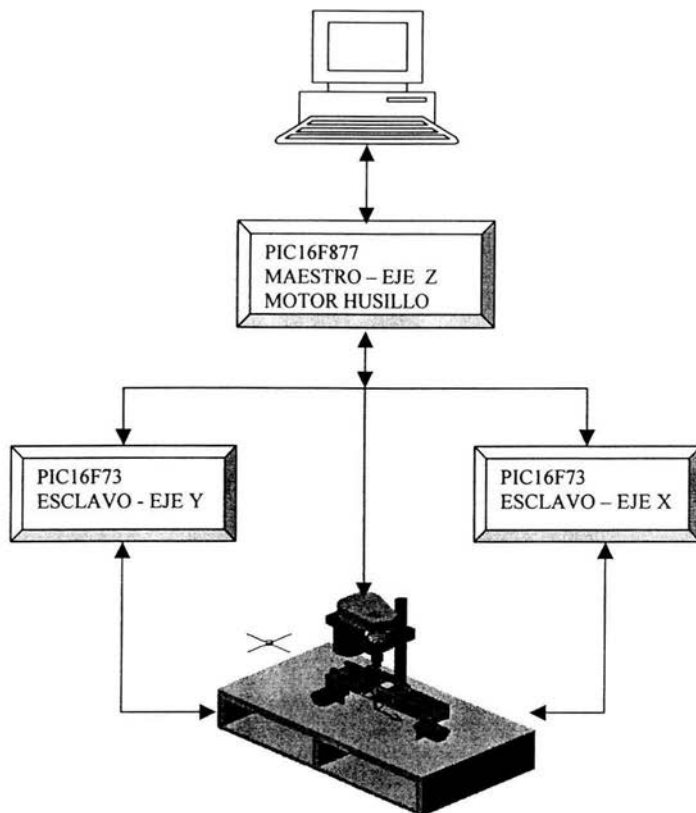


Fig 4.11 Topología de comunicación entre dispositivos

Se aprecia que la PC tiene la tarea de mandar las instrucciones, el μ C Maestro encargado de traducir estas instrucciones y reenviarlas a los μ Cs esclavos. Hecho esto los μ Cs esclavos mandan la ejecución hacia la máquina.

De acuerdo al diagrama anterior el μ C central tiene la mayor programación debido a que tiene que interpretar cada una de las instrucciones que manda la PC y decodificarlas. Cada μ C esclavo solo ejecuta la instrucción que se le envía, por lo tanto su código es menos robusto que el primero. En el anexo correspondiente observamos los diagramas de flujo para cada software de cada μ C.

4.3 Protocolos de Comunicación

4.3.1 Comunicación Serial Síncrona y Asíncrona

Unidad Universal de Transmisión Recepción Síncrona y Asíncrona (USART)

La comunicación entre los diferentes dispositivos electrónicos contenidos en el proyecto es fundamental. Por tal motivo damos un subcapítulo a la comprensión de este.

Como primer punto mencionaremos el tipo de comunicación que se tiene para enlazar la PC con el μ C maestro. La tarjeta de desarrollo PIC16F877, diseñada y construida en el laboratorio de Mecatrónica, tiene una comunicación serial RS - 232. Así que se tomo como primer prototipo de comunicación este medio.

¿Pero, cómo funciona el Protocolo RS-232? El RS-232 es un protocolo de comunicación serial, principalmente utilizado por las computadoras. A diferencia de la comunicación paralela, la información se envía secuencialmente, es decir un bit al tiempo, utilizando menos líneas de transmisión que en esta que transmite la información de manera paralela. Es decir, varios bits al mismo tiempo. Esta característica le otorga la capacidad de transmitir a mayores distancias pero con menor velocidad.

Este protocolo utiliza voltajes de +/- 15V. Un uno lógico es un voltaje entre -5V y -15V en el transmisor y en el receptor entre -3V y -25V. Un cero lógico entre 5V y 15V en el transmisor y entre 3V y 25V en el receptor. Estos voltajes se utilizan en vez de los valores habituales de CMOS y TTL, para evitar que se pierda la información por las inductancias producidas por el medio. Existen dos tipos de comunicación serial, la síncrona y la asíncrona. La primera emplea un pin más para enviar un reloj que los dos nodos de la comunicación utilizarán para sincronizar la lectura / escritura de los datos. En la asíncrona se utiliza un periodo establecido previamente para escribir / leer los datos (Fig 4.12).

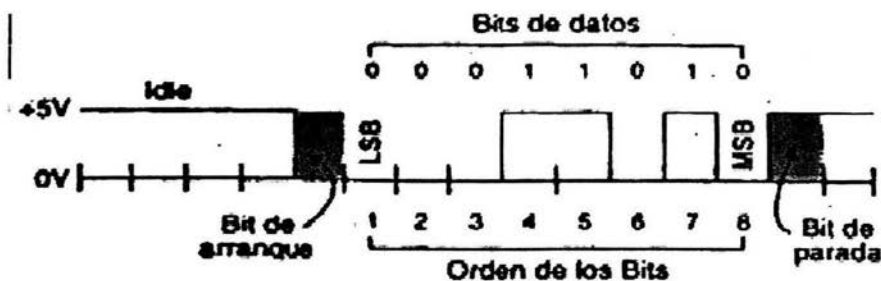


Fig 4.12 Condiciones de uso de comunicación

En la transmisión asíncrona, la línea de transmisión esta normalmente en uno. Para transmitir una palabra se envía primero una señal de cero (bit de inicio) para señalar que se va a empezar a transmitir y se mantiene así durante el periodo de tiempo establecido para

cada bit. Luego se envían de manera secuencial a intervalos marcados por el periodo. Todos los bits de los datos a enviar de el menos significativo (LSB) al más significativo (MSB). Luego para terminar la transmisión se envía opcionalmente un bit para revisar la fiabilidad de los datos, el de paridad y por último los bits de finalización que deben ser acordados antes de transmitir y quedar en uno.^{22,23} Los niveles de voltaje que se usan en el estándar RS-232, el equivalente en niveles lógicos (TTL) y la terminología correspondiente, se muestran en la tabla 7.

Voltaje >	Lógico	Control	Terminología
+3[v] a +25[v]	0	Activo	Espacio
-3[v] a -25[v]	1	Inactivo	Marca

Tabla 7 Terminología de niveles de voltaje

De la tabla se observa que un "1" lógico, equivale a un voltaje negativo (-3v a -25 v), y un "0" lógico, equivale a un voltaje positivo (+3v a +25v). Un voltaje que está entre +3v y -3v se considera como indeterminado. Cuando la línea se mantiene en "1" (Marca), está en estado de reposo. Cuando la línea está en "0" (Espacio) se toma como activa.

La velocidad a la que se envían datos en forma serial a través de una línea de comunicación, se denomina velocidad en baudios. La velocidad de baudios es expresada en unidades de bits por segundo. Una conexión RS-232 con velocidad de 19600 baudios tiene la capacidad de enviar 19600 bits de datos en 1 segundo.

Si se pueden enviar 19600 bits en un segundo, como máximo, el inverso de 19600 dará como resultado el tiempo de bit (período de un bit).

Si un receptor y un transmisor se conectan a 19600 baudios, el transmisor enviará bits de datos cada 51µs, y el receptor tomará lectura de los bits de datos cada 51 µs.

En este proyecto se emplea la comunicación serial asíncrona, en la cual la transmisión está inactiva en el estado de Marca (1 lógico). La transmisión de cada carácter en una línea de comunicación asíncrona va precedida de un bit de inicio. El bit de inicio es un espacio (0 lógico) con duración igual al tiempo de bit. En el receptor, cuando la línea cambia de marca a espacio, se interpreta como el bit de inicio. Después de este bit se reciben los bits de datos con un tiempo de bit igual a 51 µs.

Después de que el último bit de datos ha sido enviado, el transmisor pasa al nivel de marca durante un tiempo de bit. Este bit es llamado bit de paro. El bit de paro indica que todos los bits de datos han sido enviados y la transmisión del carácter se ha completado. Si el receptor detecta un bit de inicio y después los bits de datos pero no detecta el nivel de marca al final, esto indica un error en la transmisión.²³

En la figura se muestran los niveles de voltaje con el estándar RS-232 cuando se transmite un "0" ASCII. Observe que después del bit de inicio se envía el bit menos significativo (LSB) del dato (Fig 4.13).

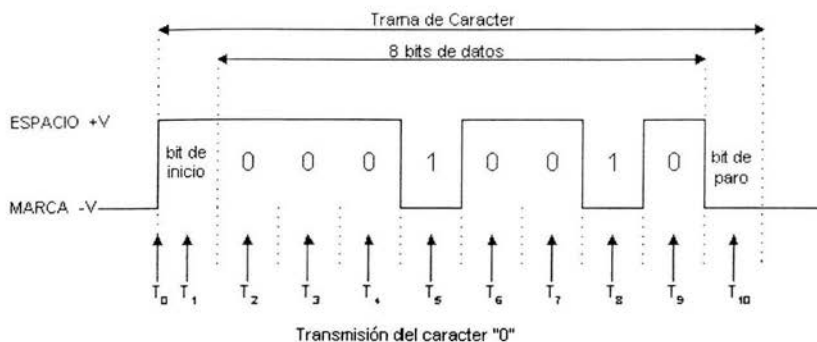


Fig 4.13 Los niveles de voltaje con el estándar RS-232

T₀: La transición del nivel marca a espacio indica al receptor que la transmisión de un nuevo carácter ha comenzado.

T₁: El receptor espera la mitad del tiempo de bit (a 19600 baudios este tiempo es 20.5 μ s) y toma otra muestra de la línea. Si la línea sigue en el nivel espacio, el bit de inicio es válido. En otro caso, si la línea de recepción regresa al nivel de marca, se trata de un bit de inicio falso que se atribuye a una línea ruidosa.

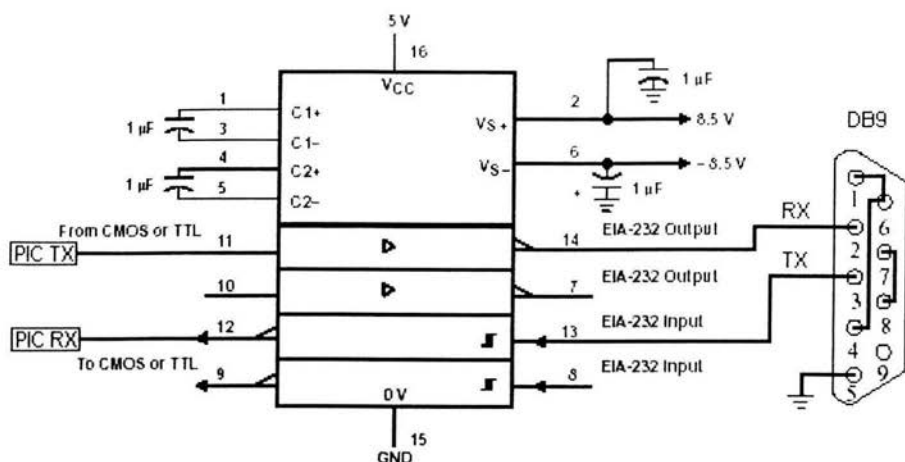
T₂: El receptor espera un tiempo de bit y hace un muestreo la línea de entrada, el nivel será el correspondiente al bit menos significativo.

T₃-T₉: Se realizan 7 muestreos más, cada 51 μ s (para 19600 baudios), y se obtienen los niveles correspondientes a los bits de datos restantes. Después de T₉ los 8 bits de datos han sido capturados.

T₁₀: Se hace un muestreo del bit de paro. Observe que la línea regresa al nivel de marca.

MAX232

El circuito integrado MAX232 cambia los niveles TTL a los del estándar RS-232 cuando se hace una transmisión y cambia los niveles RS-232 a TTL cuando se tiene una recepción. El circuito típico se muestra en la siguiente figura (Fig 4.14):



Circuito básico con el MAX232
Fig. 4.14 MAX232

Aún cuando se cuenta con una comunicación asíncrona en donde podemos transmitir y recibir información (FullDuplex), la comunicación del μC maestro y los μC esclavos se hizo también en forma asíncrona. Sin embargo al realizar comunicación entre μC central hacia los demás, observamos que en ocasiones no se realizaba la acción indicada que se le daba a los μC esclavos. Por lo tanto se siguió una manera convencional de comunicación para evitar este tipo de problemas. A continuación se explica brevemente el funcionamiento.

El mecanismo más simple utilizado en un puerto serial de una PC es de tipo unidireccional y es el que analizaremos en primer lugar. Distinguimos dos elementos: la *parte transmisora* y la *parte receptora*. La parte transmisora coloca la información en las *líneas de datos* e informa a la parte receptora que la información (los datos) están disponibles; entonces la parte receptora lee la información en las líneas de datos e informa a la parte transmisora que ha tomado la información (los datos). Observe que ambas partes sincronizan su respectivo acceso a las líneas de datos. La parte receptora no leerá las líneas de datos hasta que la parte transmisora se lo indique en tanto que la parte transmisora no colocará nueva información en las líneas de datos hasta que la parte receptora remueva la información y le indique a la parte transmisora que ya ha tomado los datos. A ésta coordinación de operaciones se le llama *acuerdo ó entendimiento* (**handshaking**).

De este modo evitamos colisiones de información o pérdida de esta.

4.3.2 Protocolo I²C

Muchos de los equipos electrónicos incluyen circuitos integrados con el bus I²C, como por ejemplo, las memorias 24Cxx, los procesadores de señal en televisores (LA7610, TA1223, DTC810,...), codificadores de video de reproductores de DVD (SAA 7128, TC 90A32F,...), preamplificadores de video en monitores (KB 2502), etc.

Lo que sigue es una breve descripción del bus I²C y algunas recomendaciones para diagnosticar fallas en el bus.

Las características más salientes del bus I²C son:

- Se necesitan solamente dos líneas, la de datos (SDA) y la de reloj (SCL).
- Cada dispositivo conectado al bus tiene un código de dirección seleccionable mediante software. Habiendo permanentemente una relación maestro / esclavo entre el micro y los dispositivos conectados.
- El bus permite la conexión de varios maestros, ya que incluye un detector de colisiones.
- El protocolo de transferencia de datos y direcciones posibilita diseñar sistemas completamente definidos por software.
- Los datos y direcciones se transmiten con palabras de 8 bits.

Funcionamiento del bus I²C

Como se menciona, las líneas SDA y SCL transportan información entre los dispositivos conectados al bus. Cada dispositivo es reconocido por su código (dirección) y puede operar como transmisor o receptor de datos.

Además, cada dispositivo puede ser considerado como maestro o esclavo. El maestro es el dispositivo que inicia la transferencia en el bus y genera la señal de reloj, el esclavo es el dispositivo direccionado.

Las líneas SDA (serial data) y SCL (serial clock) son bidireccionales, conectadas al positivo de la alimentación a través de las resistencias de pull-up. Cuando el bus está libre, ambas líneas están en nivel alto (Fig 4.15).

La transmisión bidireccional serie (8-bits) de datos puede realizarse a 100Kbits/s en el modo estándar o 400 Kbits/s en el modo rápido.

La cantidad de dispositivos que se pueden conectar al bus está limitada, solamente, por la máxima capacidad permitida: 400 pF.

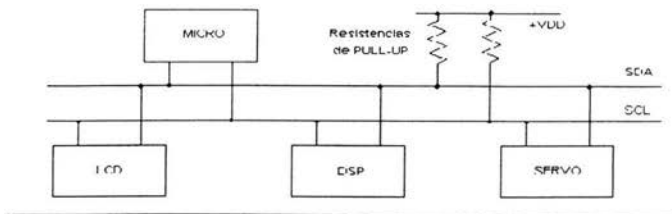


Fig. 4.15. Conexión de dispositivos via I2C

Este protocolo de comunicación no fue empleado debido a que existen conflictos de tener comunicación USART e I²C en el mismo programa de control de los microcontroladores, por lo cual se optó por comunicación USART.

4.3.3 Comunicación en Paralelo

En un esquema de transmisión de datos en **paralelo** un dispositivo envía datos a otro a una tasa de n número de bits a través de n número de cables a un tiempo. Sería fácil pensar que un sistema en *paralelo* es n veces más rápido que un sistema en *serie*, sin embargo esto no se cumple. Básicamente el impedimento principal es el tipo de cable que se utiliza para interconectar los equipos. Si bien un sistema de comunicación en paralelo puede utilizar cualquier número de cables para transmitir datos, la mayoría de los sistemas paralelos utilizan ocho líneas de datos para transmitir un byte a la vez.

Este protocolo de comunicación fue empleada para grabar el programa que se envía serialmente bit a bit. Una vez que el μC maestro interpreta la información, direcciona la información recibida y transmite los datos paralelamente a memorias EEPROM (2864).

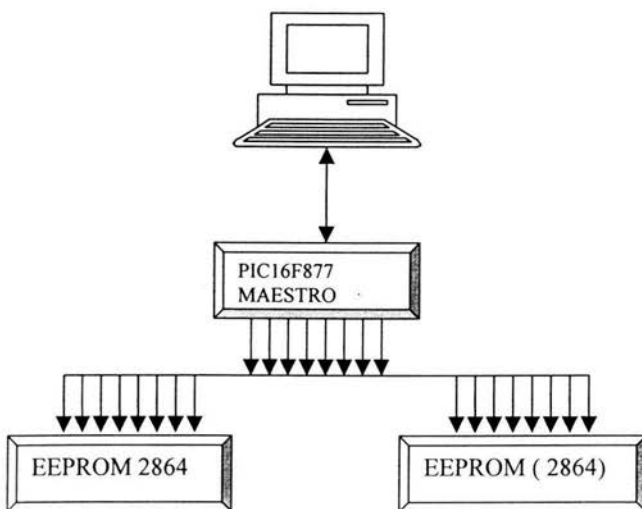


Fig. 4.16 Comunicación en paralelo de microcontrolador maestro- memoria externa

En la figura anterior puede apreciarse que se tienen dos memorias dispuestas paralelamente. Como se explicó en el capítulo anterior fue necesario concatenar dos memorias de ocho bits cada una, resultando en un mayor desplazamiento. Es decir, de acuerdo al tamaño de palabra se tiene mayor o menor capacidad de movimiento.

Conclusiones

Hemos cumplido con los objetivos y dejamos un precedente en el laboratorio de Mecatrónica que puede servir para futuros proyectos. Además existen partes del control electrónico del proyecto que pueden ser utilizadas por separado como herramientas. Por ejemplo, un programador de memorias o una etapa de potencia para motores de pasos.

La máquina tiene las siguientes características finales:

La máquina perforadora tiene la capacidad de hacer una perforación cada 0.15 [mm] (5.9 milésimas de pulgada) en ambas direcciones “X” y “Y”.

Si bien la resolución no fue la planteada al inicio del proyecto, esta resolución esta en función de la resolución del encoder lineal que se tiene montado en los tres ejes cartesianos de la máquina. Se podría incrementar la resolución con un encoder de mayor resolución, dado que el hardware y el software lo permitirían.

La máquina diseñada es capaz de perforar circuitos impresos hasta de 50 x 300 mm. Sin embargo en la sección transversal se tiene muy poca carrera de trabajo, debido a las restricciones geométricas de la máquina, no obstante esto puede ampliarse haciendo un nuevo diseño de la máquina.

Con respecto a la capacidad de memoria para almacenar el programa transmitido (32 kbytes), puede ampliarse hasta 256 kbytes, si se manejan memorias de 64 kbytes.

Si se siguen utilizando memorias paralelas (EERPOM) podríamos aún expandir mayor la capacidad de memoria, dado que se tiene en el proyecto desarrollado 13 bits de direccionamiento, pero puede incrementarse hasta 16 bits de direccionamiento, es decir, se tendría el 2 Megabit de capacidad.

El uso de tres μ Cs para controlar cada eje nos permitió tener el control de cada actuador, así como de su posicionamiento, sin embargo como se desarrollo en el presente trabajo no se logro el funcionamiento simultaneo de los ejes X y Y, y esto en gran parte al tipo de comunicación que se empleo.

El tipo de comunicación que se utilizo, comunicación universal asincrona, facilito la comunicación, entre dispositivos, no obstante creo conflictos al momento de enviar simultáneamente información, creando perdida de esta, esto se soluciono haciendo un “handshaking”, sin embargo es viable otra solución para lograr dicho objetivo, solo se necesita cambiar el protocolo de comunicación, como por ejemplo a un I²C, que permite saber si las líneas de comunicación están ocupadas y que garantiza que no se perderán instrucciones tanto transmitidas como enviadas.

El tener un posicionamiento de lazo cerrado, nos permite garantizar que el posicionamiento de las bancadas será el más optimo.

A lo largo del proyecto fuimos dándonos cuenta de que la construcción y automatización de una máquina convencional a una máquina CNC no es una tarea fácil. A medida que avanzábamos iban surgiendo problemas mecánicos, electrónicos y de programación. Ahora que hemos terminado, tenemos una noción mucho más clara de todo lo necesario para la construcción de un taladro CNC.

Desde el punto de vista mecánico nos dimos cuenta de que adaptar una máquina convencional a control numérico no es necesariamente la mejor opción. Por ejemplo, en este proyecto se tiene una limitante de carrera en el eje Y debido a la construcción de la máquina. Para aumentar el área de trabajo, podría ser más fácil construir una máquina completamente nueva que seguir realizando adaptaciones a la existente.

En cuanto a la electrónica, varias veces fue necesario agregar componentes de protección o elementos para corregir problemas. Uno de los problemas más grandes que se tuvieron fue el cómo generar diferentes voltajes para los diferentes motores y que se tuviera la capacidad de corriente solicitada. Varias veces tuvimos que rediseñar la parte de alimentación.

Se puede fabricar una máquina para realizar perforaciones en circuitos impresos de una manera sencilla y barata. Con la elección correcta de motores y elementos mecánicos, que es la parte más cara, se puede armar una máquina sencilla con tres ejes. La parte electrónica y de software ya se encuentra realizada y su costo es mucho menor.

De realizarse una inversión y teniendo personas interesadas en el proyecto, podría iniciarse la fabricación de máquinas con éste propósito. Tan sólo habría que hacerle ciertas adaptaciones ligeras al proyecto presentado en ésta tesis.

Además que se abre la posibilidad de hacer el cambio de taladro automático a máquina de fresado automático, considerando cambiar los motores con un torque adecuado y cambiando los motores de paso por motores de corriente directa ya que el software y hardware son flexibles.

Aún siendo taladro automático puede hacerse este cambio de motores y tener mayor velocidad en el proceso de barrenado dado que como se menciona anteriormente el software y hardware así lo permiten.

Por último cabe resaltar que no se logro la precisión deseada debido a problemas de vibración en la máquina. Los encoders utilizados son muy susceptibles a vibración por lo que no siempre se lleva a la posición deseada. Para resolver este problema se podrían utilizar encoders de metal pero tendrían que cambiarse los sensores a unos de tipo retroreflectivos.

Anexo A

Código fuente del programa Torito

Subrutina para cambio de herramienta

```
Private Sub CmmBCambioH_Click()  
Baudio PuertoComm Output = "1"  
Baudio Retardo  
Baudio PuertoComm Output = "g"  
End Sub
```

Subrutina para modo automatico

```
Private Sub EmpezaR_Click()  
Baudio PuertoComm Output = "1"  
Baudio Retardo  
Baudio PuertoComm Output = "T"  
  
Baudio Retardo  
  
Baudio PuertoComm Output = "2"  
Baudio Retardo  
Baudio PuertoComm Output = "T"  
Baudio Retardo  
  
Baudio PuertoComm Output = "3"  
Baudio Retardo  
Baudio PuertoComm Output = "T"  
Baudio Retardo  
Baudio Timer1.Interval = 100
```

```
For i = 0 To 5  
MoToR.Command1.Item(i).Enabled = False  
Next i  
MoToR.Homsito.Enabled = False  
MoToR.CeroPieza.Enabled = False  
End Sub
```

Subrutina para inicializar variables

```
Private Sub Form_Load()  
CmmBCambioH.Enabled = False  
End Sub
```

```
Dim Buffer, Basura As String  
Dim baud, Serial, i, b As Integer  
Dim sendi As Variant  
Public iExtToHex As String
```

Subrutina de inicializar variables para modo conexión

```
Private Sub Form_Load()  
"cerrando Puerto  
bandera = False  
bandera2 = False  
bandera1 = 0  
Alo = False  
leer = False
```

```
If PuertoComm.PortOpen = True Then PuertoComm.PortOpen = False
```

```
MoToR.ControlBotones.Enabled = False
```

```
For i = 2 To 4  
Toolbar1.Buttons(i).Enabled = False  
Next  
flag4 = False  
flag5 = False  
End Sub  
Public Sub CargarDatosPuerto()  
Select Case baud
```

```
Case 1200:
```

```
PuertoComm.Settings = "19200,N,8,1"
```

```
Case 2400:
```

```
PuertoComm.Settings = "19200,N,8,1"
```

```
Case 9600:
```

```
PuertoComm.Settings = "19200,N,8,1"
```

```
Case 19200:
```

```
PuertoComm.Settings = "19200,N,8,1"  
End Select
```

```
End Sub  
Private Sub Form_Unload(Cancel As Integer)
```

```
kkancel)
```

```
End Sub
```

Subrutina para recepcion de datos

```
Sub Puertocomm_OnComm()  
"comenreceive = 2, este recepcion va ligada con RThreshold  
Static v As Integer  
If PuertoComm.CommEvent = comEvReceive Then  
Basura = PuertoComm.Input  
MoToR.TxtVisua.Text = MoToR.TxtVisua.Text & Basura  
If Alo = True Then " if para modo automatico  
If Basura = "T" Then  
MsgBox "Cambiar Herramienta ", vbCritical, herramienta  
AutoMaTico.CmmBCambioH.Enabled = True  
ElseIf Basura = ">" Then  
MsgBox "Proceso Terminado", vbExclamation, "Fin Programa"  
End If  
Else " condicion negativa si no esta en modo automatico  
BASURA1 = Val(Asc(Basura))
```

```
Do While (banderas <> True)  
If Basura = "o" Then "direccion de PC H'0x70'  
Barra1.Panels(1).Text = "Conectado"  
bandera = True  
End If  
Exit Do  
Loop
```

```
If leer = True Then "if para modo lectura
```

```
MoToR.InfPro.Text = MoToR.InfPro.Text & Basura  
If Basura = "@" Then " if para modo terminar lectura  
leer = False  
Else
```

```
End If "cierra el if de terminar lectura  
Else
```

```
Select Case Basura
```

```
Case "N"  
otra = True  
Case "Y"  
otra = True  
Case "s"  
otra = True  
Case "x"  
otra = True  
Case "y"  
otra = True  
Case "t"  
otra = True  
Case "w"  
traNdaTos.COMDMandar.Enabled = True
```

```
Case "r"  
flag3 = True  
Case "n"  
flag3 = True  
Case "T"  
"señalizadores de  
flag3 = True 'Home ok  
Case "J"  
flag4 = True  
Retardo  
Case "I"  
flag5 = True  
Retardo  
Case "S"  
ceroz = True  
Case "I"  
ceroz = True  
flag7 = True  
Case "#"  
ConOPieza.convertir  
MoToR.IblZ.Caption = conv * 0.15  
ceroz = False  
flag6 = True  
convertir = ""  
Retardo  
Case "X"  
convertir = Mid(convertir, 2, 2)  
ConOPieza.convertir  
MoToR.IblX.Caption = conv * 0.15
```

```

ceroz = False
flag7 = True
convertir = ""
Basura = ""
Retardo
Case "Y"
convertir = Mid(convertir, 2, 2)
ConDPieza convertir
MoToR libY Caption = conv * 0.15
ceroz = False
convertir = ""
Basura = ""
flag6 = False
flag7 = False
Retardo
Case "F"
otra = False
Case ""
MsgBox "Limite Superior Z", vbExclamation, "Peligro"
Case "-"
MsgBox "Limite Inferior Z", vbExclamation, "Peligro"
Case "T"
AutoMaTico CmmBCambioH.Enabled = True
Case Else

End Select
End If "cierra el if de modo lectura

If ceroz = True Then ' if de transformar cero pieza
convertir = convertir & Basura
DoEvents
Else

End If "cierra el if de cero pieza
End If "cierra el if de automatico
End If "cierra el if de command events

' nota: ' = linea que no sirve
' nota: " = linea inactivas que se estan depurando
End Sub

```

```
Private Sub Timer1_Timer()
```

```
bandera01 = True
bandera02 = True
```

```
Timer1.Enabled = False
```

```
End Sub
```

Subrutina de Menus para interfase usuario

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
```

```
'Static bandera1 As Integer
Select Case Button.Key
```

```
Case "conexion"
ButConectar
```

```
Case "manual"
BotManual
```

```
Case "TranDatos"
BotTransmitir
Barra1.Panels(2).Text = "Transmission"
traNdaTos Show
MoToR.ControlBotones.Enabled = True
```

```
Case "automatico"
Ato = True
Barra1.Panels(2).Text = "Automatico"
AutoMaTico Show
```

```
End Select
End Sub
```

Subrutina de conexión

```
Public Sub ConECT()
'conectando al puerto escogido
PuertoComm.CommPort = Serial
'abriendo puerto
PuertoComm.PortOpen = True
Buffer = "O"
PuertoComm.Output = Buffer
'seleccionando dos caracter para recibir

```

```
PuertoComm.InputLen = 1
'recibe 1 caracter antes de irse a la interrupcion
PuertoComm.RThreshold = 1
End Sub
```

```
Public Sub ParAmeTros()
Indicador = 0
For Indicador = 0 To 3
If OptComm[Indicador].Value = True Then
Serial = Indicador + 1
Exit For
End If
Next Indicador
```

```
If OptBau1 = True Then baud = 1200
If OptBau2 = True Then baud = 2400
If OptBau3 = True Then baud = 9600
If OptBau4 = True Then baud = 19200
CargarDatosPuerto
```

```
End Sub
```

```
Public Sub Temporizador()
Do While (bandera02 = False)
DoEvents
Loop
End Sub
```

```
Public Sub ButConectar()
```

```
Static bandera1 As Integer
```

```
If bandera1 <> 1 Then
```

```
ParAmeTros
ConECT
bandera1 = 1
Timer1.Enabled = True
Do While (bandera01 = False)
DoEvents
Loop
```

```
If bandera01 = True Then
For i = 2 To 4
Toolbar1.Buttons(i).Enabled = True
Next
'Toolbar1.Buttons(1).Value = tbrPressed 'boton presionado
'Toolbar1.Buttons(1).Enabled = false
'Toolbar1.Buttons(1).Caption = "Desconectar"
'Barra1.Panels(1).Text = "Conectado"
MoToR.Cls
Else
MsgBox "No hubo conexion"
End If

```

```
Else
```

```
If PuertoComm.PortOpen = False Then
Barra1.Panels(1).Text = "Desconectado"
Else
kkancel
Baudio.Temporizador
DoEvents
'If Basura = "k" Then
Barra1.Panels(1).Text = "Desconectado"
PuertoComm.PortOpen = False
For i = 2 To 4
Toolbar1.Buttons(i).Enabled = False
Next
'Toolbar1.Buttons(1).Caption = "Conectar"
Else
MsgBox "No responde la conexion"
End If
bandera = False
bandera1 = 0
End If
End If
End Sub
```

Subrutina para modo manual

```
Public Sub BotManual()
```

```
PuertoComm.Output = "I"
PuertoComm.Output = "H"
Retardo
```

```
If Basura = "h" Then 'Or Basura = "o" Then
```

```
PuertoComm.Output = "3"
PuertoComm.Output = "H"
```

```
Do While (flag4 <> True)
DoEvents
Loop
```

```

Retardo
PuertoComm Output = "2"
PuertoComm Output = "H"

Do While (flag5 <> True)
DoEvents
Loop

If flag4 = True And flag5 = True Then

Barra1 Panels(2).Text = "Manual"
MoToR BarraXYZ.Panels(1).Text = "Manual"
MoToR Show
MoToR ControlBotones.Enabled = True
Else
MsgBox "No responde modo manual"
End If
End If
End Sub

Public Sub BotTransmitir()
Dim send As String

PuertoComm Output = "1"
PuertoComm Output = "W"

```

Subrutina para cancelar movimientos y proceso

```

Public Sub kcancel()

PuertoComm Output = "2" 'direccion 0x31 PIC1
PuertoComm Output = "K"
Retardo
PuertoComm Output = "3"
PuertoComm Output = "K"
Retardo
PuertoComm Output = "1"
PuertoComm Output = "K"

End Sub

Public Sub Retardo()
bandera02 = False
Audio.Timer1.Enabled = True
Audio.Temporizador

End Sub

Public Sub Con0Pieza(obtener)
If Len(obtener) = 1 Then
conv = Val(Asc(Mid(obtener, 1, 1)))
Else
conv = Val(Asc(Mid(obtener, 1, 1))) * 256 + Val(Asc(Mid(obtener, 2, 1)))
End If
End Sub

Private Sub Form_Load()
TmInicio.Enabled = True
TmInicio.Interval = 100
End Sub

Private Sub TmInicio_Timer()
pp = 1
CargandoBar.Min = pp
CargandoBar.Max = 20000
CargandoBar.Visible = True

For pp = pp To 20000
CargandoBar.Value = pp
Next pp
Codigo.Show
TmInicio.Enabled = False
Form1.Visible = False
End Sub

```

Subrutina para pedir Cero Pieza

```

Dim PosX, PosY, PosZ, PosW As Integer
Private Sub CeroPieza_Click()
Basura = ""
ceroz = True
Audio.PuertoComm Output = "1"
Audio.PuertoComm Output = "7"

Do While (flag6 <> True)
DoEvents
Loop

```

```

"ceroz = True
Audio.PuertoComm Output = "2"
Audio.PuertoComm Output = "7"
"ceroz = True

Do While (flag7 <> True)
DoEvents
Loop

"ceroz = True
Audio.PuertoComm Output = "3"
Audio.PuertoComm Output = "7"

End Sub
'Private Sub Command1_MouseDown(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)

Private Sub CmmLimpiar_Click()
TxtVisua.Text = ""
End Sub

Private Sub CmmStop_Click()
Audio.PuertoComm Output = "1"
Audio.PuertoComm Output = "K"
Audio.Retardo
Audio.PuertoComm Output = "2"
Audio.PuertoComm Output = "K"
Audio.Retardo
Audio.PuertoComm Output = "3"
Audio.PuertoComm Output = "K"

If Ato = True Then

For i = 0 To 5
MoToR.Command1.Item(i).Enabled = True
Next i
MoToR.Homsito.Enabled = True
MoToR.CeroPieza.Enabled = True

Else
End If
End Sub

```

Subrutina de Operaciones modo Manual

```

Private Sub Command1_MouseDown(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)
Select Case Index

Case 0
If Button = vbLeftButton Then
Audio.PuertoComm Output = "3"
Audio.PuertoComm Output = Chr(5)
Audio.Retardo
Audio.PuertoComm Output = "3" 'direccion 0x33 PIC3
Audio.PuertoComm Output = "Q" 'accion a mandar 0x51
Audio.Retardo
PosY = PosY + 0.1
BarraXYZ.Panels(5).Text = PosY

ElseIf Button = vbRightButton Then
Audio.PuertoComm Output = "3"
Audio.PuertoComm Output = Chr(5)
Audio.Retardo
Audio.PuertoComm Output = "3" 'direccion 0x33 PIC3
Audio.PuertoComm Output = "q" 'accion a mandar movimiento
continuo
Audio.Retardo
End If

Case 1
If Button = vbLeftButton Then
Audio.PuertoComm Output = "3"
Audio.PuertoComm Output = Chr(5)
Audio.Retardo
Audio.PuertoComm Output = "3" 'direccion 0x32 PIC2
Audio.PuertoComm Output = "P" 'accion a mandar 0x71
Audio.Retardo
PosY = PosY - 0.1
BarraXYZ.Panels(5).Text = PosY

ElseIf Button = vbRightButton Then
Audio.PuertoComm Output = "3"
Audio.PuertoComm Output = Chr(5)
Audio.Retardo
Audio.PuertoComm Output = "3" 'direccion 0x32 PIC2
Audio.PuertoComm Output = "p" 'accion a mandar 0x71
Audio.Retardo
End If

Case 2

```

```

If Button = vbLeftButton Then
    Baudio PuertoComm Output = "2"
    Baudio PuertoComm Output = Chr(5)
    Baudio Retardo
    Baudio PuertoComm Output = "2" 'direccion 0x32 PIC2
    Baudio PuertoComm Output = "M" 'accion a mandar 0x4D
    Baudio Retardo
    PosX = PosX + 0.1
    BarraXYZ.Panels(3).Text = PosX

ElseIf Button = vbRightButton Then
    Baudio PuertoComm Output = "2"
    Baudio PuertoComm Output = Chr(5)
    Baudio Retardo
    Baudio PuertoComm Output = "2" 'direccion 0x32 PIC2
    Baudio PuertoComm Output = Chr(8) 'accion a mandar 0x4D
    Baudio Retardo
End If

Case 3
If Button = vbLeftButton Then
    Baudio PuertoComm Output = "2"
    Baudio PuertoComm Output = Chr(5)
    Baudio Retardo
    Baudio PuertoComm Output = "2" 'direccion 0x32 PIC2
    Baudio PuertoComm Output = "N" 'accion a mandar 0x6D
    Baudio Retardo
    PosX = PosX - 0.1
    BarraXYZ.Panels(3).Text = PosX

ElseIf Button = vbRightButton Then
    Baudio PuertoComm Output = "2"
    Baudio PuertoComm Output = Chr(5)
    Baudio Retardo
    Baudio PuertoComm Output = "2" 'direccion 0x32 PIC2
    Baudio PuertoComm Output = Chr(7) 'accion a mandar 0x4D
    Baudio Retardo
End If

Case 4
If Button = vbLeftButton Then
    Baudio PuertoComm Output = "S" Chr(5)
    Baudio Retardo
    Baudio PuertoComm Output = "1" 'direccion 0x31 PIC1
    Baudio PuertoComm Output = "U" 'accion a mandar 0x55
    PosZ = PosZ + 0.1
    BarraXYZ.Panels(7).Text = PosZ
ElseIf Button = vbRightButton Then
    Baudio PuertoComm Output = "1" 'direccion 0x31 PIC1
    Baudio PuertoComm Output = "U" 'accion a mandar 0x55

End If

Case 5
If Button = vbLeftButton Then
    Baudio PuertoComm Output = "S" Chr(5)
    Baudio Retardo
    Baudio PuertoComm Output = "1" 'direccion 0x31 PIC1
    Baudio PuertoComm Output = "D" 'accion a mandar 0x75
    PosZ = PosZ - 0.1
    BarraXYZ.Panels(7).Text = PosZ
ElseIf Button = vbRightButton Then
    Baudio PuertoComm Output = "1" 'direccion 0x31 PIC1
    Baudio PuertoComm Output = "D" 'accion a mandar 0x75

End If

End Select
End Sub

```

Subrutina de variador de velocidad

```

Private Sub Command3_Click(Index As Integer)
    Static h As Integer
    Select Case Index
    Case 0
        'For h = 1 To 3
        Baudio PuertoComm Output = "1" 'direccion 0x33 PIC3
        Baudio PuertoComm Output = "V" 'accion a mandar 0x51
        Baudio Retardo

        Baudio PuertoComm Output = "2" 'direccion 0x33 PIC3
        Baudio PuertoComm Output = "V" 'accion a mandar 0x51
        Baudio Retardo

        Baudio PuertoComm Output = "3" 'direccion 0x33 PIC3
        Baudio PuertoComm Output = "V" 'accion a mandar 0x51
        Baudio Retardo
        'Next h
    Case 1

```

```

        'For h = 1 To 3
        Baudio PuertoComm Output = "1" 'direccion 0x33 PIC3
        Baudio PuertoComm Output = "V" 'accion a mandar 0x51
        Baudio Retardo

        Baudio PuertoComm Output = "2" 'direccion 0x33 PIC3
        Baudio PuertoComm Output = "V" 'accion a mandar 0x51
        Baudio Retardo

        Baudio PuertoComm Output = "3" 'direccion 0x33 PIC3
        Baudio PuertoComm Output = "V" 'accion a mandar 0x51
        Baudio Retardo

        'Next h
    End Select
End Sub

```

```

Private Sub Check0_Click(Index As Integer)
If Check0(0).Value = 1 Then
    Baudio PuertoComm Output = "1" ' direccion del PIC 1 0x31
    Baudio PuertoComm Output = "R"
    Baudio Retardo
Else
    Baudio PuertoComm Output = "1"
    Baudio PuertoComm Output = "L"
End If
End Sub

```

```

Private Sub Form_Load()
    BarraXYZ.Panels(2).Text = "Coordenada X"
    BarraXYZ.Panels(4).Text = "Coordenada Y"
    BarraXYZ.Panels(6).Text = "Coordenada Z"
    PosX = 0
    PosY = 0
    PosZ = 0
    flag3 = False
    flag5 = False
    flag7 = False
End Sub
Private Sub Hornito_Click()

```

```

    Baudio PuertoComm Output = "1"
    Baudio PuertoComm Output = "&"

```

```

Do While (flag3 <> True)
DoEvents
Loop
flag3 = False

```

```

Baudio PuertoComm Output = "2"
Baudio PuertoComm Output = "&"

```

```

Do While (flag3 <> True)
DoEvents
Loop
flag3 = False

```

```

Baudio PuertoComm Output = "3"
Baudio PuertoComm Output = "&"

```

```

Do While (flag3 <> True)
DoEvents
Loop

```

```

MsgBox "Home OK", vbExclamation, "Home"
flag3 = False

```

Subrutina para cambiar información

```

Private Sub casos(recletra)
    Select Case recletra

```

```

        Case "A"
            letra = 10
        Case "B"
            letra = 11
        Case "C"
            letra = 12
        Case "D"
            letra = 13
        Case "E"
            letra = 14
        Case "F"
            letra = 15
        Case "T"
            letra = 84

```

```

Case 10
letra = 10
Case "x"
letra = 120
Case "y"
letra = 121
Case "@"
letra = 64
Case "M"
letra = 77
End Select

```

End Sub

Subrutina para leer programa de memoria

```

Private Sub ReadProg_Click()
leer = True
Baudio PuertoComm Output = "1"
Baudio PuertoComm Output = "T"

```

End Sub

```

Dim SeñalX, SeñalY, SeñalResiduoX, SeñalResiduoY As Single
Dim SeñalEnteraX, SeñalEnteraY As Integer
Dim NumHexX, NumHexY As Integer
Dim concatena, concatena2 As String
Const ResEnc = 0.15 'unidades en mm
Private Sub ArcExc_Click()
If ArcExc.Value = 1 Then
End If
End Sub

```

```

Private Sub ArcTex_Click()
If ArcTex.Value = 1 Then
End If
End Sub

```

```

Private Sub Baudaje_Click()
Baudio Show
End Sub

```

```

Private Sub cargar_Click()
SavePrg

```

End Sub

```

Private Sub MosHerr_Click()
Dim herrera As Integer
ReDim herramienta(1 To herr) As String
Dim i As Integer
Dim valorgetDiaHerr, getDiaHeR, valorherr As String
Dim Diametro As String
Dim k As Integer, vH, apuntador As Integer

```

subrutina para determinar las características de la herramienta

```

Label1.Visible = True
Label2.Visible = True
NoHerr.Visible = True
DiaHerr.Visible = True
k = 1
apuntador = 0
Diametro = ""
getDiaHeR = ""
Do While valorgetDiaHerr <> ""

```

```

    valorgetDiaHerr = Chr(Asc(Mid$(datoreciente, k, 4)))
    getDiaHeR = getDiaHeR + Replace(getDiaHeR, getDiaHeR,
valorgetDiaHerr)

```

```

    If valorgetDiaHerr = "C" Then

```

```

        For vH = 1 To 4
            valorgetDiaHerr = Chr(Asc(Mid$(datoreciente, k + 1, vH)))
            Diametro = Diametro + Replace(Diametro, Diametro,
valorgetDiaHerr)
            k = k + 1
            Next vH
            apuntador = apuntador + 1
            herramienta(apuntador) = Diametro

            Diametro = ""
        End If
        k = k + 1
    Loop

```

Loop

For i = 1 To herr

```

DiaHerr.ForeColor = QBColor(0)
DiaHerr.AddItem " " & i & " " & (herramienta(i)) & " " &
"[MM]"
Next i

```

End Sub

```

Private Sub Form_Load()
Picture1.Cls

```

```

Label1.Visible = False
Label2.Visible = False
NoHerr.Visible = False
DiaHerr.Visible = False
Command2.Enabled = False

```

```

MayorX = False
MayorY = False
GiroNeg = False
MasMenos = False
HH = False
HHY = False
desplegarxy Panels(1).Text = "Desconectado"
desplegarxy Panels(2).Text = "Coordenadas (X,Y)"
desplegarxy Panels(3).Text = "[mm]"
desplegarxy Panels(4).Text = "[CentroX_CentroY]"
desplegarxy Panels(5).Text = "Codigo.dialogo FileTitle
'flag2 = False

```

End Sub

```

Private Sub Picture1_MouseDown(Button As Integer, Shift As Integer, x As
Single, y As Single)

```

```

If Button = 1 Then
CoorX = x: CurrentX = x
CoorY = y: CurrentY = y

```

End If
End Sub

```

Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, x As
Single, y As Single)

```

```

'Picture1.ToolTipText = Round(EScalaX_min - X, 3) & ", " &
Round(Picture1.ScaleHeight - Y, 3)
'desplegarxy Panels(2) = "(" & Round(EScalaX_min - X, 3) & ", " &
Round(Picture1.ScaleHeight - Y, 3) & ")"

```

```

Picture1.ToolTipText = Round(x, 3) & ", " & Round(Picture1.ScaleHeight - y,
3)
desplegarxy Panels(2) = "(" & Round(x, 3) & ", " &
Round(Picture1.ScaleHeight - y, 3) & ")"

```

End Sub

Subrutina para convertir archivo a aun archivo codificado

```

HexTraPic.txt
Public Sub DatosTransmitir()
Dim Nt As Integer

```

```

Open "C:\HexTrnPic.txt" For Output As #3

```

```

Print #3, "T" & 1
Nt = 2
For x = 1 To Codigo grid Rows - 3

```

```

    If x = 1 Then
        DatoFinalX = Val(Cx(x))
        DatoFinalY = Val(Cy(x))
        ValorAbsoluto
    End If

```

```

    ElseIf Cx(x) = 0 And Cy(x) = 0 Then
        Print #3, "T" & Nt
        Nt = Nt + 1
        HH = True
        HHY = True
    Else

```

```

        If Cx(x + 1) = 0 And Cy(x + 1) = 0 Then
            DatoFinalX = Val(Cx(x)) - Val(Cx(x - 1)) 'haciendo la diferencia
entre la coordenada actual
            DatoFinalY = Val(Cy(x)) - Val(Cy(x - 1)) 'y la coordenada
siguiente
            ValorAbsoluto
            flag2 = True
            GoTo 2
        Else

```

```

If flag2 = True Then
  DatoFinalX = Val(Cx(x)) - Val(Cx(x - 2)) 'haciendo la
diferencia entre la coordenada actual
  DatoFinalY = Val(Cy(x)) - Val(Cy(x - 2)) 'y la coordenada
siguiente
  flag2 = False
  ValorAbsoluto
Else
  DatoFinalX = Val(Cx(x)) - Val(Cx(x - 1)) 'haciendo la
diferencia entre la coordenada actual
  DatoFinalY = Val(Cy(x)) - Val(Cy(x - 1)) 'y la coordenada
siguiente
  ValorAbsoluto
End If
2
End If
End If
Next x
Print #3, "M30@"
Close #3

End Sub

Public Sub AcompletarHex(SeñalFinalX, SeñalFinalY, concatena,
concatena2)
  If Len(Hex(SeñalFinalX)) = 3 And Len(Hex(SeñalFinalY)) = 3
Then
  Print #3, "x", concatena & Hex(SeñalFinalX), "y",
concatena2 & Hex(SeñalFinalY)
  ElseIf Len(Hex(SeñalFinalX)) = 3 And
Len(Hex(SeñalFinalY)) = 2 Then
  Print #3, "x", concatena & Hex(SeñalFinalX), "y",
concatena2 & 0 & Hex(SeñalFinalY)
  ElseIf Len(Hex(SeñalFinalX)) = 2 And
Len(Hex(SeñalFinalY)) = 3 Then
  Print #3, "x", concatena & 0 & Hex(SeñalFinalX), "y",
concatena2 & Hex(SeñalFinalY)
  ElseIf Len(Hex(SeñalFinalX)) = 3 And
Len(Hex(SeñalFinalY)) = 1 Then
  Print #3, "x", concatena & Hex(SeñalFinalX), "y",
concatena2 & 0 & 0 & Hex(SeñalFinalY)
  ElseIf Len(Hex(SeñalFinalX)) = 1 And
Len(Hex(SeñalFinalY)) = 3 Then
  Print #3, "x", concatena & 0 & 0 & Hex(SeñalFinalX),
"y", concatena2 & Hex(SeñalFinalY)
  ElseIf Len(Hex(SeñalFinalX)) = 3 And
Len(Hex(SeñalFinalY)) = 0 Then
  Print #3, "x", concatena & Hex(SeñalFinalX), "y",
concatena2 & 0 & 0 & Hex(SeñalFinalY)
  ElseIf Len(Hex(SeñalFinalX)) = 0 And
Len(Hex(SeñalFinalY)) = 3 Then
  Print #3, "x", concatena & 0 & 0 & Hex(SeñalFinalX),
"y", concatena2 & Hex(SeñalFinalY)
  ElseIf Len(Hex(SeñalFinalX)) = 2 And
Len(Hex(SeñalFinalY)) = 2 Then
  Print #3, "x", concatena & 0 & Hex(SeñalFinalX), "y",
concatena2 & 0 & 0 & Hex(SeñalFinalY)
  ElseIf Len(Hex(SeñalFinalX)) = 0 And
Len(Hex(SeñalFinalY)) = 2 Then
  Print #3, "x", concatena & 0 & 0 & Hex(SeñalFinalX),
"y", concatena2 & 0 & 0 & Hex(SeñalFinalY)
  ElseIf Len(Hex(SeñalFinalX)) = 1 And
Len(Hex(SeñalFinalY)) = 2 Then
  Print #3, "x", concatena & 0 & 0 & Hex(SeñalFinalX),
"y", concatena2 & 0 & 0 & Hex(SeñalFinalY)
  ElseIf Len(Hex(SeñalFinalX)) = 2 And
Len(Hex(SeñalFinalY)) = 1 Then
  Print #3, "x", concatena & 0 & Hex(SeñalFinalX), "y",
concatena2 & 0 & 0 & Hex(SeñalFinalY)
  ElseIf Len(Hex(SeñalFinalX)) = 1 And
Len(Hex(SeñalFinalY)) = 1 Then
  Print #3, "x", concatena & 0 & 0 & Hex(SeñalFinalX),
"y", concatena2 & 0 & 0 & Hex(SeñalFinalY)
Else
  ElseIf Len(Hex(SeñalFinalX)) = 2 And
Len(Hex(SeñalFinalY)) = 1 Then
  Print #3, "x", concatena & 0 & 0 & Hex(SeñalFinalX),
"y", concatena2 & 0 & 0 & Hex(SeñalFinalY)
ElseIf Len(Hex(SeñalFinalX)) = 1 And
Len(Hex(SeñalFinalY)) = 1 Then
  Print #3, "x", concatena & 0 & 0 & Hex(SeñalFinalX),
"y", concatena2 & 0 & 0 & Hex(SeñalFinalY)
Else
  ElseIf Len(Hex(SeñalFinalX)) = 1 And
Len(Hex(SeñalFinalY)) = 0 Then
  Print #3, "x", concatena & Hex(SeñalFinalX), "y",
concatena2 & 0 & 0 & 0 & 0 & Hex(SeñalFinalY)
ElseIf Len(Hex(SeñalFinalX)) = 0 And
Len(Hex(SeñalFinalY)) = 1 Then
  Print #3, "x", concatena & 0 & 0 & 0 & 0 & Hex(SeñalFinalX),
"y", concatena2 & Hex(SeñalFinalY)
ElseIf Len(Hex(SeñalFinalX)) = 0 And
Len(Hex(SeñalFinalY)) = 0 Then
  Print #3, "x", concatena & 0 & 0 & 0 & 0 & Hex(SeñalFinalX),
"y", concatena2 & 0 & 0 & 0 & 0 & Hex(SeñalFinalY)
Else
  Print #3, "x", concatena & Hex(SeñalFinalX), "y",
concatena2 & Hex(SeñalFinalY)
End If
End Sub

```

```

Print #3, "x", 0 & 0 & 0 & 0 & 0, "y", 0 & 0 & 0 & 0 & 0
End If
End Sub

Public Sub ConcatenaHex()
If HH = False Then
If MasMenos = False Then
If GiroNeg = False Then
If MayorX = True Then
If Cx(x - 1) >= Cx(x) Then
concatena = "1"
MayorX = False
End If
Else
If Cx(x + 1) >= Cx(x) Then
concatena = "0"
MayorX = False
Else
If Cx(x + 1) > Cx(x + 2) Then
concatena = "0"
GiroNeg = True
Else
concatena = "0"
End If
End If
End If
Else
If Cx(x - 1) >= Cx(x) Then
concatena = "1"
ElseIf Cx(x + 1) >= Cx(x) Then
concatena = "0"
GiroNeg = False
MayorX = False
Else
concatena = "0"
GiroNeg = False
MayorX = False
MasMenos = True
End If
End If
Else
If Cx(x - 1) >= Cx(x) Then
concatena = "1"
Else
concatena = "0"
MasMenos = False
End If
End If
Else
HH = False
MayorX = False
GiroNeg = False
If Cx(x - 2) > Cx(x) Then
concatena = "1"
Else
concatena = "0"
End If
End If
End If
If HHY = False Then
If MasMenosY = False Then
If GiroNegY = False Then
If MayorY = True Then
If Cy(x - 1) >= Cy(x) Then
concatena2 = "1"
MayorY = False
End If
Else
If Cy(x + 1) >= Cy(x) Then
concatena2 = "0"
MayorY = False
Else
MayorY = True
If Cy(x + 1) > Cy(x + 2) Then
concatena2 = "0"
GiroNegY = True
Else
concatena2 = "0"
End If
End If
End If
Else
If Cy(x - 1) >= Cy(x) Then
concatena2 = "1"
ElseIf Cy(x + 1) >= Cy(x) Then
concatena2 = "0"
End If
End If
End If
End Sub

```



```

        concatenana2 = "0"
        GiroNegY = False
        MayorY = False
    Else
        concatenana2 = "0"
        GiroNegY = False
        MayorY = False
        MasMenosY = True
    End If
End If
Else
    If Cy(x - 1) >= Cy(x) Then
        concatenana2 = "1"
    Else
        concatenana2 = "0"
        MasMenosY = False
    End If
End If
Else
    HHY = False
    MayorY = False
    GiroNegY = False
    If Cy(x - 2) > Cy(x) Then
        concatenana2 = "1"
    Else
        concatenana2 = "0"
    End If
End If
End Sub

Public Sub DatosSalida()

    SeñalX = Val(DatoFinalX) / ResEnc
    SeñalY = Val(DatoFinalY) / ResEnc

    SeñalEnteraX = Int(SeñalX)
    SeñalEnteraY = Int(SeñalY)

    SeñalResiduoX = (SeñalX - SeñalEnteraX) * 10
    SeñalResiduoY = (SeñalY - SeñalEnteraY) * 10

    If SeñalResiduoX >= 5 And SeñalResiduoY >= 5 Then
        ConcatenaHex
        AcompletarHex SeñalEnteraX + 1, SeñalEnteraY + 1,
concatena, concatenana2

        ElseIf SeñalResiduoX >= 5 And SeñalResiduoY < 5 Then
        ConcatenaHex
        AcompletarHex SeñalEnteraX + 1, SeñalEnteraY,
concatena, concatenana2

        ElseIf SeñalResiduoX < 5 And SeñalResiduoY >= 5 Then
        ConcatenaHex
        AcompletarHex SeñalEnteraX, SeñalEnteraY + 1,
concatena, concatenana2

    Else
        ConcatenaHex
        AcompletarHex SeñalEnteraX, SeñalEnteraY, concatena,
concatena2

    End If

End Sub

Public Sub ValorAbsoluto()

    If DatoFinalX >= 0 And DatoFinalY >= 0 Then
        DatosSalida

    ElseIf DatoFinalX >= 0 And DatoFinalY < 0 Then
        DatoFinalY = Abs(DatoFinalY)
        DatosSalida

    ElseIf DatoFinalX < 0 And DatoFinalY >= 0 Then
        DatoFinalX = Abs(DatoFinalX)
        DatosSalida

    Else
        DatoFinalX = Abs(DatoFinalX)
        DatoFinalY = Abs(DatoFinalY)
        DatosSalida

    End If

End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Key
Case "flechalzquierda"
Codigo.Show
Case "AlmacenarPrg"
SavePrg
Case "BAD"
Baudio.Show

```

```

End Select
End Sub

Public Sub SavePrg()
Dim x, NH, BitEnteroX, BitEnteroY As Single
Dim PuX, PuY, ResiduoX, ResiduoY As Double
'seleccion de archivo de salida de texto

If ArcTex.Value = 1 Then

Open "C:\dat_tran.txt" For Output As #1
Open "C:\dat_tranPic.txt" For Output As #2

NH = 2
Print #1, Codigo.dialogo.FileTitle
Print #1, "N0", "T1"
For x = 1 To Codigo.grid.Rows - 3

If Cx(x) = 0 And Cy(x) = 0 Then
    Print #1, "N" & x, "T" & NH

    NH = NH + 1

    Else
        'rutina para calcular el numero de bits necesarios
        'para llegar a la posicion deseada
        'Numero de señales para la posicion deseada

        PuX = Log(Val(Cx(x)) / ResEnc) / Log(2)
        PuY = Log(Val(Cy(x)) / ResEnc) / Log(2)
        BitEnteroX = Int(PuX)
        BitEnteroY = Int(PuY)
        ResiduoX = (PuX - BitEnteroX) * 10
        ResiduoY = (PuY - BitEnteroY) * 10

        If ResiduoX >= 5 And ResiduoY >= 5 Then

            If Cx(x) = Cx(x + 1) Then

                Print #2, "N" & x, Tab, "Y", (BitEnteroY + 1)
                'Print #3, "N"

            Else

                Print #2, "N" & x, "X", (BitEnteroX + 1), Tab, "Y", (BitEnteroY +
1)

                'Print #2, "N" & X, Tab, "Y", (BitEnteroY + 1)
                End If

            ElseIf ResiduoX >= 5 And ResiduoY < 5 Then
                If Cy(x) = Cy(x + 1) Then
                    'Print #2, "N" & X, "X", (BitEnteroX + 1), Tab, "Y",
(BitEnteroY)

                    Print #2, "N" & x, "X", (BitEnteroX + 1)

                Else

                    'Print #2, "N" & X, "X", (BitEnteroX + 1), Tab, "Y",
(BitEnteroY)

                    Print #2, "N" & x, "X", (BitEnteroX + 1), Tab, "Y",
(BitEnteroY)

                End If

                ElseIf ResiduoX < 5 And ResiduoY >= 5 Then

                    Print #2, "N" & x, "X", BitEnteroX, Tab, "Y", (BitEnteroY +
1)

                Else

                    Print #2, "N" & x, "X", BitEnteroX, Tab, "Y", BitEnteroY

                End If

                Print #1, "N" & x, "X", Cx(x), Tab, "Y", Cy(x)

            End If

        Next x
        Print #1, "N" & x, "M30"
        Close #1
        Close #2
    End If

'seleccion de salida a archivo de excell

If ArcExc.Value = 1 Then

    Set objExcell = CreateObject("Excel.Sheet")

    'Establecer CancelError a True
    dialogo1.CancelError = False
    dialogo1.Flags = cdiOFNCreatePrompt
    ' Establecer los filtros

```

```

dialogo1 Filter = "Archivos de Excel (*.xls)*.xls"
' Especificar el filtro predeterminado
dialogo1 FilterIndex = 1
' Presentar el cuadro de diálogo Abrir
dialogo1 ShowSave
' Presentar el nombre del archivo seleccionado

objExcel1 Application.Cells(1, 1).Value = "Coordenada X"
objExcel1 Application.Cells(1, 2).Value = "Coordenada Y"
objExcel1 Application.Cells(1, 3).Value = "Coordenada Z"
objExcel1 Application.Cells(1, 4).Value = "Herramienta"

For x = 1 To Codigo.grid.Rows - 3

    If Cx(x) = 0 And Cy(x) = 0 Then

        objExcel1 Application.Cells(x + 1, 4).Value = "T" & NH
        objExcel1 Application.Cells(x + 1, 1).Value = Cx(x + 1)
        objExcel1 Application.Cells(x + 1, 2).Value = Cy(x + 1)
        objExcel1 Application.Cells(x + 1, 3).Value = -zeta
        NH = NH + 1
        Else

        objExcel1 Application.Cells(x + 1, 1).Value = Cx(x)
        objExcel1 Application.Cells(x + 1, 2).Value = Cy(x)
        objExcel1 Application.Cells(x + 1, 3).Value = -zeta

    End If

Next x

objExcel1.SaveAs FileName = dialogo1.FileName

DatosTransmitir
End Sub

Private Sub CmBDatos_Click()

Open "c:\shair\HexTrnPic.txt" For Input As #1 ' Abre el archivo.
Do While Not EOF(1) ' Repite el bucle hasta el final del archivo.
    tExtoHex = tExtoHex & Input(1, #1) ' Obtiene un carácter.
Loop

Close #1 ' Cierra el archivo
TxtDaTra.Text = tExtoHex

End Sub

Private Sub CmBTraDatos_Click()
BarraProgresiva.Min = 1
BarraProgresiva.Max = Len(TxtDaTra.Text)
BarraProgresiva.Visible = True
LblLoad.Visible = True

For i = 1 To Len(TxtDaTra.Text)
    send = Mid$(TxtDaTra.Text, i, 1)
    normales = send
    RichDatos.Text = RichDatos.Text & send
    DoEvents
    If Asc(normales) + Asc(Mid(TxtDaTra.Text, i + 1, 1)) = 23 Then
        RichDatos.Text = RichDatos.Text & send + vbCrLf
        i = i + 1
    Else
        MoToR.casos (send)
        Baudio PuertoComm Output = Chr(letra)

        Do While (otra <> True)
            DoEvents
            Loop
        DoEvents
        otra = False
    End If
    otra = True
    BarraProgresiva.Value = i

Next i
LblLoad.Visible = False
BarraProgresiva.Visible = False
otra = True

Do While (otra <> True)
DoEvents
Loop
MsgBox "Transmisión completa", vbExclamation, "Transmisión de Datos"

End Sub

```

```

Private Sub CMDMandar_Click()
tHex = TxtDato.Text
MoToR.casos (tHex)
Baudio PuertoComm Output = Chr(letra)
tBIDatos.Caption = letra
End Sub

Private Sub Form_Load()
otra = False
LblLoad.Visible = False
CMDMandar.Enabled = False
BarraProgresiva.Visible = False
End Sub

Public CentroY, CentroX As String
Public Cx(), Cy() As Double, Hta() As String
Public Xmin, Ymin, CoorX, CoorY As Single
Public ColorHerr, herr, NumReng As Integer
Public herramienta() As String
Public k, Cuadro As Integer
Public datoreciente As String
Public ValorMaxY() As String
Public ValorMaxY() As String
Public p, bar As Integer
Public flag2, HH, HHY, MasMenos, MasMenosY, MayorX, MayorY,
GiroNeg, GiroNegY, bandera01, bandera02 As Boolean
Public cambio, DatoFinalX, DatoFinalY As Single
Public Indicador, x As Integer
Public bandera03, bandera04, ceroz, cerox, ceroy As Boolean
Public basurita As String
Public conta, j, letra As Integer
Public ConHex, DesOZ As String
Public totales, letras, recietra As Double
Public send As String

Public Sub Tamaño_Picture()

Dim x, ii, rowactual
Dim repetirY As Integer, rowactual As Integer, centroy_1 As String
ReDim Cx(1 To Codigo.grid.Rows), Cy(1 To Codigo.grid.Rows) As Double
ReDim ValorMaxX(1 To Codigo.grid.Rows) As String
ReDim ValorMaxY(1 To Codigo.grid.Rows) As String
Dim EScalaX, EScalaX_min As Variant
Dim EScalaY, EScalaY_min As Variant
Dim promedio As Single
Dim j As Integer

j = 1
ColorHerr = 1
Codigo.grid.Row = 2
NumReng = 1

For x = 1 To Codigo.grid.Rows - 4
    Codigo.grid.Col = 2
    CentroX = Codigo.grid.Text

    If CentroX <> "" Then

        Codigo.grid.Col = 3
        CentroY = Codigo.grid.Text

        rowactualx = Codigo.grid.Row

    Else

        rowactualx = Codigo.grid.Row

    ii = 1

        'verificando si hay cambio de herramienta

    Codigo.grid.Col = 3
    CentroY = Codigo.grid.Text

    If CentroX = "" And CentroY = "" Then
        ColorHerr = ColorHerr + 1
        GoTo 1

    Else
        Codigo.grid.Col = 2

        Do While CentroX = ""
            Codigo.grid.Row = Codigo.grid.Row - 1
            CentroX = Codigo.grid.Text
            Loop

        End If
        End If
        Codigo.grid.Row = rowactualx

```

```

Codigo grid Col = 3
CentroY = Codigo grid Text

If CentroY <> "" Then

Codigo grid Row = rowactual

Else

ii = 1
Do While CentroY = ""
Codigo grid Row = Codigo grid Row - ii
CentroY = Codigo grid Text
Loop

End If

Codigo grid Row = rowactual

ValorMaxX(j) = CentroX
ValorMaxY(j) = CentroY

j = j + 1

1 rowactual = x + 2
Codigo grid Row = rowactual

NumReng = NumReng + 1 ' este ciclo nos permite determinar el tamaño
del arreglo ' de los valores de los centros
If CentroX = "" And CentroY = "" Then
Codigo grid Row = rowactual
Codigo grid Col = 2
If Codigo grid Text = "" Then
Codigo grid Col = 3
End If
If Codigo grid Text = "" Then
Exit For
End If
End If
Next x

Call Codigo.get_ancho(EScalaX, EScalaX_min)

```

```

Call Codigo.get_alto(EScalaY, EScalaY_min)

promedio = (Val(EScalaX + EScalaY) / 1.5)
Simulacion Picture1 ScaleHeight = promedio
Simulacion Picture1 ScaleWidth = promedio

End Sub

Public Sub InformacionCeldas()
Codigo grid Col = 0
Codigo grid Row = 1
Codigo grid Text = "N" & 1

Codigo grid Col = 1
Codigo grid Row = 0
Codigo grid ColWidth(1) = 1300
Codigo grid Text = "Herramienta"

Codigo grid Col = 2
Codigo grid Row = 0
Codigo grid ColWidth(2) = 1300
Codigo grid Text = "Coordenada X"

Codigo grid Col = 3
Codigo grid Row = 0
Codigo grid ColWidth(3) = 1300

Codigo grid Text = "Coordenada Y"

Codigo grid Col = 4
Codigo grid Row = 0
Codigo grid ColWidth(4) = 1300
Codigo grid Text = "Coordenada Z"
Codigo grid Rows = 2
End Sub

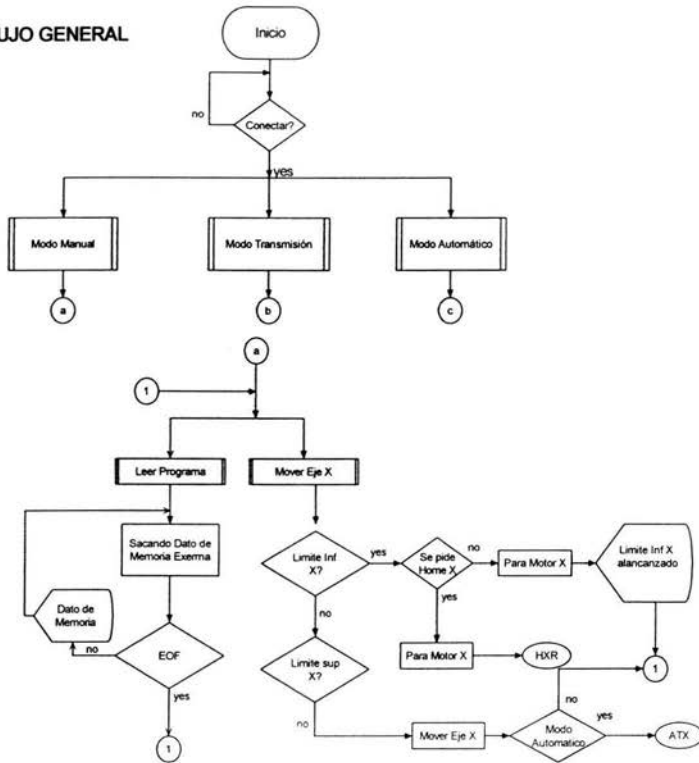
Public Ato, leer, bandera, flag3, flag4, flag5, flag6, flag7 As Boolean
Public otra, bandera03, bandera04 As Boolean
Public basurita As String
Public i, conta, j, k As Integer
Public ConHex, DesOZ As String
Public totales, letras, recletra, conv As Double
Public tExtHex As String
Public convertir As String
Public pp As Integer

```

Anexo B

Diagramas de flujo para el programa de Interfase y μC Maestro PIC16F877

DIAGRAMA DE FLUJO GENERAL



CONTINUACION MODO MANUAL

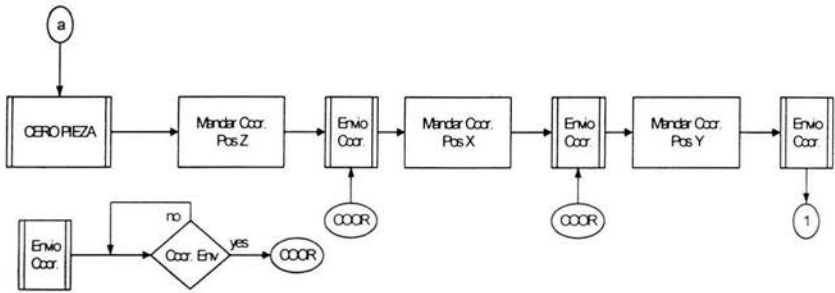
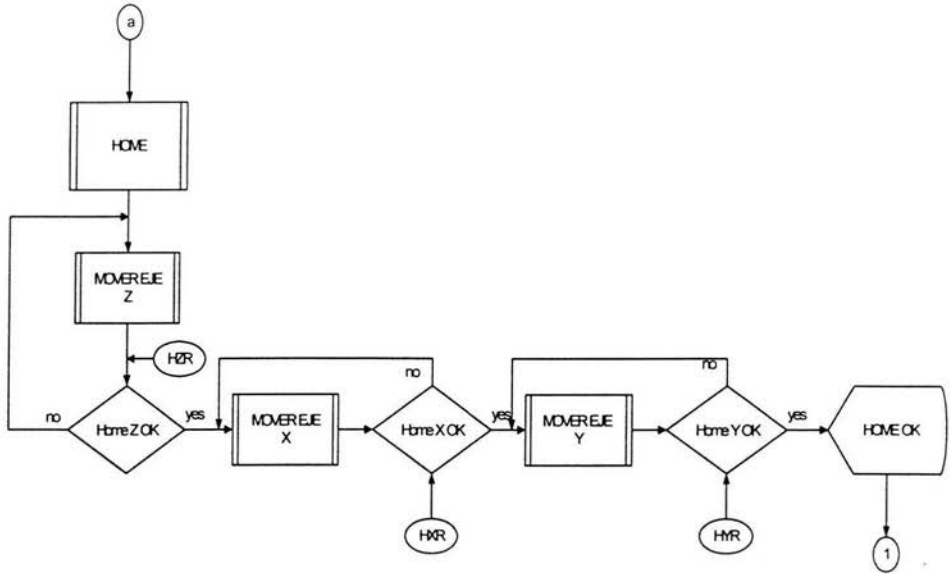
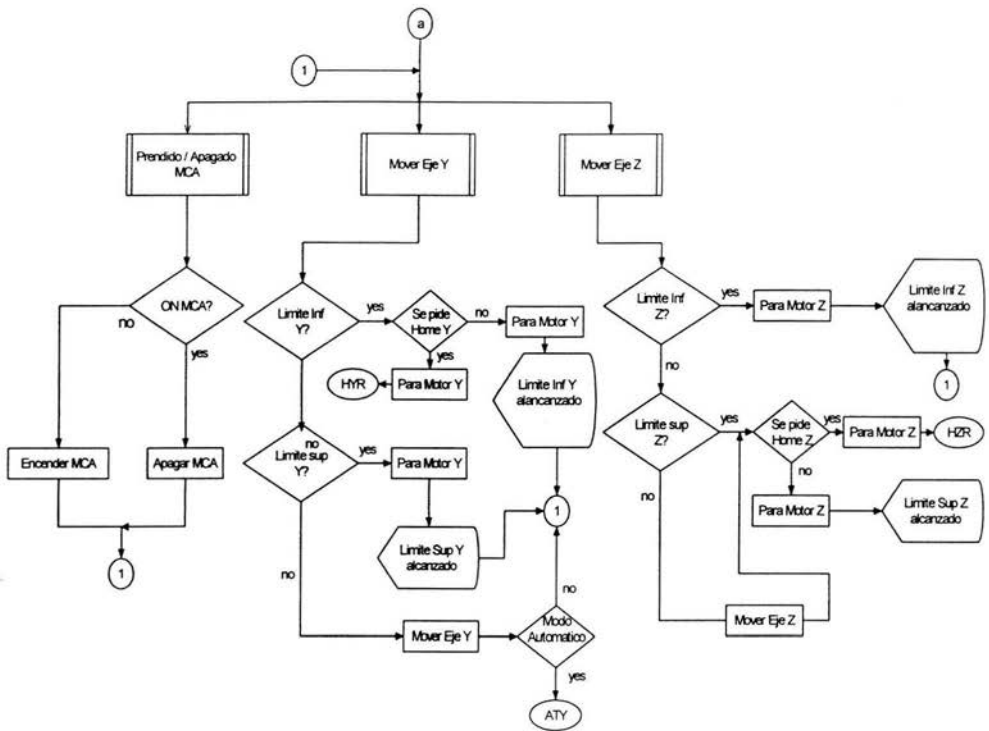
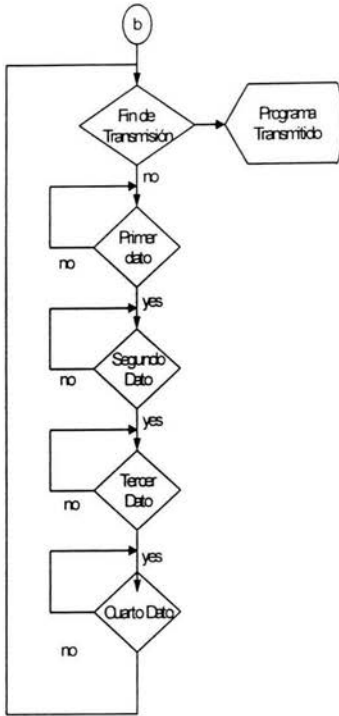


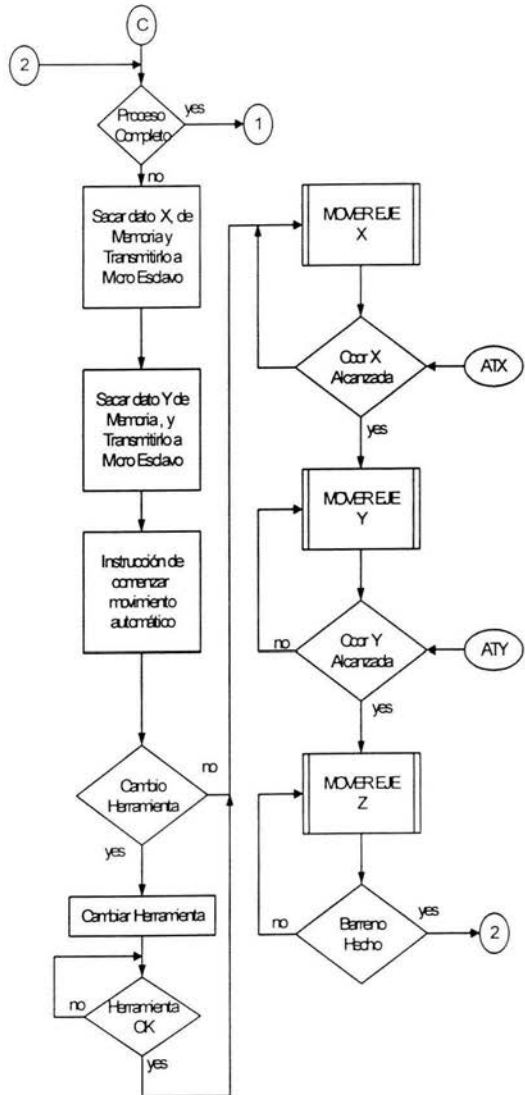
DIAGRAMA DE FLUJO EN MODO MANUAL



MODO TRANSMISIÓN DE DATOS



MODO AUTOMÁTICO



Código fuente µC PIC16F877

LIST	P=16F877				
	RADIX	HEX			
	INCLUDE	"PIC16F877.INC"			
	_CONFIG	H3D3A			

-----Registros de uso general-----					
BANDERA	EQU	0x20			
CONTROL	EQU	0x21			
FLAG	EQU	0x22			
REGPIC1	EQU	0x23			
REGHI	EQU	0x24			
REGLW	EQU	0x25			
TMP1	EQU	0x26			
ZH	EQU		0x27		
ZL	EQU		0x28		
CONDAT	EQU	0x29			
ADSHIX	EQU	0x30			
ADSLWX	EQU	0x31			
ADSHIY	EQU	0x32			
ADSLWY	EQU	0x33			
REGAPO	EQU	0x34			
RESTALW	EQU	0x35			
RESTAHI	EQU	0x36			
VELOCI	EQU	0x37			
REGLEER	EQU	0x38			
AUTO23	EQU	0x39			
TEMP23	EQU	0x40			
ALTURAZ	EQU	0x2A			
COMPARAZ	EQU	0x3A			
VALOR	EQU	0x2B			

		ORG		0X00	
		GOTO	INICIO		
		ORG		0x05	

-----Atencion al vector de Interrupcion-----					
INTER	BTSS	PIR1,RCIF	:	interrupcion por recepcion?	
		GOTO	CONTI	:	no, regresa
		BCF	PIR1,RCIF	:	si, restaura flag de
		BTSS	BANDERA,0	:	ya se conecto por 1
era vez?		GOTO	CONECTAR		
		GOTO	VERDIR		
CONECTAR	MOVLW	A'0			
		SUBWF	RCREG,W		
		BTSS	STATUS,Z	:	se pide conectar?
		GOTO	CONTI		
		BSF	BANDERA,0	:	indicacion de
conexion extosa		GOTO	CONTI		
VERDIR	BTSS	BANDERA,1	:	comunicacion PC-PIC1?	
		GOTO	PC_PIC1		
		GOTO	VERMENU		
PC_PIC1	MOVLW	0x31	:	direccion 0x01	
		SUBWF	RCREG,W		
		BTSS	STATUS,Z	:	se pide PC-PIC1
		GOTO	VERPIC2		
		BSF	BANDERA,1	:	indicacion de
comunicacion PC-PIC1		GOTO	CONTI		
VERPIC2	BTSS	FLAG,2	:	se pidio comunicacion con PIC2?	
		GOTO	DIR2		
		MOVF	RCREG,W		
		MOVWF	TEMP23		
		MOVLW	AT		
		SUBWF	RCREG,W		
		BTSS	STATUS,Z		
		GOTO	NORMAL2		
		MOVLW	AT		
		CALL	TRANSMITIR		
		BSF	AUTO23,0	:	indicando que mando
informacion de automatico		BCF	FLAG,2		
		GOTO	CONTI	:	a pic2
NORMAL2	MOVF	TEMP23,W			
		CALL	TRANSMITIR		
		BCF	FLAG,2		
		GOTO	CONTI		
DIR2	MOVLW	0x32			

		SUBWF	RCREG,W		
		BTSS	STATUS,Z	:	comunicacion con
PIC3?		GOTO	VERPIC3		
		MOVF	RCREG,W		
		CALL	TRANSMITIR		
		BSF	FLAG,2		
		GOTO	CONTI		
VERPIC3	BTSS	FLAG,3			
		GOTO	DIR3		
		MOVF	RCREG,W		
		MOVWF	TEMP23		
		MOVLW	AT		
		SUBWF	RCREG,W		
		BTSS	STATUS,Z		
		GOTO	NORMAL3		
		MOVLW	AT		
		CALL	TRANSMITIR		
		BCF	FLAG,3		
		BSF	AUTO23,1	:	indicando que mando
informacion de automatico		GOTO	CONTI	:	a pic3
NORMAL3	MOVF	TEMP23,W			
		CALL	TRANSMITIR		
		BCF	FLAG,3		
		GOTO	CONTI		
DIR3	MOVLW	0x33			
		SUBWF	RCREG,W		
		BTSS	STATUS,Z	:	comunicacion con
PIC3?		GOTO	CANCELAR		
		MOVF	RCREG,W		
		CALL	TRANSMITIR		
		BSF	FLAG,3		
		GOTO	CONTI		
CANCELAR	MOVLW	A'S'			
		SUBWF	RCREG,W		
		BTSS	STATUS,Z	:	cancelar movimiento
continuo?		GOTO	CONTI		
		BCF	REGAPO,4		
		BCF	REGAPO,5		
		GOTO	CONTI		
VERMENU	BTSS	BANDERA,3	:	estas en modo manual?	
		GOTO	MENUS		
		GOTO	SIGMENU		
MENUS	BTSS	REGAPO,2	:	modo automatico?	
		GOTO	CHKMENU		
		GOTO	HX		
SIGMENU	BTSS	FLAG,1	:	modo de grabar	
		GOTO	VER_EEP		
		GOTO	CACHAR		
VER_EEP	MOVLW	A'W			
		SUBWF	RCREG,W		
		BTSS	STATUS,Z	:	se pide grabar datos
a eeprom?		GOTO	LEERMEM		
		GOTO	HX		
		BSF	FLAG,1	:	indicador que se pide
grabar datos		CLRF	ADSLWX		
		CLRF	ADSLWY		
		CLRF	ADSHIX		
		CLRF	ADSHIY		
		MOVLW	A'w		
		CALL	TRANSMITIR	:	indicador de ok para
grabar dato		GOTO	CONTI		
LEERMEM	MOVLW	AT			
		XORWF	RCREG,W		
		BTSS	STATUS,Z		
		GOTO	HX		
		BSF	CONTROL,7	:	indicador de leer
memoria		BSF	BANDERA,2		
		GOTO	CONTI		
HX		MOVLW	A'n		
		SUBWF	RCREG,W		
		BTSS	STATUS,Z	:	llego a home X?

		GOTO HY			BTFFSS	STATUS.Z	,se pide fin de
		MOVLW A'n		escritura?	GOTO	VER_HER	
		CALL TRANSMITIR			,BCF	CONDAT.0	
		CALL CLRDIR			BSF	BANDERA.2	
		GOTO CONTI			BSF	FLAG.7	,indicador de fin de
HY		MOVLW 0x04		grabar programa	BTFFSS	REGAPO.1	,indicador de fin de
		SUBWF RCREG.W			BSF	REGAPO.1	,indicador de fin de
		BTFFSS STATUS.Z		programa	MOVF	RCREG.W	
		GOTO SPEDMAS			MOVWF	REGLW	
		MOVLW A'T			BSF	PORTE.1	
		CALL TRANSMITIR			GOTO	CONTI	
		GOTO CONTI					
SPEOMAS	BTFFSS	REGAPO.7		VER_HER	MOVLW	A'T	
		GOTO VELMAS			SUBWF	RCREG.W	
		GOTO SPEDNEG			BTFFSS	STATUS.Z	
VELMAS	MOVLW	A'V			GOTO	VER_X	
		XORWF RCREG.W			MOVF	RCREG.W	
		BTFFSS STATUS.Z			MOVWF	REGLW	
velocidad					BSF	FLAG.4	,indicador de llegada
		GOTO SPEDNEG		de herramienta	BSF	BANDERA.2	
		MOVLW 1			BSF	CONDAT.0	
		SUBWF TMP1.F			GOTO	CONTI	
		MOVF TMP1.W					
		MOVWF VELOC		VER_X	BTFFSS	FLAG.5	,indicador que llego x
		BTFFSS STATUS.Z			GOTO	REV_X	
		GOTO SEGVEL			GOTO	WRI_EE	
		BSF REGAPO.7					
		GOTO SEGVEL					
SPEDEG	MOVLW	A'V		REV_X	MOVLW	A'X	
		XORWF RCREG.W			SUBWF	RCREG.W	
		BTFFSS STATUS.Z			BTFFSS	STATUS.Z	
		GOTO SEEMOVPOS		de coordenada x	GOTO	VER_Y	
		MOVLW 1			BSF	FLAG.5	,indicador de llegada
		ADDWF TMP1.F			GOTO	BANDERA.2	
		MOVF TMP1				CONTI	
		MOVWF VELOC		VER_Y	BTFFSS	FLAG.6	,esta recogiendo datos de y?
		BTFFSS STATUS.Z			GOTO	REV_Y	
		GOTO SEGVEL			GOTO	WRI_EE	
		BCF REGAPO.7					
		MOVLW 255		REV_Y	MOVLW	A'Y	
		MOVWF TMP1			SUBWF	RCREG.W	
SEGVEL	CALL	CLRDIR			BTFFSS	STATUS.Z	
		GOTO CONTI			GOTO	CONTI	
SEEMOVPOS		BTFFSS REGAPO.2		de coordenada y			
		GOTO SEEMOVPOS1			BSF	BANDERA.2	
		GOTO READIX			BSF	FLAG.6	,indicador de llegada
SEEMOVPOS1		MOVLW A'u		WRI_EE	BTFFSS	CONDAT.1	
		XORWF RCREG.W			GOTO	DATO1	
		BTFFSS STATUS.Z			BTFFSS	CONDAT.2	
		GOTO SEEMOVNEG			GOTO	DATO2	
		BSF REGAPO.4			BTFFSS	CONDAT.3	
movimiento continuo positivo					GOTO	DATO3	
		BSF BANDERA.2			GOTO	DATO4	
		MOVLW A'U		DATO1	MOVF	RCREG.W	
		MOVWF REGPIC1			MOVWF	REGHI	
		GOTO CONTI			SWAPF	REGHI.F	
SEEMOVNEG		MOVLW A'd		primer dato	BSF	CONDAT.1	,bit de indicacion de
		XORWF RCREG.W			MOVLW	A'T	
		BTFFSS STATUS.Z			CALL	TRANSMITIR	
		GOTO CERO		DATO2	MOVF	RCREG.W	
		BSF REGAPO.5			IORWF	REGHI.F	
indiar movimiento continuo negativo				primer dato	BSF	CONDAT.2	,bit de indicacion de
		BSF BANDERA.2			MOVLW	A'J	
		MOVLW A'D			CALL	TRANSMITIR	
		MOVWF REGPIC1			GOTO	CONTI	
		GOTO CONTI					
-----comando datos para grabar en eeprom-----							
CACHAR	BTFFSS	FLAG.7		DATO3	MOVF	RCREG.W	
		GOTO FINGP			MOVWF	REGLW	
		BCF FLAG.1		primer dato	SWAPF	REGLW.F	
señal de grabar datos					BSF	CONDAT.3	,bit de indicacion de
		BCF BANDERA.2			MOVLW	A'S	
		BCF CONDAT.0			CALL	TRANSMITIR	
		MOVF RCREG.W			GOTO	CONTI	
		MOVWF ALTURAZ		DATO4	MOVF	RCREG.W	
		MOVLW A'F			IORWF	REGLW.F	
		CALL TRANSMITIR					
		CALL CLRDIR					
		BCF FLAG.7					
		GOTO CONTI					
FINGP	MOVLW	A'@			BCF	CONDAT.1	
		SUBWF RCREG.W			BCF	CONDAT.2	
					BCF	CONDAT.3	
					BSF	BANDERA.2	

tienen los datos a grabar

BSF	CONDAT,0	.indicador de que se
BCF	FLAG,6	
BCF	FLAG,5	
GOTO	CONTI	

BTFSS STATUS,Z .dato bajo transmitido X?

GOTO SEGVEL

BSF REGLEER,2

GOTO SEGVEL

CHKMENU MOVLW A'H
SUBWF SUBWF RCREG,W
BTFSS BTFSS STATUS,Z .Modo
manual?
GOTO CONTI
BSF BANDERA,3 .bit de
indicacion en modo manual
BSF BANDERA,2
GOTO CONTI

DATOX1 MOVLW A'b
DATOY11 XORWF RCREG,W
BTFSS BTFSS STATUS,Z .peticion de dato ok

DATHIX recibido?
GOTO SEGVEL
BSF REGLEER,1 .dato alto transmitido
GOTO SEGVEL

AUTO MOVLW AT
SUBWF RCREG,W
BTFSS STATUS,Z .modo automatico?
GOTO VERKANCEL
BSF BANDERA,2 .bit de indicacion
accion BCF BANDERA,3
BSF REGAPO,2
GOTO CONTI

READIX1 MOVLW A'a
XORWF RCREG,W
BTFSS BTFSS STATUS,Z .ok para mandar datos

a X ?
GOTO READY
BSF REGLEER,0
GOTO SEGVEL

VERKANCEL MOVLW A'K
SUBWF RCREG,W
BTFSS STATUS,Z .modo automatico?
GOTO CONPIC1
BCF BANDERA,1
BCF BANDERA,2
BCF REGAPO,4
BCF REGAPO,5
GOTO CONTI

READYI BTFSS REGLEER,3
GOTO READYI
BTFSS BTFSS REGLEER,1 .ya mando dato alto X

o de Y?
GOTO DATOY1
MOVLW A'C
XORWF RCREG,W
BTFSS BTFSS STATUS,Z .dato bajo transmitido

CONPIC1 MOVF RCREG,W
MOVWF REGPIC1
BSF BANDERA,2
GOTO CONTI

X?
GOTO SEGVEL
BSF REGLEER,2
BSF REGLEER,4 .dato x and y

transmitidos
BCF REGLEER,3
GOTO SEGVEL

DATOS DE cero pieza x o y

CERO BTFSS BANDERA,5 .mandando datos de cero pieza x
MANDAR MOVF MANDADY
RCREG,W
CALL TRANSMITIR
INCF CONTROL
BTFSS CONTROL,1 .se han mandado los
dos bytes de 0pzaX?
GOTO CONTI
GOTO SEGUIR1

DATOY1 MOVLW A'B
GOTO DATOY11
READY1 MOVLW A'A
XORWF RCREG,W
BTFSS BTFSS STATUS,Z .ok para mandar datos

a X ?
GOTO AUTO
GOTO FINX
BSF REGLEER,3
GOTO SEGVEL

SEGUIR1 BTFSS BANDERA,5 .se enviaron datos de CeroPiezaX?
GOTO YYYY
GOTO XXXX

FINX MOVLW A'd
SUBWF RCREG,W
BTFSS BTFSS STATUS,Z .coordenada X

alcanzada?
GOTO FINY
BSF REGLEER,5
GOTO SEGVEL

XXXX MOVLW 'X'
CALL TRANSMITIR
GOTO BORRAR

FINY MOVLW 11
XORWF RCREG,W
BTFSS BTFSS STATUS,Z
GOTO CAMHERR
BSF REGLEER,6 .coordenada Y

YYYY MOVLW 'Y'
CALL TRANSMITIR

alcanzada?
GOTO SEGVEL

BORRAR BCF BANDERA,5
BCF BANDERA,6
CLRf CONTROL
CALL CLDIR
GOTO CONTI

CAMHERR MOVLW A'g'
XORWF RCREG,W
BTFSS BTFSS STATUS,Z .cambio de

MANDADY BTFSS BANDERA,6 .bit de indicacion para mandar
0pzaY
GOTO READIX
GOTO MANDAR

herramienta listo?
GOTO KANATO
BSF AUTO23,3
GOTO SEGVEL

READIX BTFSS REGAPO,2 .modo automatico?
GOTO AUTO
MOVLW Z0
XORWF RCREG,W
BTFSS STATUS,Z .llego
BTFSS REGLEER,0 .ok para leer X datos?
GOTO READIX1
BTFSS REGLEER,1 .ya mando dato alto

KANATO MOVLW A'K'
SUBWF RCREG,W
BTFSS BTFSS STATUS,Z .modo automatico?
GOTO SEGVEL
CLRf AUTO23
BCF BANDERA,1
BCF BANDERA,2
BCF REGAPO,2
BSF AUTO23,5
GOTO SEGVEL

X?
GOTO DATOX1
MOVLW A'c
XORWF RCREG,W

Transmission de datos
TRANSMITIR MOVWF TXREG .dato a mandar
BSF STATUS,RP0 .banco 1
BTFSS TXSTA,TRMT .byte transmitido?
GOTO \$-1 .no, espera

ESTA TESIS NO SALE DE LA BIBLIOTECA

```

transmision      BCF      STATUS.RP0 ,banco 0
                  BCF      PIR1,TXIF  ,limpiando bit de fin de
                  RETURN
                  GOTO     CONTI
CONTI            RETFIE
    
```

```

INICIO          BSF      STATUS.RP0 ,banco 1
                  BCF      STATUS.RP1
                  CLRF     TRISD   ,puerto D como
salidas         CLRF     TRISB
                  BSF      TRISB.3 ,entrada para limite
superior        BSF      TRISB.4 ,entrada para limite
inferior        BSF      TRISC.0
                  BSF      TRISC.7
                  BSF      TRISC.6
                  CLRF     TRISE
                  CLRF     TRISA
    
```

```

digitales       MOVWLW  B'00000110' ,configuracion de PORTA<0-6>=I/O
entrada analogica MOVWF  ADCON1 ,porta<0> como
    
```

```

1:256          MOVWLW  B'10010100' ,CONFIGURACION DEL TIMER 0
MOVWF  OPTION_REG
MOVWLW  64
MOVWF  SPBRG ,a 19200 baudios
MOVWLW  B'00100100' ,Configuracion del USART:
    
```

```

Transmision     MOVWF  TXSTA ,Modo asincrono a alta velocidad
                  BSF      PIE1,RCIE ,Habilita interrupción por recepción
                  BCF
                  STATUS.RP0 ,banco 0
                  CLRF     BANDERA
                  CLRF     FLAG
                  CLRF     REGSPIC1
                  CLRF     PORTD
                  CLRF     PORTB
                  CLRF     PORTC
                  CLRF     PORTE
                  CLRF     REGHI
                  CLRF     REGLW
                  CLRF     CONDAT
                  CLRF     ADSHIX
                  CLRF     ADSLWX
                  CLRF     ADSHIY
                  CLRF     ADSLWY
                  CLRF     CONTROL
                  CLRF     RESTALW
                  CLRF     RESTAHI
                  CLRF     ZL
                  CLRF     ZH
                  CLRF     REGAPO
                  CLRF     TEMP23
                  CLRF     AUTO23
                  MOVWLW  25
                  MOVWF  VALOR
                  MOVWLW  .180
                  MOVWF  TMP1
                  CLRF     REGLEER
    
```

```

para RC0        MOVWLW  B'00000111' ,bit 3, modo contador de pulsos por
                  MOVWF  T1CON ,bit 0, activacion del timer1
                  CLRF     TMR1H
                  CLRF     TMR1L
USART recepción MOVWLW  B'10010000' ,Configuración del
recepción, continua MOVWF  RCSTA ,Por puerto serial (7), 8 bits de
MOVWLW  B'11000000' ,Habilitación de las interrupciones
MOVWF  INTCON
    
```

-----Ciclo de trabajo-----

```

CICLO          BTFSF  BANDERA.0 ,ya se conecto?
                  GOTO   CICLO   ,no, espera
    
```

```

GOTO           COMPC1
COMPC1         BTFSF  FLAG.0 ,conexion hecha?
                  GOTO   CHKDIR
                  MOVWLW A'0 ,mandando byte de accion
                  CALL   TRANSMITIR
                  BSF    FLAG.0 ,indicador de conexion hecha
CHKDIR         BTFSF  BANDERA.1 ,comunicacion con PIC1?
                  GOTO   CICLO ,no, espera que se le llame
                  BTFSF  BANDERA.2 ,bit de indicacion de accion PIC1
                  GOTO   CHKDIR
                  BTFSF  BANDERA.3 ,modo manual?
                  GOTO   ATMATIC
                  GOTO   MANUAL
ATMATIC        BTFSF  REGAPO.2 ,modo automatico?
                  GOTO   $-1
                  CLRF  ADSHIX ,limpiando direcciones de memoria
                  CLRF  ADSLWX ,alta y baja
                  CLRF  ADSHIY
                  CLRF  ADSLWY
                  CALL  CLRDIR
                  GOTO  LEERDAT
    
```

-----MODO MANUAL-----

```

MANUAL memoria BTFSF  FLAG.1 ,modo grabador de
                  GOTO   FOLLHAND
                  GOTO   WDATOS
FOLLHAND       BTFSF  BANDERA.4 ,ya se conecto a modo manual?
                  GOTO   CONMANU
                  GOTO   ACCION
CONMANU        BSF    BANDERA.4
                  MOVWLW A'1 ,mandando dato de confirmacion
                  GOTO   BACK
ACCION         BTFSF  CONTROL.7 ,se pide modo lectura?
                  GOTO   HANDZ
                  GOTO   READALL
HANDZ          MOVWLW A'1
                  SUBWF  REGPIC1,W
                  BTFSF  STATUS.Z ,se pide levantar eje
Z?             GOTO   ZNEGA
                  BSF    PORTB.0
                  BCF    PORTB.1
                  CALL  HOLD
                  INCF  RESTALW,F ,registro
de control para mandar dato
                  MOVWLW 255 ,correcto
                  SUBWF  RESTALW,W
                  BTFSF  STATUS.Z
                  GOTO   ZMASA
                  INCF  RESTAHI
ZMASA          BTFSF  PORTB.3 ,limite superior
                  GOTO   ZNEGATIVO
                  MOVWLW A''
                  CALL  TRANSMITIR ,señal de limite
superior alcanzado en modo
                  GOTO   APAGAR ,manual
ZNEGATIVO      BTFSF  PORTB.4 ,limite inferior
                  GOTO   VERCONTI
                  MOVWLW A''
                  CALL  TRANSMITIR ,señal de limite
superior alcanzado en modo MANUAL
                  GOTO   APAGAR
VERCONTI       BTFSF  REGAPO.4 ,movimiento continuo en Zpos?
                  GOTO   VERCONTNEG
                  CALL  CLRDIR
                  GOTO   ZMASA
VERCONTNEG     BTFSF  REGAPO.5 ,movimiento continuo
en Zneg?
                  GOTO   APAGAR
                  CALL  CLRDIR
                  GOTO   ZMASA
APAGAR         BCF    PORTB.0
                  BCF    PORTB.1
                  GOTO   BACK1
BACK           CALL  TRANSMITIR
BACK1          CALL  CLRDIR
                  GOTO   CICLO
    
```

-----Retardo de movimiento-----
 HOLD DEL1 CLRf TMR0
 OTRA1 BTfSS TMR0,5 ;TESTEO DEL BIT 7 DEL TEMP 0
 = 1?

SALTA GOTO OTRA1 HASTA QUE SEA VERDAD
 DECFSSZ TMP1,F ;DECREMENTA TMP1 Y TMP1=1?
 GOTO DEL1
 RETURN

ZNEGA MOVLW A'D
 SUBWF REGPIC1,W
 BTfSS STATUS,Z ;se pide bajar eye Z?
 GOTO ONMCA
 PORTB,0
 BSF PORTB,1
 CALL HOLD
 GOTO ZMASA

ONMCA MOVLW A'R'
 SUBWF REGPIC1,W
 BTfSS STATUS,Z ;se pide prender
 GOTO OFFMCA
 BSF PORTB,2 ;prendiendo MCA
 GOTO BACK1

OFFMCA MOVLW A'L'
 SUBWF REGPIC1,W
 BTfSS STATUS,Z ;se pidio apagar MCA?
 GOTO WDATOS
 GOTO OFF
 OFF BCF PORTB,2 ;apagando MCA
 GOTO BACK1

-----Grabando memoria-----
 WDATOS BTfSS CONDAT,0 ;se pide grabar dato?
 GOTO CEROPZA
 BTfSC FLAG,4 ;llego dato de

herramienta?
 GOTO FILLT
 BTfSC FLAG,5 ;llego dato de coordenada X?
 GOTO VERXX
 BTfSC FLAG,6 ;llego dato de Y?

COMPT1 GOTO VERY
 BSF PORTE,1
 MOVF ADSLWX,W
 GOTO PASAR
 FILLT BCF FLAG,4
 MOVF ADSLWX,W ;cargando direccion a W parte baja
 BSF PORTE,1
 CLRf REGHI
 GOTO PASAR

VERXX BCF BANDERA,2
 MOVLW A'X
 CALL TRANSMITIR
 BSF PORTE,1
 GOTO CICLO

VERY BTfSC CONDAT,6 ;primera y?
 GOTO COORDY
 GOTO FSTY

COORDY BCF BANDERA,2
 MOVLW A'Y
 CALL TRANSMITIR
 BSF PORTE,1
 BSF BANDERA,7 ;peticion de grabar Y en ADShI
 GOTO CICLO

FSTY BSF CONDAT,6 ;indicador de recibida fst y
 MOVLW 0x10 ;parte alta de la direccion de y
 MOVWF ADShIY
 BCF BANDERA,2
 MOVLW A'Y
 CALL TRANSMITIR
 GOTO NORMALY

PASAR BTfSS BANDERA,7 ;se pide grabar en parte alta?
 GOTO PARLW
 GOTO DATOY

-----Mandando parte baja de informacion de coordenada X-----

PARLW CALL DIRECC

CALL WRITEE
 BCF PORTE,0

-----Mandando parte alta de informacion de coordenada X-----

MOVF ADSLWX,W
 BCF PORTE,1 ;seleccionando eeprom alta con
 write enable
 CALL DIRECC
 CALL WRITEE
 BCF REGAPO,0
 BCF PORTA,2
 BCF PORTE,0
 BCF PORTE,1

-----Incrementado direcciones-----

MOVLW A'T
 SUBWF REGLW,W
 BTfSS STATUS,Z
 GOTO NOR
 BSF BANDERA,7 ;grabando tambien la T en Y
 MOVLW 0x10 ;parte alta de la direccion de y
 MOVWF ADShIY
 BCF BANDERA,2
 INCF ADSLWX ;REVISAR ESTA DIRECCION
 GOTO COMPT1

NOR MOVLW 0xFF
 SUBWF ADSLWX,W
 BTfSS STATUS,Z ;parte baja (memoria baja) llena?
 GOTO CONLWX ;o, continua llegando parte baja
 INCF ADShIY,F ;incrementa direccion parte alta
 GOTO CONLWX ;GOTO CHIDO

CONLWX MOVLW 0xFF
 SUBWF ADSLWX,W
 BTfSS STATUS,Z ;llena la direccion baja?
 GOTO CHIDO
 INCF ADShIY,F

CHIDO BTfSS BANDERA,7 ;datos Y?
 GOTO INCREX
 INCREX INCF INCREY
 GOTO OTRASS
 INCREY INCF ADSLWX,F
 OTRASS MOVLW A'W ;señal para mandara otro dato
 CALL TRANSMITIR
 BCF BANDERA,2
 BCF PORTE,0 ;poniendo en bajo negado output
 enable BCF BANDERA,7 ;apagando bandera de programar
 memoria alta GOTO CICLO

-----Datos de coordenada Y-----

DATOY MOVF ADSLWX,W
 BSF PORTE,1
 CALL DIRECC
 CALL WRITEE
 BCF PORTE,0

MOVF ADSLWX,W
 BCF PORTE,1 ;seleccionando

eeprom alta
 CALL DIRECC
 CALL WRITEE
 BCF PORTA,2
 BCF REGAPO,0
 GOTO CHIDO

-----Direccionando datos-----

DIRECC MOVWF PORTD
 BSF PORTA,1 ;enable latch
 NOP
 NOP
 NOP
 NOP
 BCF PORTA,1 ;disable latch 74LS373

BTfSS BANDERA,7 ;grabando datos en memoria alta?
 GOTO PARHIX ;no, continua parte baja de dato X
 GOTO PARHIY ;si, continua parte alta de datos Y

PARHIX MOVF ADShIX,W

```

GOTO PARHI
PARHIY MOVF ADSHIY,W
GOTO PARHI
PARHI MOVWF PORTD
BSF PORTA,5 ;enable latch de ads hi de eeprom
baja NOP
NOP
NOP
NOP
NOP
BCF PORTA,5 ;disable latch de ads hi
CALL LATCHEO
RETURN

```

-----Lacheando las direcciones-----

```

LATCHEO BTFS BANDERA,7 ;lacheando direcciones de datos
GOTO LACHX
GOTO LACHY
LACHX BTFS PORTE,1 ;se pide grabar en memoria alta?
GOTO EEHI
GOTO EELW
LACHY BTFS REGAPO,0
GOTO EELW
GOTO EEHI
EELW MOVLW B'00001000' ;Pines PORTA<0>->Conversor
MOVWF PORTA ;Pines PORTA<1>->Latch
CALL GRABAR1 ;activacion de latch adress con
flanco descendente del CS (negado)
BSF PORTA,3
RETURN REGAPO,0
EEHI MOVLW B'00000100' ;lacheando direcciones en eepromhi
MOVWF PORTA ;posible cambio por b'00001000'
CALL GRABAR1
RETURN

```

-----Ciclo de Escritura-----

```

WRITEE BTFS PORTE,1
GOTO WRHI
GOTO WRLW
WRLW MOVF REGLW,W
MOVWF PORTD
BCF PORTA,3 ;flanco ascendente para activacion
de latch de datos
BSF PORTE,0 ;bit para iniciar escritura OEN
CALL GRABAR1
RETURN
WRHI MOVF REGHI,W
MOVWF PORTD
BCF PORTA,2 ;latcheado informacion
BSF PORTE,0 ;porta 3 en lugar de 2
BSF PORTA,2 ;Seleccionando Chip Select para
EE2 CALL GRABAR1
RETURN

```

-----Retardo para el ciclo de escritura-----

```

GRABAR1 CLRF TMR0
BTFS TMR0,6
GOTO $-1
RETURN

```

-----TRANSMITIENDO DATOS DE MEMORIA A EEPROM A

```

PC-----
;--Parte baja de datos X
READALL CLRF ADSHX
CLRF ADSHIY
CLRF ADSLWX
CLRF ADSLWY
READALL1 MOVLW A'X'
CALL TRANSMITIR
MOVF ADSLWX,W
BSF PORTA,1 ;seleccionando memoria baja
NOP
NOP
CALL CHANGE
MOVWF B'00000000'
BSF PORTA ;-(Chip Enable1) activado

```

```

BCF PORTE,0 ;-(output enable1) activado
BCF PORTE,1 ;-(write enable1) activado
NOP
NOP
MOVF PORTD,W ;modo lectura
MOVWF REGLW ;datoXLW leído
CALL TRANSMITIR
CALL PTODEXIT ;Puerto D como salidas
MOVLW A@
SUBWF REGLW,W
BTFS STATUS,Z
GOTO VAAA
CALL CLRDIR
BCF CONTROL,7 ;apagando modo

```

```

lectura PC GOTO CICLO
;--Parte alta de datos X
VAAA MOVF ADSLWX,W
NOP
NOP
CALL CHANGE
MOVLW B'00000100'
MOVWF PORTA ;-(Chip Enable2)
activado BSF PORTE,1 ;-(write enable2)
activado
NOP
NOP
MOVF PORTD,W ;modo lectura
MOVWF REGHI ;datoXHI leído
CALL TRANSMITIR
BCF PORTE,1 ;modo normal
BCF PORTA,2
CALL PTODEXIT

```

```

;--Parte baja de datos Y
MOVLW A'Y'
CALL TRANSMITIR
MOVLW 0x10
MOVWF ADSHIY
MOVWF ADSLWY,W
BSF BANDERA,7 ;leyendo dato de Y
BSF PORTE,1 ;seleccionando
memoria baja BCF REGAPO,0
NOP
NOP
CALL CHANGE

```

```

MOVLW B'00000000'
MOVWF PORTA ;-(Chip Enable1)
BCF PORTE,0 ;-(output
enable1) activado
activado BCF PORTE,1 ;-(write enable1)
NOP
NOP
MOVF PORTD,W ;modo lectura
MOVWF REGLW ;datoXLW leído
CALL TRANSMITIR
CALL PTODEXIT ;Puerto D como
salidas

```

```

;--Parte alta de datos Y
NOP
NOP
MOVF ADSLWY,W
CALL CHANGE
MOVLW B'00000100'
MOVWF PORTA ;-(Chip Enable2)
activado BSF PORTE,1 ;-(write enable2)
activado
NOP
NOP
MOVF PORTD,W ;modo lectura
MOVWF REGHI ;datoXHI
leído
CALL TRANSMITIR

```

```

BCF PORTE,1 ;modo normal
BCF PORTA,2
BCF BANDERA,7

CALL PTODEXIT

INCF ADSLWX,F
MOVLW 0xFF
SUBWF ADSLWX,W
BTFS STATUS,Z ;leyo la parte baja de
ocho bit EE1?
GOTO AUMENYRE
INCF ADSHX,F
GOTO AUMENYRE

```

```

AUMENYRE INCF ADSLWY,F
MOVLW 0xFF
SUBWF ADSLWY,W
BTFS STATUS,Z
GOTO READALL1
INCF ADSHY,F
GOTO READALL1

CHANGE leer CALL DIRECC ;mandando direccion a
entradas NOP
NOP
CALL CAMBIAR ;Puerto D como
RETURN

```

-----LEYENDO DATOS DE MEMORIA PARA MANDAR A PICS-----

```

LEERDAT BTFS AUTO23,0 ;ya se mando informacion auto a
PIC2?
GOTO $-1
BTFS AUTO23,1 ;ya se mando
informacion auto a PIC3?
GOTO $-1
MOVLW A@'
CALL TRANSMITIR

```

-----Direccionando parte baja de informacion X-----

```

CICLOT MOVF ADSLWX,W
BSF PORTE,1 ;seleccionando memoria baja
NOP
NOP
CALL CHANGE

MOVLW B'00000000'
MOVWF PORTA ;-(Chip Enable1)
activado BCF PORTE,0 ;-(output enable1)
activado BCF PORTE,1 ;-(write enable1)
activado NOP
NOP
MOVF PORTD,W ;modo lectura
MOVWF REGLW ;datoXLW leído

CALL PTODEXIT ;Puerto D como salidas

MOVLW AT'
XORWF REGLW,W
BTFS STATUS,Z ;se pide cambio de herramienta?
GOTO FINPROG
MOVLW AT'
CALL TRANSMITIR
CALL PTODEXIT ;Puerto D como salidas
BCF PORTB,2
BCF PORTB,0
BCF PORTB,1

superior? BTFS PORTB,3 ;llego a limite
GOTO $-1
BCF PORTB,0 ;parando el motor
NOP
NOP
NOP
NOP
BSF PORTB,1 ;girando el motor en sentido
contrario ;para salir del sensor

CALL HOLDATO

```

```

BCF PORTB,1 ;parando el motor
CALL CLEARCONT
BTFS AUTO23,3 ;ya se cambio la
herramienta?
GOTO $-1
BCF AUTO23,3
GOTO VAVALU ;va a seguir sacando informacion de
ambas coordenadas

```

-----FIN DE PROGRAMA-----

```

FINPROG MOVLW A@'
XORWF REGLW,W
BTFS STATUS,Z
GOTO NOTOOL
CALL CLRDIR
BCF REGAPO,2
BSF BANDERA,3
BCF PORTB,2 ;prendiendo motor de
CA salidas CALL PTODEXIT ;Puerto D como
ejecutado MOVLW A>'
CALL TRANSMITIR ;fin de programa
GOTO CICLO

```

-----Mandando parte alta de informacion de X a PIC2

```

NOTOOL BSF PORTB,2
NOP
NOP
NOP
MOVF ADSLWX,W
NOP
NOP
CALL CHANGE

MOVLW B'00001000'
MOVWF PORTA ;-(Chip Enable2)
activado BSF PORTE,1 ;-(write enable2)
activado NOP
NOP
MOVF PORTD,W ;modo lectura
MOVWF REGHI ;datoXHI leído
BCF PORTE,1 ;modo normal
BCF PORTA,2
CALL PTODEXIT

MOVLW A?'
CALL TRANSMITIR
CALL HOLD
MOVLW A'X
CALL TRANSMITIR
BTFS REGLEER,0 ;dato ok X transmitido?
GOTO $-1
GOTO OKX
AHORAY BTFS REGLEER,3 ;dato ok de Y
transmitido?
OKX GOTO $-1
MOVF REGHI,W
CALL TRANSMITIR
CALL HOLD
BTFS REGLEER,1 ;dato XHI ó YHI recibido?
GOTO $-1
MOVF REGLW,W
CALL TRANSMITIR
CALL HOLD
BTFS REGLEER,2 ;dato XLW recibido?
GOTO $-1
BTFS REGLEER,4 ;dato X y Y recibido?
GOTO TRANSY
GOTO ESPERANDO

```

-----Transmitiendo datos de Y-----

-----Mandando parte baja de informacion Y a PIC3

```

TRANSY BTFS AUTO23,2 ;siempre direccionando para
0x1000
GOTO LEEFSTY
GOTO LEENORY ;sigue leyendo normal

```


CLEARCONT	BCF T1CON.2		CALL GOTO	TRANSMITIR CICLO	
	NOP		BAJARSUBIR	BCF PORTB.0	,parando el motor
	NOP		NOP	NOP	
	NOP		NOP	NOP	
	BCF T1CON.0		BSF	PORTB.1	,girando el motor en sentido contrario
	NOP				
	NOP				
	NOP				
	CLRF TMR1L		CALL BCF RETURN	HOLDATO PORTB.1	para salir del sensor
	NOP				
	NOP				
	NOP				
	CLRF TMR1H		COPZAY	MOVLW A';	
	BSF T1CON.2		SUBWF REGPIC1.W	STATUS.Z	
	NOP		BTFSS STATUS.Z	LIMNEGX	
	NOP		GOTO BANDERA.6		,indicador de señal de transmision de datos de OpzaY continua
	NOP		BSF BANDERA.2		
	NOP		BCF BANDERA.2		
	NOP		MOVLW A';		
	NOP		CALL TRANSMITIR		
	BSF T1CON.0		GOTO	CICLO	
	RETURN				
-----Cero pieza Z-----					
CEROPZA	MOVLW A'7		LIMNEGX x?	MOVLW A'm'	,llego fin de limite inferior alcanzado
para eje Z?	SUBWF REGPIC1.W		SUBWF REGPIC1.W		
	BTFSS STATUS.Z	,se pide grabar la posicion de O pza	BTFSS STATUS.Z		
	GOTO HOME		GOTO LIMNEGY		
	MOVF TMR1H.W		CALL CLRDIR		
	MOVWF ZH		MOVLW A'm'		
	MOVF RESTAHL.W		CALL TRANSMITIR		,mandando informacion a PC para registrar el
	SUBWF ZH.W	,mandando dato real parte alta	GOTO	CICLO	,limite inferior alcanzado
	CALL TRANSMITIR		LIMNEGY y?	MOVLW 0x03	,llego fin de limite inferior alcanzado
	MOVF TMR1L.W		SUBWF REGPIC1.W		
	MOVWF ZL		BTFSS STATUS.Z		
	MOVF RESTALW.W		GOTO HANDX		
	SUBWF ZL.W	,mandando dato real parte baja	CALL CLRDIR		
	CALL TRANSMITIR		MOVLW A'm'		
	MOVLW A'8		CALL TRANSMITIR		,mandando informacion a PC para registrar el
	CALL TRANSMITIR		GOTO	CICLO	,limite inferior alcanzado
	CALL CLEARCONT		HANDX	MOVLW A'7	
	GOTO BACK1				
HOME	MOVLW A'8		PC	SUBWF REGPIC1.W	
	SUBWF REGPIC1.W			BTFSS STATUS.Z	
	BTFSS STATUS.Z			GOTO HANDY	
	GOTO COPZAX			MOVLW A'7	,mandando dato de OK modo manual PIC2
	BSF PORTB.0			CALL TRANSMITIR	
	BCF PORTB.1			BSF PORTB.5	
	CALL CLRDIR			GOTO BACK1	
superior?	BTFSS PORTB.3	,llego a limite	HANDY	MOVLW A'7	
	GOTO \$-1				
	CALL BAJARSUBIR				
	CALL CLEARCONT				
	MOVLW A'7				
	GOTO BACK				
CLRDIR	BCF BANDERA.1		OK modo manual PIC3 PC	XORWF REGPIC1.W	
	BCF BANDERA.2			BTFSS STATUS.Z	
	RETURN			GOTO LIMPIAR	
				MOVLW A'7	
				CALL TRANSMITIR	,mandando dato de
				GOTO BACK1	PORTB.5
				BSF	
COPZAX	MOVLW A'5		LIMPIAR	CALL CLRDIR	
	SUBWF REGPIC1.W			GOTO	CICLO
	BTFSS STATUS.Z	,se pide transmitir opzax?		END	
	GOTO COPZAY				
	BSF BANDERA.5				
	BCF BANDERA.2				
	MOVLW A'5				

Programas en ensamblador para el μC PIC16F73 (eje X)

```
LIST P=16F73
RADIX HEX
INCLUDE "P16F73.INC"
__CONFIG H'3D3A'
```

-----Registros de uso general-----

```
BANDERA2 EQU 0x20
RESTALW EQU EQU 0x21
RESTAHI EQU EQU 0x22
REGPIC2 EQU EQU 0x23
VELOC1 EQU EQU 0x24
TMP1 EQU EQU 0x25
TMP0 EQU EQU 0x26
BPIC2 EQU EQU 0x27
XH EQU EQU 0x28
XL EQU EQU 0x29
DATHIX EQU EQU 0x30
DATLWX EQU EQU 0x31
COMPARALW EQU EQU 0x32
COMPARAHI EQU EQU 0x33
CONTAR EQU EQU 0x34
DATLWXN EQU EQU 0x35
TIEMPO EQU EQU 0x36
SALHOME EQU EQU 0x2A

ORG 0x00 INICIO
GOTO INICIO
ORG 0x05
```

-----Atencion al vector de Interrupcion-----

```
INTER BTFSS PIR1,RCIF ;interrupcion por recepcion?
GOTO CONTI
BCF PIR1,RCIF ;DOS ;si, restaura flag de
BTFSS BPIC2,0 ;ya llego la direccion

0x32 GOTO REVDIR
GOTO PKDATO

REVDIR MOVLW 0x32 ;direccion PIC2 H'0x32?
SUBWF RCREG,W
BTFSS STATUS,Z
GOTO CONTI ;indicador de direccion ok!
BSF BPIC2,0
GOTO CONTI

PKDATO MOVLW A'K ;se pidio cancel
SUBWF RCREG,W
BTFSS STATUS,Z
GOTO MENUS
CLRF BANDERA2 ;activacion de modo
BSF BANDERA2,1

manual BSF BANDERA2,0
CLRF PORTA
BCF PORTC,4
BSF CONTAR,3
GOTO CONTI

MENUS BTFSS BANDERA2,0 ;esta en modo
manual? GOTO CHKATO
GOTO SPEDMAS

CHKATO BTFSS BANDERA2,2 ;esta en modo
automatico? GOTO CHKMEN2
GOTO DATMEMX

SPEDMAS BTFSS BPIC2,7 ;esta en maxima velocidad?
GOTO VELMAS
GOTO SPEDNEG

VELMAS MOVLW A'V ;se pide aumentar velocidad
XORWF RCREG,W
BTFSS STATUS,Z
GOTO SPEDNEG
MOVLW 1
SUBWF TMP1,F
MOVF TMP1,W
MOVWF VELOC1 ;velocidad maxima?
BTFSS STATUS,Z
GOTO SEGVEL
BSF BPIC2,7
GOTO SEGVEL
```

```
SPEDNEG MOVLW A'V
XORWF RCREG,W
BTFSS STATUS,Z
GOTO SEEMOVPOS
MOVLW 1
ADDWF TMP1,F
MOVF TMP1,W
MOVWF VELOC1
BTFSS STATUS,Z ;velocidad minima?
GOTO SEGVEL
BCF BPIC2,7
MOVLW 255
MOVWF TMP1

SEGVEL CALL CLRDIR2
GOTO CONTI

SEEMOVPOS MOVLW 8
XORWF RCREG,W
BTFSS STATUS,Z ;movimiento continuo positivo?
GOTO SEEMOVNEG
BSF BANDERA2,5 ;indicando movimiento
continuo positivo
BSF BPIC2,1 ;accion indicada
BCF BANDERA2,6 ;apagando bandera de
movimiento continuo negativo
CALL CLEAR
MOVLW A'M
MOVWF REGPIC2
GOTO CONTI

SEEMOVNEG MOVLW 7
XORWF RCREG,W
BTFSS STATUS,Z ;movimiento continuo negativo?
GOTO CANCELAR
BSF BANDERA2,6 ;indicando movimiento
continuo negativo
BSF BPIC2,1 ;accion indicada
BCF BANDERA2,5 ;apagando
movimientos positivos continuos
CALL CLEAR
MOVLW A'N
MOVWF REGPIC2
GOTO CONTI

CANCELAR MOVLW 5
XORWF RCREG,W
BTFSS STATUS,Z ;se pide cancelar movimiento
continuos
GOTO AUTO2 ;DATOMEMX
BCF BANDERA2,5 ;apagando movimientos continuos
BCF BANDERA2,6 ;apagando movimientos continuos
CALL CLEAR
CALL CLRDIR2
GOTO CONTI

CLEAR BCF BANDERA2,4 ;limpiando datos de primera vez
BCF BPIC2,5 ;que paso por movimiento continuo
RETURN

DATO1 MOVF RCREG,W
MOVWF DATHIX
BSF BPIC2,3
MOVLW A'1
CALL TRANSMITIR
MOVWF A'b ;mandando dato que esta listo para
recibir CALL TRANSMITIR ;datos
GOTO CONTI

DATO2 MOVF RCREG,W
MOVWF DATLWX
MOVLW 0
SUBWF DATLWX,W
BTFSS STATUS,Z ;se pide quedarse en su misma
posicion? GOTO DIFERR
GOTO NOMOVE

DIFERR MOVF 1
SUBWF DATLWX,W
MOVWF DATLWXN

NOMOVE MOVLW A'1
CALL TRANSMITIR ;mandando dato que esta listo para
MOVWF A'c recibir CALL TRANSMITIR ;datos
BCF BPIC2,3 ;apagando bandera de ultimo dato
cachado BCF BPIC2,2 ;apagando bandera de grabando
dato de memoria BSF BANDERA2,3 ;indicador de que estamos en un
ciclo automatico
```

```

CALL CLDIR2
GOTO CONTI

START MOVWL 2
SUBWF RCREG.W
BTSS STATUS.Z
GOTO SEGVEL
BSF CONTA.R.2 ,comenzar movimiento automatico
CALL CLDIR2
GOTO CONTI

DATMEMX BTSS BPIC2.2 ,se estan grabando datos?
GOTO GRABAR
BTSS BPIC2.3 ,ya llego dato bajo de informacion?
GOTO DATO1
GOTO DATO2

GRABAR MOVWL A'x
XORWF RCREG.W
BTSS STATUS.Z ,llego peticion de grabar datos de
memoria ?
GOTO START
BSF BPIC2.2 ,bandera de
colectando datos
MOVLW A'1
CALL TRANSMITIR
MOVLW A'a ,mandando dato que esta listo para
recibir
CALL TRANSMITIR
GOTO CONTI ,datos

SEG_MAN MOVF RCREG.W
MOVWF REGPIC2
BSF BPIC2.1
GOTO CONTI

CHKMEN2 MOVWL A'H
SUBWF RCREG.W
BTSS STATUS.Z ,modo manual?
GOTO CONTI
BSF BANDERA2.0
BSF BPIC2.1 ,indicador de accion
GOTO CONTI

AUTO2 MOVWL A'T ,peticion de modo
automatico
SUBWF RCREG.W
BTSS STATUS.Z
GOTO SEG_MAN
BSF BANDERA2.2 ,indicador de modo automatico
BSF BANDERA2.0
BSF BPIC2.1 ,accion indicada
GOTO CONTI

-----Transmision de Datos-----

TRANSMITIR MOVWF TXREG ,dato a mandar
BSF STATUS.RP0 ,banco 1
BTSS TXSTA.TRMT ,byte transmitido?
GOTO $+1 ,no, espera
BSF STATUS.RP0 ,banco 0
BSF PIR1.TXIF ,limpiando bit de fin de
transmision
RETURN

CONTI RETFIE

```

```

INICIO BSF STATUS.RP0 ,banco 1
CLRF TRISB
MOVLW B'10000011' ,CONFIGURACION
DEL TIMER 0 A UNA ESCALA DE 1 128
MOVWF OPTION_REG

MOVLW B'00000101' ,SALIDAS COOMO
ANALOGICA PUERTO RA0,RA1
MOVWF ADCON1 ,Y VOLTAJE DE
REFERENCIA VREF+ VDD,PIN RA3

,Salidas Digitales, RA2,RA5,RE0,RE1
,POR RA2,RA5, salidas a motor de pasos
justificacion left los 6 bits mas significativos
,son leidos en ADRESH

```

```

CLRF TRISA
MOVLW 0xFF

```

```

MOVWF TRISC
BCF TRISC.4 ,salida a driver
.BSF TRISC.7 ,Configuracion para recepcion y
.BSF TRISC.6 ,transmision de datos
.BSF TRISC.0 ,Pin para recibir los pultos del
encoder
.BCF TRISC.2 ,limite inferior x-
.BCF TRISC.3 ,limite superior x+

MOVLW 64 ,a 19200 baudios
MOVWF SPBRG
MOVLW B'00100100' ,Configuracion del USART
Transmision MOVWF TXSTA ,Modo asincrono a alta velocidad
BSF PIE1.RCIE ,Habilita interrupcion por recepcion

```

-----Configuracion del TIMER1 como entrada digitales-----

```

BCF STATUS.RP0 ,banco 0
pata RC0 MOVWL B'00000111' ,bit 3, modo contador de pulsos por
timer1 MOVWF T1CON ,bit 0, activacion del
CLRF TMR1H
CLRF TMR1L
CLRF BANDERA2
CLRF REGPIC2
CLRF DATLWX
CLRF DATHIX
CLRF PORTB
CLRF PORTA
CLRF PORTC
CLRF BPIC2
CLRF TMP1
CLRF RESTAHI
CLRF RESTALW
CLRF VELOC1
CLRF XL
CLRF XH
CLRF CONTA.R
CLRF DATLWXN
CLRF COMPARALW
CLRF COMPARAH
CLRF SALHOME
BCF PORTC.4 ,deshabilitando el driver del motor
MOVLW B'10010000'
MOVWF RCSTA ,Por puerto serial (7), 8 bits de
recepcion, continua
MOVLW B'11000000' ,Habilitacion de las
interrupciones, y por cambio en PB<4-7>
MOVWF INTCON
MOVLW B'10000101' ,FOSC/32
SELECCION DE CANAL 0 (RA0/AN0)
MOVWF ADCON0 ,habilita la conversion
MOVLW 100 ,velocidad de motores
MOVWF TMP1
MOVWF VELOC1
MOVWF TIEMPO
MOVLW 255
MOVWF SALHOME

```

-----Ciclo de trabajo-----

```

CICLO BTSS BPIC2.0 ,se pide comunicacion con PIC2?
GOTO CICLO ,no, espera
GOTO COMPC2

COMPC2 BTSS BPIC2.1 ,bit de accion indicado?
GOTO CICLO
BTSS BANDERA2.0 ,modo manual?
GOTO VERAUTO
BTSS BANDERA2.1 ,ya se conecto a modo actual?
GOTO CONMAN2
GOTO MANUAL2

CONMAN2 MOVWL A'1
CALL TRANSMITIR
MOVLW A'1 ,dato para indicar que se comunico
a modo manual
CALL TRANSMITIR
CALL CLDIR2
BSF BANDERA2.1 ,indicador de modo
actual
GOTO CICLO

VERAUTO BTSS BANDERA2.2 ,se pide modo
automatico?
GOTO CICLO
CALL CLDIR2
GOTO CONAUTO

```

-----TABLA DE DATOS DE MOTOR X-----

TABLAX	ADDWF	PCL.F	
	RETLW	36	,12
	RETLW	32	,9
	RETLW	0	,3
	RETLW	4	,6

-----MODO AUTOMATICO-----

REG_AUTO AUTOMATICO	CALL BTFS	CLEARCONT DATHIX,4	, sentido horario de
moteres?	GOTO	CONTNEGA	,no. controles de
posicion negativo	GOTO	SEGATO	,si. controles de
posicion positivo			
SEGATO	BTFS	CONTAR,3	,cancelar automatico?
	GOTO	OKIPOS	
	CALL	CANTTT	
	GOTO	CICLO	
OKIPOS	BTFS	BPIC2,4	,alcanza parte baja de informacion?
	GOTO	LOADLW	
	GOTO	LOADHI	
LOADLW	MOVF	TMR1L,W	
	MOVWF	COMPARALW	
	MOVF	COMPARALW,W	
	SUBWF	DATLWX,W	
	BTFS	STATUS,Z	; DATLWX=COMPARALW (Z =
1)??	GOTO	MAYOR	
	CALL	MANTEN	
	GOTO	LOADHI	
MAYOR	BTFS	STATUS,C	,DATLW > TMR1L (C = 1)??
	GOTO	MENOR	,No. -> DATLW > TMR1L
	CALL	MANTEN	
	GOTO	LOADHI	,Si. -> TMR1L > DATLW
MENOR	MOVF	TMR1L,W	
	MOVWF	COMPARALW	
	MOVF	COMPARALW,W	
	SUBWF	DATLWX,W	
	BTFS	STATUS,Z	,alcanzo el dato con MENOS
resolucion?	GOTO	SIGNO	
	CALL	MANTEN	
LOADHI informacion	BSF	BPIC2,4	,a alcanzado parte baja de
	MOVF	TMR1H,W	
	MOVWF	COMPARAHI	
	MOVF	COMPARAHI,W	
	SUBWF	DATHIX,W	
	BTFS	STATUS,Z	,alcanzo COMPARAHI=DATHIX?
	GOTO	ALAMAYOR	
	CALL	MANTEN	
	GOTO	FINCOOR	
ALAMAYOR = 0)?	BTFS	STATUS,C	,TMR1H > DATHIX (C
TMR1H	GOTO	SIGNO	,No. DATHIX >
	GOTO	FINCOOR	,Si. TMR1H > DATHIX
SIGNO	BTFS	CONTAR,1	,Movimiento negativo?
	GOTO	AVANXPOS	,No. va a movimiento
positivo	GOTO	AVANXNEG	,Si. va a movimiento
negativo			
FINCOOR	MOVLW	A'1	,si. trasmite a
PICMAESTRO	CALL	TRANSMITIR	,informacion sobre su
llegada	MOVLW	A'd	,a la coordenada
solicitada	CALL	TRANSMITIR	
	CALL	CLEARCONT	
	BCF	BPIC2,4	,resetendo informacion
	BCF	BANDERA2,3	
	BCF	PORTC,4	,diable driver motor
	BCF	CONTAR,1	,limpiando el signo
	BCF	CONTAR,2	,cancelar movimiento
automatico	CALL	CLRDIR2	
	GOTO	CONAUTO	

AVANXPOS	CALL	HOMEX1	,va a avanzar X+
automatico	GOTO	SEGATO	,va a continuar el ciclo de
AVANXNEG	CALL	XNEGATO1	
	GOTO	SEGATONEG	
CONTNEGA	BSF	CONTAR,1	,movimiento negativo de motores
	MOVLW	0x0F	
	ANDWF	DATHIX,F	
SEGATONEG	BTFS	CONTAR,3	,cancelar automatico?
	GOTO	OKINEGA	
	CALL	CANTTT	
	GOTO	CICLO	
OKINEGA	MOVF	TMR1L,W	
	MOVWF	COMPARALW	
	MOVF	COMPARALW,W	
	SUBWF	DATLWX,W	
	BTFS	STATUS,Z	,alcanzo DATLWX == TMR1L (Z =
1)??	GOTO	MAYOR	
	CALL	MANTEN	
	GOTO	LOADHI	
CANTTT	CALL	CLRDIR2	
	BCF	BANDERA2,2	
	BCF	BANDERA2,3	
	BSF	BANDERA2,0	
	BCF	CONTAR,2	
	BCF	CONTAR,3	
	RETURN		
MANTEN	CALL	HOLD	
	DECF	TIEMPO,F	
	GOTO	MANTEN	
	MOVLW	.7	
	MOVWF	TIEMPO	
	RETURN		

-----Ciclo de espera en modo automatico-----

CONAUTO automatico?	BTFS	BANDERA2,3	,modo de ciclo
	GOTO	\$-1	
	BTFS	CONTAR,2	,orden de comenzar movimientos?
	GOTO	\$-1	
	GOTO	AUTOMATICO	

-----MODO MANUAL-----

MANUAL2	MOVLW	A'M	
	SUBWF	REGPIC2,W	
	BTFS	STATUS,Z	,se pide mover eje x
+?	GOTO	XNEGA	
XNEG	BSF	PORTC,4	,enable driver motor
HOMEX1	DECF	TMPO,F	
HOMEX	GOTO	MOTORX	
BACK	BTFS	BANDERA2,7	
	GOTO	BACK1	
BACK2	CALL	TRANSMITIR	
BACK1	BCF	PORTC,4	,diable driver motor
	CALL	CLRDIR2	
	GOTO	CICLO	
MOTORX	CALL	HOLD	
	MOVF	TMPO,W	
	ANDLW	3	
	CALL	TABLAX	
	MOVWF	PORTA	
	INF	BTFS	PORTC,2
	GOTO	SUPX	
	BCF	PORTC,4	,diable motor driver
	BCF	BANDERA2,7	
	MOVLW	A'1	,no hya necesidad de direccionar
la informacion	CALL	TRANSMITIR	,dado que unicamente
se comunica con PIC1	MOVLW	A'm	,indicando que llego al limite de +X
	CALL	TRANSMITIR	
	BTFS	BANDERA2,3	,estoy en modo
automatico?	GOTO	MANX	
MANX	GOTO	BACK	
SUPX	BTFS	CONTAR,6	,ya paso por el sensor de home?
	GOTO	FSTHOME	

```

GOTO EXITHOM
FSTHOME BTFSS PORTC,3
GOTO HOX
BCF PORTC,4 ,disable motor driver
BCF BANDERA2,7
BSF CONTAR,6 ,indicador de home
pasado CLR

EXITHOM DECFSZ SALHOME
GOTO XNEG
BCF CONTAR,6

MOVW A'1'
CALL TRANSMITIR
MOVW A'n' ,dato que llego a
CALL TRANSMITIR
CLR RESTAHI
CLR RESTALW
CALL CLEARCONT

GOTO BACK
HOX BTFSS BANDERA2,3 ,estoy en modo
automatico?
GOTO REVHOME
RETURN

REVHOME BTFSS BANDERA2,7 ,se pide home?
GOTO VERCONPOS
GOTO HOMXNEG

VERCONPOS BTFSS BANDERA2,5 ,se pide
movimiento continuo positivo?
GOTO VERCONNNEG
GOTO FST

FST BTFSS BPIC2,5 ,ya paso por aqui primera vez?
GOTO FST2
GOTO HOMEX

FST2 BSF BPIC2,5
CALL CLRDIR2
GOTO HOMEX

VERCONNNEG BTFSS BANDERA2,6 ,se pide
movimiento continuo negativo?
GOTO BACK1
BACK1 BTFSS BANDERA2,4 ,ya paso por aqui primera vez?
GOTO FSTVEZ
GOTO XNEGATO

FSTVEZ BSF BANDERA2,4
CALL CLRDIR2
GOTO XNEGATO

-----Rutina de Retardo y modulo de control de avance de motores-----
---
HOLD CLR TMR0
BTFSS TMR0,5 ,TESTEO DEL BIT 7 DEL TEMP 0
= 1?
GOTO $-1 ,HASTA QUE SEA VERDAD
BCF PIR1,ADIF ,LIMPIA LA BANDERA DE
INTERRUPCION DEL
BSF ADCON0,GO ,TERMINO DE LA
CONVERSION, inicia la conversin

.ESPERA BTFSS PIR1,ADIF ,TERMINO LA CONVERSION?
GOTO $-1
MOVF ADRES,W
MOVW 200
MOVWF TMP1 ,REGISTRO DE TRABAJO PARA
RETARDADOR

DEL1 CLR TMR0
BTFSS TMR0,5 ,TESTEO DEL BIT 7 DEL TEMP 0
= 1?
GOTO $-1 ,HASTA QUE SEA VERDAD
DECFSZ TMP1,F ,DECREMENTA TMP1 Y TMP1=17,
SALTA
GOTO DEL1
MOVF VELOC1,W
MOVWF TMP1
RETURN

-----Limpiando el contador-----
CLEARCONT BCF T1CON,2

NOP
NOP
NOP

```

```

BCF T1CON,0
NOP
NOP
NOP
CLR TMR1L
NOP
NOP
NOP
CLR TMR1H
BSF T1CON,2
NOP
NOP
NOP
BSF T1CON,0
RETURN

-----
XNEGA MOVW A'N'
SUBWF REGPIC2,W
BTFSS STATUS,Z
GOTO CERPZA2 ,se pide mover eje x -?

XNEGATO1 BSF PORTC,4
XNEGATO INCF TMP0,1
HOMXNEG INCF RESTALW,F
MOVW 255
SUBWF RESTALW,W
BTFSS STATUS,Z
GOTO MOTORX
INCF RESTAHI
GOTO MOTORX

CERPZA2 MOVW A'7'
SUBWF REGPIC2,W
BTFSS STATUS,Z ,se pide grabar la posicion de O pza
para eje Z?
GOTO HOME
MOVW A'1'
CALL TRANSMITIR
MOVW A'3'
CALL TRANSMITIR ,señal de cero pieza x
CALL HOLD
MOVF TMR1H,W
MOVWF XH
MOVWF RESTAHI,W
SUBWF XH,W
CALL TRANSMITIR
CALL HOLD
MOVF TMR1L,W
MOVWF XL
MOVWF RESTALW,W
MOVWF XL,W
SUBWF XL,W
CALL TRANSMITIR

HOME MOVW A'8'
SUBWF REGPIC2,W
BTFSS STATUS,Z
GOTO BACK
BSF BANDERA2,7 ,indicador de home
BSF PORTC,4 ,enable driver motor
CALL CLRDIR2
GOTO HOMXNEG

CLRDIR2 BCF BPIC2,0
BCF BPIC2,1

END

```

Programas en ensamblador para el μ C PIC16F73 (eje Y)

```
LIST      P=16F73
RADIX    HEX
INCLUDE  "P16F73.INC"
__CONFIG H'3D3A'
```

-----Registros de uso general-----

```
BANDERA3 EQU      0x20
RESTALW  EQU      0x21
RESTAHI  EQU      0x22
REGPIC3  EQU      0x23
VELOC1   EQU      0x24
TMP1     EQU      0x25
TMP0     EQU      0x26
BPIC3    EQU      0x27
YL       EQU      0x28
YH       EQU      0x29
DATHIY   EQU      0x30
DATLWY   EQU      0x31
DATLWYN  EQU      0x32
CONTAR   EQU      0x33
COMPARALW EQU      0x34
COMPARAHI EQU      0x35
TIEMPO   EQU      0x36
SALHOME  EQU      0x2A
```

```
ORG      0X00
GOTO     INICIO
ORG      0x05
```

-----Atencion al vector de Interrupcion-----

```
INTER    BTFS    PIR1,RCIF      ;interrupcion por
recepcion?
        GOTO    CONTI           ;no,
regresa
        BCF    PIR1,RCIF      ;si, restaura flag de
0x32     BTFS    BPIC3,0        ;ya llego la direccion
        GOTO    REVDIR
        GOTO    PKDATO
REVDIR   MOVLW   0x33           ;direccion PIC3
SUBWF   RCREG,W
BTFS    STATUS,Z
H0x33?   GOTO    CONTI
        BSF    BPIC3,0        ;indicador de direccion
ok!      GOTO    CONTI
PKDATO   BTFS    CONTAR,4      ;grabar registro de memoria?
        GOTO    REVTRES
        GOTO    DATMEMY
REVTRES  MOVLW   A'3           ;llego otro tres?
SUBWF   RCREG,W
BTFS    STATUS,Z
        GOTO    OKI
        GOTO    CONTI
OKI      MOVLW   A'K           ;se pidio kancel
SUBWF   RCREG,W
BTFS    STATUS,Z
        GOTO    MENUS
        CLRF   BANDERA3
        BSF   BANDERA3,1
        BSF   BANDERA3,0
        CLRF   PORTA
        BCF   PORTC,4
        BSF   CONTAR,3
        GOTO   CONTI
MENUS    BTFS    BANDERA3,0    ;esta en
modo manual?
        GOTO   CHKATO
        GOTO   VELMAS
CHKATO   BTFS    BANDERA3,2    ;esta en
modo automatico?
        GOTO   CHKMEN3
        GOTO   DATMEMY
VELMAS   MOVLW   A'V           ;se pide aumentar velocidad
XORWF   RCREG,W
BTFS    STATUS,Z
```

```
GOTO    SPEDNEG
MOVLW   1
SUBWF   TMP1,F
MOVF    TMP1,W
MOVWF   VELOC1
GOTO    SEGVEL
SEPNEG   MOVLW   A'V
XORWF   RCREG,W
BTFS    STATUS,Z
GOTO    SEEMOVPOS
MOVLW   1
ADDWF   TMP1,F
MOVF    TMP1,W
MOVWF   VELOC1
GOTO    SEGVEL
SEGVEL   CALL    CLRDIR3
        GOTO    CONTI
SEEMOVPOS MOVLW   A'q'
XORWF   RCREG,W
BTFS    STATUS,Z ;movimiento continuo positivo?
GOTO    SEEMOVNEG
        BSF    BANDERA3,5 ;indicando movimiento
continuo positivo
        BSF    BPIC3,1 ;accion indicada
        BCF    BANDERA3,6 ;apagando bandera de
movimiento continuo negativo
        CALL   CLEAR
        MOVLW A'Q'
        MOVWF REGPIC3
        GOTO   CONTI
SEEMOVNEG MOVLW   A'p'
XORWF   RCREG,W
BTFS    STATUS,Z ;movimiento continuo negativo?
GOTO    CANCELAR
        BSF    BANDERA3,6 ;indicando movimiento
continuo negativo
        BSF    BPIC3,1 ;accion indicada
        BCF    BANDERA3,5 ;apagando
movimientos positivos continuos
        CALL   CLEAR
        MOVLW A'P'
        MOVWF REGPIC3
        GOTO   CONTI
CANCELAR MOVLW   5
XORWF   RCREG,W
BTFS    STATUS,Z ;se pide cancelar movimiento
continuos
        GOTO   AUTO3
        BCF   BANDERA3,5 ;DATMEMY
        ;apagando
movimientos continuos
        BSF    BANDERA3,6 ;apagando
movimientos continuos
        CALL   CLEAR
        CALL   CLRDIR3
        GOTO   CONTI
CLEAR    BCF    BANDERA3,4
        BCF    BPIC3,5
RETURN
DATOY1   MOVF    RCREG,W
        MOVWF DATHIY
        BSF    BPIC3,3 ;recibiendo dato alto
        MOVLW A'1'
        CALL   TRANSMITIR
        MOVLW A'B' ;mandando dato que esta listo para
recibir
        CALL   TRANSMITIR ;datos
        GOTO   CONTI
DATOY2   MOVF    RCREG,W
        MOVWF DATLWY
        MOVLW 0
        SUBWF DATLWY,W
        BTFS    STATUS,Z
        GOTO   DIFERR
        GOTO   NOMOVE
DIFERR   MOVLW   .1
SUBWF   DATLWY,W
MOVWF   DATLWYN
NOMOVE   MOVLW   A'1'
        CALL   TRANSMITIR
        MOVLW A'C' ;mandando dato que esta listo para
recibir
        CALL   TRANSMITIR ;datos
        BCF   BPIC3,3 ;apagando bandera de ultimo dato
cachado
```

```

BCF      BPIC3,2      ;apagando bandera de grabando
dato de memora
BSF      BANDERA3,3  ;indicador de que
estamos en un ciclo automatico
CALL    CLRDIR3
GOTO    CONTI

START    MOVLW      3
SUBWF   RREG.W
BTFSS   STATUS,Z
GOTO    SEGVEL
BSF     CONTAR,2 ;inicia movimientos automaticos
CALL    CLRDIR3
GOTO    CONTI

DATMEMY BTFSS   BPIC3,2 ;,se estan grabando datos?
GOTO    GRABAR
BTFSS   BPIC3,3 ;,llego dato bajo de informacion?
GOTO    DATOY1
GOTO    DATOY2

GRABAR  MOVLW      A'Y
XORWF   RREG.W
BTFSS   STATUS,Z ;,llego peticion de grabar datos de
memoria ?
GOTO    START
BSF     BPIC3,2 ;,bandera de colectando datos
MOVLW   A'1'
CALL    TRANSMITIR
MOVLW   A'A' ;,mandando dato que
esta listo para recibir
CALL    TRANSMITIR ;,datos
BSF     CONTAR,4
GOTO    CONTI

SEG_MAN MOVF     RREG.W
MOVWF   REGPIC3
BSF     BPIC3,1
GOTO    CONTI

CHKMEN3 MOVLW      A'H
SUBWF   RREG.W
BTFSS   STATUS,Z ;,modo manual?
GOTO    CONTI
BSF     BANDERA3,0
BSF     BPIC3,1 ;,indicador de accion
GOTO    CONTI

AUTO3   MOVLW      A'T
SUBWF   RREG.W
BTFSS   STATUS,Z
GOTO    SEG_MAN
BSF     BANDERA3,2 ;,indicador de modo
automatico
BCF     BANDERA3,0
BSF     BPIC3,1 ;,accion indicada
GOTO    CONTI

-----Transmision de Datos-----

TRANSMITIR  MOVWF   TXREG ;,dato a mandar
BSF     STATUS,RP0 ;,banco 1
BTFSS   TXSTA,TRMT ;,byte transmitido?
GOTO    S-1 ;,no, espera
BCF     STATUS,RP0 ;,banco 0
BCF     PIR1,TXIF ;,limpiando bit de fin de transmision
RETURN

CONTI      RETFIE

-----
INICIO     BSF     STATUS,RP0 ;,banco 1
          CLRF   TRISB ;,puerto D como salidas
          MOVLW  B'10000011' ;,CONFIGURACION DEL TIMER 0
1:1        MOVWF  OPTION_REG
          MOVLW  B'00000101' ;,SALIDAS COMO ANALOGICAS
RA0,RA1   MOVWF   ADCON1 ;,Y VOLTAJE DE REFERENCIA
VREF+ VDD,PIN RA3 ;,justificacion left los 6 bits mas
significativos ;,son leidos en ADRESH
          ;,RA2,RA3 salidas a Motor
-----
          CLRF   TRISA
          MOVLW  0xFF

```

```

MOVWF    TRISC ;,salida a driver
BCF      TRISC,4
BCF      TRISC,7 ;,Configuracion para recepcion y
          TRISC,6
          TRISC,0 ;,Pin para recibir los pultos del
encender ;,BCF   TRISC,2 ;,limite inferior Y-
          ;,BCF   TRISC,3 ;,limite superior Y+
          MOVLW  64 ;,a 19200 baudios
          MOVWF  SPBRG
          MOVLW  B'00100100' ;,Configuracion del USART
Transmision
          MOVWF  TXSTA ;,Modo asincrono a alta
velocidad
          BSF    PIE1,RCIE ;,Habilita interrupción por recepción
-----Configuracion del TIMER1 como entrada digitales-----
          BCF    STATUS,RP0 ;,banco 0
          MOVLW  B'00000111' ;,bit 3, modo contador de pulsos por
pata RC0
          MOVWF  T1CON ;,bit 0, activacion del
timer1
          CLRF   TMR1H
          CLRF   TMR1L
          CLRF   BANDERA3
          CLRF   REGPIC3
          CLRF   DATLWY
          CLRF   DATHY
          CLRF   PORTB
          CLRF   PORTA
          CLRF   PORTC
          CLRF   BPIC3
          CLRF   TMP1
          CLRF   RESTAHI
          CLRF   RESTALW
          CLRF   VELOC1
          CLRF   YL
          CLRF   YH
          CLRF   CONTAR
          CLRF   DATLWYN
          CLRF   COMPARALW
          CLRF   COMPARAHI
          CLRF   SALHOME
          BCF    PORTC,4
          MOVLW  B'10010000'
          MOVWF  RCSTA ;,Por puerto serial (7), 8
bits de recepción, continua
          MOVLW  B'11000000' ;,Habilitación de las
interrupciones, y por cambio en PB<4-7>
          MOVWF  INTCON
          MOVLW  B'10000101' ;,FOSC/32, SELECCION DE CANAL
0 (RA0/AN0)
          MOVWF  ADCON0 ;,habilita la conversion
          MOVLW  100 ;,velocidad de motores
          MOVWF  TMP1
          MOVWF  VELOC1
          MOVLW  4
          MOVWF  TIEMPO
          MOVLW  50
          MOVWF  SALHOME
-----Ciclo de trabajo-----
CICLO     BTFSS   BPIC3,0 ;,se pide comunicacion con PIC2?
          GOTO   CICLO ;,no, espera
          GOTO   COMPC3

COMPC3   BTFSS   BPIC3,1 ;,bit de accion indicado?
          GOTO   CICLO
          BTFSS   BANDERA3,0 ;,modo manual?
          GOTO   VERAUTO
manual?   BTFSS   BANDERA3,1 ;,ya se conecto a modo
          GOTO   CONMAN3
          GOTO   MANUAL3

CONMAN3  MOVLW   A'1'
          CALL    TRANSMITIR
          MOVLW   A'T ;,indicador de que se
comunico en modo ;,manual OK
          CALL    TRANSMITIR
          CALL    CLRDIR3
          BSF     BANDERA3,1 ;,indicador de modo
actual
          GOTO    CICLO

VERAUTO  BTFSS   BANDERA3,2 ;,se pide modo
automatico?
          GOTO    CICLO

```

CALL GOTO	CLRDIR3 CONAUTO		
-----TABLA DE DATOS DE MOTOR Y-----			
TABLAY ADDWF	PCL F		
RETLW	36		.10
RETLW	32		.9
RETLW	0		.5
RETLW	4		.6
-----MODO AUTOMATICO-----			
REG_AUTO AUTOMATICO	CALL BTFS GOTO	CLEARCONT DATHIY 4 CONTNEGA	.sentido horario de motores? .No, controles de posicion negativo .Si, controles de
posicion positiva SEGATO automaticos?	BTFS GOTO CALL GOTO	CONTAR,3 OKIPOS CANTTT CICLO	.kancelar movimientos
OKIPOS	BTFS GOTO GOTO	BPIC3,4 LOADLW3 LOADHI3	.alcanza parte baja de informacion?
LOADLW3	MOV MOVWF SUBWF BTFS GOTO CALL GOTO	TMR1L,W COMPARALW DATLWY,W STATUS,Z MAYOR MANTEN LOADHI3	.alcanzo DATLWX=COMPARALW
MAYOR	BTFS GOTO	STATUS,C MENOR	. DATLWY > TMR1L (C = 1) ?? .No, DATLWY >
TMR1	.CALL GOTO	MANTEN LOADHI3	
MENOR	MOV MOVWF SUBWF BTFS	TMR1L,W COMPARALW DATLWY,W STATUS,Z	.alcanzo el dato con MENOR
resolucion?	GOTO CALL	SIGNO MANTEN	
LOADHI3 informacion	BSF MOV MOVWF SUBWF BTFS GOTO CALL GOTO	BPIC3,4 TMR1H,W COMPARAH DATHIY,W STATUS,Z ALTMAYOR MANTEN FINCOOR	.a alcanzado parte baja de
ALTMAYOR = 0??	BTFS GOTO	STATUS,C SIGNO	.TMR1H > DATHIY (C .No, DATHIY >
TMR1H	GOTO	FINCOOR	.Si, TMR1H > DATHIY
SIGNO	BTFS GOTO GOTO	CONTAR,1 AVANYPOS AVANYNEG	.Movimiento negativo? .No, va a movimiento positivo .Si, va a movimiento negativo
FINCOOR	MOV CALL	A'1 TRANSMITIR	.si, transmite a PICMAESTRO .informacion sobre su
llegada solicitada	MOV CALL	.11 CLRDIR3 CONAUTO	.a la coordenada
tres 3y->3	CALL BTFS BCF BCF BCF BCF BCF	CLEARCONT BPIC3,4 BANDERA3,3 PORTC,4 CONTAR,1 CONTAR,2 CONTAR,4	.reseteo informacion .Limpiando el signo .cancelar movimientos automaticos .limpiando bandera de llegada de
AVANYPOS automatico	CALL GOTO	HOMEY1 SEGATO	.va a avanzar X+ .va a continuar el ciclo de
AVANYNEG	CALL	YNEGATO1	

GOTO	SEGATONEG		
CONTNEGA BSF MOV ANDWF	CONTAR,1 0x0F DATHIY,F		.movimiento negativo de motores
SEGATONEG	BTFS GOTO CALL GOTO	CONTAR,3 OKINEGA CANTTT CICLO	.cancelar automatico?
OKINEGA	MOV MOVWF SUBWF BTFS GOTO CALL GOTO	TMR1L,W COMPARALW DATLWY,W STATUS,Z MAYOR MANTEN LOADHI3	.alcanzo COMPARAH=DATHIX?
CANTTT	CALL BCF BCF BSF BCF RETURN	CLRDIR3 BANDERA3,2 BANDERA3,3 BANDERA3,0 CONTAR,2 CONTAR,3	
-----Ciclo de espera en modo automatico-----			
CONAUTO automatico?	BTFS GOTO BTFS	BANDERA3,3 \$-1 CONTAR,2	.modo de ciclo .orden de iniciar
movimientos automaticos?	GOTO GOTO	\$-1 AUTOMATICO	
-----MODO MANUAL-----			
MANUAL3 mover eje Y +?	MOV SUBWF BTFS GOTO	A'Q REGPIC3,W STATUS,Z YNEGA	.se pide
YNEG HOMEY1 driver motor	BSF HOMEY	PORTC,4	.enable
	DEC GOTO	TMP0,F MOTORY	
BACK	BTFS GOTO GOTO CALL BCF	BANDERA3,7 BACK1 BACK2 TRANSMITIR PORTC,4	
BACK2 driver motor	CALL GOTO	CLRDIR3 CICLO	.disable
MOTORY	CALL MOV ANDLW CALL MOVWF BTFS GOTO BCF BCF MOV CALL	HOLD TMP0,W 3 TABLAY PORTA PORTC,2 SUPY PORTC,4 BANDERA3,7 A'1 TRANSMITIR	
INF	BTFS GOTO BCF BCF MOV CALL	TRANSMITIR BANDERA3,3 MANY BACK	.no hay necesidad de .dado que unicamente
direccionar la informacion habla al PIC1	CALL MOV	TRANSMITIR 0x03	.indicando que llego al
limite de -X automatico?	CALL BTFS GOTO	TRANSMITIR BANDERA3,3 MANY	.estoy en modo
MANY	GOTO	BACK	
SUPY	BTFS GOTO GOTO	CONTAR,6 FSTHOME EXITHOM	.ya paso por el sensor de home?
FSTHOME	BTFS GOTO BCF BCF BSF	PORTC,3 HOY PORTC,4 BANDERA3,7 CONTAR,6	.disable motor driver .indicador de home pasado
EXITHOM	DECFSZ	SALHOME	

```

GOTO YNEG
BCF CONTAR,6
MOVLW 63
MOVWF SALHOME

limite superior
MOVLW A'1'
CALL TRANSMITIR
MOVLW 0x04 ;informacion de que se alcanzo

CALL TRANSMITIR
CLRF RESTAHI
CLRF RESTALW
CALL CLEARCONT

GOTO BACK

HOY BTFFS BANDERA3,3 ;estoy en modo
automatico?

GOTO REVHOME
RETURN

REVHOME BTFFS BANDERA3,7 ;se pide home?
GOTO VERCONPOS
GOTO HOMYNEG

VERCONPOS BTFFS BANDERA3,5 ;se pide
movimiento continuo positivo?
GOTO VERCONNNEG
GOTO FST

FST BTFFS BPIC3,5 ;ya paso por aqui primera vez?
GOTO FST2
GOTO HOMEY

FST2 BSF BPIC3,5
CALL CLRDIR3
GOTO HOMEY

VERCONNNEG BTFFS BANDERA3,6
;se pide movimiento continuo negativo?
GOTO BACK1
BTFFS BANDERA3,4 ;ya paso por aqui
primera vez?
GOTO FSTVEZ
GOTO YNEGATO

FSTVEZ BSF BANDERA3,4
CALL CLRDIR3
GOTO YNEGATO

```

-----Rutina de Retardo y modulo de control de avance de motores-----

```

HOLD DEL1 CLRF TMR0
= 1? BTFFS TMR0,5 ;TESTEO DEL BIT 7 DEL TEMP 0

VERDAD GOTO $-1 ;HASTA QUE SEA

SALTA DECSZ TMP1,F ;DECREMENTA TMP1 Y TMP1=1?.

GOTO DEL1
MOVWF VELOC1,W
MOVWF TMP1
RETURN

```

-----Limpiando el contador-----

```

CLEARCONT BCF T1CON,2

NOP
NOP
NOP

BCF T1CON,0

NOP
NOP
NOP

CLRF TMR1L

NOP
NOP
NOP

CLRF TMR1H

BSF T1CON,2

NOP
NOP
NOP

```

```

BSF T1CON,0
RETURN

-----
YNEGA MOVLW A'P'
SUBWF REGPIC3,W
BTFFS STATUS,Z ;se pide mover eje x -?
GOTO CERPZA3

YNEGATO1 YNEGATO BSF PORTC,4 ;enable driver motor
HOMYNEG INCF TMP0,1
INCF RESTALW,F
MOVLW 255
SUBWF RESTALW,W
BTFFS STATUS,Z
GOTO MOTORY
INCF RESTAHI
GOTO MOTORY

CERPZA3 MOVLW A'?
SUBWF REGPIC3,W
BTFFS STATUS,Z ;se pide grabar la posicion de 0 pza
para eje Z?

GOTO HOME
MOVLW A'1'
CALL TRANSMITIR
MOVLW A'1'
CALL TRANSMITIR ;señal de cero pieza Y
CALL HOLD
MOVWF TMR1H,W
MOVWF YH
MOVWF RESTAHI,W
SUBWF YH,W
CALL TRANSMITIR
CALL HOLD
MOVWF TMR1L,W
MOVWF YL
MOVWF RESTALW,W
SUBWF YL,W
CALL TRANSMITIR

CALL CLEARCONT

GOTO BACK

HOME MOVLW A'8'
SUBWF REGPIC3,W
BTFFS STATUS,Z
GOTO BACK
BSF BANDERA3,7 ;indicador de home)

BSF PORTC,4
CALL CLRDIR3
GOTO HOMYNEG

CLRDIR3 BCF BPIC3,0
BCF BPIC3,1
RETURN

END

```


Anexo C

Diagrama Electrónico de etapa de control de motores

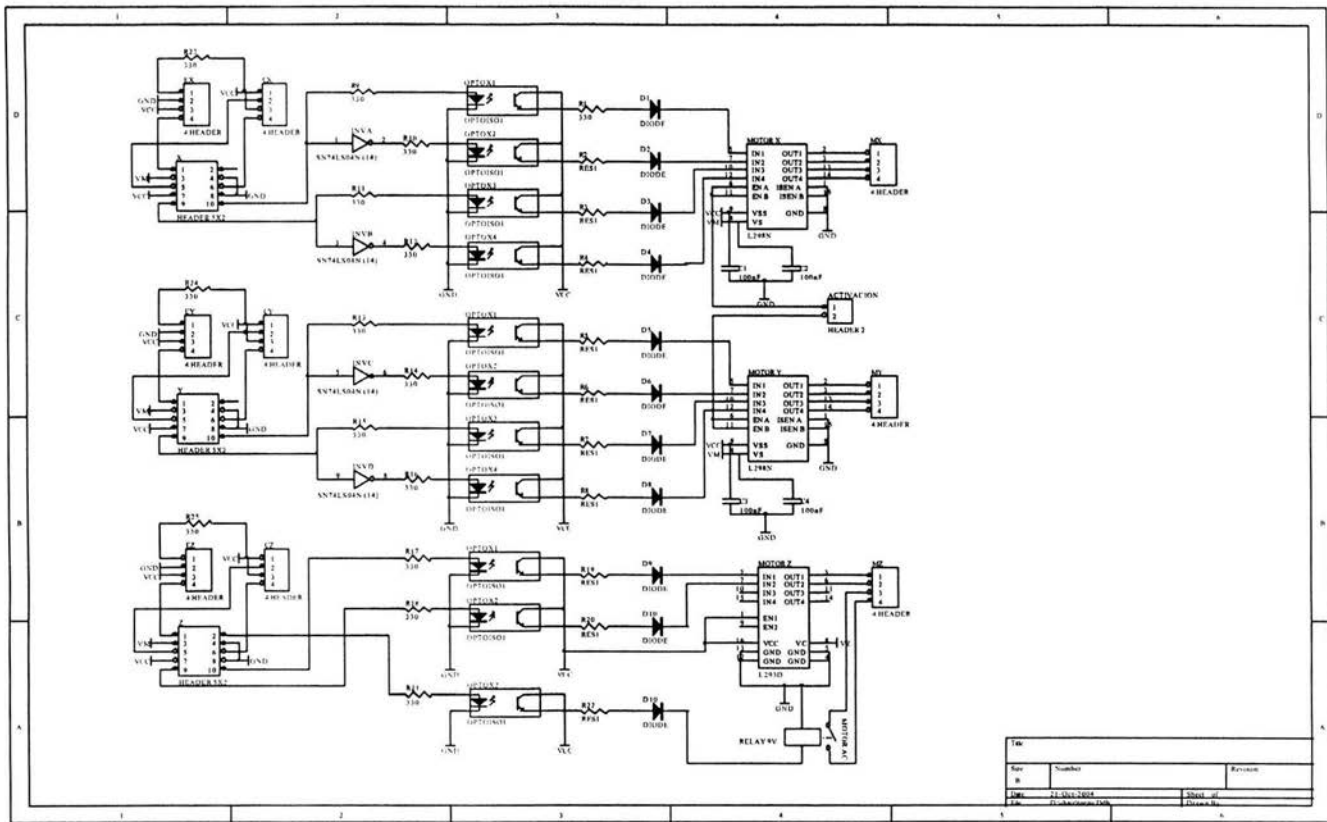
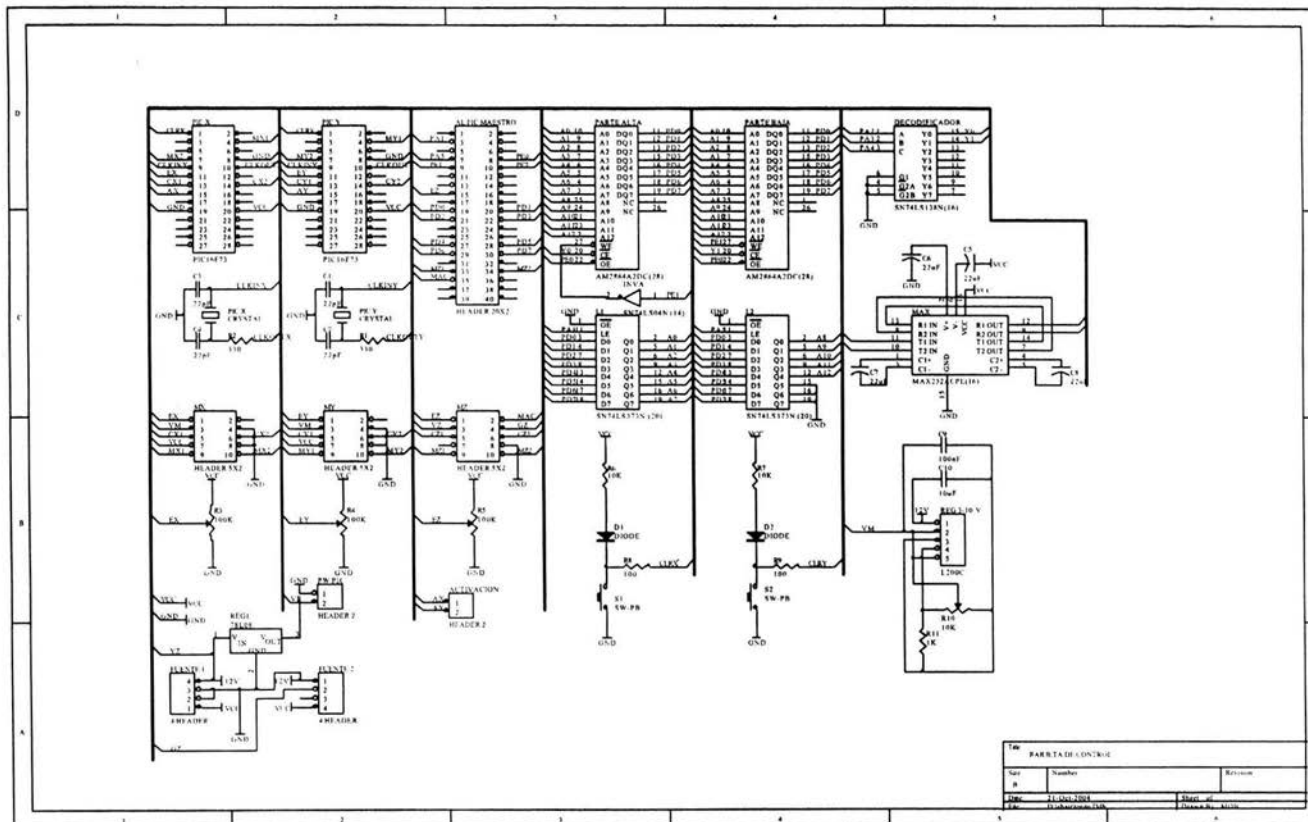


Diagrama Electrónico de la tarjeta de control



Bibliografía

- [1] Boon, G. K.; Mercado, A.
"Automatización Flexible en la Industria".
Editorial LIMUSA, Noriega. 1991
- [2] Martino, R.L.
"Sistemas Integrados de Fabricación".
Editorial LIMUSA, Noriega. 1990
- [3] Yoram Korem
"Computer Control of Manufacturing Systems"
Editorial McGraw-Hill inc.1983
- [4] Mike Lynch
"Computer Numerical Control for Machining"
Editorial McGraw-Hill inc.1992
- [5] Roger S. Pressman, John E. Williams
"Numerical Control and Compute aided Manufacturing"
Editorial John Wiley & Sons 1997
- [6] Robert L. Norton
"Diseño de Máquinas"
Editorial Pearson, 1999
- [7] José Ma. Angulo, Susana Romero Yesa
"Microcontroladores PIC, Diseño práctico de aplicaciones"
Segunda parte: PIC16F8XX
Editorial McGraw-Hill inc.2000
- [8] Manual de MICROCHIP
"PIC16F87X Data Sheet"
- [9] Manual de MICROCHIP
"PIC16F7X Data Sheet"
- [10] Sterling Instruments
"Handbook of timing belts and Pulleys"
Páginas de internet consultadas
- [11] Publicación
"Transmisiones flexibles, correas y cadenas"
Universidad Jaume, España

[12] Microcontroladores PIC

<http://www.todopic.org>

[14] Sistemas flexibles de transmisión

<http://www.ujes.es/publ>

[15] Simco Technologies

<http://www.simcotech.com/sensors/sensorslideshow1.htm>

[17] Siemens photoelectric sensors

<http://www.sea.siemens.com/step/templates/lesson.mason?sensors:6:1:1>

[18] Stepper motor basics

<http://library.solarbotics.net/pdflib/pdf/motorbas.pdf>

[19] Electronics Club : switches

<http://www.kpsec.freeuk.com/components/switch.htm>

[20] The educational encyclopedia

<http://users.pandora.be/educyclopedia/electronics/motorstep.htm>

[21] DataSheet X28HC64 EEPROM

[22] Comunicación Serial

<http://www.tecnotopia.com.mx/mecatronica/puertoserialasincrono.htm>

[23] Comunicación Serial Asíncrona

<http://juandeg.tripod.com/comserial.htm>