

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD TECNICA
CARRERA ELECTRÓNICA Y TELECOMUNICACIONES



Trabajo de Aplicación

“Control Numérico Computarizado para tallado de texto y formas”

Nivel Licenciatura

Postulante: Univ. Juan Mamani Amaru

La Paz – Bolivia

agosto-2012

DEDICATORIA

A mis padres q me apoyaron en todo momento, mis amigos y en especial a Fabiola que sin tus palabras y tu apoyo no estaría presente en este momento, este trabajo va dedicado hacia ti...

Juan

INDICE

CONTENIDO	Pag.
1. RESUMEN	1.
2. PROBLEMÁTICA	2.
3. JUSTIFICACION	3.
3.1. JustifiacionTecnologica	3.
3.1.1. Impacto teorico	4.
3.1.2. Impacto Social	4.
3.1.3. Impacto Económico	4.
3.1.4. Impacto Ambiental	5.
4. OBJETIVOS	5.
4.1. General	5.
4.2. Específicos	5.
5. FUNDAMENTO TEORICO	6.
5.1. Definiciones	8.
5.1.1. Interfaz directa	9.
5.1.2. Supervisión	9.
5.1.3. Control	10.
5.1.4. Interfaz indirecta	10.
5.2. Códigos de programación CNC	12.
5.2.1. "HP-GL" lenguaje gráfico	13.
5.2.2. Sintaxis de HP-GL (Lenguaje Gráfico de Hewlett Packard)	15.
5.3. Maquinas taladradoras	16.
5.3.1. Clasificación de las maquinas según el tipo de Mecanizado	16.
5.3.1.1. Punto a Punto	16.
5.3.1.2. Paraxial	17.
5.3.2. Trayectoria Continua	17.
5.4. Clasificacion de las maquinas según el tipo de control	17.

5.4.1. NC	17.
5.4.2. CNC	18.
5.4.3. DNC	18.
5.5. Sistemas de Control	19.
5.5.1. Puerto de datos	19.
5.5.2. Puertos de estado	19.
5.5.3. Puertos de control	19.
5.5.4. Enviando datos por el puerto paralelo	20.
5.5.5. Características del microcontrolador AVR Atmega16	22.
6. DESARROLLO DEL TRABAJO	25.
6.1. Programas en Visual Basic 6.0	27.
6.2. Programa CNC	28.
6.3. Programa DECODIFICADOR	32.
6.4. Programa POSICIONADOR	35.
6.5. Esquema del controlador del motor	38.
6.6. Esquema del control manual de la maquina CNC	39.
6.7. Consideraciones sobre el uso del AutoCad	45.
6.8. Diseño de la maquina	46.
7. CONCLUSIONES	50.
8. BIBLIOGRAFIA	51.
9. ANEXOS	
Código fuente del Programa CNC	52.
Código fuente del Programa DECODIFICADOR	61.
Código fuente del Programa POSICIONADOR	66.

INDICE DE FIGURAS

Figuras	Pag.
Figura 1. Árbol de problemas	3.
Figura 2. Maquina CNC (Control Numérico Computarizado)	6.
Figura 3. Programa de maquinado	7.
Figura 4. Supervision y control	10.
Figura 5. Bloque de instrucciones del lenguaje "HP-GL"	15.
Figura 6. Mecanizado Punto a Punto	17.
Figura 7. Maquinas	18.
Figura 8. Taladradora CNC	18.
Figura 9. El puerto paralelo	19.
Figura 10. Configuración del puerto paralelo	20.
Figura 11. El integrado 74LS245	21.
Figura 12. Circuito de control del puerto paralelo	22.
Figura 13. Forma física del microcontrolador	22.
Figura 14. Diagrama en bloques general Maquina CNC	25.
Figura 15. Diagrama en bloques de la interface	25.
Figura 16. Diagrama en bloques de control	26.
Figura 17. Maquina CNC casera	27.
Figura 18. Ventana de presentación del programa CNC	29.
Figura 19. Ventana de presentación del programa DECODIFICADOR	32.
Figura 20. Proceso de decodificación	34.
Figura 21. Ventana de presentación POSICIONADOR	35.
Figura 22. Circuito de control de motores	38.

Figura 23. Circuito de control de motores auxiliar	39.
Figura 24. Diagrama de flujo del código fuente	40.
Figura 25. Detalles del programa MicroC	41.
Figura 26. Mesa convencional	46.
Figura 27. Desplazamiento de la mesa en dirección	47.
Figura 28. movimiento de material en Y	48.
Figura 29. Maquina CNC en proceso de construcción	49.

CONTROL NUMERICO COMPUTARIZADO PARA TALLADO DE TEXTO Y FORMAS

1. RESUMEN

El presente trabajo de aplicación trata del diseño y construcción de un sistema de tallado de textos y formas de control numérico computarizado, de bajo costo; en la que haciendo uso del puerto paralelo controla la posición y la velocidad de los motores paso a paso que accionan los ejes de la máquina, así como también el motor del cabezal taladrador, a partir de un diseño gráfico, mecanizado de esta manera piezas taladradas en serie, así como también fresado de contornos. Mejorando la calidad de mecanizado, disminuyendo los costos de producción y acortando los tiempos de diseño y producción.

Para la realización del mecanizado en este prototipo de maquina CNC, se han de realizar las siguientes tareas:

La herramienta informática que permite realizar el diseño grafico (cad), seleccionara opciones de mecanizado, simular una entidad o diseño, elegir una herramienta, tipo de acabado y como característica fundamental la generación de un archivo de codificación (cam), así como la posición de la herramienta en las diferentes tareas a mecanizar que en este caso está desarrollado en Visual Basic.

Debido a que la información obtenida de la codificación es muy abundante, se realizó un programa en el lenguaje Visual Basic con la finalidad de agrupar la información que nos genera el software (CAD) que en el trabajo de aplicación se utiliza el "AutoCAD", referente a la ubicación de la herramienta y la de generar una coordenada constante, en el caso de que se produzca un cambio de operación. Esta nos servirá de matriz fundamental para el algoritmo desarrollado en una herramienta de instrumentación virtual.

2. PROBLEMÁTICA.-

Observando las microempresas que existen en nuestra ciudad de La Paz específicamente por la zona del cementerio, que se encargan de realizar el diseño de placas, trofeos, recuerdos, etc. Donde el material que utilizan el cobre, plata, etc. Realizan el grabado de diferentes mensajes, imágenes de acuerdo al gusto de sus clientes.

Es ahí donde se observa que el grabado de sus productos lo realizan manualmente, no teniendo una precisión en el diseño de letras, o una buena imagen.

El presente trabajo de aplicación tratara de automatizar este proceso dando lugar a crear una de estas máquinas que será útil en esta rama o dándole aplicación en distintas áreas ya que el hardware puede cambiar pero no el software que se mantendrá.

El problema central de nuestro proyecto en la automatización de tallado de textos y formasson:

- La realización de este proceso manualmente.
- El tiempo que lleva realizar manualmente
- La salud de quien lo realiza
- Entrar a la era de la modernización

Los efectos que llegara a tener este proyecto son

- El tallado de textos y formas automático
- El tiempo es mínimo de la realización del tallado
- Solo necesitara mantenimiento el equipo y no afecta la salud del obrero

Ahora mostraremos el árbol de problemas general planteado

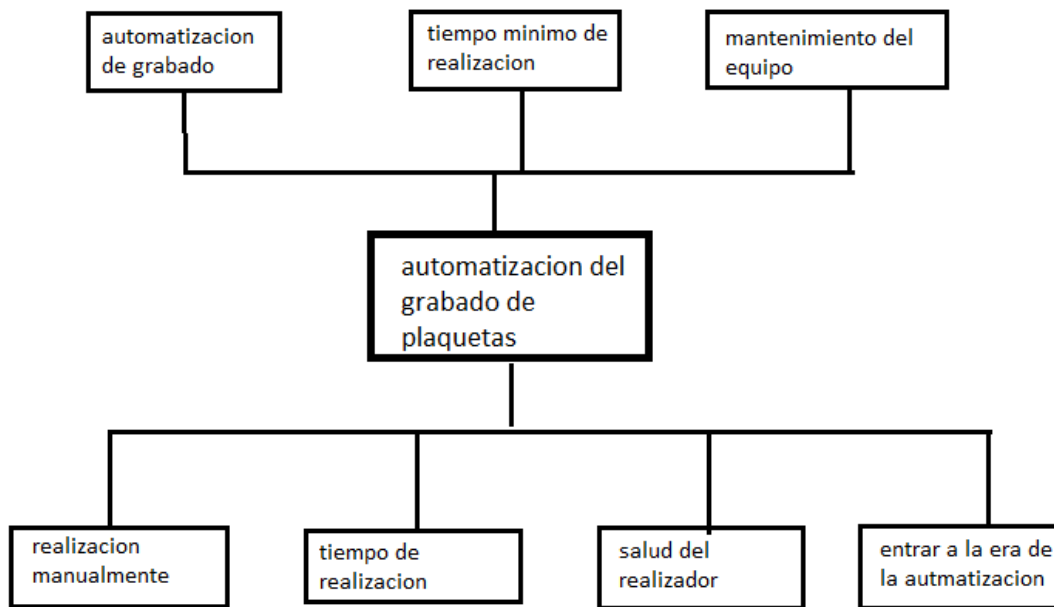


Figura. 1. Árbol de problemas general

3. JUSTIFICACION.-

Este proyecto está enfocado a la automatización del tallado de textos y figuras en procesos de producción y manufactura orientado a la pequeña y mediana empresa, permitiendo personalizar el producto para los clientes y tener un mejor tiempo de respuesta a la hora de realizar pedidos en la demanda.

Otro concepto que se pone a prueba es la Manufactura Asistida por Computadora (CAM) que tiene como objetivo la interface humano-maquina a través de una computadora hace más fácil el proceso de comunicación con la máquina para trasladar ideas y diseños.

3.1 Justificación tecnológica.-

Mencionando, las cuatro variables fundamentales que inciden en la bondad de un automatismo son: productividad, rapidez, precisión y velocidad.

Se compone por los siguientes impactos:

3.1.1 Impacto teórico.-

Este proyecto tendrá un impacto teórico debido a la utilización de la “teoría del control numérico” implementado en nuestra máquina, demostraremos que automatizaremos el grabado de plaquetas con una maquina CNC donde es muy poco utilizado la teoría del control numérico.

Sabemos que la automatización se basa en la actualidad por el control de PLC's donde la implementación de este control se hace de manera más eficiente, pero en nuestro proyecto lograremos realizar este proceso mediante microcontroladores implementando el ya mencionado control numérico.

3.1.2 Impacto social.-

El presente proyecto automatizara el grabado de plaquetas, beneficiando de esta manera el trabajo que realizan nuestros artesanos, por la precisión que tiene nuestra maquina por el manejo del control numérico

Sabemos que nuestra realidad observamos el proceso que se realiza para el grabado de placas donde se realiza un esfuerzo físico, además que afecta en la salud debido a la aspiración del metal donde se realiza el grabado.

Será de gran ayuda a nuestros artesanos donde con un poco de capacitación del software y el manejo de la maquina el trabajo se hará mas fácil, incrementando la producción de sus productos.

3.1.3 Impacto económico.-

Con el avance de la tecnología los sistemas con la electrónica han ido evolucionando además de una actualización en su respuesta en la comunicación del software con el hardware, es así que el proyecto agilizara el proceso.

Automatizando este proceso se tiene una mejor producción y una pérdida considerable en tiempo y un aumento de dinero así como incrementar la producción de la empresa y un aumento económico para los que invierten en la empresa.

El proyecto como prototipo tendrá un costo alto pero realizando las respectivas pruebas y corrigiendo los defectos que llegaran a pasar su precio será más accesible

para aquellos que quieran adquirirlo especialmente a quienes van dirigidos a nuestros artesanos.

Es por eso que nuestros artesanos invirtiendo en nuestro proyecto incrementaran su producción que eso implica más ingresos en sus empresas o microempresas.

3.1.4 Impacto ambiental.-

En la actualidad todo los proyectos se basan en la protección de nuestro medio ambiente debido al calentamiento global, el presente proyecto pensando justamente en nuestro medio ambiente se basa en el uso de energía eléctrica, no implementando baterías, combustible, etc.

Es de esta manera la única manera que se encuentra en la posibilidad de proteger nuestro medio ambiente, esperando que sea considerado el trabajo que se está realizando.

4. OBJETIVOS.-

4.1. General

- Diseñar y construir un prototipo de “Maquina CNC” (Control Numérico Computarizado) para el tallado de textos y figuras sobre madera.

4.2 Específicos

- Desarrollar un sistema automatizado bajo el concepto CAM (Manufactura Asistido por Computadora) y generar una interfaz amigable para el usuario.
- Desarrollar un software para lograr una comunicación adecuada entre el usuario y la maquina con la facilidad de ingreso de información.
- Diseñar un circuito auxiliar para el control de los motores empleando microcontroladores.
- Realizar un diseño mecánico en la estructura para el correcto funcionamiento de la maquina CNC.
- Interpretar el código “HP-GL” de los archivos .plt y codificarlos a un código CNC.
- Explicar la importancia y el uso que se le da al puerto paralelo.

5. FUNDAMENTO TEORICO

El CNC tuvo su origen a principios de los años cincuenta. En esta época las computadoras estaban en sus inicios y eran tan grandes que el espacio ocupado por la computadora era mayor que el de la máquina.

Hoy día las computadoras son cada vez más pequeñas y económicas, con lo que el uso del CNC se ha extendido a todo tipo de maquinaria: taladradoras, tornos, fresadoras, rectificadoras, electroerosionadoras, máquinas de coser, etc.

En una máquina CNC, a diferencia de una máquina convencional o manual, una computadora controla la posición y velocidad de los motores que accionan los ejes de la máquina. Gracias a esto, puede hacer movimientos que no se pueden lograr manualmente como círculos, líneas diagonales y figuras complejas tridimensionales.

Las máquinas CNC son capaces de mover la herramienta al mismo tiempo en los tres ejes para ejecutar trayectorias tridimensionales como las que se requieren para el maquinado de complejos moldes y troqueles como se muestra en la imagen



Figura 2. Máquina CNC ("control numérico computarizado").

En una máquina CNC una computadora controla el movimiento de la mesa, el carro y el husillo. Una vez programada la máquina, ésta ejecuta todas las operaciones por sí sola, sin necesidad de que el operador esté manejándola. Esto permite aprovechar mejor el tiempo del personal para que sea más productivo.

El término "control numérico" se debe a que las órdenes dadas a la máquina son indicadas mediante códigos numéricos. Por ejemplo, para indicarle a la máquina que mueva la herramienta describiendo un cuadrado de 10 mm. Por lado se le darían los siguientes códigos:

```
G90 G71
G00 X0.0 Y0.0
G01 X10.0
G01 Y10.0
G01 X0.0
G01 Y0.0
```

Un conjunto de órdenes que siguen una secuencia lógica constituyen un programa de maquinado, figura 3. Dándole las órdenes o instrucciones adecuadas a la máquina, ésta es capaz de maquinar una simple ranura, una cavidad irregular, la cara de una persona en altorrelieve o bajorrelieve, un grabado artístico, un molde de inyección de una cuchara o una botella... lo que se quiera.

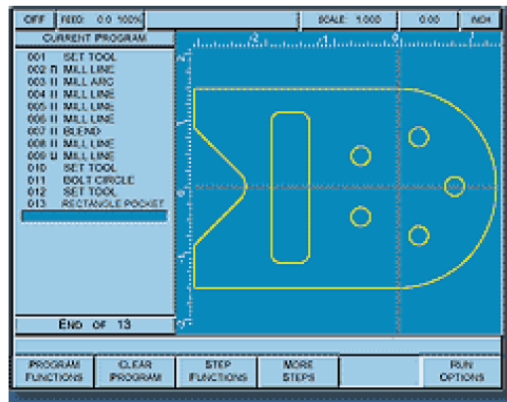


Figura 3 Programa de maquinado

Al principio hacer un programa de maquinado era muy difícil y tedioso, pues había que planear e indicarle manualmente a la máquina cada uno de los movimientos que tenía que hacer. Era un proceso que podía durar horas, días, semanas. Aun así era un ahorro de tiempo comparado con los métodos convencionales.

Actualmente muchas de las máquinas modernas trabajan con lo que se conoce como "lenguaje conversacional" en el que el programador escoge la operación que desea y la máquina le pregunta los datos que se requieren. Cada instrucción de este lenguaje conversacional puede representar decenas de códigos numéricos. Por ejemplo, el maquinado de una cavidad completa se puede hacer con una sola instrucción que especifica el largo, alto, profundidad, posición, radios de las esquinas, etc. Algunos controles incluso cuentan con traficación en pantalla y funciones de ayuda geométrica. Todo esto hace la programación mucho más rápida y sencilla.

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 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.

5.1 Definiciones.-

El diseño y la fabricación asistidos por ordenador (CAD/CAM) es una disciplina que estudia el uso de sistemas informáticos como herramienta de soporte en todos los procesos de diseño y fabricación de cualquier tipo de producto. Se ha convertido en un requisito indispensable para la industria actual que enfrenta la necesidad de mejorar la calidad, disminuir costes y acortar tiempos de diseño y producción.

El uso cooperativo de herramientas de diseño y fabricación ha dado lugar a la aparición de una nueva tecnología denominada „Fabricación Integrada por Ordenador“ e incluso se habla de la „Gestión Integrada por Ordenador“ como el

último escalón de automatización hacia el que todas las empresas deben orientar sus esfuerzos.

CAD es el acrónimo de „Computer Aided Design“ o diseño asistido por computador. Se trata de la tecnología implicada en el uso de ordenadores para realizar tareas de creación, modificación, análisis y optimización de un diseño. Las herramientas de CAD abarcan desde herramientas de modelado geométrico hasta aplicaciones a medida para el análisis u optimización de un producto específico. Entre estos dos extremos se encuentran herramientas de modelado y análisis de tolerancias, cálculo de propiedades físicas (masa, volumen, momentos, etc.), modelado y análisis de elementos finitos, ensamblado, etc. La función principal en estas herramientas es la definición de la geometría del diseño (pieza mecánica, arquitectura, circuito electrónico, etc.) ya que la geometría es esencial para las actividades subsecuentes en el ciclo de producto descrito en la figura 4.

El término CAD se puede definir como el uso de sistemas informáticos en la creación, modificación, análisis u optimización de un producto, así como para la planificación, gestión y control de las operaciones de una planta de fabricación mediante una interfaz directa o indirecta entre el sistema informático y los recursos de producción.

Así pues, las aplicaciones del CAM se dividen en dos categorías:

5.1.1 Interfaz directa: son aplicaciones en las que el ordenador se conecta directamente con el proceso de producción para monitorizar su actividad y realizar tareas de supervisión y control. Así pues estas aplicaciones se dividen en dos grupos:

5.1.2 Supervisión: implica un flujo de datos del proceso de producción al computador con el propósito de observar el proceso y los recursos asociados y recoger datos.

5.1.3 Control: supone un paso más allá que la supervisión, ya que no solo se observa el proceso, sino que se ejerce un control basándose en dichas observaciones.

5.1.4 Interfaz indirecta: se trata de aplicaciones en las que el ordenador se utiliza como herramienta de ayuda para la fabricación, pero en las que no existe una conexión directa con el proceso de producción. La figura 1.3 muestra de forma gráfica la diferencia entre estos dos tipos de aplicaciones.

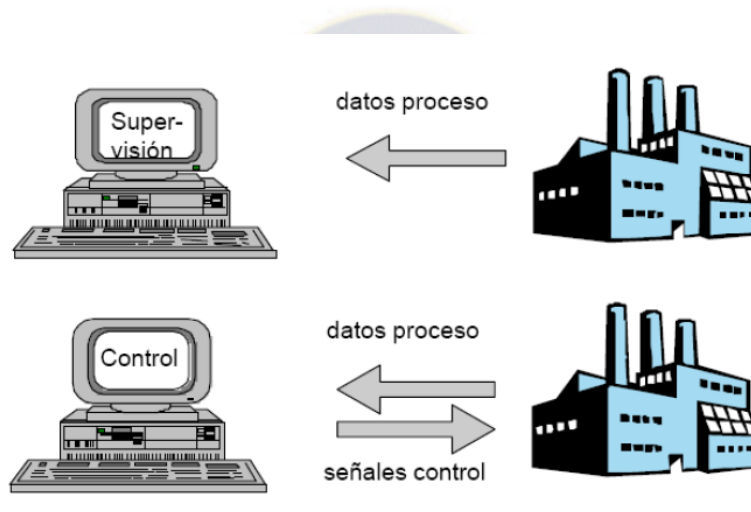


Figura 4 Supervisión y control

La planificación de procesos es la tarea clave para conseguir la automatización deseada, sirviendo de unión entre los procesos CAD y CAM. El plan de procesos determina de forma detallada la secuencia de pasos de producción requeridos para fabricar y ensamblar, desde el inicio a la finalización del proceso de producción. Las piezas se organizan en familias y cada nueva pieza se clasifica dentro de una familia, según las características o los elementos que la componen. Además, los sistemas informáticos pueden usarse para determinar el aprovisionamiento de materias primas y piezas necesarias para cumplir el programa de trabajo de la manera más eficiente, minimizando los costes financieros y de almacenaje. Esta actividad se denomina Planificación de Recursos Materiales (Material Requirement Planning o MRP). También es posible ejercer tareas de monitorización y control de la

actividad de las maquinas del taller que se integran bajo el nombre de Planificación de Recursos de Manufacturación (Manufacturing Requirement Planning o MRP II).

La Ingeniería Asistida por Ordenador (Computer Aided Engineering o CAE) es la tecnología que se ocupa del uso de sistemas informáticos para analizar la geometría generada por las aplicaciones de CAD, permitiendo al diseñador simular y estudiar el comportamiento del producto para refinar y optimizar dicho diseño. Existen herramientas para un amplio rango de análisis. Los programas de cinemática, por ejemplo, pueden usarse para determinar trayectorias de movimiento y velocidades de ensamblado de mecanismos. Los programas de análisis dinámico de grandes desplazamientos se usan para determinar cargas y desplazamientos en productos complejos como los automóviles. Las aplicaciones de temporización lógica y verificación simulan el comportamiento de circuitos electrónicos complejos.

El método de análisis por ordenador más ampliamente usado en ingeniería es el método de elementos finitos o FEM (de Finite Element Method). Se utiliza para determinar tensiones, deformaciones, transmisión de calor, distribución de campos magnéticos, flujo de fluidos y cualquier otro problema de campos continuos que serían prácticamente imposibles de resolver utilizando otros métodos. En este método, la estructura se representa por un modelo de análisis constituido de elementos interconectados que dividen el problema en elementos manejables por el ordenador, requiere más un modelo abstracto de descomposición espacial que la propia geometría del diseño, dicho modelo se obtiene eliminando los detalles innecesarios de dicha geometría o reduciendo el número de dimensiones. Una vez creado dicho modelo, se genera la malla de elementos finitos para poder aplicar el método. Al software que se encarga de generar el modelo abstracto y la malla de elementos finitos se le denomina pre-procesador. Después de realizar el análisis de cada elemento, el ordenador ensambla los resultados y los visualiza. Las regiones con gran tensión se destacan, por ejemplo, mostrándose en color rojo. Las herramientas que realizan este tipo de visualización se denominan post-procesadores.

La ventaja del análisis y optimización de diseños es que permite a los ingenieros determinar cómo se va a comportar el diseño y eliminar errores sin la necesidad de gastar tiempo y dinero construyendo y evaluando prototipos reales. Ya que el coste de reingeniería crece exponencialmente en las últimas etapas del desarrollo de un producto y en la producción, la optimización temprana que permiten las herramientas CAE supone un gran ahorro de tiempo y una notable disminución de costes.

Así pues, CAD; CAM y CAE son tecnologías que tratan de automatizar ciertas tareas del ciclo de producto y hacerlas más eficientes. Dado que se han desarrollado de forma separada, aun no se han conseguido todos los beneficios potenciales de integrar las actividades de diseño y fabricación del ciclo de producto. Para solucionar este problema ha aparecido una nueva tecnología: la fabricación integrada por ordenador o CIM (de Computer Integrated Manufacturing). Esta tecnología tiene el objetivo de aunar las *islas de automatización* conjuntándolas para que cooperen en un sistema único y eficiente. El CIM trata de usar una única base de datos que integre toda la información de la empresa y a partir de la cual se pueda realizar una gestión integral de todas las actividades de la misma, repercutiendo sobre todas las actividades de administración y gestión que se realicen en la empresa, además de las tareas de ingeniería propias del CAD y el CAM. Se dice que el CIM es más una filosofía de negocio que un sistema informático.

5.2 Códigos de programación CNC

Para generar un programa CNC es necesario identificar el código de letras utilizado para este fin. En la tabla 1. Podemos ver las letras que caracterizan las funciones programables, se trata de un extracto de la norma DIN que sigue siendo válida para la mayoría de los controladores de máquinas-herramientas de control numérico.

Comando Descripción	
N	Número de Secuencia

G	Funciones Preparatorias
X	Comando para el Eje X
Y	Comando para el Eje Y
Z	Comando para el Eje Z
R	Radio desde el Centro Especificado
A	Ángulo contra los Punteros del Reloj desde el Vector +X
I	Desplazamiento del Centro del Arco del Eje X
J	Desplazamiento del Centro del Arco del Eje Y
K	Desplazamiento del Centro del Arco del Eje Z
F	Tasa de Alimentación
S	Velocidad de Giro
T	Número de Herramienta
M	Funciones Misceláneas

Tabla 1. Funciones que caracterizan las maquinas CNC

La tabla que se observa a continuación es una traducción de la información respecto a este lenguaje gráfico.

5.2.1 "HP-GL" lenguaje gráfico

Instrucciones de trazado y Control del Lápiz:

Instrucción Función	
PU	Subir el lápiz (Pen Up).
PD	Bajar el lápiz (Pen Down)
PA	Posicionamiento Absoluto de Trazado (Plot Absolute).
PR	Posicionamiento relativo de Trazado (Relative Coordinate Pen Move).
AA	Trazado de Arcos en Modo Absoluto (Absolute Arc Plot).
AR	Trazado de Arcos en Modo Relativo (Relative Arc Plot).
CI	Círculo.

Tabla2. Sentencias comunes q se utilizan para todas la maquinas CNC

Otras instrucciones

Instrucción Función	
DF	Valores por defecto del set de Instrucciones.
IN	Inicializar set de Instrucciones.
SC	Escala.
SP	Selección de Lápiz.

Tabla 3. Otras instrucciones utilizadas en el lenguaje “HP-GL”

Existen otras instrucciones pero estas son las más comunes.

5.2.2 Sintaxis de HP-GL (Lenguaje Gráfico de Hewlett Packard)

Aquí se toma un ejemplo muy sencillo para describir la forma en que se escribe un bloque de instrucciones. Cada bloque está formado por una o dos instrucciones de dos letras cada una, que suele ser un mnemónico de los que vimos en las tablas anteriores.



Figura 5. Bloque de instrucciones del lenguaje "HP-GL"

La primera instrucción está referida a si se realiza el trazado o no, en este caso la instrucción es **PD** (*bajar el lápiz*), eso significa que sí, se realizará trazado. La segunda instrucción, está referida al modo de desplazamiento, que como ya vimos anteriormente, puede ser en modo Absoluto (PA) o Relativo (PR), en este caso será Desplazamiento o Posicionamiento Absoluto (**PA**). Lo que sigue son dos números separados por una coma, el primer número es la coordenada para el Eje X, el segundo para el Eje Y, estas coordenadas determinan el punto al cual debe desplazarse, en nuestro ejemplo será hacia el punto **X=1001, Y=566**.

Finalmente, el signo ";" indica **Fin de Bloque de Instrucción**, igual que en el lenguaje C o C++.

El procedimiento que se realiza para la escritura se muestra de la siguiente manera:

```
.(;.l81;;17:.N;19:IN;SC;PU;RO0;IP;IW;VS15;VS15;VS15;VS15;VS15;  
VS15;SP1;PU;PA0,0;SP1;LT;PA53,421;
```

Las instrucciones que se observa son las instrucciones de configuración del Trazador, de cómo se va a realizar el trazado, el sistemas de coordenadas, el área

de impresión, etc., estas instrucciones aparecerán siempre, son las que encabezan cualquier código de trazado, puede variar según el trazador o la forma en que se deba trazar una imagen, puede depender de varios factores. Luego aparecen un par de instrucciones que ya conocemos, como verás, la primera es **PU**; lo único que se hace con esta instrucción es levantar el taladro. La que sigue, ya es un desplazamiento **PA0, 0**; y según la instrucción, lo hará en Modo Absoluto hacia la coordenada **0,0**, es decir, la punta del lápiz se dirige a la coordenada de origen.

Veamos ahora las dos que siguen, la primera de ellas **SP1**; es **Select Pen 1** (Selección del Lápiz 1).

La Segunda es **LT**; (**Line Type**) que vendría a ser Tipo de línea, hay un par de números que deberían acompañar a LT y que indican el tipo de línea a trazar, que podrían ser; línea llena, sólo puntos, línea y punto (en sus distintas variedades), simples segmentos separados por un espacio determinado (también en sus distintas variedades), etc. Por defecto cuando aparece esta instrucción sólo, sin ningún número, significa que el trazado se hará con línea llena. Luego aparece otro desplazamiento en Modo Absoluto **PA53, 421**; hacia el punto X=53, Y=421....

Finalmente el código que se genera para realizar el grabado de un objeto es de la siguiente manera

PDPA53,397,77,372,125,372,174,397,174,469;PUPA149,469;PDPA101,566;PUPA125,372; PDPA149,397,149,445,101,518; ...

5.3 Maquinas taladradoras.-

5.3.1 Clasificación de las maquinas según el tipo de mecanizado

Según los movimientos que se pueden controlar sobre una máquina herramienta, los CNC se clasifican en los siguientes sistemas: Punto a punto (PTP); Paraxial; De trayectoria continua o contorneado.

5.3.1.1 Punto a Punto (Point To Point).- Este sistema controla el posicionamiento de la herramienta en los sucesivos puntos donde debe efectuarse una o varias

posiciones de mecanizado. La trayectoria seguida de un punto a otro no tiene importancia, además la herramienta no siempre está en contacto con la pieza a lo largo de la trayectoria. El posicionado se realiza normalmente a la velocidad máxima que soporta la máquina. Se utiliza en taladradoras, punteadoras, punzonadoras, etc.

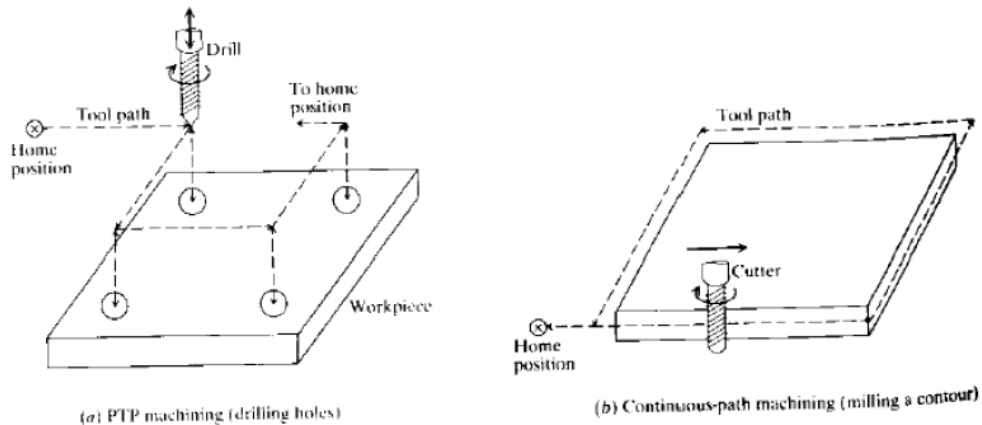


Figura 6. Mecanizado Punto a Punto

5.3.1.2 Paraxial.- Con este sistema es posible controlar, además de la posición del órgano móvil, la trayectoria seguida por el mismo. La velocidad de avance es programable y puede efectuarse un mecanizado durante el desplazamiento. Se utiliza en ciertas fresadoras simples, algunas mandriladoras – fresadoras, taladradoras que pueden realizar pequeños fresados, etc.

5.3.2 Trayectoria Continua o Contorneado.- Se trata de una forma de posicionado que permite generar curvas, no solo se controla la posición final sino el movimiento en cada instante. De esta forma podemos conseguir recorridos tales como: rectas con cualquier pendiente, arcos y cualquier curva definible matemáticamente. Pueden usarse también como PTP.

5.4 Clasificación de las máquinas según el tipo de control

Según la forma que se pueden controlar una máquina herramienta, se clasifican en: NC, CNC, DNC

5.4.1 NC.- DPU lector de cinta perforada. Cada vez que se mecaniza una pieza se debe leer la cinta.

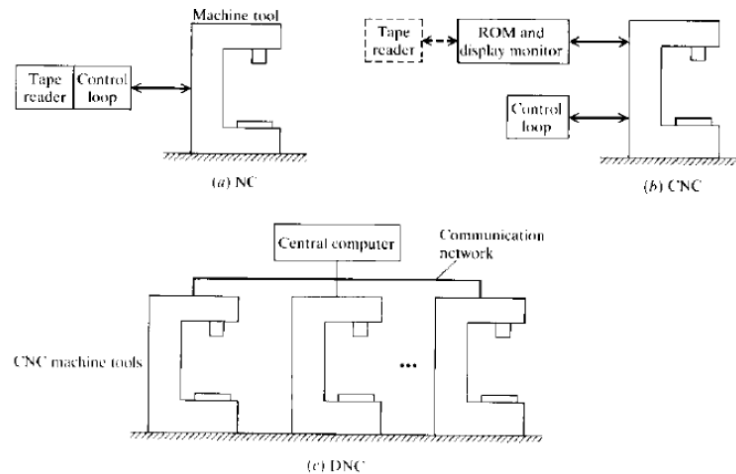


Figura 7 Máquinas: a) NC, b) CNC, c) DNC

5.4.2 CNC.- DPU es una ROM con un display, el programa NC se lee una vez. Capacidad de diagnóstico para mantenimiento y reparación de la máquina. Cada máquina con su programa. Sin realimentación. Incapaz generación informes (ratios producción, trabajo en curso, rechazos, etc.).



Figura 8 Taladradora CNC

5.4.3 DNC.- Host + máquinas CNC + red Host almacena programas NC y los carga en cada una de las máquinas. Realimentación. Generación de informes. Diferentes

niveles jerárquicos de ordenadores y redes. Ventaja: centraliza la información de control del sistema.

5.5 Sistemas de control.-

El puerto paralelo ha sido y sigue siendo uno de los puertos donde se realiza la conexión de nuestras impresoras, su nombre es Conector DB25.

En la siguiente figura observamos el puerto DB25

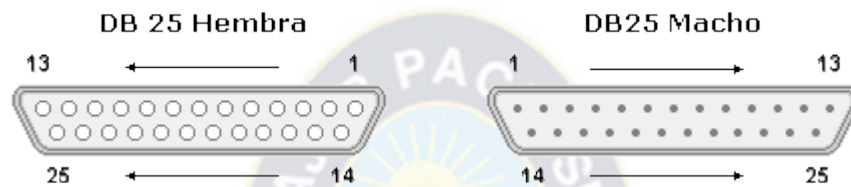


Figure 9. El Puerto paralelo

Este puerto dispone de tres registros de 8 bit cada uno (un byte).

Cada uno de estos registros se denomina puertos o PORT., y cada uno de sus bits, representa un pin determinado del puerto. Los pines que van del 18 al 25 (ambos inclusive): Son para masa, y sirven para conectar las descargas de los circuitos.

5.5.1 Puerto de datos (Pin 2 al 9): Es el PORT 888 y es de solo escritura, por este registro enviaremos los datos al exterior de la pc, nose envía señales eléctricas al ordenador por estos pines.

5.5.2 Puerto de estado (Pin 15, 13, 12, 10 y 11): Es el PORT 889 y es de solo lectura, por aquí enviaremos señales eléctricas al ordenador, de este registro solo se utilizan los cinco bits de más peso, que son el bit 7, 6, 5, 4 y 3 teniendo en cuenta que el bit 7 funciona en modo invertido.

5.5.3 Puerto de control (Pin 1, 14, 16 y 17): Es el correspondiente al PORT 890, y es de lectura/escritura, es decir, podremos enviar o recibir señales eléctricas, según nuestras necesidades. De los 8 bits de este registro solo se utilizan los cuatro de

menor peso o sea el 0, 1, 2 y 3, con un pequeño detalle, los bits 0, 1, y 3 están invertidos.

En la siguiente imagen puedes ver los tres registros, sus bits y los pines asignados a cada uno de ellos. La imagen corresponde a un conector DB-25 (Hembra).

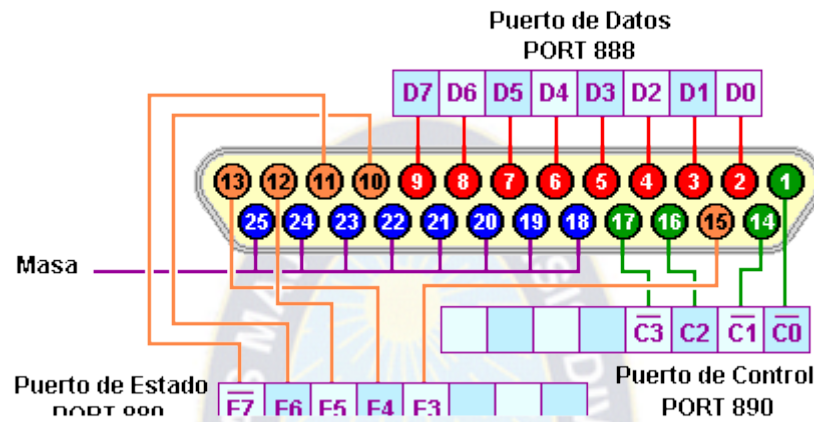


Figura 10. Configuración del Puerto paralelo.

La tensión de trabajo del puerto es de 5 voltios, por lo que se necesita una fuente estabilizada o regulada de tensión, esto es importante tenerlo en cuenta, ya que estaremos enviando señales al puerto. Por otro lado, si bien puedes utilizar la PC para enviar señales al exterior sin necesidad de una fuente externa, es recomendable utilizarla y así no se exige demasiado al puerto.

Ahora bien, si se realiza la activación de un bit de salida por el puerto, este permanecerá así hasta que se realice el cambio, es decir que se estará enviando 5V de forma continua hasta que lo pongas a 0.

5.5.4 Enviando datos por el puerto paralelo

El siguiente circuito nos permitirá enviar señales por el puerto paralelo, utilizaremos un buffer como el 74HC244 o el 74HCT245, se recomienda este último

Non-inverting 3-state outputs

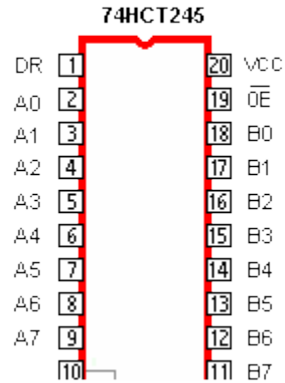


Tabla de Verdad

INPUTS		INPUTS/OUTPUTS	
\overline{OE}	DIR	A_n	B_n
L	L	A = B	inputs
L	H	inputs	B = A
H	X	Z	Z

Nota:
 X = Sin importancia
 Z = Alta impedancia

Figura 11. El integrado 74LS245 actúa como Buffer

Este integrado tiene la ventaja de ser bidireccional, es decir todos los pines A pueden ser entradas y los B salidas si DIR (pin1 DR) se encuentra a nivel bajo (L), el modo invertido se obtiene con DR a nivel alto. Si el terminal OE (pin 19) se encuentra a nivel alto (H) el integrado pone los pines A y B con alta impedancia, es decir ni entra ni sale señal alguna, en ese caso el estado de DR no tiene importancia. En la siguiente tabla se muestra los nombres de los pines y sus funciones correspondientes.

No. PIN	Nombre	Función
1	DIR	Control de dirección
2, 3, 4, 5, 6, 7, 8, 9	A0-A7	Entrada/Salida de datos
10	GND	Fuente (0V)
18, 17, 16, 15, 14, 13, 12, 11	B0-B7	Entrada/Salida de datos
19	\overline{OE}	Habilitación (Activo (L))
20	Vcc	Fuente (+5V)

Tabla 4. Nombre de los pines del Puerto paralelo.

En el siguiente esquema no se representaron todos los pines del puerto, sino los correspondientes al puerto de datos y los de masa.

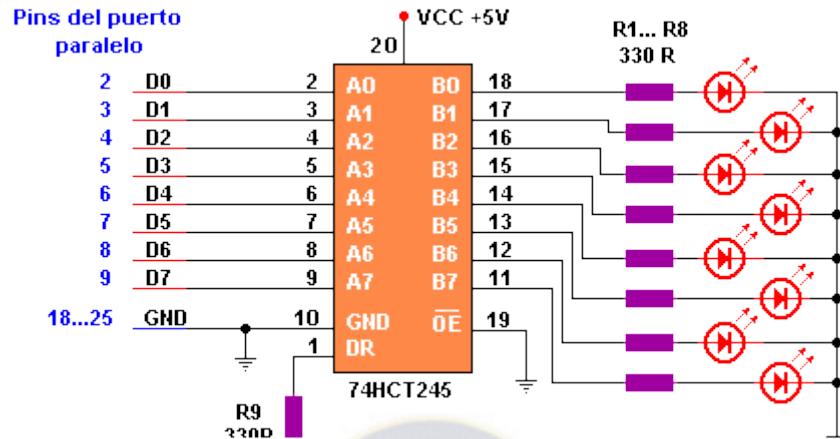


Figura 12. Circuito de control del Puerto paralelo y el integrado 74LS245

Este tipo de puerto es el más utilizado entre todas las maquinas CAM pero especialmente en las fresadoras, pantógrafos, y maquinas CNC debido a la facilidad q tiene en la sincronización de los motores además tomando en cuenta el código de interpretación del código CNC que si es almacenado en un microcontrolador no será optimo en el funcionamiento.

5.5.5 Características del microcontrolador AVR Atmega 16.-

El **AVR** es una arquitectura Harvard modificada de 8-bit RISC chip único microcontrolador que fue desarrollado por Atmel en 1996. El AVR fue una de las familias de microcontroladores primeros para utilizar en el chip de memoria flash para almacenamiento de programas, en oposición a una ROM de tiempo programable , EPROM , o EEPROM utilizado por otros microcontroladores en el momento.



Figura 13. Forma física del microcontrolador Atmega 16

Microcontrolador AVR de 8 bit de alto rendimiento y bajo consumo.

- Arquitectura RISC avanzada.
 - 131 instrucciones. La mayoría de un simple ciclo de clock de ejecución.
 - 32 x 8 registros de trabajo de propósito general.
 - Capacidad de procesamiento de unos 16 MIPS a 16 MHz.
 - Funcionamiento estático total.
 - Multiplicador On-Chip de 2 ciclos
- Memorias de programa y de datos no volátiles.
 - 16K bytes de FLASH autoprogramable en sistema.
Resistencia: 1.000 ciclos de escritura / borrado.
 - Sección de código añadida opcional con bits de bloqueo independientes.
Programación en sistema con el programa añadido On-Chip.
Operación de lectura durante la escritura.
 - 512 bytes de EEPROM.
Resistencia: 100.000 ciclos de escritura / borrado.
 - 1K bytes de SRAM interna.
 - Bloqueo (cerradura) programable para la seguridad del software.
- Interface JTAG.
 - Mantenimiento de eliminación de errores On-Chip.
 - Programación de FLASH, EEPROM, fusibles y bits de bloqueo a través de la interface JTAG.
 - Capacidades de Boundary Scan de acuerdo con el Standard (norma) JTAG.
- Características de los periféricos.
 - Dos Timer/Contadores de 8 bits con prescaler separado y modo comparación.
 - Un Timer/Contador de 16 bits con prescaler separado, modo comparación y modo de captura.
 - Comparador analógico On-Chip.

- Timer watchdog programable con oscilador separado On-Chip.
- Interface serie SPI maestro/esclavo.
- USART serie programable.
- Contador en tiempo real con oscilador separado.
- ADC de 10 bit y 8 canales.
 - 8 canales de terminación simple
 - 7 canales diferenciales sólo en el encapsulado TQFP.
 - 2 canales diferenciales con ganancia programable a 1x, 10x o 200x.
- 4 canales de PWM.
- Interface serie de dos hilos orientada a byte.
- Características especiales del microcontrolador.
 - Reset de Power-on y detección de Brown-out programable.
 - Oscilador RC interno calibrado.
 - Fuentes de interrupción externas e internas.
 - 6 modos de descanso: Idle, reducción de ruido ADC, Power-save, Power-down, Standby y Standby extendido.
- I/O y encapsulados
 - 32 líneas de I/O programables.
 - PDIP de 40 pines, TQFP y MLF de 44 pines.
- Tensiones de funcionamiento.
 - 2.7 - 5.5V (ATmega16L).
 - 4.5 - 5.5V (ATmega16).
- Niveles de velocidad.
 - 0 - 8 MHz (ATmega16L).
 - 0 - 16 MHz (ATmega16).

6. DESARROLLO DEL TRABAJO.-

La siguiente figura muestra en general todo el proceso realizado para nuestro trabajo de aplicación.

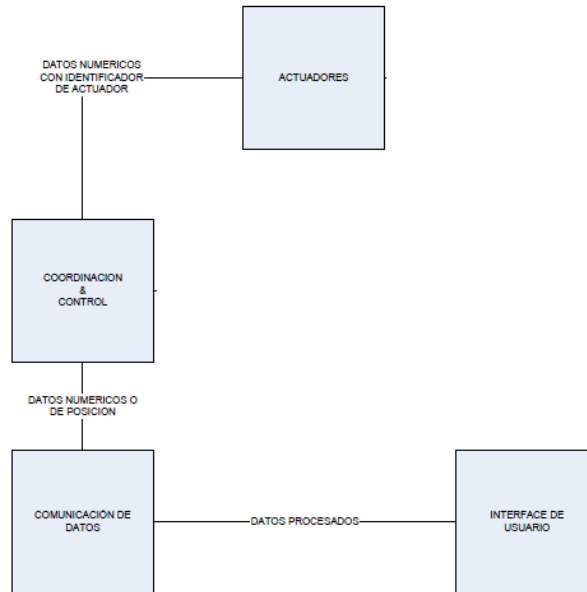


Figura 14 .Diagrama en bloques general de la Máquina CNC.

La interfaz utilizada para la realización de nuestro trabajo de aplicación se muestra en la siguiente figura.

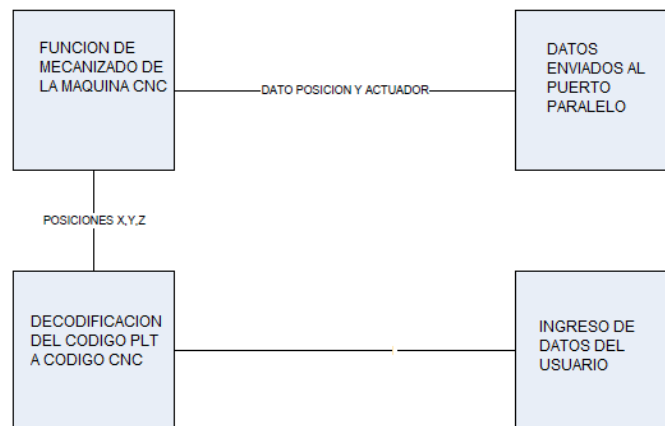


Figura 15. Diagrama en bloques de la interfaz.

El proyecto CNC consta de la “máquina” propiamente dicha, que tiene tres ejes de movimientos: el plano de trabajo con direcciones X e Y, y el eje vertical Z.

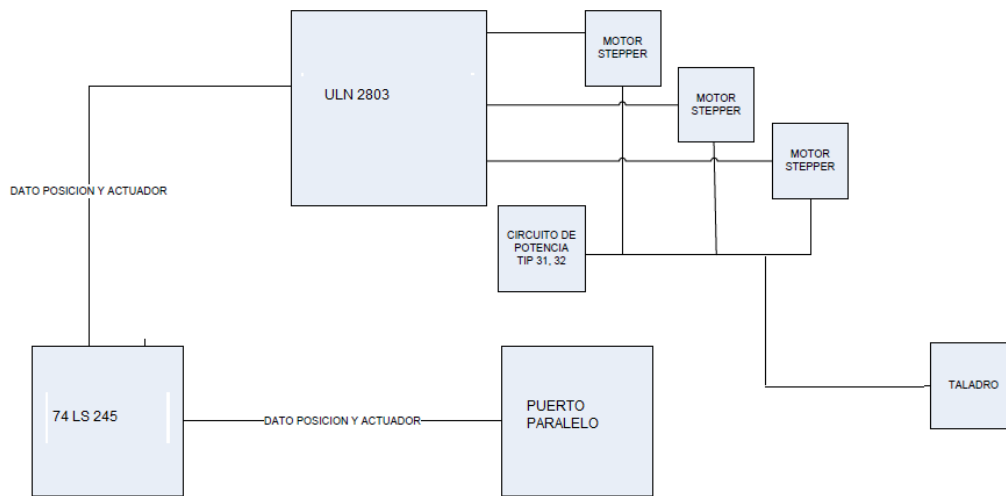


Figura 16. Diagrama en bloques de comunicación, control y coordinación

Para cada eje de movimiento, tiene un motor paso a paso con un tornillo de paso 1 mm. El motor requiere 96 pasos para girar una vuelta, de manera que un paso del motor, equivale a un desplazamiento cercano a la centésima de mm, lo que es una precisión muy interesante.

Puede realizar tallados en bajo o alto relieve.

Los tres motores son controlados por un programa en la PC, llamado CNC, que manda señales a una interface electrónica, que es la encargada de manejar la parte de potencia del CNC.

Los datos al programa CNC, son pasados a través de un archivo de texto con extensión .CNC.

El diseño del trazado, para este proyecto, tal como está creado, se debe hacer en AutoCAD, que es un programa usado muchísimo en diseño y CNC.

Se pueden usar otros programas de diseño, pero requieren de la modificación del programa decode.

Los programas de diseño generan archivos de impresión, de acuerdo a los drivers que usen. El AutoCAD genera archivos tipo PLT, con el driver que propongo usar. Este archivo .PLT es decodificado por decode.exe, para transformarlo en .CNC, que reconoce el programa de PC.

El programa CNC, simplemente carga el archivo.CNC, y ejecuta la tarea, parando al llegar a la última instrucción, avisando con un cartel.

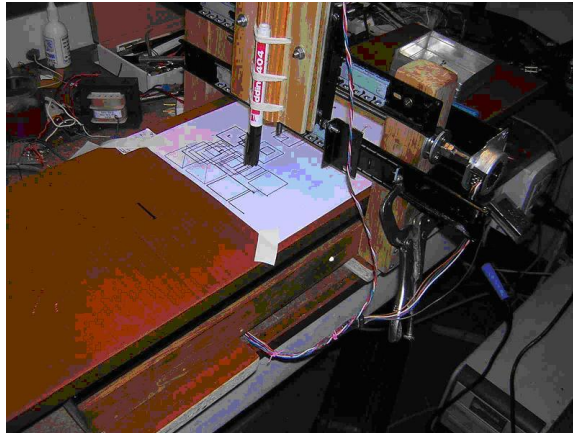


Figura 17. Maquina CNC casera

Esta imagen muestra este CNC en la etapa de construcción, trabajando como plotter.

El marcador está fijado con elásticos a una plaquita de 10 x 15 x 0.5 cm, a su vez fijada al soporte de la agujereadora (Que está desmontada, al igual que su almohadilla de masilla epoxi de apoyo).

6.1 Programas en Visual Basic 6.0

Para esta aplicación se crearon estos 3 programas:

CNC

DECODE

POSICIONADOR

El programa único es el CNC, que usa a los otros dos como subprogramas, para convertir el archivo PLT a CNC y el otro permite posicionar manualmente la herramienta en el origen elegido, para comenzar el trabajo.

Este CNC (máquina) no tiene un origen fijo, como por ejemplo el scanner. De manera que el origen se elige a voluntad, primero en el AutoCAD y luego en el CNC, mediante el posicionador manual.

El procedimiento usual de trabajo es:

- 1) Generar el dibujo AutoCAD.
- 2) Generar el archivo.PLT
- 3) Arrancar CNC.EXE
- 4) Generar el archivo.CNC
- 5) Cargar el archivo.CNC
- 6) Posicionar la herramienta en el origen
- 7) Ejecutar el Trazado o Trazar.

En el caso de un circuito impreso, el origen se puede ubicar en una esquina de la plaqueta, para que sea fácil identificarlo al posicionar.

6.2 Programa CNC

El dibujo en ACAD, se hace siempre en escala 1:1 y se plotea con esa escala, solo por practicidad. En este CNC de 96 pasos por mm, se debe usar una Escala de CNC de 2.4. Si por cualquier causa se desea achicar o agrandar el Trazado, se cambia su valor, menor o mayor a 1, para achicar o agrandar.

La Profundidad marcada en el TextBox, se refiere a la posición en que dejamos la punta de la herramienta cuando usamos el posicionador. Tener en cuenta que es conveniente dejarla 50 o 100 pasos por encima de la superficie a trabajar.

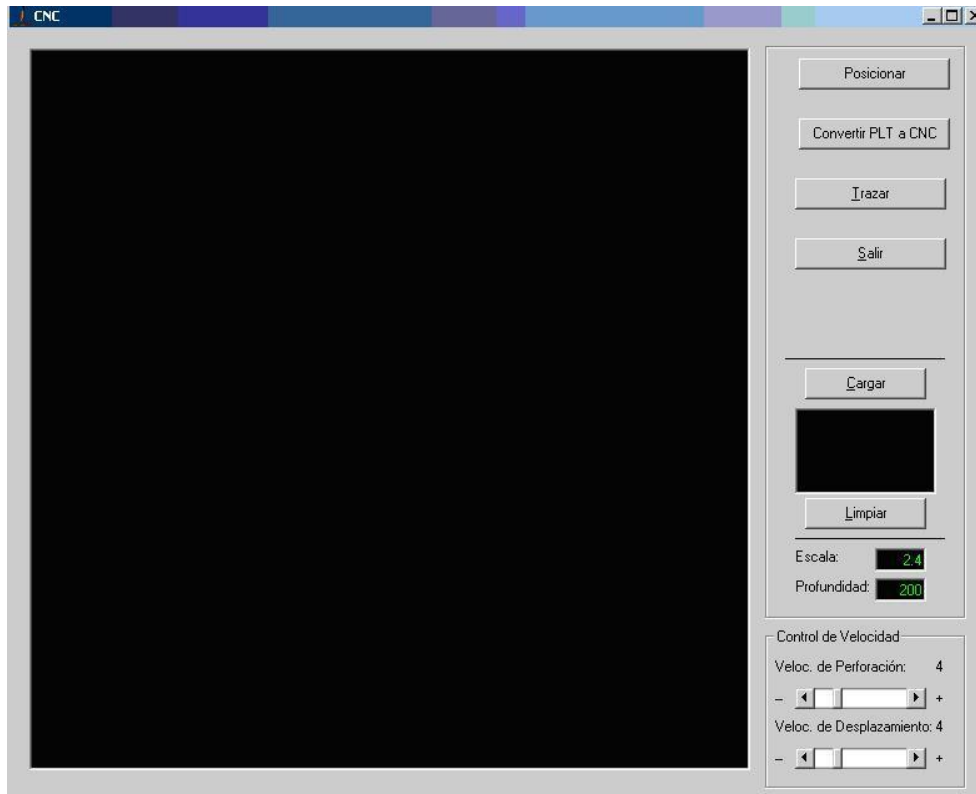


Figura 18. Ventana de presentación del programa CNC

La mayoría de los archivos PLT, tienen la misma estructura, me refiero más que nada al encabezado, que es muy similar en todos ellos, siempre teniendo en cuenta que las coordenadas deben estar en modo Absoluto, por supuesto.

Lo que nos interesa aquí es cómo realizar la lectura de esos archivos, así que tomaremos el ejemplo de un código para estudiarlo.

La intención de hacer este ordenamiento es con la idea de facilitarle la lectura del código al programa principal de la máquina (el programa de control) es decir, le dará mayor velocidad de lectura, mayor velocidad de desplazamiento, menor utilización de recursos de la PC. Para hacerlo deberemos tener en cuenta cuáles serán las

instrucciones que reconocerá nuestra máquina,

Los comandos que se utilizara en este proyecto y son las listadas a continuación

Comando Descripción	
PA	Absolute Position (Posicionamiento Absoluto): En algunos casos, especialmente al iniciar el código suele presentarse esta instrucción, para comenzar a graficar, así que nos puede ser de utilidad
PU	Pen-Up (Subir el Lápiz): Puede que sea un lápiz (cuando lo convierta en un Plotter) o puede que sea un Mini-Taladro, así que para no entrar en detalles lo llamaremos Herramienta de Trazado, o simplemente Herramienta, Entonces será "Subir la Herramienta"
PD	Pen-Down: (Bajar el Lápiz): Para nosotros, Bajar la Herramienta
LT	Line-Type (Tipo de Línea): En realidad no será de mucha importancia, ya que el tipo de línea que se utiliza por lo general es continua, ahora que... si el gráfico incluye otro tipo de línea, tendremos problemas. Lo importante en nuestro programa es que marca el inicio del trazado de la imagen.
SP	Select-Pen (Selección de lápiz): Bueno, esta instrucción no está implementada en nuestra máquina, pero nos puede servir para saber en qué momento terminó de trazar, ya que lo vi al final de código en la mayoría de los archivos PLT.

Tabla 4. Instrucciones utilizadas en el proyecto

Veamos cómo viene el inicio de un archivo PLT, y cuáles son los bloques que hay que quitar...

.(;.l81;;17:.N;19:IN;SC;PU;RO0;IP;IW;VS15;VS15;VS15;VS15;VS15;VS15;SP1;
PU;PA0,0;SP1;LT;PA53,421;PDPA53,397,77,372,125,372,174,397,174,469...

queda de la siguiente manera:

PUPA2645,445;PDPA2645,348,2621,58;PUPA2548,445;PDPA2596,445;**PU;PA0,0;
SP;**

Estos archivos están escritos en una sola línea de código, no existe un retorno de carro, no, nada de eso, es una sola línea, ahora se realizara un análisis de todo este código.

Los más importante del código que se genera son los bloques que siguen

PU;PA0,0; resulta que da la casualidad que se repiten al final del código, y tiene que ser así, ya que la máquina debe saber cuál será el punto de origen (coordenada 0,0) a demás ese será el punto de referencia principal y todos los otros desplazamientos estarán referidos a ese punto,

Si lo vemos con más detalle PU es levantar la herramienta, y PA0,0 desplazarse al punto Cero, que bien podría resumirse en un sólo bloque.

PU0, 0;

De acuerdo, esto será obligatorio tenerlo al comienzo y al final de nuestro código.

El bloque que sigue es **SP1;** en realidad este bloque no tiene importancia, ya que estamos trabajando con una sola herramienta, por lo tanto será ignorado.

Y apareció un bloque más **LT;** que no es tan importante, ya que el tipo de línea será siempre llena, es decir siempre será LT; pero es importante tenerlo en cuenta ya que **a partir de ahora comenzarán los desplazamientos**

Y comenzamos con **PA53, 421;** como ya saben PA es Posicionamiento Absoluto, sólo debe desplazarse a ese punto (53,421), sin subir ni bajar la herramienta de

trazado, así, tal como está.

De ahora en más no volverá a aparecer un bloque PA solo, sino acompañado de una instrucción PU o PD, tal como se ve en el resto del código, bueno la verdad es que sí aparece, pero al final (*esto ya está previsto*).

Otra observación importante, es que un bloque que tenga como instrucción PU, no aparecerá nunca como uno PD que suele venir con quinientas mil coordenadas en un solo bloque, y esto se debe a que ninguna máquina se la pasará paseando por la superficie de trabajo al divino gas (sin hacer ningún trazado), se supone que está destinada a trabajar y no a andar de juerga por allí, no es como nosotros Bueno, vamos a comenzar con nuestro primer programa.

6.3 Programa DECODIFICADOR



Figura 19. Ventana de presentación del programa DECODIFICADOR

Primeramentese carga todo el código en una variable, luego se busca la instrucción LT; la función que nos permite hallarla es:

InStrRev(líneaReg, "LT")

Esta función devuelve la posición de "L" en realidad, lo que hace es buscar en la variable **líneaReg** (Variable que contiene nuestro código) los caracteres LT, una vez lo encuentra, devuelve la posición del carácter anterior a L, esa posición la podría guardar en una variable, por ejemplo "ini", pero el bloque completo es LT; así que saltamos 2 caracteres más para pasar a las instrucciones que nos interesan.

ini=ini+2

Necesitamos un List para almacenar cada bloque de instrucciones, y la primera instrucción que necesitamos es **PU0,0**, a pesar de que sabemos que la siguiente instrucción será PA, vamos a ir leyendo carácter por carácter, y a concatenarlo para armar cada bloque, recuerda que cada bloque termina en un ;
También es necesario conocer la longitud del código, para así leer hasta el último carácter:

Longitud=Len(líneaReg)

Como **ini** es la posición actual del carácter a leer, entonces podemos leer mientras **ini<>Longitud**

Ahora, si el carácter leído es distinto al ; se irá concatenando en otra variable en mi caso será en (**PInstrucc**), el primer carácter a concatenar será **P**, el segundo puede ser **A, U o D**.

Si **PInstrucc="PA"** comenzamos a concatenar sus números hasta él;

Si **PInstrucc="PU"** no me interesa y sigo concatenando

Si **PInstrucc="PD"** tampoco, pero sigo concatenando

Si es **PInstrucc="PUPA"** entonces sí, Hago **PInstrucc="PU"** y comienzo a concatenar sus valores hasta el ;

Si es ***PInstrucc="PDPA"*** hago lo mismo que antes, ***PInstrucc="PD"*** y comienzo a concatenar sus valores hasta el ; Lo curioso aquí, es que pueden haber muchas coordenadas en un solo bloque PD, entonces hay que separarlos de dos en dos, es decir...

Si encuentro la primer coma, la tendré en cuenta. Si encuentro la segunda, terminé con un bloque, lo cargo en el List y hago ***PInstrucc=""*** es decir, limpio mi variable para comenzar de nuevo, y como estoy en un bloque "PD" hago ***PInstrucc="PD"***, y a este le concatenaré otro par de números, y así hasta que me encuentre con un ;

Finalmente el último **SP;** será ignorado y cuando hayamos alcanzado la longitud total de la variable, lo que haremos será agregarle nuevamente **SP;**, como dije antes... podríamos haberle puesto un **The End**,

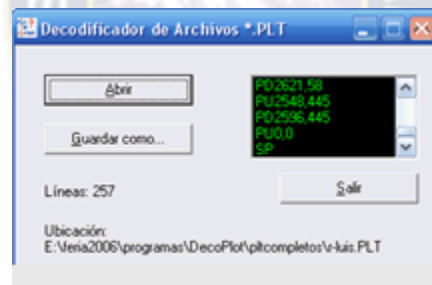


Figura 20. Proceso de decodificación

En el código final del programa hay otras cosas más, como llamadas al cuadro de diálogo Abrir, Guardar Como, un arreglo para evitar que se modifique el tamaño del formulario, un par de Labels para mostrar la ubicación y nombre del archivo, la cantidad de líneas de instrucciones, el List con sus propiedades, la limpieza del List una vez guardado el archivo y otras insignificancias

6.4 Programa POSICIONADOR

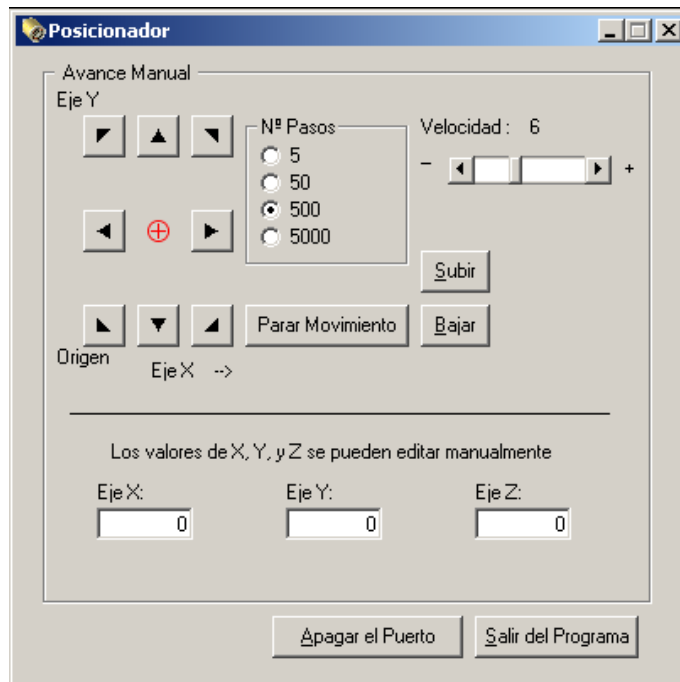


Figura 21. Ventana de presentación POSICIONADOR

Este programalo utilizaremos para mover los motores de cualquier eje, indicándole la cantidad de pasos que deberá dar, con posibilidades de apagar el puerto en cualquier momento si es que fuese necesario.

Su mayor utilidad, es desplazar la herramienta sobre la superficie de trabajo, especialmente cuando se desea ubicarla en el punto **Cero** o mejor aún, cuando necesitas saber cuál será la altura inicial, es decir, para cuando desees calibrar la máquina.

Las bases del funcionamiento del programa ya lo vimos anteriormente, aquí solo le agregaremos el control de los motores, que será muy similar al que usamos en los que describimos en páginas anteriores.

Para mover los motores, o mejor dicho, para girar el motor dos pasos consecutivos, debe haber un cierto retardo, que como ya vimos, un timer es demasiado lento, lo

que hacemos es utilizar un bucle For...Next.

La Cantidad de pasos a dar que en este caso sería la cantidad de puntos a desplazarse, se cargan en una variable llamada "**cuentapasos**" luego esta se va decrementando hasta llegar a cero, en ese caso se habrán alcanzado la cantidad de pasos que se deseaban dar.

Los valores que se muestran en los 3 Text inferiores, son simplemente la cantidad de pasos que se están avanzando, esto es para tener una idea de la distancia que se logra alcanzar para una determinada cantidad de pasos, que por supuesto, puedes ponerlos a cero en cualquier momento, y así reiniciar la cuenta.

Avance en los ejes X, Y

Como los avances en los ejes x,y son uno-a-uno, los valores para incrementar en cada coordenada serán **-1**, **0**, ó **1**, esto es debido a que los desplazamientos son horizontales, verticales, o diagonales (a 45°), por ejemplo, cuando el desplazamiento es vertical u horizontal, uno de los ejes tomará el valor **0** y el otro será **-1** ó **1**. Si el desplazamiento es en diagonal (para este programa siempre será a 45°) ninguno de los ejes podrá tomar el valor **0** y los posibles desplazamientos surgen de la combinación entre **-1** y **1**, bien, estos valores se cargan en una variable llamada plusX,plusY que también la vimos anteriormente.

Para entender mejor los desplazamientos, vamos a ir de adelante para atrás, y comencemos con las llamadas al Puerto

Call PortOut(888, (EjeX(Xi) + EjeY(Yi)))

Es importante recordar que estamos manejando matrices para cada motor, las variables ***Xi, Yi*** como ya sabemos, son los índices de los elementos de cada matriz, que deben ser 8 (por que estamos trabajando a medios pasos), ahora bien, para dar una secuencia de pasos estos deben irse incrementando o decrementando según el

caso, pues para ello es que se le suman los valores **plusX,plusY**

'incrementa la secuencia de pasos

$X_i = X_i + plusX$

$Y_i = Y_i + plusY$

If $plusX > 0$ And $X_i = 7$ Then $X_i = -1$

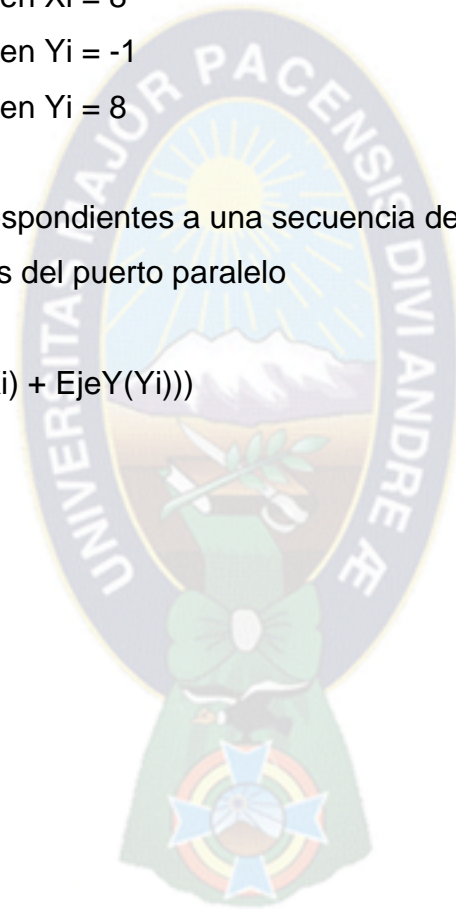
If $plusX < 0$ And $X_i = 0$ Then $X_i = 8$

If $plusY > 0$ And $Y_i = 7$ Then $Y_i = -1$

If $plusY < 0$ And $Y_i = 0$ Then $Y_i = 8$

suma de los valores correspondientes a una secuencia de pasos a dar, y que se envían al registro de datos del puerto paralelo

Call PortOut(888, (EjeX(X_i) + EjeY(Y_i)))



6.5 Esquema del controlador de los motores

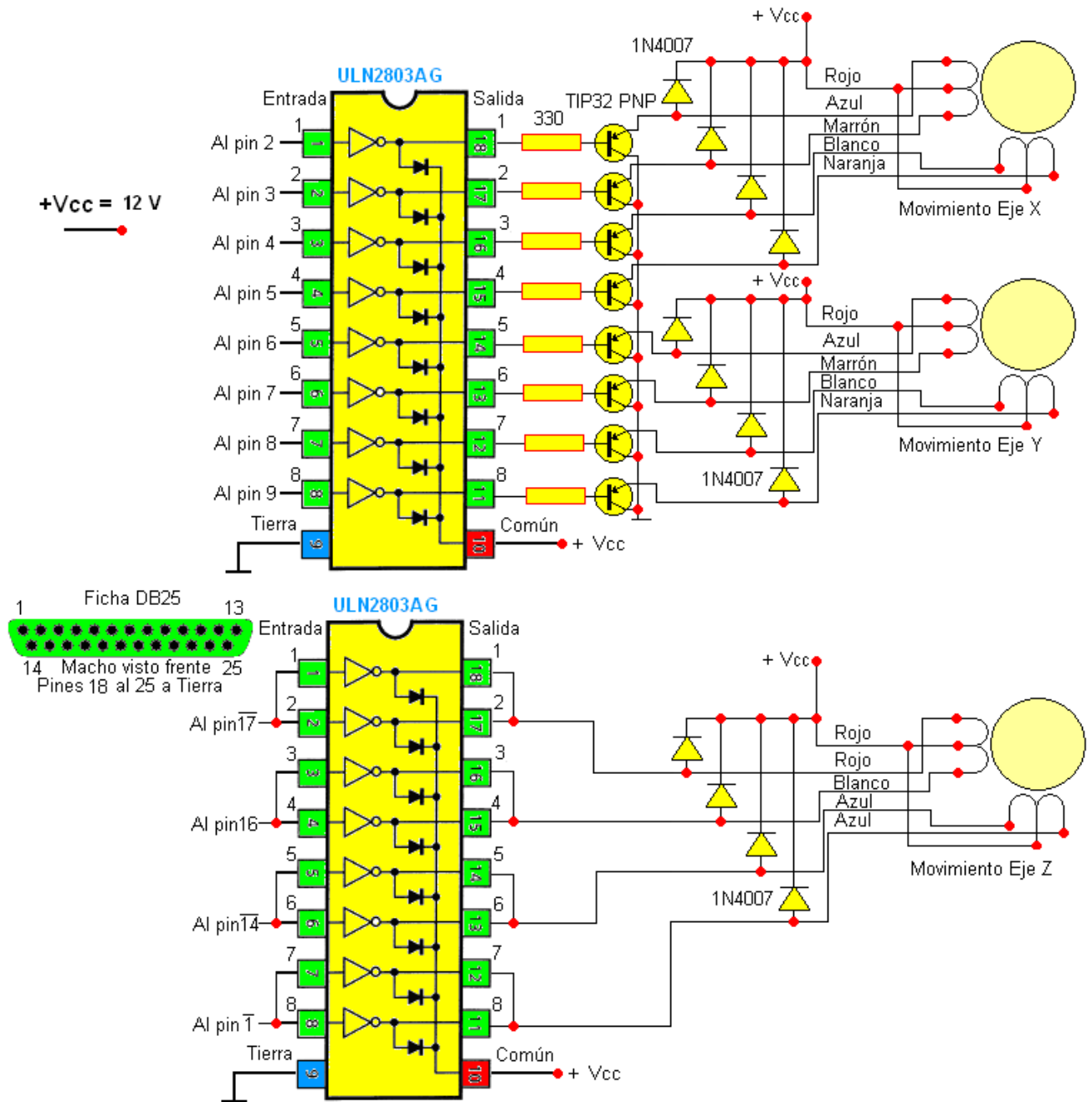


Figura 22. Circuito de control de motores

Los tres motores, como trabajan con medio paso, exceden la corriente máxima que soportan los ULN2803, por ello, los motores más grandes (1.4 Amp para Eje x y Eje

y) están controlados por los transistores, y el más chico (Del Eje z) tiene dos salidas puestas en paralelo. Toma 0.7 Amp.

Los diodos son para proteger a los transistores del pico de la tensión inducida al cortar la corriente.

La señal que llega a los ULN2803 está generada por la PC, a través del puerto paralelo, de Datos 0 a 7 y de Control 0 a 3.

Los motores son recuperados de dos impresoras en desuso, y trabajan satisfactoriamente con 12 Vcc.

El controlador está armado sobre una plaqueta perforada de 5 x 10 cm.

La fuente de alimentación es un trafo de 12 V, 3 Amp., con un puente rectificador y condensador electrolítico de 1000 uF 25V.

6.6 Esquema del control manual de la maquina CNC

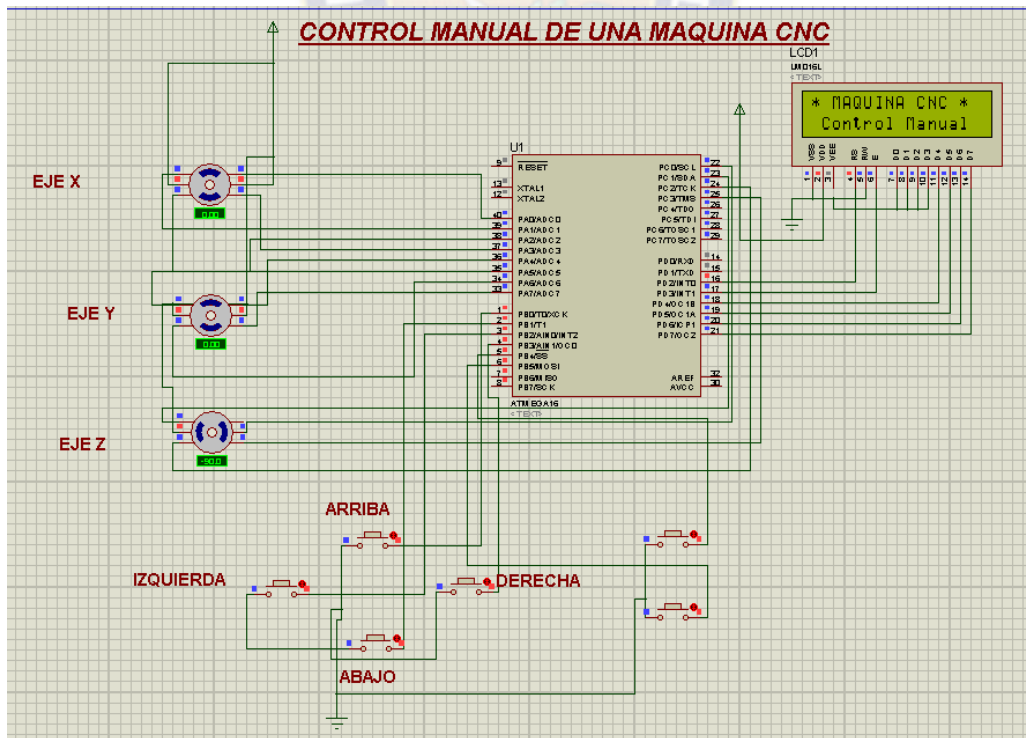


Figura 23. Circuito de control de motores auxiliar

Este circuito se encarga de controlar los motores de una manera manual es decir que no será el tallado de una manera continua si mediante la activación de los motores a la necesidad del usuario.

En la siguiente figura se muestra el diagrama de flujo donde muestra los pasos que sigue el código realizado en el programa "Micro C AVR".

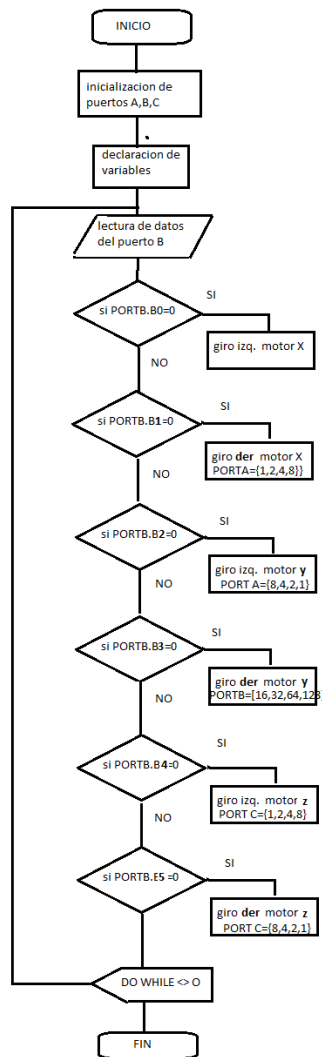


Figura24. Diagrama de flujo del código fuente del microcontrolador

Posteriormente mostramos el código realizado en Micro C



Figura 25. Detalles del programa MicroC.

donde se muestra las sentencias utilizadas para el funcionamiento del Atmega 16.

```
int vector[]={1,2,4,8} ;
int vector2[]={8,4,2,1};
int vector3[]={16,32,64,128};
int vector4[]={128,64,32,16};
int vector5[]={1,2,4,8} ;
int vector6[]={8,4,2,1};
int i,c=0;
```

```
sbit LCD_RS at PORTD2_bit;
sbit LCD_EN at PORTD3_bit;
sbit LCD_D4 at PORTD4_bit;
sbit LCD_D5 at PORTD5_bit;
sbit LCD_D6 at PORTD6_bit;
```

```

sbit LCD_D7 at PORTD7_bit;

sbit LCD_RS_Direction at DDD2_bit;
sbit LCD_EN_Direction at DDD3_bit;
sbit LCD_D4_Direction at DDD4_bit;
sbit LCD_D5_Direction at DDD5_bit;
sbit LCD_D6_Direction at DDD6_bit;
sbit LCD_D7_Direction at DDD7_bit;
// End LCD module connections

void main() {
    DDRA=0xff;
    DDRC=0xff;
    DDRB=0;
    PORTA=0;
    PORTC=0;
    PORTB=255;

    Lcd_Init();
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Cmd(_LCD_CURSOR_OFF);
    Lcd_Out(1, 1, "1");
    Lcd_Out(1, 1, "* MAQUINA CNC *");
    Lcd_Out(2, 1, " Control Manual ");
while(1)
    {
if (PINB.B2==0)
    {
        int i;
        PORTA=vector[i];
        Delay_ms(200);
        i=i+1;
if (i==4) i=0;

```



```

        Lcd_Cmd(_LCD_CLEAR);
        Lcd_Out(1, 1, "* MAQUINA CNC *");
        Lcd_Out(2, 1, "Motor x (izq).");
    }
    if(PINB.B3==0)
        {
            int i;
            PORTA=vector[i];
            Delay_ms(200);
            i=i+1;
        }
    if(i==4)i=0;
        Lcd_Cmd(_LCD_CLEAR);
        Lcd_Out(1, 1, "* MAQUINA CNC *");
        Lcd_Out(2, 1, "Motor x (der).");
    }
    if(PINB.B0==0)
        {
            int i;
            PORTA=vector3[i];
            Delay_ms(200);
            i=i+1;
        }
    if(i==4)i=0;
        Lcd_Cmd(_LCD_CLEAR);
        Lcd_Out(1, 1, "* MAQUINA CNC *");
        Lcd_Out(2, 1, "Motor y (ariba)");
    }
    if(PINB.B1==0)
        {
            int i;
            PORTA=vector4[i];
            Delay_ms(200);
            i=i+1;
        }
    if(i==4)i=0;
        Lcd_Cmd(_LCD_CLEAR);
        Lcd_Out(1, 1, "* MAQUINA CNC *");

```

```

        Lcd_Out(2, 1, "Motor y (abajo)");
    }
    if(PINB.B4==0)
    {
        int i;
        PORTC=vector5[i];
        Delay_ms(200);
        i=i+1;
    if(i==4) i=0;
        Lcd_Cmd(_LCD_CLEAR);
        Lcd_Out(1, 1, "* MAQUINA CNC *");
        Lcd_Out(2, 1, "Motor z (sube)");
    }
    if(PINB.B5==0)
    {
        int i;
        PORTC=vector6[i];
        Delay_ms(200);
        i=i+1;
    if(i==4) i=0;
        Lcd_Cmd(_LCD_CLEAR);
        Lcd_Out(1, 1, "* MAQUINA CNC *");
        Lcd_Out(2, 1, "Motor z (baja)");
    }
}
}
}

```

Este es un circuito de emergencia que responderá en los siguientes casos:

- Problemas en el computador.
- Problemas con el software CAM.
- Graficos o textos difíciles de realizar.
- Cambios de ultimo momento en el disenio del grafico a tallar, etc.

6.7 Consideraciones sobre el uso del AutoCAD

Se usa el AutoCAD, porque pareciera que es el programa más usado para diseño tanto por mecánicos como arquitectos y permite obtener archivos de impresión compatibles con CNC.

Se lo usa también en los CNC “en serio”, para pasar del diseño directamente a la fabricación de piezas.

Todos saben obtener una copia. Se recomienda el ACAD 2000, porque es muy versátil y no tan complicado. Se puede usar casi totalmente con los menús de pantalla.

Los dibujos deben ser hechos con líneas, poli líneas, curvas, textos.

El círculo no es una poli línea, y no es convertible a CNC con el driver ACAD mostrado. Para subsanar este inconveniente, si es requerido un círculo, se lo dibuja, y luego con BREAK se le corta un sector lo más chico posible, de una décima de mm, por ejemplo. Eso la convierte en poli línea.

Hay un driver que al crear el PLT convierte el círculo en una figura facetada, pero no lo uso porque al círculo lo destino exclusivamente para hacer las perforaciones. En este caso uso un círculo de 1 mm de radio.

Lo otro importante, es que el CNC sigue la secuencia del dibujo, de manera que conviene hacerlo ordenado, si se puede, para que el CNC no se la pase viajando con la herramienta levantada, de un lado para otro, sin hacer nada.

A continuación, viene un procedimiento lo más detallado posible, para obtener los archivos CNC necesarios para fabricar una plaqueta de circuito impreso.

El mismo procedimiento, sin las perforaciones, se puede usar para cualquier diseño de tallado en madera u otros materiales.

6.8 Diseño de la maquina

Será necesario entonces establecer las necesidades de movimiento en el espacio x, y, z, tanto de la pieza como de la herramienta para obtener la forma final de la estructura.

Las maquinas convencionales son de mesa móvil en la cual la pieza a mecanizar se mueve en los x, y , la herramienta en el z.

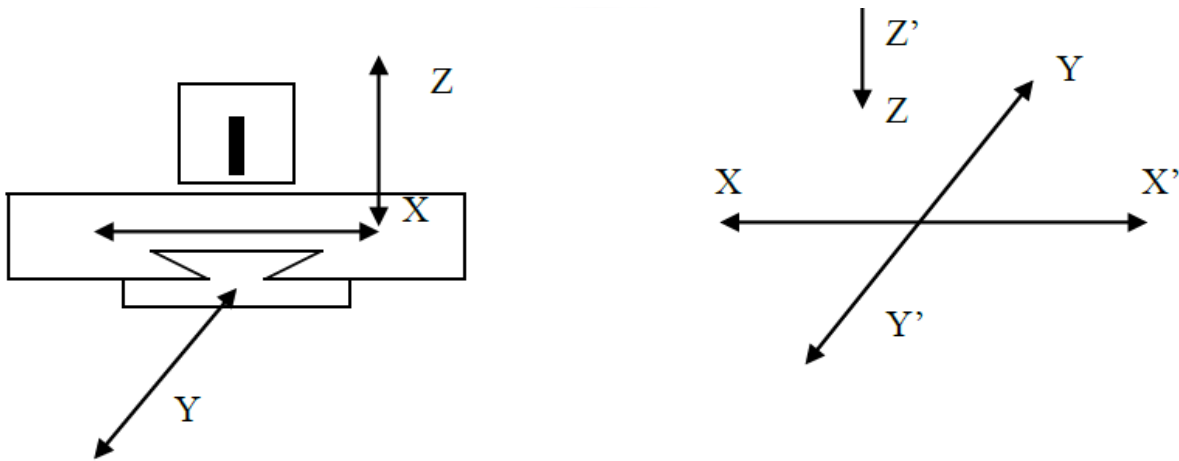


Figura 26Mesa Convencional, movimiento de material en X, Y.

Para que pueda desplazarse X sobre Y se dispone de guías en forma de cola de Milano que asegura el paralelaje de mecanizado pero tienen el inconveniente de su elevada fricción por su gran superficie de contacto, evita vibraciones, los motores para este caso deben ser de un par de arranque elevado para vencer la fricción inicial y producir el corte. Para el proyecto de estudio no será útil.

Se debe pensar en una estructura en la cual la mesa se desplace en dirección a un solo eje x ó y, y la herramienta se poseione en el sentido de los ejes restantes, así tenemos

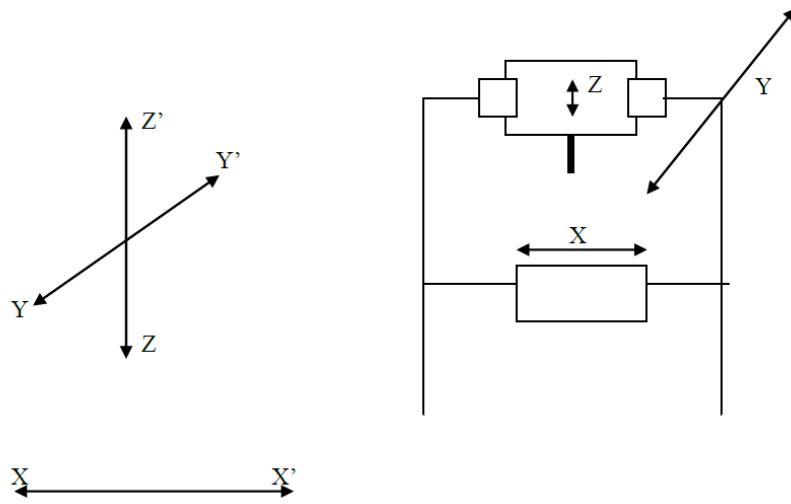
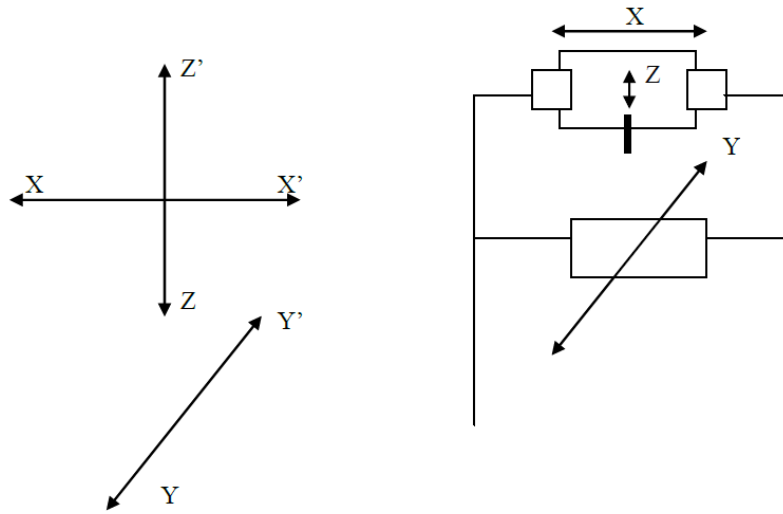


Figura27.Desplazamiento de la mesa en dirección X o Y.

Analizando las estructuras anteriores con sus ventajas y desventajas se elige finalmente la estructura (Figura 4.3) que a continuación se detalla:

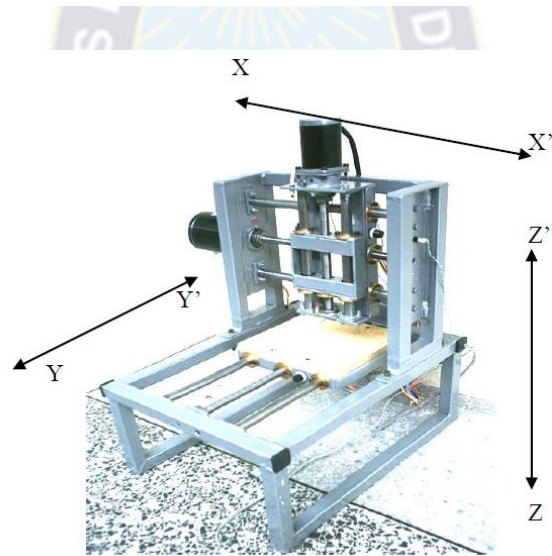
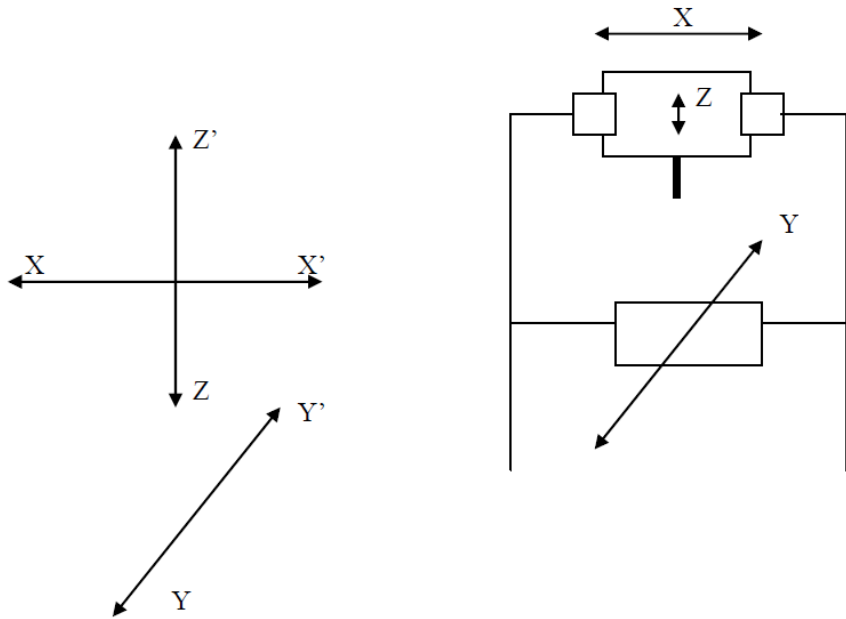


Figura28. Mesa Convencional, movimiento de material en Y, y la herramienta en y Z.

La fabricación es totalmente distinta se empleó la madera como material de fabricación en la siguiente figura se muestra en proceso de construcción la máquina que será puesta en funcionamiento

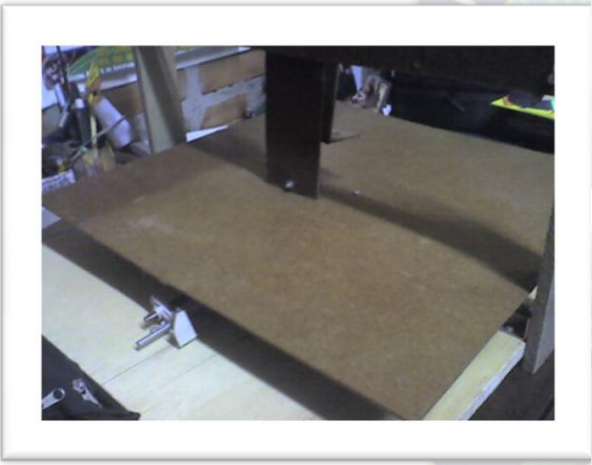
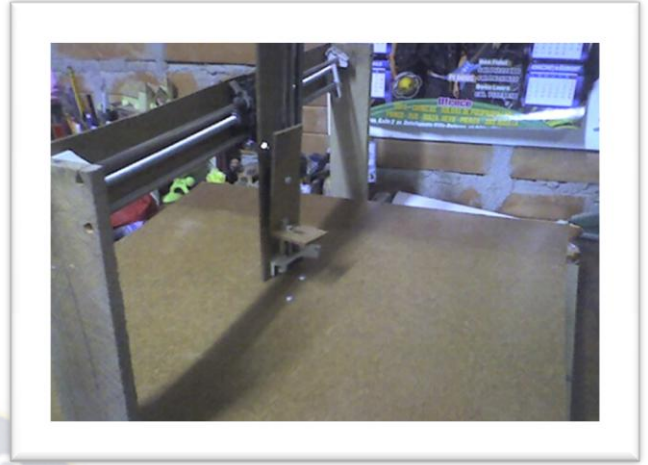


Figura 29. Maquina CNC en proceso de construcción

CONCLUSIONES.-

- ✓ El desarrollo del diseño y construcción de este tipo de prototipos, garantiza que con la tecnología e investigación de personal técnico nacional es posible abaratar los costos que involucra el adquirir maquinaria de origen internacional, con la finalidad de estar acorde a los avances tecnológicos que la industria del control numérico nos brinda.
- ✓ Este tipo de proyecto CNC elaborado a bajo costo servirá para que instituciones educativas de nivel medio y superior así como de capacitación profesional, tengan la posibilidad de adquirir y poder impartir este tipo de conocimientos que en la actualidad la industria de la automatización nos facilita.
- ✓ La construcción de maquinaria CNC, garantiza precisión y exactitud en la elaboración de piezas fabricadas en serie, disminuyendo el tiempo de mecanizado e incrementando la producción
- ✓ La construcción de este prototipo de Máquina Talladora CNC, servirá para que los alumnos de las diferentes Carreras entiendan, como funciona un sistema CAD / CAM, y cómo es el mecanizado automático de operaciones tan fundamentales como son el taladrado y el fresado vertical en forma didáctica.
- ✓ El programa elaborado para el control de posición y de velocidad, tanto de los motores paso a paso para cada una de las coordenadas de desplazamiento, como para el motor del cabezal taladrador, así como la visualización de algunos parámetros, se encuentra centralizado en el software CNC realizado en Visual Basic. El mismo que manejará los puertos tanto de entrada como de salida para los diferentes periféricos que en la máquina se encuentran.
- ✓ Para una mejor visualización en el sitio de mecanizado, tanto del desplazamiento de los ejes de coordenadas, variación de velocidad de los motores paso a paso así como del motor del cabezal taladrador; se ubicó un “visualizador de gráficos” en el software del prototipo
- ✓ El uso de motores paso a paso, respecto al de servomotores, se basa en que aquellos se los puede manejar digitalmente, sin realimentación, su velocidad se la puede controlar, son pequeños y poseen un elevado torque en bajas

revoluciones, lo que permite un bajo consumo de energía tanto en vacío como en plena carga, su mantenimiento es mínimo debido a que no tiene escobillas.

- ✓ Existe en la Web, algunos programas CAD/CAM gratuitos que generan codificaciones para el mecanizado en forma confusa.
- ✓ Se ha optado por utilizar el puerto paralelo ya que las instrucciones de la interpretación de código es muy complejo para que un microcontrolador lo tenga en su memoria. En los lenguajes de ensamblador de un microcontrolador no posee la interpretación o las instrucciones necesarias para decodificar un código HPGL del AutoCAD debido a ese problema se utilizó el puerto paralelo, se sabe que ya no es común este puerto pero la mayoría de las maquinas CNC q existen en el mercado funcionan con este puerto.
- ✓ Por esta razón y que la maquina tenga un circuito de respaldo se optó por desarrollar un circuito con el microcontrolador Atmega 16 que actúa como un controlador manual de la maquina dando asi dos opciones de control sobre esta.

7. BIBLIOGRAFIA.-

<http://r-luis.xbot.es/cnc>

<http://www.luberth.com/plotter/ditwasplotter.htm>

<http://www.jtronics.de/werkstatt/cnc-pcb-fraese.html>

(Anexos)

Código Fuente del programa CNC.-

```
Option Explicit
Dim cCadena As String
Dim cLetra As String * 1
Dim cCodigoP As String * 2
Dim cX As String, cY As String
Dim nComa As Integer, nLargoCad As Integer
Dim n As Integer, nFin As Integer, Rta As Integer
Dim xi As Single, yi As Single
Dim bArriba As Boolean, bSubir As Boolean
Dim BitEjeZ(8) As Byte, BitEjeX(8) As Byte, BitEjeY(8) As Byte
Dim nBitx As Integer, nBity As Integer, nBitz As Integer
Dim VelocDesp As Integer, VelocPerf As Integer, TT As Integer, T As Integer
'variables para retardos
Dim nNewX As Single, nNewY As Single, nNewZ As Integer           'variables
para cálculo de desplazamiento
Dim nPlusx As Single, nPlusy As Single                           'variables
para cálculo de desplazamiento
Dim nPrevx As Single, nPrevy As Single                           'variables
para cálculo de desplazamiento
Dim Profund As Integer, nEscala, nPrevz As Integer              'variables
para cálculo de desplazamiento
Dim Puerto                                     ' Para cambiar de Puerto
LPT1 o LPT2
Dim Dibujar As Boolean                                     ' Para dibujar solamente
en pantalla
Private Sub B_Convertidor_Click() ' Arranca el programa para Convertir .PLT
a .CNC
Shell "decode.exe", vbNormalFocus
End Sub
Private Sub B_Posicionar_Click() ' Arranca el programa que permite
posicionar manualmente el inicio del trazado
Shell "Posicionador.exe", vbNormalFocus
End Sub
Private Sub B_Cargar_Click()
Timer1.Enabled = False
```

```

lstCodigo.Clear          ' Borra el contenido del ListBox
Picture1.BackColor = &H0&  ' Borra la imagen o pinta el picture de negro
    On Error GoTo solucion ' CancelError es True.
    CommonDialog1.Filter = "All Files (*.*)|*.*|Ploter (*.cnc)|*.cnc" '
Establece los filtros.
    CommonDialog1.FilterIndex = 2 ' Especifica el filtro predeterminado.
    CommonDialog1.ShowOpen ' Presenta el cuadro de diálogo Abrir.
Dim lineareg As String
    Open CommonDialog1.FileName For Input As #2
n = 0
' Carga el archivo plt en un list
    While Not EOF(2) ' Carga el archivo plt en un list y lee mientras el
archivo no termine
Line Input #2, lineareg ' lee desde la primera línea
lineareg = Trim(lineareg)
lstCodigo.AddItem lineareg, n ' Agrega la línea en el list box
    n = n + 1 ' Incrementa una línea
    lstCodigo.Refresh ' Actualiza el ListBox
Wend
    Close #2
    n = 0
Label15.Caption = "Archivo cargado a ejecutar es " + CommonDialog1.FileName
solucion:
Exit Sub
End Sub
Private Sub B_Mecanizar_Click()
nPrevz = 0
nPrevx = 0
nPrevy = 0
lstCodigo.ListIndex = 1 'Va a la posición 1 del Código CNC
Picture1.BackColor = &H0&  ' Borra la imagen o pinta el picture de negro
bSubir = True
nEscala = Val(txtEscala.Text) ' Carga la escala
Label14.Caption = "Para hacer otra pasada, volver a pulsar ""Mecanizar""."
Timer1.Enabled = True
End Sub
Private Sub B_Dibujar_Click()
nPrevz = 0

```

```

nPrevx = 0
nPrevy = 0
lstCodigo.ListIndex = 1      'Va a la posición 1 del Código CNC
Picture1.BackColor = &H0&    ' Borra la imagen o pinta el picture de negro
bSubir = True
nEscala = Val(txtEscala.Text)      ' Carga la escala
Dibujar = True                    ' Dibuja en pantalla solamente
Label14.Caption = "Cuando Dibuja Solamente, no se tienen en cuenta las
Velocidades."
Timer1.Enabled = True
End Sub
Private Sub Form_Load()
Dibujar = False                  ' No, dibuja en pantalla solamente
If Option2 = True Then Puerto = 60416
If Option1 = True Then Puerto = 888
Out Puerto, 0                    ' Apaga el puerto de datos
Out Puerto + 2, 11               ' Apaga el puerto de control
hscPerforacion.Value = 1        ' carga valores iniciales para la velocidad de
motores
hscDesplazamiento.Value = 1     ' carga valores iniciales para la velocidad de
motores
hscProfundidad.Value = 150      ' Carga valor inicial de profundidad
' Cargar las matrices con los valores de puerto para los tres ejes.
Corresponden a giro de a medio paso.
BitEjeZ(0) = 10: BitEjeZ(1) = 8:  BitEjeZ(2) = 9:  BitEjeZ(3) = 13
BitEjeZ(4) = 15: BitEjeZ(5) = 7:  BitEjeZ(6) = 3:  BitEjeZ(7) = 2
BitEjeX(0) = 1:  BitEjeX(1) = 3:  BitEjeX(2) = 2:  BitEjeX(3) = 6
BitEjeX(4) = 4:  BitEjeX(5) = 12: BitEjeX(6) = 8:  BitEjeX(7) = 9
BitEjeY(0) = 16: BitEjeY(1) = 48: BitEjeY(2) = 32: BitEjeY(3) = 96
BitEjeY(4) = 64: BitEjeY(5) = 192: BitEjeY(6) = 128: BitEjeY(7) = 144
End Sub
Private Sub Form_Resize()
' si se modifica la ventana y no está minimizado entonces
If WindowState <> 1 Then
'se desplazan el picture y el frame con sus controles
Picture1.Width = Consola.Width - 3210
Picture1.Height = Consola.Height - 960
Frame1.Left = Consola.Width - 2790

```

```

Frame2.Left = Consola.Width - 2790
'aquí se mantendran la coordenada de origen de y
'en la parte inferior de picture aunque este cambie de tamaño
Picture1.ScaleTop = -Picture1.Height + 100 ' Corre 100 puntos el origen
para arriba, para que se dibuje bien la línea de posicion cero
Picture1.ScaleLeft = -10 ' Corre 10 puntos el origen
para derecha, para que se dibuje bien la línea de posicion cero
End If
End Sub
Private Sub hscDesplazamiento_Change()
Label9.Caption = hscDesplazamiento.Value 'muestra el valor de la velocidad
en los ejes x,y
End Sub
Private Sub hscPerforacion_Change()
Label8.Caption = hscPerforacion.Value 'muestra el valor de la velocidad en
el eje z
End Sub
Private Sub hscProfundidad_Change()
Label11.Caption = hscProfundidad.Value
End Sub
Private Sub Timer1_Timer()
nEscala = Val(txtEscala.Text) 'carga la escala
cCadena = "" 'limpia la variable que carga la 1°
línea
cCadena = lstCodigo.Text 'lee la primer línea de código
nLargoCad = Len(cCadena) 'calcula longitud del código
cCodigoP = Left(cCadena, 2) 'toma las dos primeras letras de la
línea
On Error GoTo Solutions 'por si surge algún error

If cCodigoP = "SP" Then 'si se lee la última línea de
código
Timer1.Enabled = False 'detiene el timer
Rta = MsgBox("Grafico Finalizado", vbInformation, "Graficadora
CNC") 'y muestra un mensaje de fin de trazado
Out Puerto, 0
Out Puerto + 2, 11
Dibujar = False

```

```

End If
    lstCodigo.ListIndex = lstCodigo.ListIndex + 1
    If nLargoCad > 3 Then
nComa = InStr(cCadena, ",")
cX = Mid(cCadena, 3, nComa - 3)
nNewX = Int(Val(cX) * nEscala)
cY = Mid(cCadena, nComa + 1, nLargoCad - Len(cX) - 1) 'carga el valor para
el eje y
nNewY = Int(Val(cY) * nEscala)           'lo multiplica por la escala
End If
    Timer1.Enabled = False
    If Dibujar = True Then
DibujarSolamente
    Else
    Mover
End If
Solutions:
    Exit Sub
End Sub
Private Sub B_Salir_Click()
Out Puerto, 0
Out Puerto + 2, 11
End
End Sub
Private Sub Mover()
If Option2 = True Then Puerto = 60416
If Option1 = True Then Puerto = 888
'Desplazamiento para el eje z
    '-----si el taladro sube-----
If cCodigoP = "PU" Then
nBitz = 0
Subir:
DoEvents
    VelocPerf = (15 - hscPerforacion.Value) * 10    'velocidad para
subir
    If nPrevz > 0 Then           'sube hasta la posición 0
nPrevz = nPrevz - 1
nBitz = Int(nPrevz) Mod 8

```



```

        Out Puerto + 2, BitEjeZ(nBitz) 'aquí se manda el dato al puerto
'Retardo
        For TT = 1 To VelocPerf
            For T = 1 To 10000: Next T
        Next TT
        GoTo Subir
    End If
End If
    '-----si el taladro está arriba debe bajar-----
If cCodigoP = "PD" Then
nBitz = 0
Bajar:
DoEvents
    VelocPerf = (15 - hscPerforacion.Value) * 10    'velocidad para
bajar
nNewZ = hscProfundidad.Value
    If nPrevz < nNewZ Then 'baja hasta la posicion indicada en nNewZ
nPrevz = nPrevz + 1
nBitz = Int(nPrevz) Mod 8
    Out Puerto + 2, BitEjeZ(nBitz) 'aquí se manda el dato al puerto
'Retardo
        For TT = 1 To VelocPerf
            For T = 1 To 10000: Next T
        Next TT
        GoTo Bajar
    End If
'de ahora en más el taladro permanece abajo
End If
    Out Puerto + 2, 11
'*****
'Desplazamiento para los ejes x-y

'*****
If cCodigoP = "PU" Then bArriba = False 'sube lápiz
If cCodigoP = "PD" Then bArriba = True  'baja lápiz
'*****
graficar:
DoEvents

```

```

    VelocDesp = (15 - hscDesplazamiento.Value) * 10
    'busca diferencias entre origen y destino
    If nNewX <> nPrevx Then xi = Abs(nNewX - nPrevx)           ' Es el
incremento de los valores
    If nNewY <> nPrevy Then yi = Abs(nPrevy - nNewY)           ' Es el
incremento de los valores

If xi > yi Then
nPlusx = 1
nPlusy = yi / xi
    End If
If xi < yi Then
nPlusx = xi / yi
nPlusy = 1
    End If
If xi = yi Then
nPlusx = 1
nPlusy = 1
    End If
    If nPrevx < nNewX Then nPrevx = nPrevx + nPlusx
    If nPrevx > nNewX Then nPrevx = nPrevx - nPlusx
    If nPrevy < nNewY Then nPrevy = nPrevy + nPlusy
    If nPrevy > nNewY Then nPrevy = nPrevy - nPlusy

nBitx = Int(nPrevx) Mod 8
nBity = Int(nPrevy) Mod 8
    Out Puerto, BitEjeX(nBitx) + BitEjeY(nBity)
    If bArriba = True Then Picture1.PSet (nPrevx, -nPrevy), RGB(255, 0,
0)
For TT = 1 To VelocDesp
    For T = 1 To 13000: Next T    ' Modificado para ajustar velocidad en
nueva PC Alfredo
    Next TT
    'si termina de dibujar detiene el timer
If nPrevy = nNewY And nPrevx = nNewX Then
Timer1.Enabled = True
Else
    GoTo graficar

```



```

End If
End Sub
Private Sub DibujarSolamente()
'Desplazamiento para el eje z
'-----si el taladro sube-----
If cCodigoP = "PU" Then
nBitz = 0
Subir:
DoEvents
    If nPrevz > 0 Then
        nPrevz = nPrevz - 1
nBitz = Int(nPrevz) Mod 8
        GoTo Subir
    End If
End If
'-----si el taladro está arriba debe bajar-----
If cCodigoP = "PD" Then
nBitz = 0
Bajar:
DoEvents
    'VelocPerf = (15 - hscPerforacion.Value) * 10    'velocidad para
bajar
nNewZ = hscProfundidad.Value
    If nPrevz < nNewZ Then 'baja hasta la posicion indicada en nNewZ
nPrevz = nPrevz + 1
nBitz = Int(nPrevz) Mod 8
        GoTo Bajar
    End If
    'de ahora en más el taladro permanece abajo
End If

'Desplazamiento para los ejes x-y

'*****
If cCodigoP = "PU" Then bArriba = False 'sube lápiz
If cCodigoP = "PD" Then bArriba = True  'baja lápiz
graficar:
DoEvents

```

```

        'busca diferencias entre origen y destino
        If nNewX <> nPrevX Then xi = Abs(nNewX - nPrevX)           ' Es el
incremento de los valores
        If nNewY <> nPrevY Then yi = Abs(nPrevY - nNewY)           ' Es el
incremento de los valores
        'según la diferencia que exista entre x,y hay tres posibilidades
        'incrementará el eje x, luego calculará el de y
If xi > yi Then
nPlusx = 1
nPlusy = yi / xi
        End If
'incrementará eje y, luego calculará el de x
If xi < yi Then
nPlusx = xi / yi
nPlusy = 1
        End If
'incrementará los dos por igual
If xi = yi Then
nPlusx = 1
nPlusy = 1
        End If
        'se actualizan las coordenadas mientras avanza
If nPrevX < nNewX Then nPrevX = nPrevX + nPlusx
        If nPrevX > nNewX Then nPrevX = nPrevX - nPlusx
        If nPrevY < nNewY Then nPrevY = nPrevY + nPlusy
        If nPrevY > nNewY Then nPrevY = nPrevY - nPlusy
nBitx = Int(nPrevX) Mod 8
nBity = Int(nPrevY) Mod 8
'pixeliza la línea en el Picture1 (color rojo)
If bArriba = True Then Picture1.PSet (nPrevX, -nPrevY), RGB(255, 0, 0)
'si termina de dibujar detiene el timer
If nPrevY = nNewY And nPrevX = nNewX Then
Timer1.Enabled = True
        Else
                GoTo graficar 'Si no terminó, sigue graficando
End If
End Sub

```

Codigo Fuente Programa DECODIFICADOR .-

```

Option Explicit
Dim LineaReg As String
Dim Largo As Double
Dim PInstrucc As String
Dim Letra As String      '| listado de variables utilizadas
Dim Instruccion As String
Dim ComaBandera As Integer
Dim Directorio As String
Dim ini 'As Integer
Dim n          ' Contador de las filas del ListBox
Dim i          ' Contador usado en la escritura del archivo
Private Sub cmdAbrir_Click()
LineaReg = "": Largo = 0: PInstrucc = "": Letra = "": Instruccion = "" '
Poner a cero en la repetición
ComaBandera = 0: Directorio = "": ini = 0 '
Poner a cero en la repetición
lstCodigo.Clear          ' Limpiar el ListBox en la repetición
On Error GoTo solucion  ' CancelError es True.
CommonDialog1.Filter = "All Files (*.*)|*.*|Plotter (*.plt)|*.plt" '
Establece los filtros.
CommonDialog1.FilterIndex = 2 ' Especifica el filtro predeterminado.
CommonDialog1.ShowOpen      ' Presenta el cuadro de diálogo Abrir.
lstCodigo.AddItem "PU0,0", 0 'carga la primera línea de código en el
list
Open CommonDialog1.FileName For Input As #2 'Abre el archivo y lo carga en
la variable LineaReg
Directorio = CommonDialog1.FileName
Line Input #2, LineaReg
Close #2
' busca el inicio de código con LT
n = 1
LineaReg = Trim(LineaReg) ' quita espacios en blanco
Largo = Len(LineaReg)    ' obtiene la longitud del código
ini = InStrRev(LineaReg, "LT") ' busca LT en lieareg, devuelve la posición
de L
ini = ini + 2            ' salta hasta después del ;

```

```

While ini <> Largo          ' lee mientras la longitud del archivo no
termine
    'ahora comienza a armar el código letra por letra
    Letra = Mid(LineaReg, ini, 1)
ini = ini + 1
    'si la letra no es ";" va formando el código
If Letra <> ";" Then PInstrucc = PInstrucc + Letra
'***** en caso de que sea PA toma los valorea para X y para Y
If PInstrucc = "PA" Then
    Letra = ""
'mientras no encuentre ";" va concatenando la instrucción
Do While Letra <> ";"
    PInstrucc = PInstrucc + Letra
Letra = Mid(LineaReg, ini, 1)
ini = ini + 1
    Loop
    'cuando llega al ";" carga la instrucción en el list
lstCodigo.AddItem PInstrucc, n
    n = n + 1
lstCodigo.Refresh
    PInstrucc = ""          'limpia la variable para comenzar de nuevo
End If
'***** En caso de que sea PDPA
If PInstrucc = "PDPA" Then
Reinipd:
PInstrucc = "PD"          'lo deja como PD
    Letra = ""
    ComaBandera = 0
    'mientras no llegue al ";" va concatenando la instrucción
    Do While Letra <> ";"
        PInstrucc = PInstrucc + Letra          'comienza a cargar las
coordenadas
        Letra = Mid(LineaReg, ini, 1)          'una por una
        'si encuentra la primera coma, la tendrá en cuenta
        If Letra = "," Then ComaBandera = ComaBandera + 1
        'si encuentra la segunda, debe cargar la instrucción en el
list
        If Letra = "," And ComaBandera = 2 Then

```

```

        ComaBandera = 0
        lstCodigo.AddItem PInstrucc, n 'agrega la instrucción
en el list
        n = n + 1
ini = ini + 1
        GoTo Reinipd 'vuelve por la
siguiente coordenada
End If
ini = ini + 1
        Loop
lstCodigo.AddItem PInstrucc, n
n = n + 1
lstCodigo.Refresh
        PInstrucc = "" 'limpia la variable para cargar nueva
instrucción
End If
'***** en caso de que sea PU
If PInstrucc = "PUPA" Then
        PInstrucc = "PU" 'lo deja como PU
        Letra = ""
        Do While Letra <> ";"
                PInstrucc = PInstrucc + Letra 'carga las coordenadas
                Letra = Mid(LineaReg, ini, 1) 'letra por letra
ini = ini + 1
        Loop
        lstCodigo.AddItem PInstrucc, n 'pasa la instrucción al
list
        n = n + 1
        lstCodigo.Refresh
        PInstrucc = "" 'limpia la variable para cargar nueva
instrucción
End If

'***** en caso de que sea CI40 o sea Círculo de 1 mm, para
perforaciones
If PInstrucc = "CI40" Then

```

```

        PInstrucc = "PD"                ' lo deja como PD, para
perforar
        Letra = ""                      ' No agrega coordenadas
        Do While Letra <> ";"           ' Busca el ";"
            PInstrucc = PInstrucc + Letra ' carga las coordenadas (No
carga)
            Letra = Mid(LineaReg, ini, 1) ' letra por letra
        ini = ini + 1                   ' Se deja el do while, para que incremente
el ini
        Loop                            ' Cierre del do while
        lstCodigo.AddItem PInstrucc, n   ' pasa la instrucción al list
        n = n + 1                       ' Incrementa numero de fila
        lstCodigo.AddItem "PU", n       ' Agrega la subida del Eje Z
        n = n + 1                       ' Incrementa numero de fila
        lstCodigo.Refresh               ' Actualiza la vista en
pantalla
        PInstrucc = ""                  ' limpia la variable para
cargar nueva instrucción
    End If                               ' Termina esta subrutinita
'*****
*****
Wend
lstCodigo.AddItem "SP", n               ' si ya terminó agrega SP al final del list
    lstCodigo.Refresh                   ' Actualiza la vista en pantalla
    labell1.Caption = "Líneas: " & lstCodigo.ListCount ' muestra la
cantidad de líneas de código
    n = 0                               ' Pone a cero el contador de líneas
    '***** Escribe la dirección del archivo.PLT
    *****
        Label2.Caption = "Ubicación PLT: " & Directorio ' muestra la ubicación
del archivo
        Largo = Len(CommonDialog1.FileName) ' Calcula el largo de la dirección
del PLT
        CommonDialog1.FileName = Left(CommonDialog1.FileName, Largo - 4) ' Le sacó
el punto y la extensión PLT
        Open CommonDialog1.FileName + ".CNC" For Output As #2 'abre nuevo
archivo para escritura

```

```

        For i = 0 To lstCodigo.ListCount      ' Graba el archivo.CNC y le
agrega el contenido
            Print #2, lstCodigo.List(i) 'carga el código de cada línea del
list
        Next
    Close #2                                'termina de escribir
    '''''' Escribe la dirección del archivo.CNC ''''''''''''''''''''''''''''''''
Label3.Caption = "Ubicación CNC: " & CommonDialog1.FileName + ".CNC" '
muestra la ubicación del archivo
Label4.Caption = "Tarea terminada. Archivo CNC guardado" ' Mensaje
solucion:
Exit Sub
End Sub
Private Sub cmdSalir_Click()
End
End Sub
Private Sub Form_Load()
Label5.Caption = "Convierte PLTs para trazados y perforaciones." + Chr(13)
+ Chr(13) + _
"Para volver a convertir, volver a abrir otro archivo." + Chr(13) + Chr(13)
+ _
"Convierte el PLT a CNC y lo graba con igual nombre pero .CNC."
End Sub

```

Codigo fuente Pograma POSICIONADOR.-

```

Option Explicit
Dim Xi As Integer, Yi As Integer, Zi As Integer

```

```

Dim T As Integer, TT As Integer
Dim plusY As Integer, plusX As Integer, plusZ As Integer
Dim cuentapasos As Long
Dim VelocDesp As Integer
Dim EjeX(8) As Byte, EjeY(8) As Byte, EjeZ(8) As Byte
Dim Puerto ' Número del Puerto seleccionado, LPT1 o
LPT2
Dim PararMovimiento As Boolean ' Para parar los movimientos
Private Sub B_PuestaCero_Click()
txtXmanual.Text = 0
    txtYmanual.Text = 0
    txtZmanual.Text = 0
End Sub
Private Sub cmdApagar_Click() 'apaga el puerto
Out Puerto, 0
    Out Puerto + 2, 11
End Sub
Private Sub cmdSalir_Click() ' Botón Salir del programa
    Out Puerto, 0 ' Pone a cero el puerto Datos
    Out Puerto + 2, 11 ' Pone a cero el puerto Control
    End ' Cierra el programa
End Sub
Private Sub cmdXY_Click(Index As Integer) ' Botonera de X - Y
    OpcionPasos ' Leer las opciones de cantidad de pasos a dar
    Select Case Index ' Selecciona de acuerdo al botón
pulsado
Case 0
plusY = 1
plusX = 0
    Case 1
plusY = 1
plusX = 1
    Case 2
plusY = 0
plusX = 1
    Case 3
plusY = -1
plusX = 1

```



```

    Case 4
plusY = -1
plusX = 0
    Case 5
plusY = -1
plusX = -1
    Case 6
plusY = 0
plusX = -1
    Case 7
plusY = 1
plusX = -1
    End Select
plusZ = 0
    Mover
End Sub
Private Sub cmdZ_Click(Index As Integer) ' Botón Subir - Bajar
    If Index = 0 Then ' Index = 0 = Bajar, Index = 1
= Subir
plusZ = 1 ' Baja
    Else
plusZ = -1 ' Sube
    End If
plusX = 0: plusY = 0 ' Los ejes x,y no se moverán
    OpcionPasos ' Leer las opciones de cantidad
de pasos a dar
    Mover ' llamada al subprocedimiento
Mover()
End Sub
Private Sub Command1_Click()
PararMovimiento = True
End Sub
Private Sub Form_Load() ' Comienza con el puerto en limpio
'Puerto = 888 ' LPT1
'Puerto = 60416 ' EC00 LPT1 en la nueva compu '59520 '48128
' LPT2 es tarjeta LPT agregada en mi máquina
PararMovimiento = False ' Parar los movimientos
Out Puerto, 0 ' Pone a cero los Datos

```

```

Out Puerto + 2, 11          ' Pone a cero los Control
hscVelocidad.Value = 2     ' Velocodades de avance
' Definición de todos los valores posibles del Puerto. Cargar las matrices
EjeX(0) = 1: EjeX(1) = 3: EjeX(2) = 2: EjeX(3) = 6
EjeX(4) = 4: EjeX(5) = 12: EjeX(6) = 8: EjeX(7) = 9
EjeY(0) = 16: EjeY(1) = 48: EjeY(2) = 32: EjeY(3) = 96
EjeY(4) = 64: EjeY(5) = 192: EjeY(6) = 128: EjeY(7) = 144
EjeZ(0) = 10: EjeZ(1) = 8: EjeZ(2) = 9: EjeZ(3) = 13
EjeZ(4) = 15: EjeZ(5) = 7: EjeZ(6) = 3: EjeZ(7) = 2
End Sub

Private Sub Mover()
If Option15 = True Then Puerto = 888
If Option16 = True Then Puerto = 60416 ' EC00 LPT1 en la nueva compu
OtraVez:
DoEvents
If cuentapasos = 0 Then GoTo salir ' Para evitar repetición de la
orden de mover
If PararMovimiento = True Then GoTo salir ' Para evitar repetición de la
orden de mover
    VelocDesp = (15 - hscVelocidad.Value) * 10 ' Actualiza la
velocidad
    'verifica que no exista desbordamiento de las matrices
If plusX > 0 And Xi = 7 Then Xi = -1
    If plusX < 0 And Xi = 0 Then Xi = 8
    If plusY > 0 And Yi = 7 Then Yi = -1
    If plusY < 0 And Yi = 0 Then Yi = 8
    If plusZ > 0 And Zi = 7 Then Zi = -1
    If plusZ < 0 And Zi = 0 Then Zi = 8
'incrementa la secuencia de pasos
    Xi = Xi + plusX
Yi = Yi + plusY
    Zi = Zi + plusZ

'mueve los motores
    Out Puerto, EjeX(Xi) + EjeY(Yi)
    Out Puerto + 2, EjeZ(Zi)

'inhabilita el movimiento de los ejes no seleccionados

```

```

If plusX = 0 And plusY = 0 Then Out Puerto, 0
    If plusZ = 0 Then Out Puerto + 2, 11

'actualiza los valores en los cuadros de texto
txtXmanual.Text = Val(txtXmanual.Text) + plusX
    txtYmanual.Text = Val(txtYmanual.Text) + plusY
    txtZmanual.Text = Val(txtZmanual.Text) + plusZ
cuentapasos = cuentapasos - 1      'va decrementando el contador
    If cuentapasos = 0 Then          'si llegó a cero, termina apagando
el puerto
Out Puerto, 0
    Out Puerto + 2, 11
    GoTo salir
End If
'retardo
For TT = 1 To VelocDesp
    For T = 1 To 10000: Next T
Next TT
GoTo OtraVez      'sigue con el desplazamiento
salir:
PararMovimiento = False
cuentapasos = 0      'Pone a cero el contador
Out Puerto, 0      'Apaga datos
Out Puerto + 2, 11      'Apaga control
End Sub

Private Sub hscVelocidad_Change()
lblVelocidad.Caption = hscVelocidad.Value ' Muestra la velocidad
seleccionada
End Sub

Private Sub OpcionPasos()
If cuentapasos = 0 Then      ' Para evitar repetición de la orden de
mover
If Option1 = True Then cuentapasos = 5      ' 0.052 mm
If Option2 = True Then cuentapasos = 48      ' 0.5 mm; 96 pasos es 1 mm.
If Option3 = True Then cuentapasos = 480      ' 5 mm
    If Option4 = True Then cuentapasos = 4800      ' 50 mm
If Option5 = True Then cuentapasos = 96      ' 1 mm para poner a cero la
profundidad inicial

```

End If
mover
End Sub

' Para evitar repetición de la orden de

