



BENEMÉRITA UNIVERSIDAD AUTONOMA DE PUEBLA

**FACULTAD DE CIENCIAS DE LA
ELECTRONICA.**

**PARA OBTENER EL GRADO DE
LICENCIATURA EN ELECTRONICA**

**“AMBIENTE GRAFICO E INTERFAZ
ELECTRONICA PARA CNC”**

**PRESENTA:
Constantino Cruz Molina**

**Asesor:
Dr. Antonio Michua Camarillo**

H. Puebla de Zaragoza, 7 de Enero de 2015.

Índice

Introducción	I
Justificación	II
Objetivos	III
 Capítulo I: MARCO TEORICO	1
1.1 Antecedentes	1
1.2 Bus en Serie Universal	3
1.3 Disposición de pines del USB	4
1.4 Tarjetas electrónicas de control	6
1.5 Sistemas mecánicos CNC	7
 Capítulo II: ANÁLISIS Y DISEÑO GRAFICO	8
2.1 Programa de dibujo y edición de código G	8
2.2 Programación orientada a objetos	11
2.3 OpenGL	15
2.3.1 Implementación del algoritmo para leer el código G.....	22
 Capítulo III: CODIGOS G	24
3.1 Algoritmos para el conjunto de códigos G	24
3.1.1 Preliminares matemáticos	24
3.1.2 Ecuación de la línea recta	24
3.1.3 Ecuación del círculo	25
3.1.4 Ecuación del círculo con vectores de dirección al centro	28
3.2 Códigos G implementados	29
3.2.1 Código G00	31
3.2.2 Código G01	32
3.2.3 Código G02	34
3.2.4 Código G03	35

Capítulo IV: DISEÑO DEL SISTEMA ELECTRONICO Y MECANICO	37
4.1 Circuito de control	37
4.2 Comunicación por puerto USB del microcontrolador	39
4.2.1 Programación del PIC18F2550 para la conexión USB	41
4.3 Programación del software de control	42
4.4 Etapa de potencia	43
4.5 Sistema mecánico	48
 Capítulo V: RESULTADOS Y CONCLUSIONES	 50
5.1 Resultados	50
5.2 Conclusiones	57
 Bibliografía	 58
Apéndice	59

Introducción

En la actualidad el concepto de automatización es perfectamente entendido, al menos de forma intuitiva, por la gran mayoría de las personas. Generalmente se entiende como la sustitución de obreros por maquinas en tareas más o menos especializadas, tales como pintar cortar, mover, soldar; incluso en tareas de gran complejidad como operaciones quirúrgicas.

En la industria la automatización es clave para la producción en serie de productos manufacturados ya sea en el ensamblaje de automóviles o en el proceso de ensamblado de tarjetas electrónicas, por citar algunas. Generalmente en el campo industrial el uso de Programadores Lógicos Programables (PLC) es muy común, sin embargo las empresas de manufactura utilizan control numérico por computadora (CNC) para diseñar y maquilar piezas que se requieren en alguna aplicación específica como elaboración de bujes en un automóvil, diseño de soporte para maquinas textiles, diseño de placas electrónicas, etc.

Descripción del sistema:

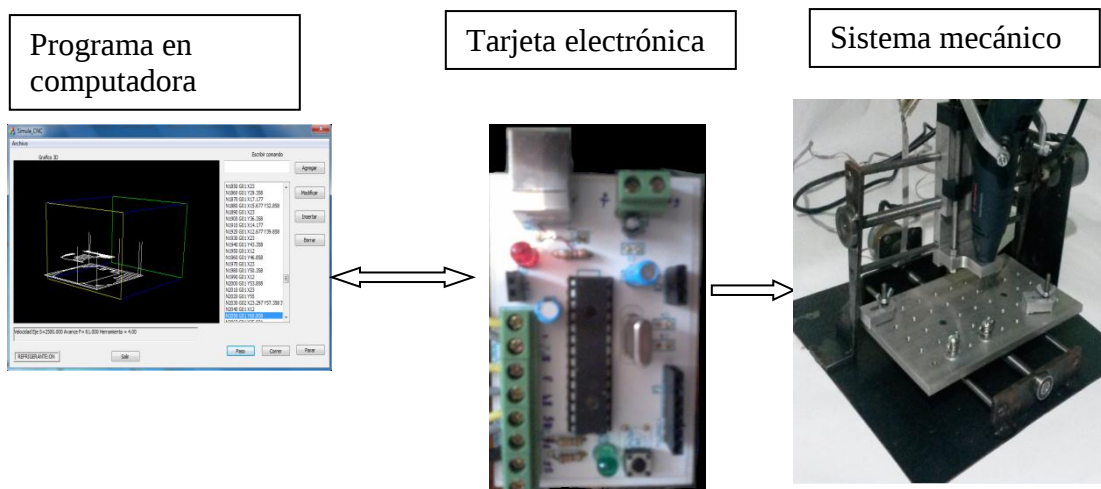


Figura A: Sistema CNC

Un sistema CNC comprende tres partes fundamentales como se muestra en la figura A que son la interfaz de usuario en la computadora en donde el programador escribe un código de uso industrial que representa un conjunto de operaciones que se ejecutaran para obtener una pieza mecánica. El programa se ejecuta secuencialmente y cada instrucción representa una acción en el sistema mecánico. Las instrucciones que se deben de ejecutar en el programa pasan a través de una interfaz electrónica la cual es la encargada de interpretar los comandos enviados por la computadora y traducirlos en movimiento de motores los cuales provocan un desplazamiento en los ejes coordenados x, y, z. Cada movimiento de los motores representa una acción de taladrado sobre una pieza, de perforación, de desbaste o de acabado.

Justificación

En la actualidad el avance tecnológico ha provocado que se puedan manufacturar más piezas mecánicas en menor tiempo, una de estas aplicaciones es el uso de CNC el cual es una máquina de uso general para el diseño de moldes, piezas y un variado conjunto de aplicaciones en la industria. Los sistemas industriales CNC son de alto costo y de difícil acceso para gente en general. Una alternativa para desarrollar piezas en baja escala y de alta calidad es diseñar un sistema CNC de tamaño reducido y prestaciones limitadas, pero que sea suficiente para una aplicación particular. Hoy en día se pueden encontrar varias aplicaciones que usan CNC's de tamaño reducido, la mayoría de estos sistemas usan el puerto paralelo de la computadora ya que es uno de los puertos más utilizados en máquinas de escritorio, sin embargo, cuando uno necesita mayor flexibilidad del sistema este tipo de interfaces resultan difíciles de mantener. Una mejora del puerto paralelo es el puerto USB de transmisión serial de alta velocidad. El puerto USB es muy adecuado cuando se requiere reducir espacio en la transmisión de información y además es muy flexible debido a su arquitectura, hoy en día la mayor parte de dispositivos móviles se conectan a través de puertos USB. En la propuesta actual se propone desarrollar una interfaz para controlar un sistema CNC de tamaño reducido usando el puerto USB. La aplicación del sistema es para la realización de circuitos impresos y maquinados de figuras en madera.

DESCRIPCION:

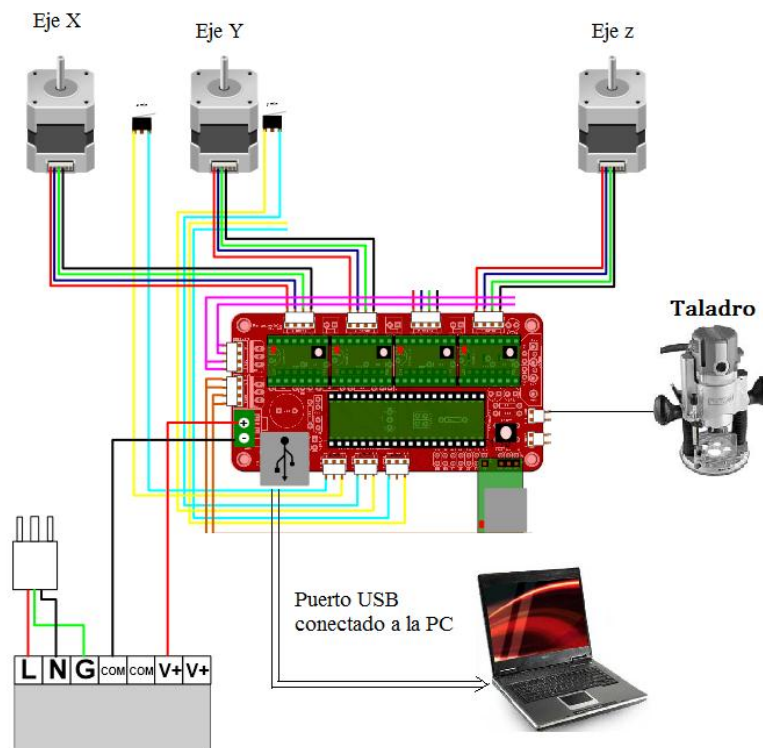


Figura B.- Diagrama general de la tarjeta electrónica

Como se muestra en la figura B la interfaz electrónica se conecta al puerto USB y un programa en la computadora le envía una serie de comandos que al llegar a la tarjeta electrónica se traducen a movimientos de tres motores de pasos. Los motores de pasos son los encargados de mover al sistema mecánico a una coordenada específica de acuerdo a un programa escrito con código G. La tarjeta electrónica traduce un comando enviado por la computadora vía el puerto USB a un movimiento físico en el sistema mecánico.

La tarjeta electrónica consta de un microcontrolador para procesar los datos, un conector USB y una etapa de potencia que sirve para mover motores a pasos. La etapa de potencia se tiene que aislar de la etapa de control debido al voltaje y corriente que se maneja, una sobre corriente de fuga que llegue al circuito de control puede provocarle daño o desprogramar al microcontrolador.

OBJETIVO:

Diseño de una tarjeta electrónica y control por computadora de la posición de un sistema mecánico de tres grados de libertad para hacer circuitos impresos.

Objetivos particulares:

- Investigar cuales son las características principales de los programas actuales en el mercado que se utilizan para maquinado de piezas.
- Diseñar un algoritmo para leer los códigos G00, G01, G02, G03 y M30 en un archivo de texto bajo el estándar EIA RS274D.
- Diseñar un programa en una computadora IBM PC para la simulación de códigos G y visualización en tres dimensiones haciendo uso de la librería grafica OpenGL.
- Investigar el protocolo de transmisión serial USB mediante el uso del microcontrolador PIC18F2550
- Programar el microcontrolador PIC18F2550 para enviar y recibir datos del puerto USB a un programa en computadora
- Implementar un algoritmo en una computadora IBM PC para mover tres motores a pasos para el sistema mecánico de tres grados de libertad.
- Diseño y construcción del sistema mecánico.
- Mostrar el funcionamiento del sistema de control por computadora en un producto terminado de circuito impreso

CAPITULO I: MARCO TEORICO

En este capítulo se presentan los antecedentes de los sistemas CNC además de establecer una tabla comparativa entre los software de escritorio usados en medianas y pequeñas empresas. Se muestran los precios comerciales del Mach 3, KCam y LinuxCNC, esto es importante ya que en el presente trabajo se desarrolla un programa que desempeña el mismo trabajo. También se presenta los diferentes tipos de conexiones USB y su disposición de pines.

1.1 Antecedentes

En los años 1940 y 1950 se diseñan las máquinas de control numérico, basadas en las máquinas existentes con motores modificados cuyos mandos se accionaban automáticamente siguiendo las instrucciones dadas en tarjetas perforadas. Estas máquinas iniciales se desarrollaron rápidamente con equipos analógicos y digitales ver figura 1.1. La miniaturización de los sistemas con microprocesadores ha llegado a ser utilizada en las máquinas herramienta, lo que dio lugar a la denominación control decimal numérico, control numérico por computadora, control numérico por computador o control numérico computarizado (CNC).

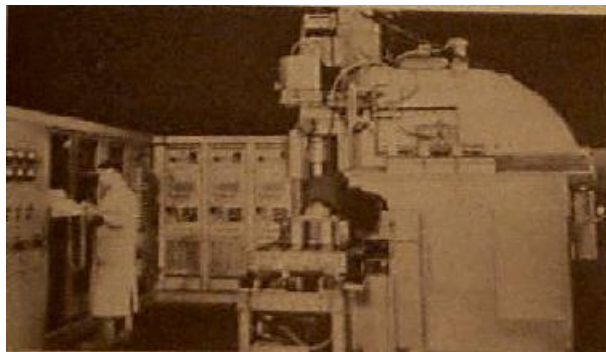


Figura 1.1: Maquina CNC de los años 50

Los precios de las maquinas CNC varían alrededor de 500 mil pesos y su precio puede ser superior o inferior esto depende del conjunto de herramientas que utiliza así como la marca modelo y capacidad de trabajo. Existen enrutadores CNC que vienen en diversas configuraciones, desde estilos pequeños hechos en casa o D.I.Y. (Do it yourself) "o de escritorio", fresadoras de mediana escala para la pequeña empresa, a enredadoras industriales grandes usadas para maquilar productos como, piezas de carpintería, piezas para aeroplanos y autos. Aunque hay muchas configuraciones, la mayoría de los CNC tienen pocas partes específicas: un controlador CNC dedicado, uno o más motores de ejes, servo motores, motores a pasos, servo amplificadores, inversores trifásicos, guías lineales, tuercas y una mesa de trabajo o mesas de trabajo. Además, las fresadoras CNC pueden tener bombas de vacío, con tableros de rejilla y ranuras para sostener partes en el lugar de corte.

La versión de código G más común en Estados Unidos es la propuesta por Electronic Industries Alliance a principio de los años 60 del siglo XX. En febrero de 1980 se aprobó una revisión final con el nombre de *RS274D*. En todo el mundo, en estándar ISO 6983 es el más usado, aunque en algunos países europeos en concreto se utilizan a veces otros estándares como el DIN 66025 alemán.

Los distintos fabricantes de máquinas CNC han añadido todo tipo de extensiones y variaciones al lenguaje, por lo que los operadores de las máquinas deben conocer las peculiaridades concretas que el fabricante ha previsto para su uso.

Existe una versión de G-code conocida como *BCL* que aunque es considerado un estándar se usa sólo en unas pocas máquinas.

El enruteador CNC es controlado por una computadora. Las coordenadas son cargadas en la máquina por un programa separado de diseño asistido por computadora (CAD). Los dueños de los CNC normalmente tienen dos aplicaciones de software, un programa hace los diseños CAD y otro programa traslada esos diseños en un programa de manufactura asistida por computadora (CAM) que usa instrucciones en código “G”, este programa es el que se comunica con la máquina para hacer la realización de la pieza diseñada. Las maquinas CNC pueden ser controladas directamente por programación manual y también tienen la opción de cargarle programas en código G para la construcción de algún diseño. Dentro de los programas CAM que se pueden usar se encuentra el Mach 3 de la empresa ArtSoft cuyo precio de la nueva versión del mes de Abril del 2014 es de 175 dolares o \$2,275 pesos aproximadamente, este software es muy versátil y tiene varios componentes para el maquilado además de asistentes ver figura 4, su uso se puede encontrar desde pequeñas aplicaciones en máquinas CNC caseras hasta el control de sistemas industriales.

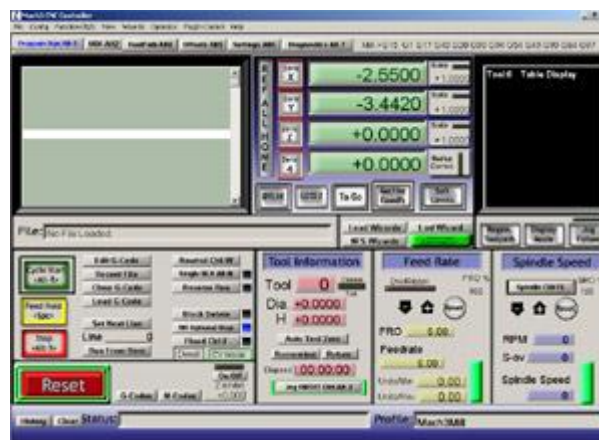


Figura 1.2: Programa Mach 3

Otro software CAM es el KCam de la empresa KellyWare su precio se encuentra en 95 dólares aproximadamente \$1,235 pesos, este software igual que el anterior tiene diversas gamas de aplicaciones la diferencia radica en los paquetes que maneja en el caso de KCam tiene menos utilidades que el Mach 3 sin embargo tienen la misma capacidad de control de una maquina CNC. Ver figura 1.3.

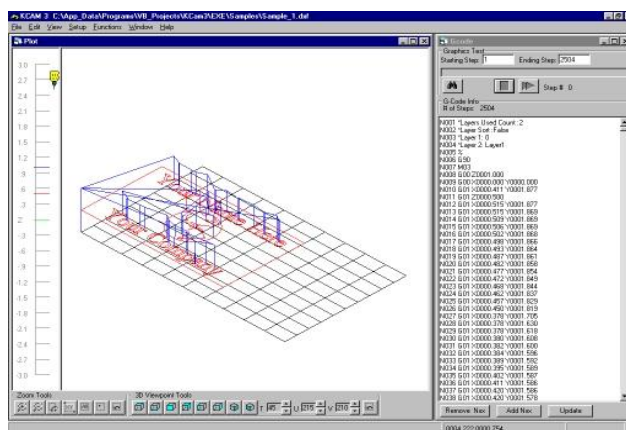


Figura 1.3: Programa KCam

En el presente trabajo se propone diseñar un programa que se pueda utilizar para interpretar los códigos G tal como lo hace el Mach 3 y el KCam con un conjunto mínimo de instrucciones que permita al usuario poder maquilar piezas diseñadas en un programa CAD. La tabla 1 muestra una comparativa de precios de los programas CAM y su interfaz utilizada.

Tabla 1: Lista de programas para la ejecución de código G

Programa CAM	Sistema Operativo	Puerto	Precios (Pesos Mexicanos) Licencia
Mach 3	Windows	Paralelo	2,275
KCam	Windows	Paralelo	1,235
CNC USBController	Windows	USB	1243.24
LinuxCNC	Linux	Paralelo	Free

1.2 Bus en Serie Universal (USB)

Una de las partes que componen una maquina CNC es la parte de la interfaz electrónica que se utiliza para controlar la CNC, el puerto que utiliza la tarjeta determina el protocolo de comunicación usado para el control del CNC, la mayoría de los proyectos para escritorio o hechos en casa utilizan el puerto paralelo, sin embargo, debido a la tendencia tecnológica se ha ampliado y popularizado el uso del puerto USB. Al igual que la tecnología de las comunicaciones entre otras; se modernizan los equipos y se deben mejorar los sistemas

electrónicos. En años atrás solo se tenían celulares que mandaban mensajes de texto y podía hacerse llamadas y estos fueron sustituidos por nuevos y modernos sistemas que pueden grabar video, audio y tienen la capacidad de conexión a sistemas de telefonía lo mismo sucede con la transmisión de datos por lo cual se ha elegido el puerto usb para el envío y recepción de información.

El Universal Serial Bus (USB) (bus universal en serie) es un estándar industrial desarrollado a mediados de los años 1990 que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores y periféricos y dispositivos electrónicos. La iniciativa del desarrollo partió de Intel que creó el USB Implementers Forum junto con IBM, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC en 1996 se lanzó la primera especificación (USB 1.0), la cual no fue popular, hasta 1998 con (USB 1.1). Actualmente agrupa a más de 685 compañías.

Su campo de aplicación se extiende en la actualidad a cualquier dispositivo electrónico o con componentes, desde los automóviles (las radios de automóvil modernas van convirtiéndose en reproductores multimedia con conector USB o iPod) a los reproductores de Blu-ray Disc o los modernos juguetes. Se han implementado variaciones para su uso industrial e incluso militar.

1.3 Disposición de Pines del USB

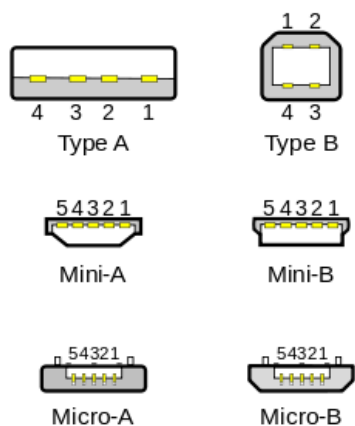


Figura 1.4: Conectores tipo A (izquierda) y B (derecha)

Pin 1	V _{CC} (+5 V)
Pin 2	Data-
Pin 3	Data+
Pin 4	Masa

Ediciones previas:

El estándar USB evolucionó a través de varias versiones antes de su lanzamiento oficial en 1996:

USB 0.7: Lanzado en noviembre de 1994.

USB 0.8: Lanzado en diciembre de 1994.

USB 0.9: Lanzado en abril de 1995.

USB 0.99: Lanzado en agosto de 1995.

USB 1.0 Release Candidate: Lanzado en noviembre de 1995.

Miniplug/Microplug

Pin	Nombre	Color	Descripción
1	VCC	Rojo	+5 V
2	D-	Blanco	Data -
3	D+	Verde	Data +
4	ID	Ninguno	Permite la distinción de Micro-A y Micro-B Tipo A: conectado a tierra Tipo B: no conectado
5	GND	Negro	Tierra y retorno o negativo

Los dispositivos USB se clasifican en cuatro tipos según su velocidad de transferencia de datos:

Baja velocidad (1.0): Tasa de transferencia de hasta 1,5 Mbit/s (188 kB/s). Utilizado en su mayor parte por dispositivos de interfaz humana (Human Interface Device, en inglés) como los teclados, los ratones (mouse), las cámaras web, etc.

Velocidad completa (1.1): Tasa de transferencia de hasta 12 Mbit/s (1,5 MB/s) según este estándar, pero se dice en fuentes independientes que habría que realizar nuevamente las mediciones. Ésta fue la más rápida antes de la especificación USB 2.0. Estos dispositivos

dividen el ancho de banda de la conexión USB entre ellos, basados en un algoritmo de impedancias LIFO.

Alta velocidad (2.0): Tasa de transferencia de hasta 480 Mbit/s (60 MB/s) pero con una tasa real práctica máxima de 280 Mbit/s (35 MB/s). El cable USB 2.0 dispone de cuatro líneas, un par para datos, y otro par de alimentación. Casi todos los dispositivos fabricados en la actualidad trabajan a esta velocidad

Superalta velocidad (3.0): Tiene una tasa de transferencia de hasta 4,8 Gbit/s (600 MB/s). La velocidad del bus es diez veces más rápida que la del USB 2.0, debido a que han incluido 5 contactos adicionales, desechando el conector de fibra óptica propuesto inicialmente, y será compatible con los estándares anteriores. En octubre de 2009 la compañía taiwanesa ASUS lanzó la primera placa base que incluía puertos USB 3.0, tras ella muchas otras le han seguido y actualmente se ve cada vez más en placas base y portátiles nuevos, conviviendo junto con el USB 2.0

El sistema de electrónico propuesto utiliza el USB versión 1.1 y un conector tipo B hembra.

1.4 Tarjetas electrónicas de control

En un sistema CNC la tarjeta de control es la encargada de aplicar energía a los motores para obtener el movimiento deseado dada una coordenada enviada desde la computadora, existen varias formas de lograr el control de la maquina las más habituales implementan en la misma tarjeta electrónica la etapa de control y la etapa de potencia para mover los actuadores en un segundo esquema se tiene la tarjeta de control separada de la etapa de potencia sin embargo trabajan en conjunto para lograr mover los actuadores . En el mercado hay varias opciones de tarjetas útiles que se pueden utilizar para el propósito del trabajo desarrollado en la tesis, en la tabla 2 se muestra el costo de la tarjeta y el motor utilizado para el control del CNC.

Tabla 2: Precios de tarjetas de control para un CNC

Tarjeta con etapa de potencia integrada	Motores utilizados	Puerto utilizado	Precio (Pesos Mexicanos)
Controladora 4 ejes	Bipolares	Paralelo	\$920
Controladora 3 Ejes	Bipolares	Paralelo	\$780
CNC USB controller Mk2/4 (4 Ejes)	Bipolares	USB	\$1786.98

Los precios de la tabla 2 son para las tarjetas de aplicación de escritorio o de pequeña escala. Los precios para tarjetas industriales son más elevados.

En la tabla 2 se muestran tarjetas donde se tiene la etapa de control junto con la etapa de potencia. Las tarjetas que vienen separadas se venden por módulos y depende del número de ejes a controlar es el número de módulos que se compran. El precio promedio de un módulo para el control de un solo motor de las mismas características que la mostrada en la tabla 2 es de \$310, así que se deben comprar 3 tarjetas para poder mover un CNC, además de la tarjeta de control en todas estas opciones se utilizan motores bipolares. Sin embargo también se pueden utilizar motores unipolares para los cuales el diseño utilizado para su control y etapa de potencia es mucho más simple y económico. En el trabajo actual se propone el diseño de la etapa de control y la etapa de potencia usando motores unipolares.

1.5 Sistemas mecánicos CNC

La parte mecánica para un sistema CNC es muy importante debido a que esta debe ser lo más robusta posible. La selección de materiales para su construcción y su ensamblado aseguran un buen funcionamiento del sistema. La mayor parte de sistemas de escritorio se construyen usando metales como el aluminio, acero etc. Sin embargo existen prototipos diseñados en madera. Los sistemas CNC contruidos en madera tienen la desventaja de cambiar su espesor de la pieza de madera cuando existe humedad y normalmente corren el riesgo de dañarse. Por lo cual el prototipo presentado en el presente trabajo se propuso inicialmente con una estructura con partes de metal y de madera. La siguiente tabla muestra los precios de sistemas mecánicos similares al construido en la tesis.

Tabla 2b: Precios de sistemas mecánicos CNC

Empresa fabricante	Área de trabajo	Precio (Pesos Mexicanos)
Minifab	300mm x 300mm x 50mm	\$18,000
Independiente	400mmx580mmx60mm	\$76,500
CNC31	600 X 900 x 150mm.	\$98,500

En la tabla 2b se muestran algunos precios de sistemas CNC comerciales. La variación de los precios depende principalmente del área de trabajo y el material de construcción, en los tres casos se usa aluminio, pero el tamaño varia. Otro punto a considerar es el programa empleado para realizar el fresado en el primer caso se usa código abierto y el programa Open CNC. En los otros dos casos se utiliza el Mach 3.

CAPITULO II: ANÁLISIS Y DISEÑO GRAFICO

La organización del capítulo 2 se inicia con el diseño del programa CAM llamado Simula CNC. Se hace uso del compilador Visual C++ 2010 y la librería grafica OpenGL. La programación está basada en objetos, se muestran los algoritmos utilizados para la programación del ambiente gráfico.

2.1 PROGRAMA DE DIBUJO Y EJECUCION DE CODIGO G

La interfaz gráfica consta de dos partes; la primera parte es la simulación de códigos G y aspecto que tienen el programa, en esta parte se editan los códigos G que se han de utilizar para que posteriormente los ejecute en la máquina. La segunda consta de la relación de los códigos G con la ejecución de los códigos en la parte mecánica, es decir la programación que se utiliza para comunicarse con el hardware. A continuación se muestra un diagrama esquemático de la parte de programación.

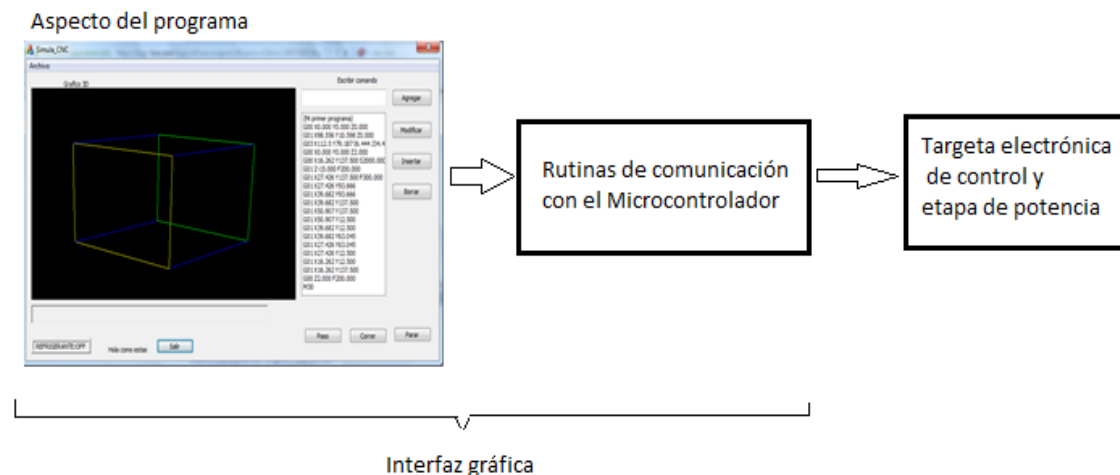


Figura 2.1: Interfaz grafica

Visual C++ es un lenguaje de programación de alto nivel, está especialmente diseñado para el desarrollo y depuración de código escrito para las API's de Microsoft Windows, DirectX y la tecnología Microsoft .NET Framework. Existen herramientas en Visual C++ para construir una aplicación grafica como la clase, atributos, métodos, mensajes y eventos.

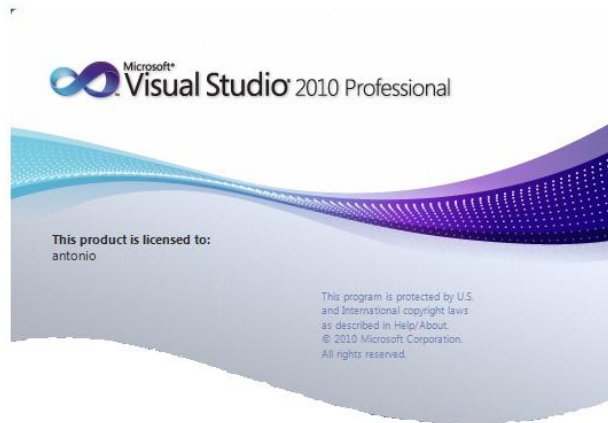


Figura 2.2: Plataforma de programación Visual C++

Visual C++ cuenta con tres tipos de proyectos que se puede generar desde el MFC los cuales son documento basado en dialogo, documento simple y múltiple documento. En el trabajo se eligió la opción de dialogo, la ventana para elegir esta opción aparece en un cuadro gris, en esta opción se deben de añadir menús, botones y cualquier objeto necesario para la interfaz, este tipo de aplicaciones son muy usadas donde se necesite presentación de datos y despliegue de información de tipo gráfica. La aplicaciones de documento y múltiple documento son parecidas a un editor de texto como el Microsoft Word. La aplicaciones de simple y múltiple documento incluye un menú principal donde aparecen las opciones abrir o cerrar documento y además menús desplegables de ayuda y edición.

Es importante señalar que el asistente de Visual C++ solo construye la aplicación gráfica y el programador es el encargado de implementar los algoritmos que activen la funcionalidad de cada elemento grafico incluido al hacer click o elegir alguna opción. En la tabla 3 se muestran el conjunto de objetos gráficos utilizados.

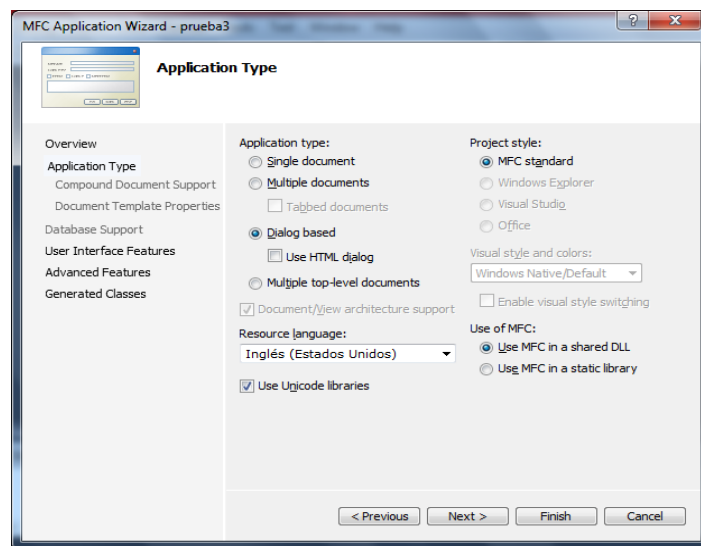


Figura 2.3: Ventana para la creación del proyecto

Tabla 3: Objetos gráficos utilizados en la programación

Objeto	Propiedad	Valor
Picture Control	Caption	IDC_VENTANA_3D
Edit Control	Edit	IDC_EDIT_COMANDO
ListBox Control	List	IDC_LISTA_DATOS
Button	Agregar	IDC_BUTTON_AGREGAR
Button	Modificar	IDC_BUTTON_MODIFICAR
Button	Insertar	IDC_BUTTON_INSERTAR
Button	Borrar	IDC_BUTTON_BORRAR
Button	Parar	IDC_BUTTON_PARAR
Button	Correr	IDC_BUTTON_CORRER
Button	Paso	IDC_BUTTON_PASO
Button	Salir	IDC_BUTTON_SALIR
Static Text	Caption	IDC_STATIC_MOSTAR4
Static Text	Caption	IDC_STATIC_COOLANT5
Menu	Caption	IDR_Menu

En la tabla 3 encontramos 3 campos el primero denominados “Objeto” se refiere al recurso grafico utilizado en el programa y localizado en la barra de herramientas de visual C++. Estos objetos son incluidos en el programa y se utilizan para darle forma al cuadro de Dialogo. El segundo campo denominado “Propiedad” se refiere a la etiqueta del objeto o la función que realiza en el programa, las propiedades de los objetos son distintas y dependen de la función que realizan. El tercer campo denominado “Valor “ se refiere al nombre que internamente usa visual C++ para poder identificar un objeto de otro y este nombre es distinto aunque el objeto es el mismo, por ejemplo se utilizan botones de pulsación los cuales en el programa se identifican con distintos nombres.

Finalmente tenemos incluido un recurso de menú, los menús tienen su propia hoja de recurso que permite incluir opciones como las de Abrir, Guardar, Nuevo, etc. En la figura 2.4 se muestra el menú utilizado. La figura 2.5 muestra el aspecto del programa.

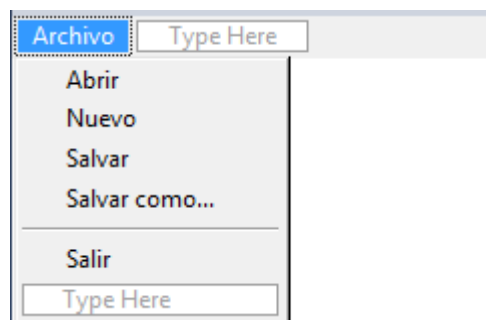


Figura 2.4: Menú para el programa Simula CNC

La apariencia grafica queda de la siguiente forma

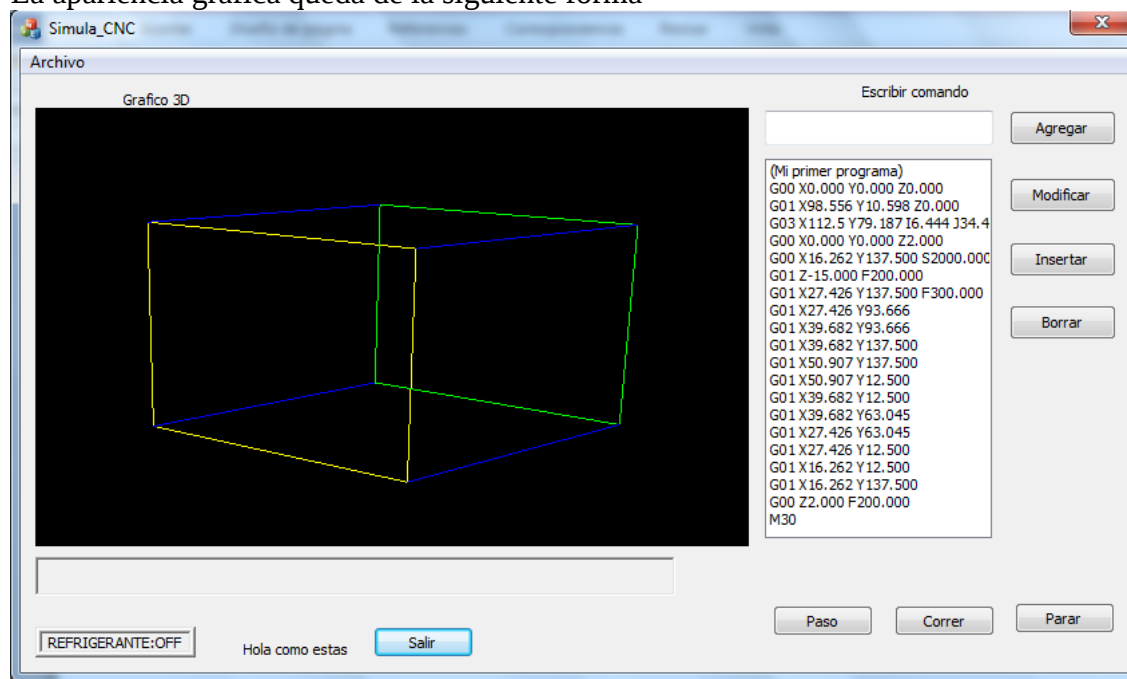


Figura 2.5: Aspecto del programa Simula CNC

2.2 Programación orientada a objetos

La *programación orientada a objetos POO* (oriented objects programming) es una evolución de los métodos previos de programación. Este tipo de programación es más estructurada y proporciona mecanismos para la abstracción y ocultación de datos, lo que da paso a la creación de código más reutilizable y con mayor facilidad para su evolución en el tiempo.

La programación orientada a objetos es un método de implementación en el que los programas se organizan como colecciones cooperativas de objetos, cada uno de los cuales representa una instancia de alguna clase y cuyas clases son, todas ellas, miembros de una jerarquía de clases unidas mediante relaciones de herencia.

Existen tres puntos importantes en esta definición:

1. En la programación orientada a objetos los bloques lógicos de construcción fundamentales son los objetos, no los algoritmos.
2. Cada objeto es una instancia de alguna clase.
3. Las relaciones entre las clases se dan por medio de relaciones de herencia.

Para realizar programación orientada a objetos es necesario emplear lenguajes de programación que apoyen este tipo de programación, estos lenguajes deben disponer de tres propiedades fundamentales: *encapsulación*, *herencia* y *polimorfismo*, dado que si un lenguaje carece de estas no apoya la programación orientada a objetos. El empleo de nuevos elementos sintácticos y sistemáticos en lenguajes de programación no es la barrera principal para crear software con esta tecnología, sino más bien la adopción de una nueva forma de pensar y diseñar los programas.

Para comprender de mejor forma la programación orientada a objetos, es necesario explicar algunos conceptos del modelado de objetos.

Abstracción: es un mecanismo utilizado para comprender y resolver un problema complejo, desde la perspectiva del observador, en entidades (objetos) bien definidas por medio de la definición de sus datos y operaciones: características esenciales que lo distinguen de todas las demás.

Al realizar una buena abstracción se resalta los detalles importantes y se omiten las que no lo tienen o que puedan provocar distracción en el observador.

Encapsulamiento: es el proceso de almacenar en un mismo contenedor los elementos de una abstracción que constituyen su estructura y su comportamiento; ocultando al mismo tiempo sus operaciones (información precisa e irrelevante para otros).

La abstracción y el encapsulamiento son conceptos complementarios, pues mientras la abstracción se centra en el comportamiento observable del objeto, el encapsulamiento se centra en la implementación que da lugar a este comportamiento.

La ocultación de información es utilizada frecuentemente, donde los elementos de un objetos que no contribuyen a sus características primordiales se ocultan.

El encapsulamiento permite reducir el potencial de errores, dado que los objetos dependen unos de otros de diversas formas. Es posible entender el encapsulamiento como: introducir un objeto dentro de una caja negra donde existe una entrada y una salida, pues al introducir datos en la entrada se obtendría un resultado en la salida, sin la necesidad de conocer el funcionamiento de la caja.

Modularidad: es la propiedad que tiene un sistema que ha sido descompuesto en un conjunto de módulos cohesivos y débilmente acoplados.

Es decir, un sistema puede ser fragmentado en módulos, donde cada módulo contiene sus propios datos y operaciones.

El *acoplamiento* se refiere a la dependencia que un sistema pueda tener.

Es decir, si el módulo A depende del módulo B, cualquier modificación en A implica modificaciones en B. Así que si un sistema tiene muchas dependencias tiene fuerte acoplamiento, y al minimizar las dependencias se dice que tiene débil acoplamiento.

Cohesión, se dice que un sistema tiene una fuerte cohesión cuando se tiene una agrupación de abstracciones que guardan cierta relación lógica.

Jerarquía: un conjunto de abstracciones forman una jerarquía y es posible emplear la identificación de estas jerarquías en el diseño, para el entendimiento de la complejidad.

La jerarquía es una clasificación de abstracciones.

Existen bloques básicos de construcción básicos en la programación orientada a objetos, por lo que a continuación se da una definición de estos.

Clase: es el mecanismo que se emplea para crear objetos, una clase define un conjunto de objetos que comparten una estructura común y un comportamiento común.

Es decir, la clase es la implementación de un objeto, define atributos (propiedades) y métodos (comportamiento) que implementan la estructura de datos y operaciones del objeto, respectivamente.

La *interfaz* de una clase proporciona su visión externa y por lo tanto enfatiza la abstracción a la vez que oculta su estructura y los secretos de su comportamiento. La interfaz de una clase se puede dividir en:

Pública (Public): Este tipo de declaración permite que el elemento declarado sea accesible a todos los objetos y funciones del programa.

Protegida (Protected): Una declaración en la que el elemento es accesible sólo a la propia clase, sus subclases y sus clases amigas.

Privada (Private): Una declaración accesible sólo a su propia clase y sus clases amigas.

Objeto: es un componente de software que encapsula un estado (propiedades) y un comportamiento (métodos), es decir un objeto es la instancia de una clase y tiene identidad.

El *estado* de un objeto cambia de acuerdo a los métodos que le son aplicados, este abarca todas las propiedades (estáticas) del mismo más los valores actuales (dinámicos) de cada una de sus propiedades. Esto es conocido como el comportamiento del objeto, puesto que ningún objeto trabaja de forma aislada, los objetos reciben acciones y actúan sobre otros objetos.

El comportamiento de un objeto es definido por un conjunto de métodos que le pueden ser aplicados, es decir, el comportamiento es la acción que realiza un objeto cuando pasa un mensaje o en respuesta a un cambio de estado

Método: es un algoritmo asociado a un objeto, cuya ejecución se desencadena tras la recepción de un mensaje, desde el punto de vista del comportamiento, es lo que el objeto puede hacer.

Las *propiedades* ó *atributos* son las características distintivas de una clase y describen un rango de valores. Son contenedores de un tipo de dato asociado a un objeto cuyo valor puede ser alterado por la ejecución de algún método.

Un *método* puede producir un cambio en las propiedades del objeto, ó la generación de un *evento* con un nuevo mensaje para otro objeto del sistema.

Evento: es un suceso en el sistema (una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente.

Mensaje: es la comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.

La *identidad* es la propiedad de un objeto que lo distingue de todos los demás objetos. A continuación se muestra la interacción del objeto con los métodos, propiedades y mensajes.

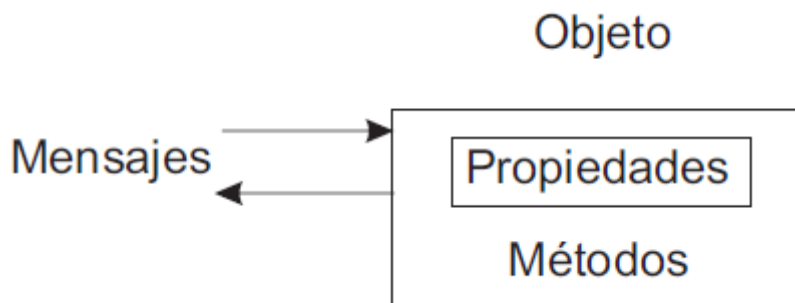


Figura 2.6: Objeto e interfaz de programación

Anteriormente mencionamos las propiedades fundamentales que apoyan la programación orientada a objetos: *Encapsulación*, *Herencia* y *Polimorfismo*, dado que ya se ha definido. Encapsulación, definiremos los dos términos restantes.

Herencia: es una relación entre clases que permite crear estructuras jerárquicas de clases, donde es posible la definición de subclases que incluyan nuevas propiedades o las modifiquen o las inhabiliten. Estas subclases pueden compartir su comportamiento sin tener que re-implementarlo. La herencia permite tomar una clase existente como base y definir una nueva clase, la clase nueva hereda todos los atributos y comportamientos presentes en la clase existente. La clase de la que otras heredan se le llama *superclase*, *clase base* o *clase madre*, análogamente las que heredan de otra u otras clases se conocen como: *subclase*, *clase derivada* ó *clase hija*.

Existen formas de heredar, conocidas como: *herencia simple*, en la que una clase comparte la estructura y/o comportamiento definidos con una sola clase.

Herencia múltiple: cuando una subclase se deriva de más de una clase.

Un término que se confunde típicamente con la herencia es la composición, esta consiste en la incrustación de una clase dentro de otra. Típicamente se confunden dado que ambos se

apoyan en otras clases para la construcción de nuevas clases. Pero enfatizamos que la *composición* significa que una clase se implementa re-utilizando variables miembro que contienen instancias de otras clases. La composición es una forma de reutilización, donde se construyen nuevas clases a partir de las existentes pero en lugar de heredar miembros, la nueva clase contiene a la otra clase íntegramente.

Polimorfismo: En los lenguajes de programación orientada a objetos se pueden tener métodos con comportamientos diferentes y asociados a objetos distintos, compartiendo el mismo nombre, al llamarlos por ese nombre se utiliza el comportamiento correspondiente al objeto que se está usando, es decir, las referencias y los objetos pueden contener objetos de diferentes tipos y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado.

Existen dos tipos de operaciones habituales que representan la infraestructura necesaria para la creación y destrucción de objetos, éstas son:

Constructor: es el método que se emplea para crear un objeto y principalmente para inicializar su estado en el momento de su definición. Tiene el mismo nombre de la clase y es llamado implícitamente cuando definimos objetos de sus clases, no regresa ningún valor y al igual que otros métodos puede llevar argumentos.

El constructor nos ayuda a crear e inicializar un objeto, sin embargo, al salir del ámbito de definición del objeto, debemos asegurar que la memoria asignada al objeto sea liberada, por lo que empleamos un método especial llamado *destructor*. Destructor es llamado una vez por cada objeto en el momento de su destrucción, la destrucción del objeto tiene lugar cuando el objeto abandona su ámbito de definición o es explícitamente destruido, cuando asignamos memoria dinámica a un objeto y lo liberamos cuando ya no es necesario.

2.3 OpenGL

OpenGL es un motor gráfico cuyas rutinas están integradas en la tarjeta gráfica, posee todas las características necesarias para la representación de escenas 3D modeladas con polígonos, desde el pintado más básico de triángulos, hasta el mapeado de texturas, transparencias, efectos de iluminación, etc.

Fue desarrollada por Silicon Graphics Inc. en 1992. Su nombre viene del inglés Open Graphics Library, cuya traducción es librería de gráficos libre. OpenGL es una especificación, es decir, un documento que describe un conjunto de funciones y su comportamiento exacto. A partir de estas especificaciones se crean implementaciones (bibliotecas de funciones creadas para enlazar con las funciones de la especificación OpenGL). La compatibilidad de OpenGL abarca desde Borland C, Visual C++, Java, Visual Fortran entre otros.

OpenGL es una especificación estándar que define una Interfaz de Programación de Aplicaciones (Application Programming Interface) multi-lenguaje y multi-plataforma para escribir aplicaciones que produzcan gráficos 2D y 3D. Se trata del conjunto de llamadas al

sistema que ofrecen acceso a los servicios del sistema desde los procesos y representa un método para conseguir abstracción en la programación, generalmente entre los niveles o capas inferiores y los superiores del software.

Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o iconos en la pantalla. De esta forma, los programadores se benefician las ventanas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio. Las API's asimismo son abstractas, el software que proporciona una cierta API generalmente es llamado la implementación de esa API.

El dibujo en OpenGL se basa en la composición de pequeños elementos, con los que vamos construyendo la escena deseada, estos elementos se llaman *primitivas*. Todas las primitivas de OpenGL son objetos de una o dos dimensiones, abarcando desde puntos simples a líneas o polígonos complejos.

Para definir el área de trabajo en OpenGL hay que imaginar un volumen con perspectiva ortonormal, como se puede observar en la figura 2.7.

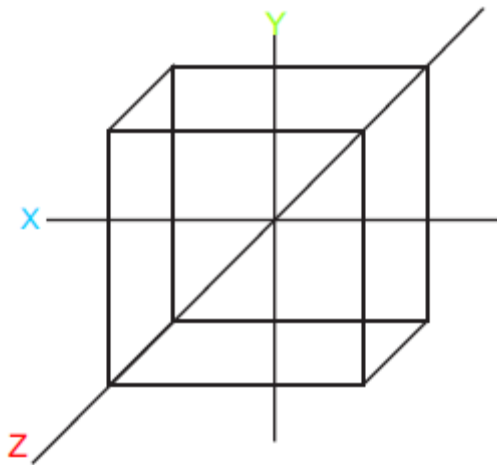


Figura 2.7: Sistema de referencia de Open GL

Es importante ubicar el sólido, una vez hecho esto, se verifica el origen del objeto y se procede a crear un nuevo proyecto en Visual C++, tomando en cuenta que debe estar incluida la librería *glut.dll* en la carpeta de *system* y *system32* de Windows; posteriormente en el proyecto se incorporan las librerías de acuerdo al diagrama a bloque de la figura 2.8.

Las librerías que se agregan al proyecto son: la *GLU* que contiene funciones gráficas de más alto nivel, que permiten realizar operaciones complejas; y la *GLUT* que es un paquete auxiliar para construir aplicaciones de ventanas, además de incluir algunas primitivas geométricas auxiliares.

GLU(Graphics Library Utility). Librería que contiene código para la Creación de objetos de uso repetitivo. Contiene funciones gráficas de más alto nivel, que permiten realizar operaciones complejas.

GLUT(GL Utility Toolkit). Librería que proporciona la mínima funcionalidad esperada de cualquier sistema moderno basado en ventanas.

Es un paquete auxiliar para construir aplicaciones de ventanas, además de incluir algunas primitivas geométricas auxiliares.

GLX Librería empleada por GLUT, da soporte a máquinas que utilizan Windows System, para iniciar una ventana, etc. WGL es el equivalente para sistemas Microsoft.

La interacción de OpenGL con Visual C++ se describe en el siguiente diagrama, figura 14.

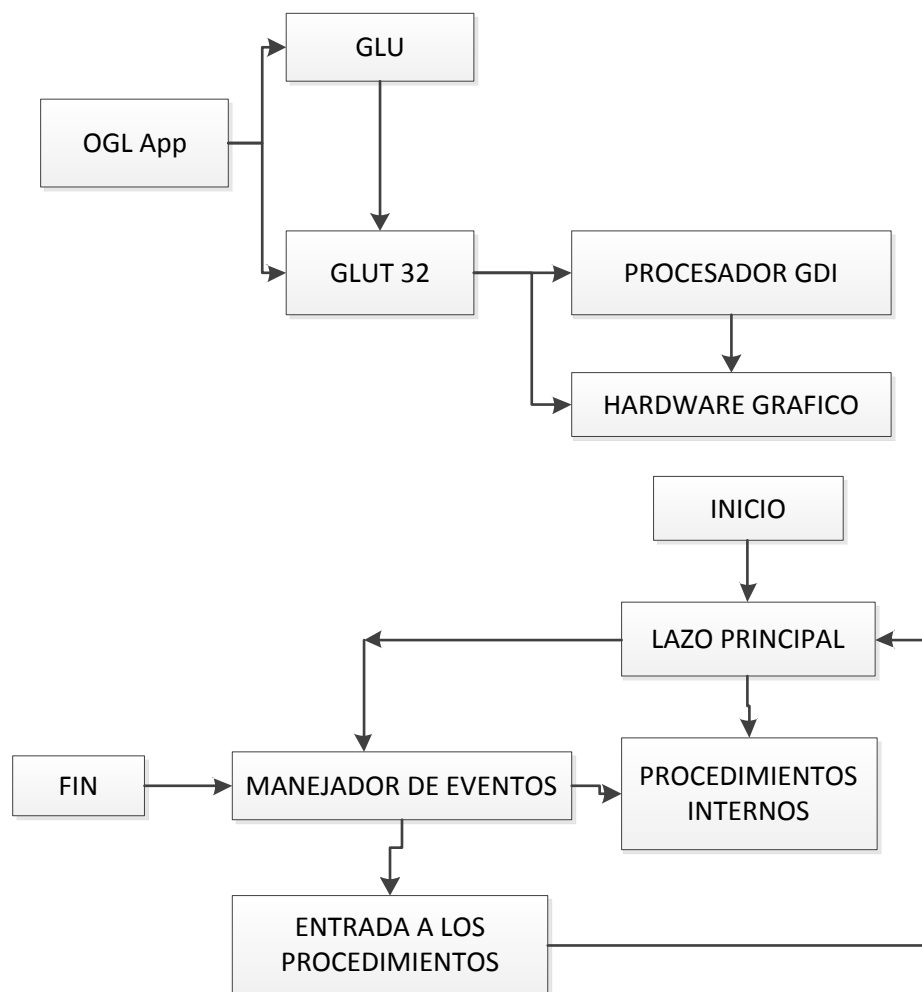


Figura 2.8: Diagrama de bloques del manejador Open GL

A continuación se enumeran algunas operaciones importantes, para inicializar el modo de visualización, estas operaciones se listan en el siguiente diagrama:

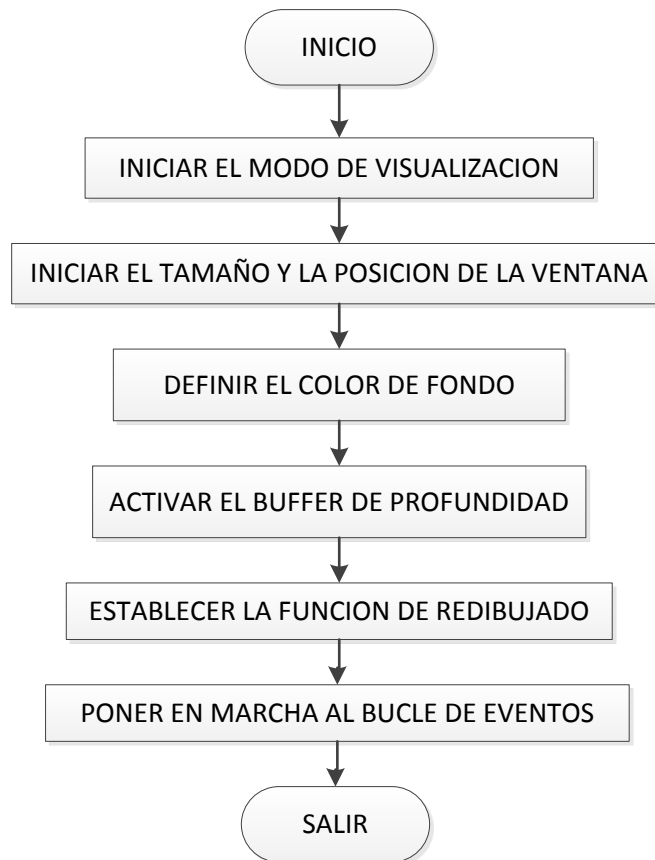


Figura 2.9: Diagrama para inicializar los gráficos en OpenGL

Existen algunas funciones básicas que se emplean en la programación del simulador, en la tabla 4 se muestran algunas de las más importantes.

Tabla 4: Funciones gráficas para OpenGL

Función	Descripción
<code>glutInit(argc,argv);</code>	Esta función inicia la GLUT, los parámetros <code>argc</code> y <code>argv</code> son iguales que en el <i>main</i> .
<code>glutInitDisplayMode</code> (GLUT SINGLE) (GLUT RGB);	Define el modo de creación de la ventana GLUT_SINGLE indica el uso de un buffer y GLUT_RGB el tipo de modelo de color.
<code>glutInitWindowPosition(0,0);</code>	Indica la posición x, y en la parte superior de la ventana.
<code>glutInitWindowSize(0,0);</code>	Indica el ancho y alto de la ventana
<code>glutCreateWindow(nombre);</code>	Función que crea la ventana y da un nombre
<code>glutMainLoop();</code>	Esta función cede el control del flujo del programa a la GLUT
<code>glClearColor(0,0,0,0);</code>	Define el color con el que se borrará el

	buffer. Los primeros 3 componentes son tonos <i>RGB</i> .
<code>glColor3f(0,0,0);</code>	Define el color actual con el que se dibuja. Los 3 componentes son tonos <i>RGB</i> .
<code>glViewport(0, 0, width, height)</code>	Define el espacio de ventana donde OGL dibuja. Los primeros dos parámetros definen la esquina superior izquierda de la ventana, <i>width</i> y <i>height</i> representan el ancho y alto.
<code>glLightfv</code>	Define una fuente de luz. Es posible definir fuentes de diversos colores.
<code>glVertex3f(10.0f, 5.0f, 3.0f)</code>	Define puntos, líneas y polígonos.

Los vértices (puntos 3D) son el denominador común en cualquiera de las primitivas OpenGL. Con ellos se definen puntos, líneas y polígonos, `glVertex` es la función que define vertices y puede tomar de dos a cuatro parámetros de cualquier tipo numérico.

Lista Display: Todos los datos que describen pixeles o geometría son almacenados aquí. La lista Display realiza procesamiento inmediato de datos, también conocido como modo inmediato.

Evaluador: Todas las primitivas geométricas son eventualmente descritas por vértices, provee un método derivado de vértices usado para representar las caras de control de puntos.

Operaciones Per-Vertex: Convierte los vértices dentro de las primitivas. Algunos datos de los vértices son transformados por matrices de punto flotante de 4 x 4.

Ensamble de primitivas: Es la eliminación de porciones de geometría donde está fuera del espacio definido por un plano.

Operaciones de Pixel: Cuando los datos geométricos toman un patrón mediante la renderización del pipeline de OpenGL, los datos del pixel toman una ruta diferente.

Ensamble de textura: Una aplicación OpenGL puede aplicar texturas a la imagen dentro de la geometría del objeto para hacerlo parecer más realista.

Rasterización: Es la conversión tanto de los datos de pixel y geometría dentro de fragmentos. Cada fragmento corresponde a un pixel en el framebuffer.

Operaciones de fragmentos Dentro del framebuffer los valores son actualizados, una serie de operaciones se realizan, las cuales pueden ser habilitadas o deshabilitadas.

OpenGL soporta dos clases de primitivas: Primitivas geométricas. Incluye puntos, segmentos de línea polígonos, curvas y superficies. Estas primitivas pueden ser manipuladas por operaciones como la traslación y rotación.

Primitivas de trama. Se emplean para añadir texturas y transparencias. Se pueden representar por medio de arreglo de pixeles, estas no pueden ser manejadas como las primitivas geométricas.

OpenGL puede manipular imágenes en tres dimensiones mediante las siguientes operaciones matriciales.

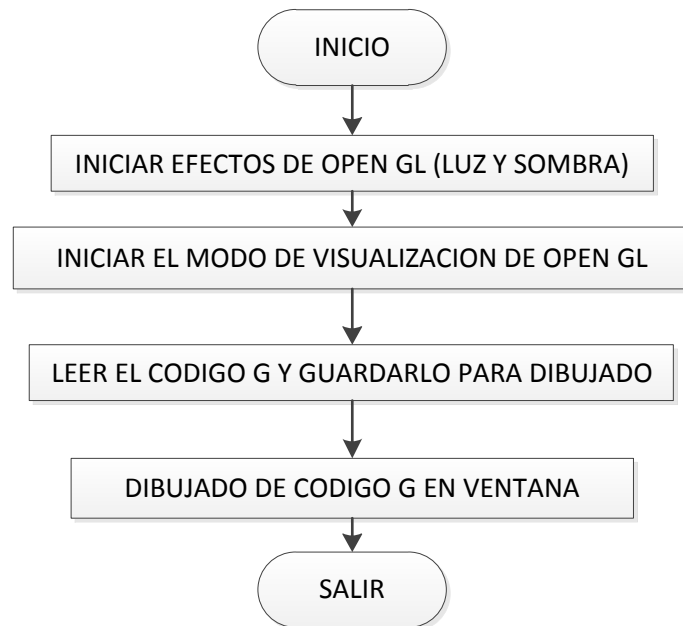


Figura 2.10: Operaciones para representar una escena

Todas estas funciones nos sirven para el tratamiento de la imagen descargada, esta descarga se realiza en una clase donde contendrá variables de tipo puntero las cuales reservarán memoria para que estas sean tratadas.

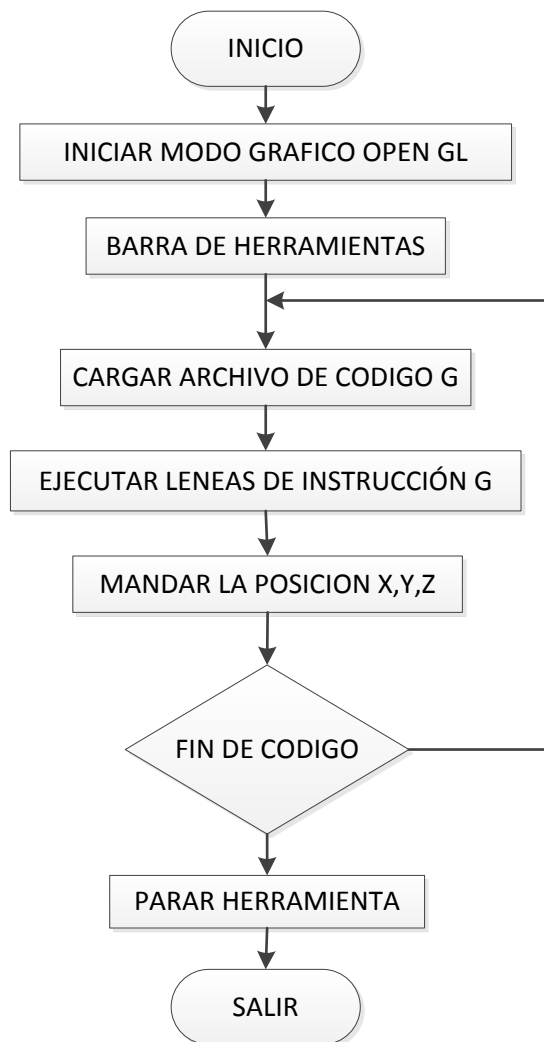


Figura 2.11: Diagrama de flujo de simulación de código G

El siguiente paso consiste en relacionar mediante un lenguaje de programación las diferentes entidades tanto gráficas como matemáticas para poder describir el comportamiento del prototipo. Los pasos para programar un simulador se describen en el diagrama de flujo de la figura 2.11.

2.3.1 Implementación del algoritmo para leer el código G

El código de programación se implementa haciendo uso de componentes de visual C++, uno de los componentes utilizados es la caja de lista de datos. Para usar sus métodos se crea una variable de tipo control y la cual nos ayuda a saber dónde se encuentra el cursor que apunta a una línea de texto. A continuación se muestra el código utilizado para acceder a una línea de texto.

```
m_listDatos.SetCurSel(indice);           //Seleccionar actual
m_listDatos.GetText( indice, str );
tam_linea=str.GetLength();
```

Figura 2.12: Acceso a una línea de texto

Una vez que se tiene acceso a la línea de texto se lee carácter a carácter la línea completa, esta línea representa una instrucción de código G, la primera línea puede ser un comentario y en tal caso no se ejecuta. La línea de texto contiene información de caracteres especiales usados en el código G por lo cual si en la línea existe un carácter se analiza y si es un carácter de comando de código G se estudia el formato de la línea.

```
//Leer una línea y dibujar el dato
dato=str.GetAt(i);
if(dato=='(') break;

if(dato[0] >='A' & dato[0]<'Z')    //Es decir solo letras
{                                  //0 Del if donde checa datos correctos
    m_dato=dato[0];

    switch(m_dato){                //1
    case 'G':                      //dentro del codigo G hay varias opciones

        a="";

        i++;
        valor=str.GetAt(i);
```

Figura 2.13: Lectura de un carácter

La figura 2.13 se lee en la variable “dato” un carácter a través de la variable string “str” que usa el método GetAt para obtener un carácter a la vez, posteriormente si ese carácter es letra entra a un ciclo de decisión que se utiliza para entrar a un conjunto de casos que identifica el tipo de código G que se está leyendo.

```

while(valor[0]>='0' & valor[0] <='9')
{
    a=a+valor;          //concatena cadena

    j=i+1;
    if(j>=tam_linea) break;
    valor=str.GetAt(j);    //Leer el siguiente dato debe ser un numero
                        // sino terminar
    i++;
}

```

Figura 2.14: Lectura de un valor numérico

Cuando se ha detectado que se tiene un código G se prosigue a saber qué tipo. Seguido del carácter G sigue un numero cuyo valor identifica la acción de la maquina si es 0 hablamos de un movimiento rápido si es 1 el código es G01 y se debe se hacer interpolación o activar el roto tool para maquilar. En la figura 2.14 se muestra la forma de leer el valor numérico.

```

////////////////////////////////////
case 0:          //Solo moverse sin ranurar
////////////////////////////////////

do{
    a="";
    i++;
    if(i>=tam_linea) break;
    dato=str.GetAt(i);

    if(dato=='X' || dato == 'Y' || dato=='Z')
    {
        i++;
        valor=str.GetAt(i);

        while(valor[0]>='0' & valor[0] <='9' || valor=='-' || valor=='.') //Obtener el numero
        {
            a=a+valor;
            j=i+1;
            if(j>=tam_linea) {i++; break;}
            valor=str.GetAt(j);
        }
    }
}

```

Figura 2.15: Obtención de valor X, Y, Z

Cuando se ha definido el tipo de código se procede a obtener los valores numéricos de las coordenadas. Se continua con la lectura carácter a carácter de la línea y se decide si es la letra X, Y o Z, en el caso del código G00. Si tenemos el carácter X, Y o Z seguido de este viene la información del valor de la coordenada, por lo cual se utiliza un ciclo para leer el valor numérico de la coordenada. En la figura 2.15 se muestra el código, en el principio de código se encuentra una estructura de control de decisión que checa si el carácter es una X, Y o Z, en tal caso se obtiene su valor y esto se realiza en ciclo “while” que lee los siguientes caracteres mientras sean numero o un punto. Esta información se dibuja y se manda al CNC para ejecutar cada línea de código.

CAPITULO III: CODIGOS G

En este capítulo se presenta la interfaz electrónica con el microcontrolador de Microchip el PIC18F2550 y la comunicación USB, además del protocolo usado para comunicarse con la computadora.

3.1 Preliminares matemáticos

En este segundo apartado se inicia con el conjunto de herramientas matemáticas para la edición de figuras geométricas que se utilizan cuando se dibuja el conjunto de instrucciones en código G. Se comienza por la ecuación de la línea recta y su algoritmo y posteriormente se presenta la ecuación del círculo.

3.1.2 Ecuación de la línea recta

La primera ecuación que se mostrara es la de la línea recta. La ecuación de una línea recta se puede encontrar si conocemos dos puntos de la recta (x_1, y_1) y (x_2, y_2) con estos dos puntos un puede encontrar la ecuación de cualquier punto usando la siguiente formula.

$$y - y_1 = m(x - x_1) \quad \text{Ecuación (1)}$$

Donde m representa la pendiente de la línea recta y (x_1, y_1) es el punto inicial. Desarrollando la ecuación (1) se llega a:

$$y = mx + b \quad \text{Ecuación (2)}$$

En la ecuación (2) se tienen la forma estándar de la línea recta. El valor de $b = -m \cdot x_1 + y_1$ se obtiene despejando la variable “ y ” de la ecuación (1). Al graficar la ecuación (2) se observa que si la pendiente es de valor positivo la recta esta inclinada hacia la derecha y en caso contrario hacia la izquierda, esto se muestra en la figura 3.1

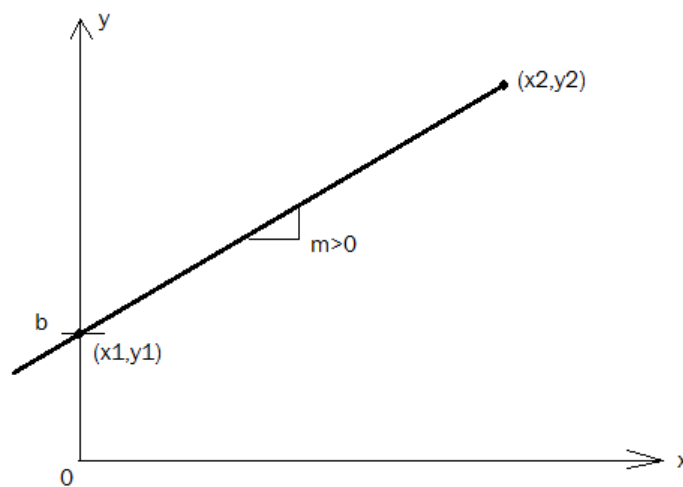


Figura 3.1: Línea recta con pendiente positiva

El valor de la constante b se utiliza para conocer la intersección de la línea recta sobre el eje y . Si $b > 0$ la recta esta sobre el eje x y si $b < 0$ la recta esta debajo del eje x . La ecuación (1) es la que se implementa en el programa y se utiliza para implementar los códigos G00 y G01.

3.1.3 Ecuación del círculo

La ecuación de un círculo que no se encuentra centrado en el origen se puede expresar de la siguiente forma:

$$(x_1 - x_0)^2 + (y_1 - y_0)^2 = R^2 \quad \text{Ecuación (3)}$$

Donde (x_1, y_1) es un valor sobre el círculo y R es el radio del círculo, el valor X_0 y Y_0 es la coordenada del centro del círculo.

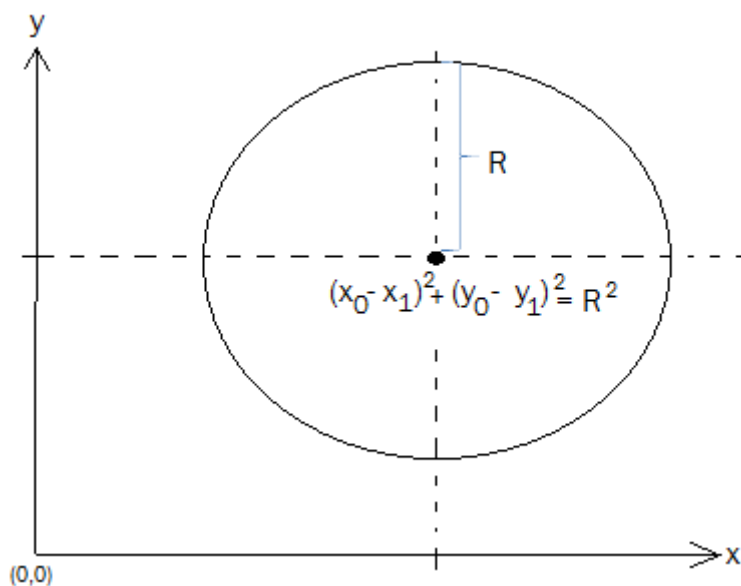


Figura 3.2: Ecuación del círculo centrado en (X_0, Y_0)

En la figura 3.2 se muestra el círculo con centro en (x_0, y_0) en la ecuación (3) se necesita conocer el centro del círculo y el radio para poder hacer la gráfica. Sin embargo, si solo se conocieran dos puntos y el radio del círculo. Se puede usar geometría analítica para poder obtener la ecuación del círculo (3). En la figura (20) se muestra el problema y la geometría que se construye para poder hallar la ecuación.

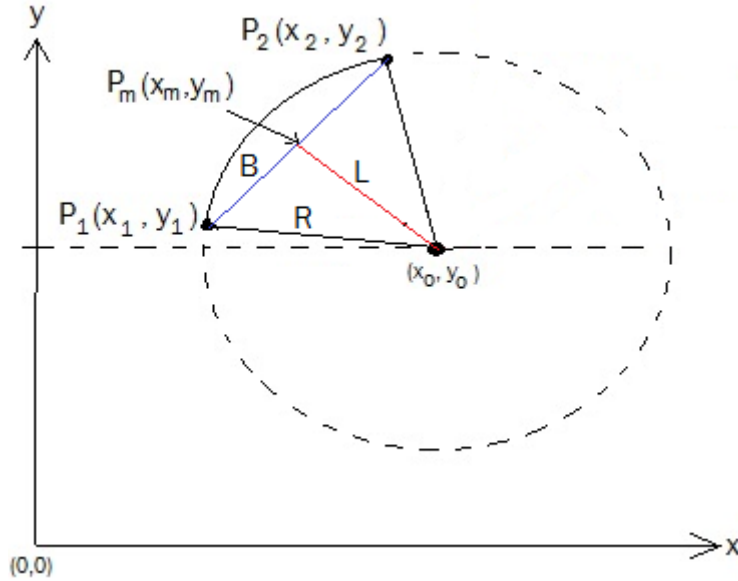


Figura 3.3: Círculo de radio R

En la figura (3.3) el dato desconocido es el centro del círculo. Para hallar el centro del círculo primero se procede a calcular el punto medio que se encuentra entre los puntos P1 y P2. Las siguientes ecuaciones se utilizan:

$$x_m = \frac{x_1 + x_2}{2} \quad \text{Ecuación (4)}$$

$$y_m = \frac{y_1 + y_2}{2} \quad \text{Ecuación (5)}$$

El punto medio denotado como Pm tiene coordenadas (Xm, Ym), una vez hallado este punto se procede a hallar la magnitud de B. La magnitud de B se obtiene calculando la mitad de distancia entre los puntos P1 y P2.

$$B = \frac{1}{2} \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad \text{Ecuación (6)}$$

El valor de B se utiliza para conocer L a partir de un triángulo que se forma entre el punto medio Pm el punto P1 y el origen del círculo.

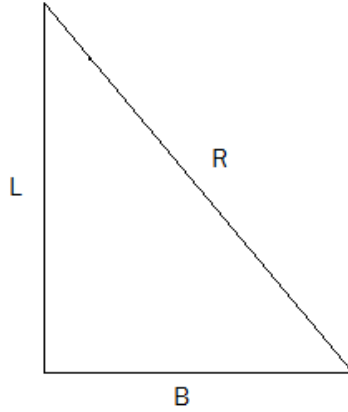


Figura 3.4: Triángulo entre los puntos Pm, P1 y (xo,yo).

De la figura (3.4) el valor de L se halla usando el teorema de Pitágoras.

$$L = \sqrt{R^2 - B^2} \quad \text{Ecuación (7)}$$

Se puede formar un triángulo tomando como vértices los puntos Pm y P1

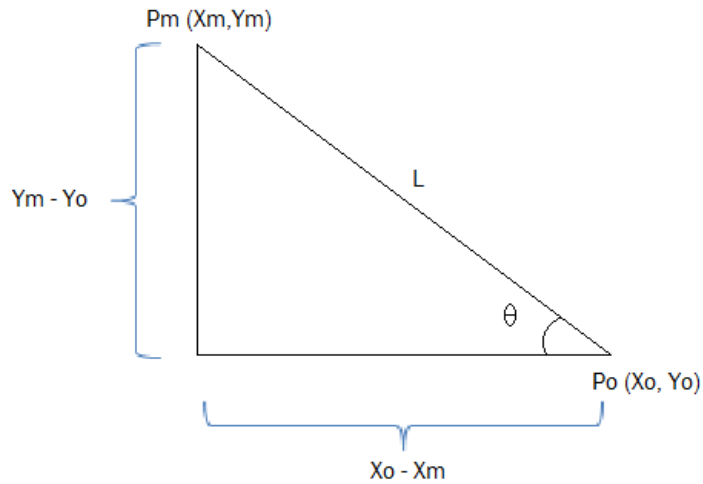


Figura 3.5: Triángulo formado por Pm y Po

De la figura (3.5) se observa que las coordenadas buscadas (Xo, Yo) se pueden hallar de las siguientes relaciones trigonométricas.

$$x_0 - x_m = L \cos(\theta) \Rightarrow x_0 = x_m + L \cos(\theta) \quad \text{Ecuación (8)}$$

$$y_0 = y_m + L \sin(\theta) \quad \text{Ecuación (9)}$$

El ángulo teta se halla a partir de los puntos sobre el círculo y se muestra la relación en la siguiente figura

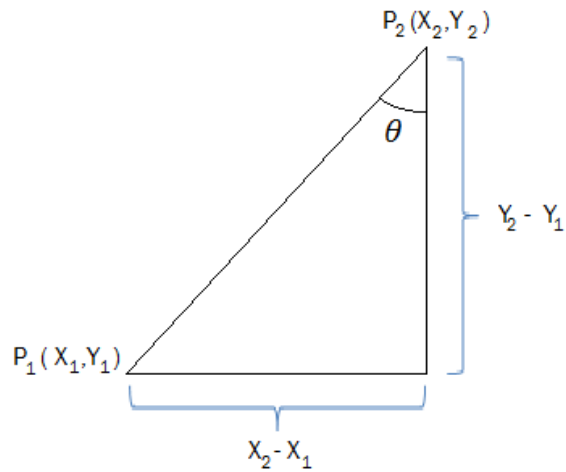


Figura 3.6: Triángulo formado por P1 y P2

$$\theta = \tan^{-1} \left(\frac{X_2 - X_1}{Y_2 - Y_1} \right) \quad \text{Ecuación (10)}$$

Las ecuaciones (8), (9) y (10) se utilizan para implementar las instrucciones G02 y G03 en el caso de que se haga un círculo de radio R y para el cual se da un punto inicial y un punto final.

3.1.4 Ecuación del círculo con vectores de dirección al centro

Se puede determinar la ecuación (3) conociendo un punto sobre el círculo y el valor que se debe de desplazar hacia el centro. La siguiente figura muestra esta situación.

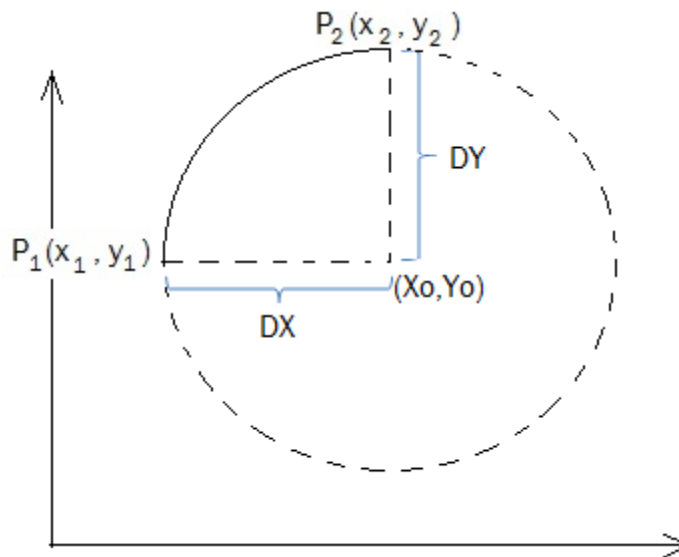


Figura 3.7: Puntos en el círculo y desplazamiento al centro

Como se muestra en la figura (3.7) si conocemos el punto inicial P1 podemos hallar el valor de las coordenadas del centro del círculo a través de los desplazamientos en X y Y respectivamente, las ecuaciones que se utilizan en este caso son las siguientes:

$$\begin{aligned} x_0 &= DX + x_1 \\ y_0 &= DY + y_1 \\ R &= \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} \end{aligned} \quad \text{Ecuación (11)}$$

A partir del sistema de ecuaciones (11) se puede hallar la ecuación del círculo (3), el valor de radio R y los desplazamientos DX, DY varían y no siempre son simétricos como se dibujan en la figura (3.7). En el caso de que solo se conozca un punto P1 y un desplazamiento en X o en Y se desplaza sobre el eje y se dibuja un círculo completo

3.2 Códigos G implementados

El código G es el nombre que habitualmente recibe el lenguaje de programación más usado en Control numérico (CNC), el cual posee múltiples implementaciones. Usado principalmente en automatización, forma parte de la ingeniería asistida por computadora. Al G-code se le llama en ciertas ocasiones lenguaje de programación G.

En términos generales, código G es un lenguaje mediante el cual las personas pueden decir a máquinas herramienta controladas por computadora qué hacer y cómo hacerlo.

Los códigos G que se utilizan comúnmente en programa son:

Tabla 5: Códigos G

CODIGO	SIGNIFICADO
• G0 o G00	Movimiento rápido
• G1 o G01	Interpolación lineal
• G2 o G02	Interpolación en el sentido del reloj
• G3 o G03	Interpolación en sentido anti-horario
G40	Apagar compensación de corte
G41	Compensación de corte a la izquierda
G42	Compensación de corte a la derecha
G92	Desplazamiento de punto cero
G87	Fresado de caja rectangular

G81, G82, G83, G84	Ciclo de taladrado
G90	Programación absoluta
G91	Programación rectangular
G25	Llamada a subprograma
G26	Separar subprogramas (disk)
G94	Velocidad de avance en mm/min
G95	Velocidad de avance en mm/revolución
G54	Punto cero 1
G55	Punto cero 2
G56	Punto cero 3
G57	Punto cero 4
G58	Punto cero 5
G59	Punto cero 6
G53	No desplazamiento de punto cero

Tabla 6: Códigos M

CODIGO	SIGNIFICADO
M0 o M00	Parar programa
• M3 o M03	Arranque de husillo en sentido horario
M4 o M04	Arranque de husillo en sentido anti horario
M5 o M05	Parar husillo
M8 o M08	Prender refrigerante
M9 o M09	Apagar refrigerante
M17	Retorno de subprograma
• M30 o M02 o M2	Fin de programa

Tabla7: Códigos complementarios

CODIGO	SIGNIFICADO
F	Velocidad de avance
S	Velocidad de husillo
N	Numero de bloque
T o TOOL CALL	Herramienta
• X	Código para el eje X
• Y	Código para el eje Y
• Z	Código para el eje Z
• I	Distancia incremental a el centro de eje x

• J	Distancia incremental a el centro de eje y
• K	Distancia incremental a el centro de eje z
R o Z	Arco <180 grados
R- o Z-	Arco >180 grados
P	Superposición de fresado
L	Numero sub programa de bloque
U	Repeticiones de subprograma

3.2.1 COGIGO G00

G00 representa un movimiento rápido en el plano x, y este comando se utiliza para moverse de un punto a otro sin hacer perforaciones, la forma de utilizar el comando es la siguiente
G00 X__ Y__ Z__

Donde

X Dirección opcional para el movimiento en el eje X

Y Dirección opcional para el movimiento en el eje Y

Z Dirección opcional para el movimiento en el eje Z

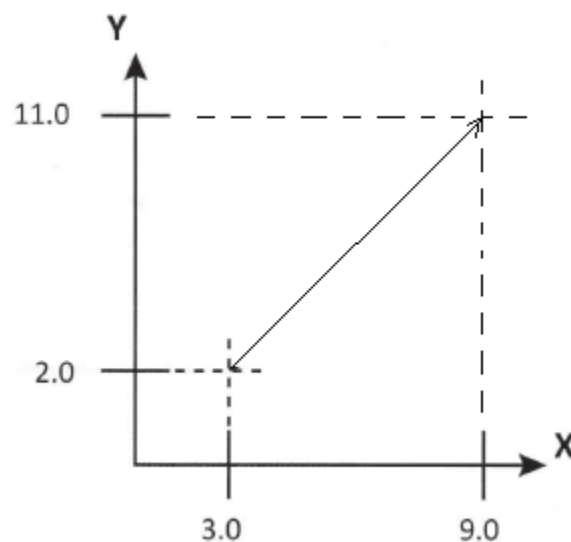


Figura 3.8: Línea que genera el código G00

En este ejemplo la maquina se encuentra en el punto $x=3.0$, $y=2.0$ y $Z=0.0$, se usa el comando G00 X9.0 Y11.0 para pasar al otro punto en línea recta, hay dos caminos que se pueden elegir para llegar al punto para lo cual se tendría que seguir la secuencia de comandos siguiente:

Ruta 1: G00 X9.0
G00 Y11.0
Ruta 2: G00 Y11.0
G00 X9.0

Para implementar el comando se utiliza la ecuación de la línea recta donde el primer punto x_1 , y_1 se toma como referencia y ya se conoce, posteriormente se da un segundo punto x_2 , y_2 y con esta información se mueve la maquina en línea recta hasta el siguiente punto.

La ecuación que se utiliza es la siguiente:

$$y = m(x - x_1) + y_1$$

Donde

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

3.2.2 COGIGO G01

Representa un movimiento de interpolación lineal, esto quiere decir que este código se utiliza cuando se activa la herramienta y se empieza a talar las placas en bruto. La forma de utilizar el comando es la siguiente:

G01 X__ Y__ Z__ F__

Donde

X Dirección opcional para el movimiento en el eje X

Y Dirección opcional para el movimiento en el eje Y

Z Dirección opcional para el movimiento en el eje Z

F Avance en pulgadas o milímetros por minuto

Se muestra a continuación un ejemplo, la maquina se encuentra en la posición $x = 3.0$, $y = 2.0$ y se utiliza el comando G01 para marcar una línea hacia el punto $x = 9.0$, $y = 7.0$
G01 X9.0 Y7.0 F10.0;

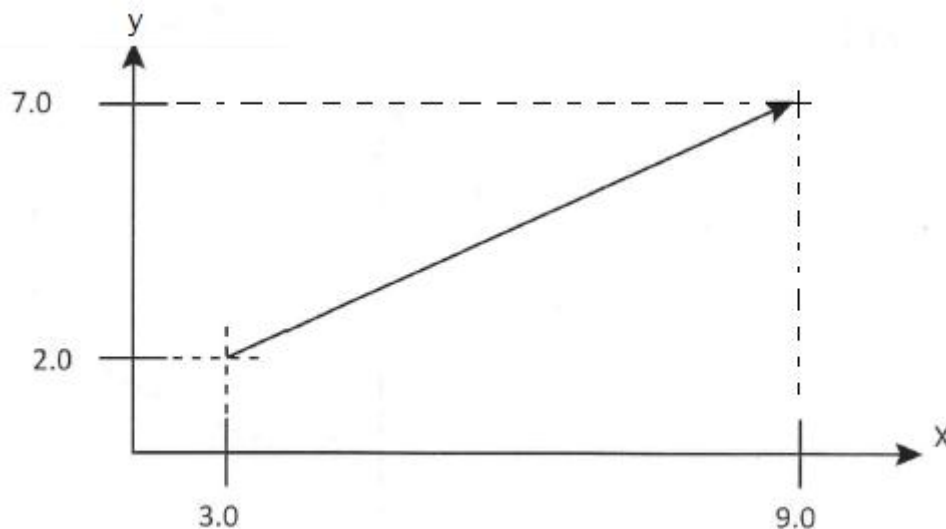


Figura 3.9: Línea que genera el código G01

En el código anterior se utiliza una avance de 10 en el programa solo nos ayudara a activar el taladro, ya que no contamos con la electrónica para variar la velocidad.

El algoritmo que se utiliza para implementar en el programa el dibujo y trayectoria de la línea recta se muestra en la figura (3.10). En este caso se muestra como desde el inicio que uno debe conocer el punto previo (x_1, y_1) para poder llegar al siguiente punto (x_2, y_2) se hace uso de la ecuación de la línea recta. Además en el caso del código G01 se debe activar la herramienta giratoria que hará el desbaste sobre el material en bruto. En el caso del código G00 no se activa el Roto Tool y solo se mueve la maquina al punto deseado haciendo una trayectoria de línea recta.

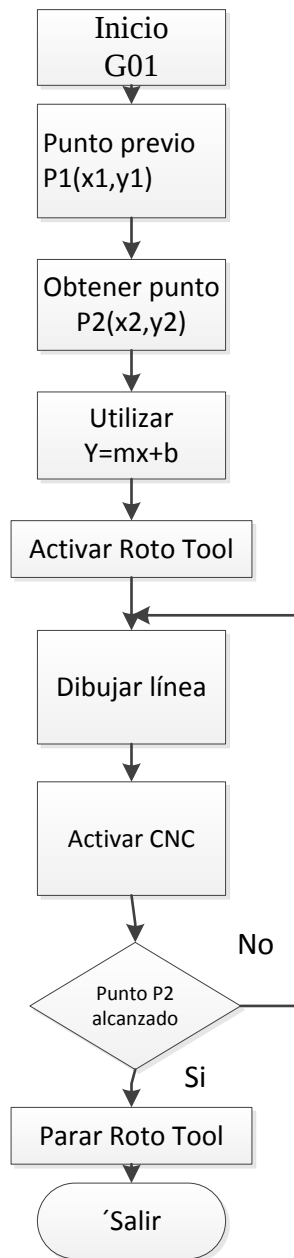


Figura 3.10: Algoritmo para implementar el código G01

3.2.3 COGIGO G02

Interpolación circular con sentido de las manecillas del reloj, este código hace círculos, arcos y semicírculos, dado un punto en el plano se elige otro para el cual se pueda llegar haciendo un círculo, cuenta con los siguientes parámetros

X Dirección opcional para el movimiento en el eje X
 Y Dirección opcional para el movimiento en el eje Y
 Z Dirección opcional para el movimiento en el eje Z

I Vector en dirección el eje X hasta el centro del círculo
J Vector en dirección el eje Y hasta el centro del círculo
K Vector en dirección el eje Z hasta el centro del círculo
R Valor del radio del círculo

A continuación se muestra un ejemplo donde se usa el comando G02 y la opción de asignar un radio del círculo.

G02 X6.0 Y3.0 R4.0

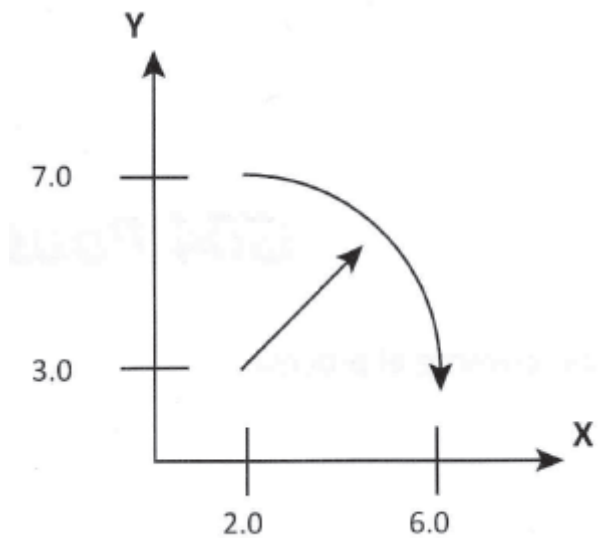


Figura 3.11: Trayectoria generada por el código G02

3.2.4 COGIGO G03

Interpolación circular en contra de las manecillas del reloj, este código hace círculos, arcos y semicírculos, dado un punto en el plano se elige otro para el cual se pueda llegar haciendo un círculo, cuenta con los siguientes parámetros

X Dirección opcional para el movimiento en el eje X
Y Dirección opcional para el movimiento en el eje Y
Z Dirección opcional para el movimiento en el eje Z
I Vector en dirección el eje X hasta el centro del círculo
J Vector en dirección el eje Y hasta el centro del círculo
K Vector en dirección el eje Z hasta el centro del círculo
R Valor del radio del círculo

A continuación se muestra un ejemplo donde se usa el comando G03 y la opción de asignar un radio del círculo.

G03 X6.0 Y3.0 R4.0

Al igual que el código G02 el diagrama que se utiliza es el mostrado en la figura (3.12) con la diferencia de que el círculo se realizara en contra de las manecillas del reloj

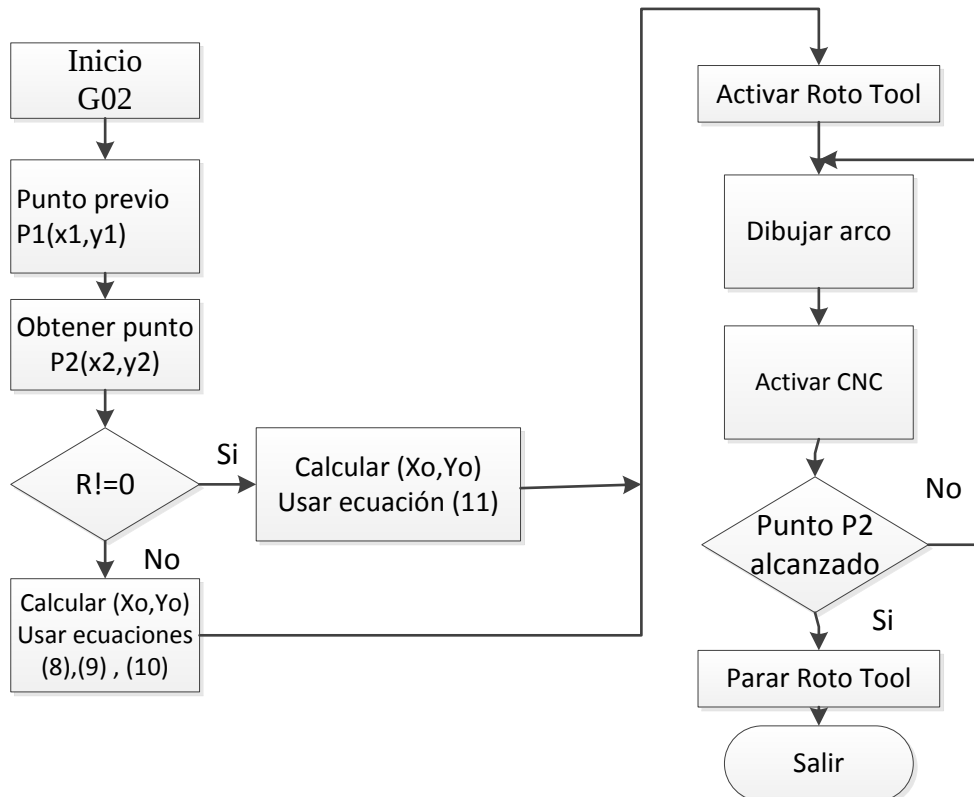


Figura 3.12: Diagrama de flujo para el código G02

CAPITULO IV

DISEÑO DEL SISTEMA ELETRONICO Y MECANICO

En este capítulo se muestran los circuitos para la etapa de potencia del CNC, la tarjeta de control con el PIC18f2550 y se muestra el diseño mecánico del CNC hecho en el programa CAD Solid Works de la empresa Dassault Systems.

4.1 CIRCUITO DE CONTROL

El circuito que controla las instrucciones de la computadora y las traduce al movimiento x, y, z se ha implementado con un PIC18F2550 de la familia de Microchip este microcontrolador tienen una interfaz vía USB por medio de la cual uno se puede comunicar al puerto USB de la computadora. Los datos de las coordenadas procesadas en la computadora se mandan al microcontrolador en cual utiliza dos pines para manejar cada eje del plano cartesiano movido por un motor a pasos unipolar. Por lo tanto se usan 6 pines del microcontrolador para ubicarse en un punto en el espacio y un pin extra que se encarga de activar la herramienta de corte y de perforación. A continuación se muestra un diagrama esquemático de la conexión de la tarjeta con las etapas de potencia de los motores.

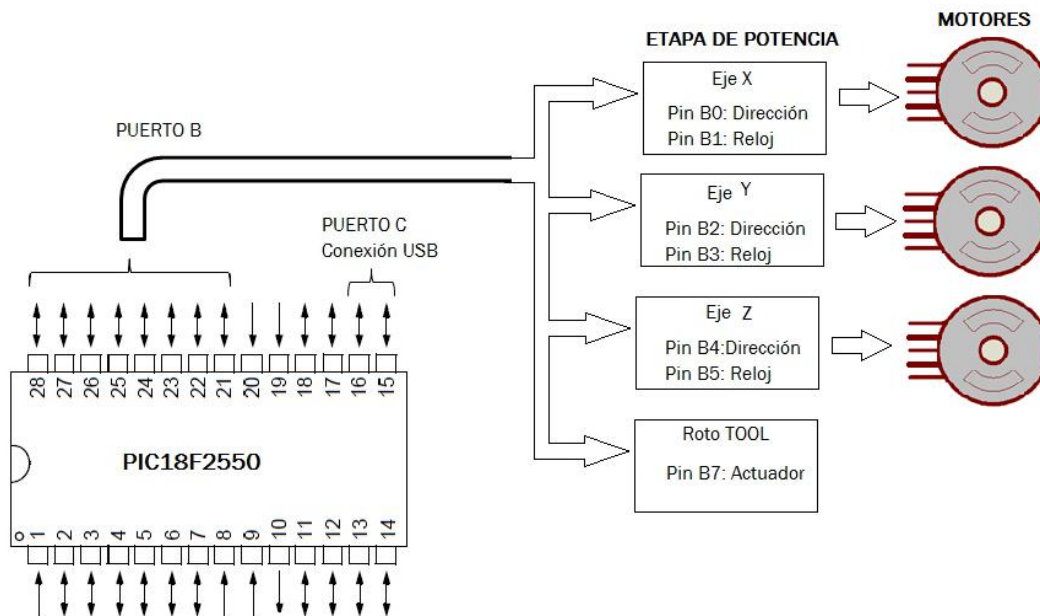


Figura 4.1: Diagrama para la conexión del PIC18F2550

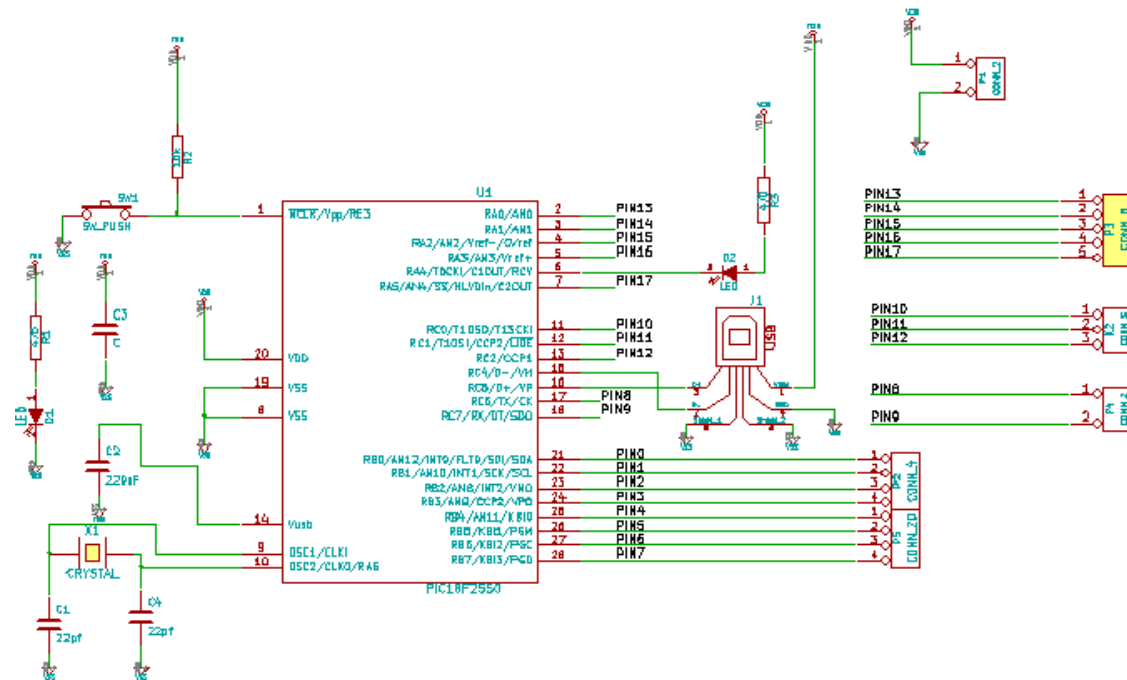


Figura 4.2: Tarjeta electrónica de control

4.2 Comunicación por puerto USB del microcontrolador

El circuito de control de comunica a la computadora y al programa a través del puerto serial universal USB, el integrado encargado de la comunicación y control de los motores es el PIC18F2550 el cual usa su puerto C para conectarse al puerto USB y transmitir datos. En la figura se muestra la disposición de pines

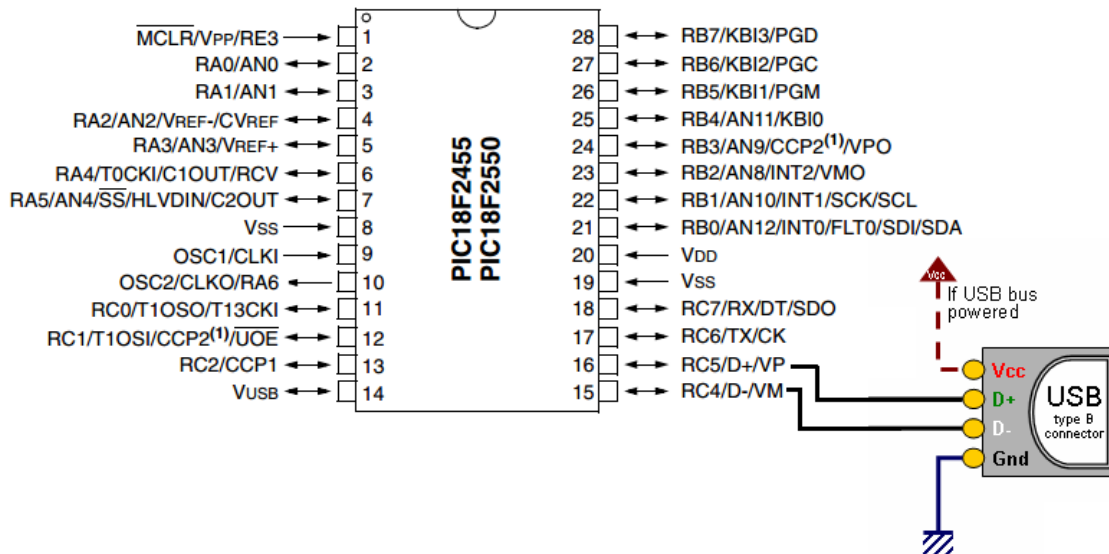


Figura 4.3: Diagrama del PIC18F2550

Los pines utilizados del microcontrolador son el pin 15 y pin 16, la comunicación necesita además un capacitor conectado al pin 14 de valor 0.47 uF como se muestra en la figura (4.3) se deben conectar a un conector USB tipo B hembra. Para la comunicación se hacen dos programas uno que se programa en el microcontrolador hecho en PIC-C o en el compilador CCS y el programa anfitrión de la PC.

4.2.1 Programación del PIC18f2550 para la conexión USB

Para la edición del código a implementar al PIC18F2550 se utilizara el compilador PIC C Compiler CCS, se utilizan 4 librerías propias al compilador:

pic18f2550.h = Librería para utilizar dicho dispositivo con todas sus características en hardware. Propia del compilador y no debe modificarse.

pic18_usb.h = Librería correspondiente al uso de la comunicación USB para PIC18. Propia del compilador y no debe modificarse.

usb.c = Librería para uso estándar de comunicación USB. Propia del compilador y no debe modificarse.

descriptor_USB.h = Librería que contiene la configuración y descripción general del dispositivo a conectar. Librería de ejemplo que fue modificada para utilizar con este proyecto en particular, con el nombre de "usb_desc_scope.h", son 2 las partes modificadas:

1-Vendor ID, VID= 04D8, código correspondiente a la firma de Microchip para utilizar comunicación USB.

2.-Product ID, PID= 0090, código cualquiera, escogido para este proyecto en particular y que debe ser igual al de la Interfaz y Driver.

Microchip proporciona un DRIVER que corre bajo Windows 7, para comunicación USB para la familia PIC18, el cual puede descargarse desde su sitio web.

El nombre del **Driver es MCHPUSB**, y está compuesto por 5 elementos:

- 1.-**Ioctl.h**
- 2.-**mchpusb.cat**
- 3.-**mchpusb.inf**
- 4.-**mchpusb.sys**
- 5.-**mchpusb64.sys**

En nuestro caso nos interesa solo en "**mchpusb.inf**", se trata del archivo a modificar pues contiene los códigos **VID y PID** que necesitamos configurar.

Necesitamos modificar el código PRODUCT ID = PID, el valor para dicho código será **0090**, el código VID no será necesario cambiarlo, pues es la firma de Microchip para el uso de comunicaciones USB, el cual corresponde a **04D8**, ya está escrito por defecto en el INF.

El programa para la interfaz USB 2.0 del microcontrolador debe cumplir con lo siguiente

- Configuración de comunicación USB: Tener parámetros de reconocimiento y transmisión de comunicación USB.
- Configurar la entrada de datos y de control: determinar el puerto de salida e interrupciones a utilizar
- Inicialización de USB: Se encarga de inicializar la comunicación USB
- Llamada de rutina de servicio: rutina cíclica que evitara la pérdida de comunicación

Rutina cíclica

- Establecer tiempo de activación de rutina
- Llamada a rutina de servicio
- Recibe el dato
- Procesamiento del dato envía y recibe información
- Notifica fin de envío de dato
- Llamada a rutina de servicio

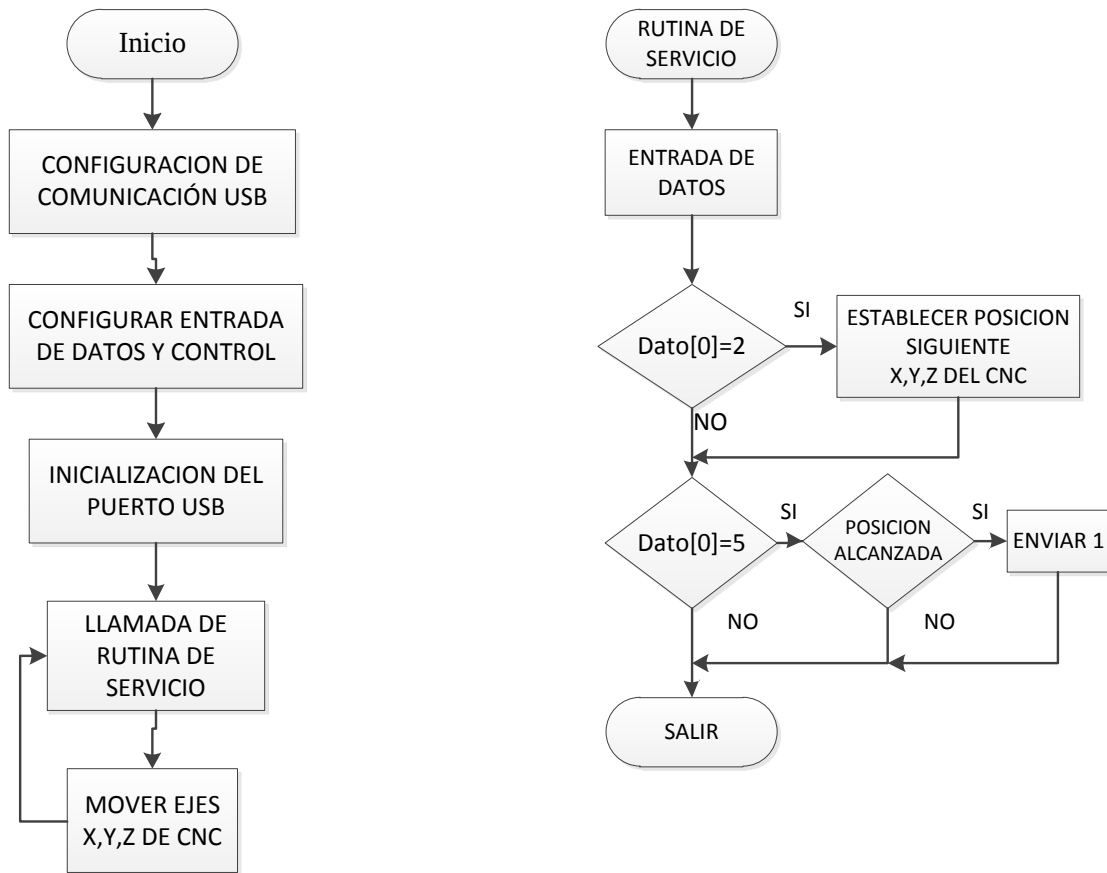


FIGURA 4.4: Algoritmo implementado en el PIC

Como se muestra en la figura (4.4) el programa comienza configurando los puertos usados para la comunicación de datos y control de los motores, posteriormente se inicializa la comunicación USB una vez establecida la comunicación se entra en lazo infinito que lo que hace es mandar datos de posición de los ejes X, Y, Z del CNC, si no hubiera datos que enviar de alguna posición distinta los motores no se mueven. Para no romper la comunicación de la tarjeta de control con la PC se utiliza una interrupción por tiempo de 1ms. Cada milisegundo se lee el puerto USB y se verifica si existen datos entrantes o si ya se ha terminado de mover al CNC a la posición deseada desde la PC.

La rutina de interrupción maneja un arreglo de tamaño 32 de tipo entero llamado “DATO”, este arreglo de datos reserva su posición inicial para saber qué tipo de instrucción debe ejecutar. En el diagrama de flujo si DATO[0]=2 se debe de leer del puerto USB 31 valores de tipo entero que indican cual es la posición siguiente del CNC, cada dato representa un valor que se manda al puerto B del microcontrolador y este dato se ejecuta cuando el motor ejecuta un paso.

Finalmente si DATO[0]=5 el microcontrolador envía el valor “1” indicando que se ha terminado de mover el sistema y está preparado para recibir más instrucciones.

4.3 Programación del Software de control

Al igual que el microcontrolador el programa en Visual C++ necesita un conjunto de librerías para comunicarse con el puerto USB. En el caso de los microcontroladores de la familia Microchip se utiliza la librería `mpusbapi.dll`.

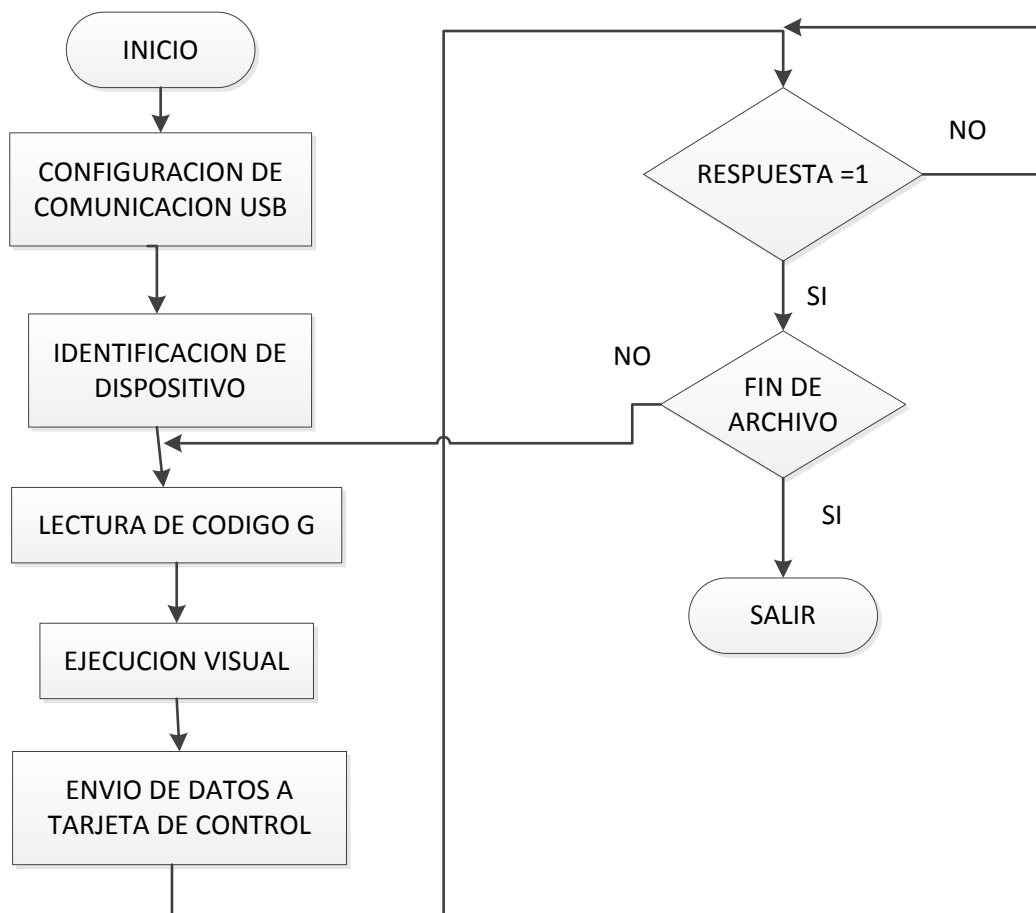


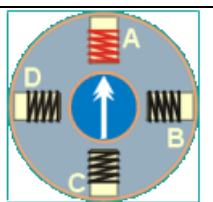
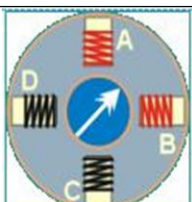
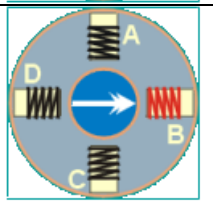
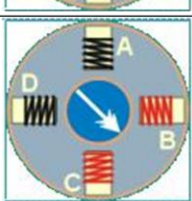
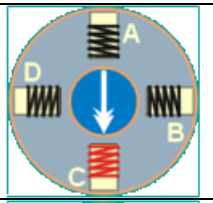
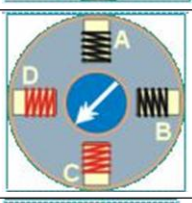
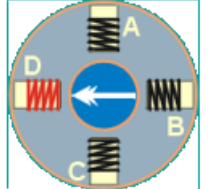
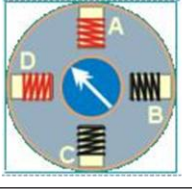
Figura 4.5: Programa de control para la PC

En la figura (4.5) se muestra el diagrama de flujo del programa de control de la computadora, la primera parte consiste en configurar el puerto para la comunicación por USB y identificar el dispositivo de comunicación el cual es un microcontrolador, posteriormente se lee una línea de código G, se grafica en pantalla y se envían los datos para que se mueva el CNC a la coordenada deseada. El microcontrolador envía una señal de respuesta cuando termina de ejecutar la instrucción, como se muestra en el diagrama de flujo el valor de esta señal debe de ser “1”, cuando se ha terminado de ejecutar una línea de código se procede a ejecutar la siguiente línea de un archivo completo y este ciclo continua hasta que se han leído todas las instrucciones.

4.4 Etapa de potencia

Para el diseño de la etapa de potencia se utilizan motores unipolares los cuales presentan un control sencillo. Los motores unipolares funcionan energizando las bobinas del motor. Existen tres formas de controlar las bobinas de un motor unipolar y están son paso simple, paso doble y medio paso, en el trabajo se expone el control por paso simple y paso medio. En paso simple consiste en lo siguiente, se va energizando una bobina a la vez y cuando se hayan terminado de energizar la cuatro bobinas se comienza de nuevo la secuencia para hacer girar al motor, en este esquema la fuerza del motor no es la máxima que puede entregar, sin embargo su control es simple comparado con los circuitos utilizados para medio paso y paso doble. La secuencia de control se muestra a continuación:

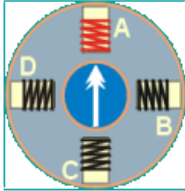
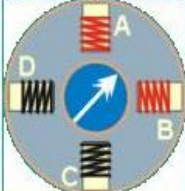
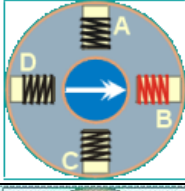
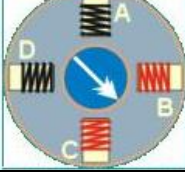
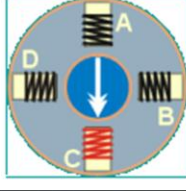
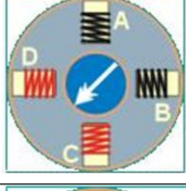
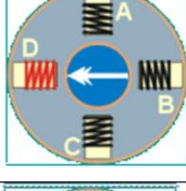
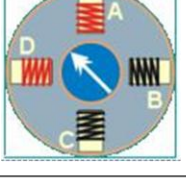
Tabla 5: Secuencia de paso simple y paso doble

Paso simple	A	B	C	D		Paso doble	A	B	C	D	
1	1	0	0	0		1	1	0	0		
2	0	1	0	0		2	0	1	1	0	
3	0	0	1	0		3	0	0	1	1	
4	0	0	0	1		4	1	0	0	1	

En la tabla 5 se muestra la secuencia de paso simple y paso doble para el control de un motor unipolar donde las letras A, B, C, D se utilizan para indicar la bobina que se está energizando, los motores unipolares cuentan con cinco hilos de conexión de los cuales cuatro son los de las bobinas y uno se utiliza como conexión común y este puede conectarse a alimentación o tierra.

En la configuración de medio paso se gana precisión debido a que en vez de mandar cuatro pulsos para llegar a la posición deseada se deben mandar ocho. La secuencia de pulso se combina con la de paso simple y pasó doble, en esta configuración se pueden llegar a posiciones intermedias que en las otras dos configuraciones no se podían, en la siguiente tabla se muestra la secuencia de pasos.

Tabla 5b: Secuencia de medio paso

Medio Paso	A	B	C	D	
1	1	0	0	0	
2	1	1	0	0	
3	0	1	0	0	
4	0	1	1	0	
5	0	0	1	0	
6	0	0	1	1	
7	0	0	0	1	
8	1	0	0	1	

En la tabla 5b se muestra la secuencia de medio paso para el control de un motor unipolar donde las letras A, B, C, D se utilizan para indicar la bobina que se está energizando, en este caso se tienen ocho secuencias, por lo que el circuito de control debe poder enviar estas secuencias. Este tipo de control permite tener posiciones intermedias como se muestra en la secuencia dos en donde se ha enviado.

Para el circuito de control se utiliza un microcontrolador en el cual se programa la secuencia de secuencia de medio paso de paso doble o medio paso, se ha utilizado la secuencia de medio paso para obtener los resultados debido a que esta secuencia alcanza posiciones intermedias. El programa se realiza en PIC-C y se muestra a continuación.

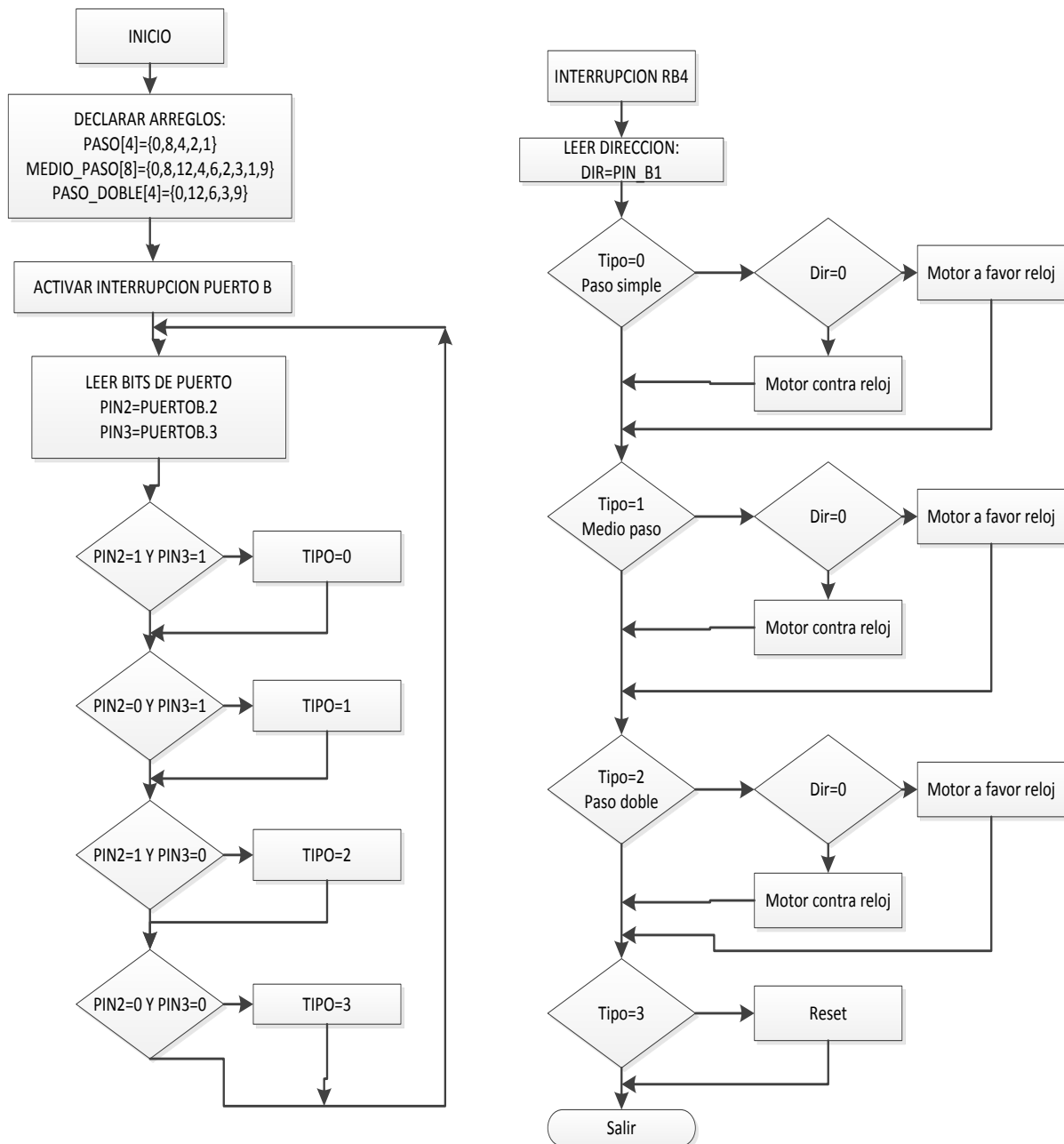


Figura 4.6: Diagrama de flujo del control de motores

En la figura 4.6 se muestra el diagrama utilizado para el control de motores en configuración de medio paso, paso doble y paso simple, la implementación del programa se encuentra en el apéndice A. La forma de controlar los motores depende de la lectura de dos bits de control en el puerto B. También se utiliza la interrupción de pin 4 del puerto B para poder saber si hay un pulso de reloj en tal caso se saca un valor de los arreglos PASO, MEDIO_PASO y PASO_DOBLE que contienen la secuencia para las distintas formas de activar el motor unipolar. Para saber la dirección del motor se lee el pin uno del puerto B.

La tarjeta de control además de enviar los datos de reloj y dirección proporciona también la elección del tipo de secuencia de control de los motores por lo tanto en el programa se ha incluido un menú con estas opciones. La figura siguiente muestra el menú.

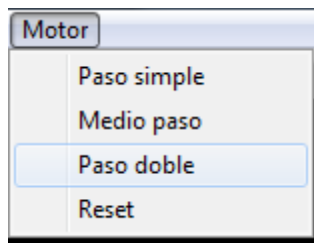


Figura 4.7: Menú de opciones de control de motores

Como parte final de esta sección se muestra el diagrama eléctrico y el circuito en PCB del circuito de control esto se ve en las figuras 4.8 y 4.9

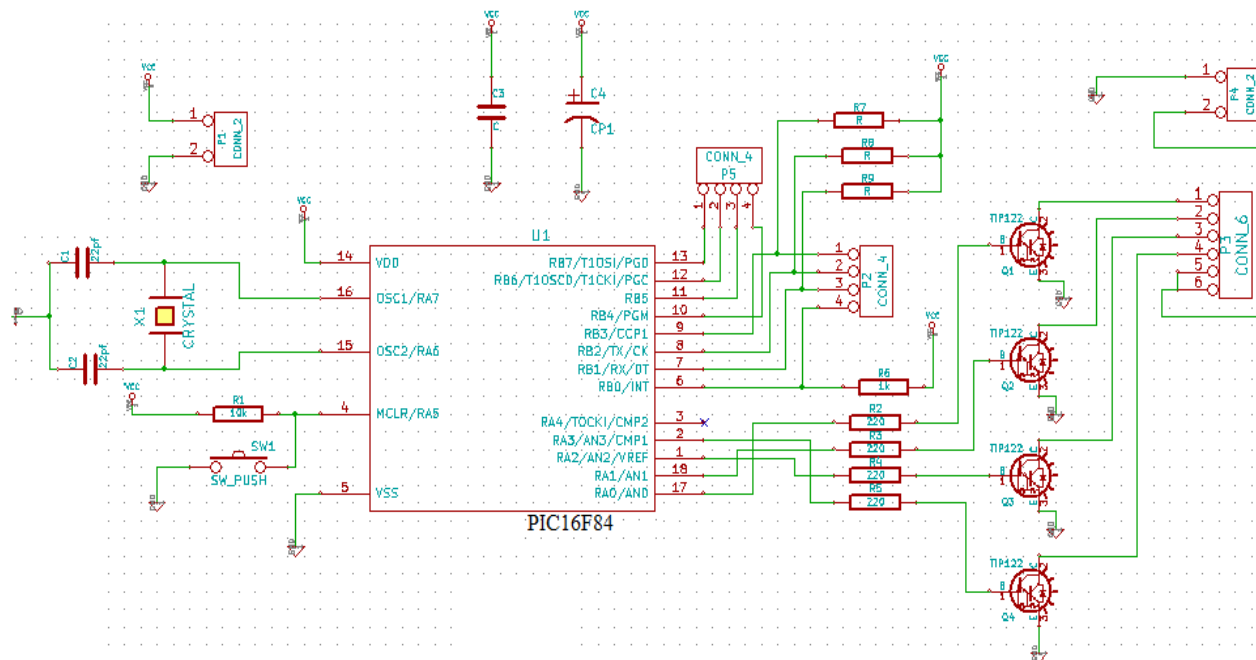


Figura 4.8: Diagrama electrónico de potencia

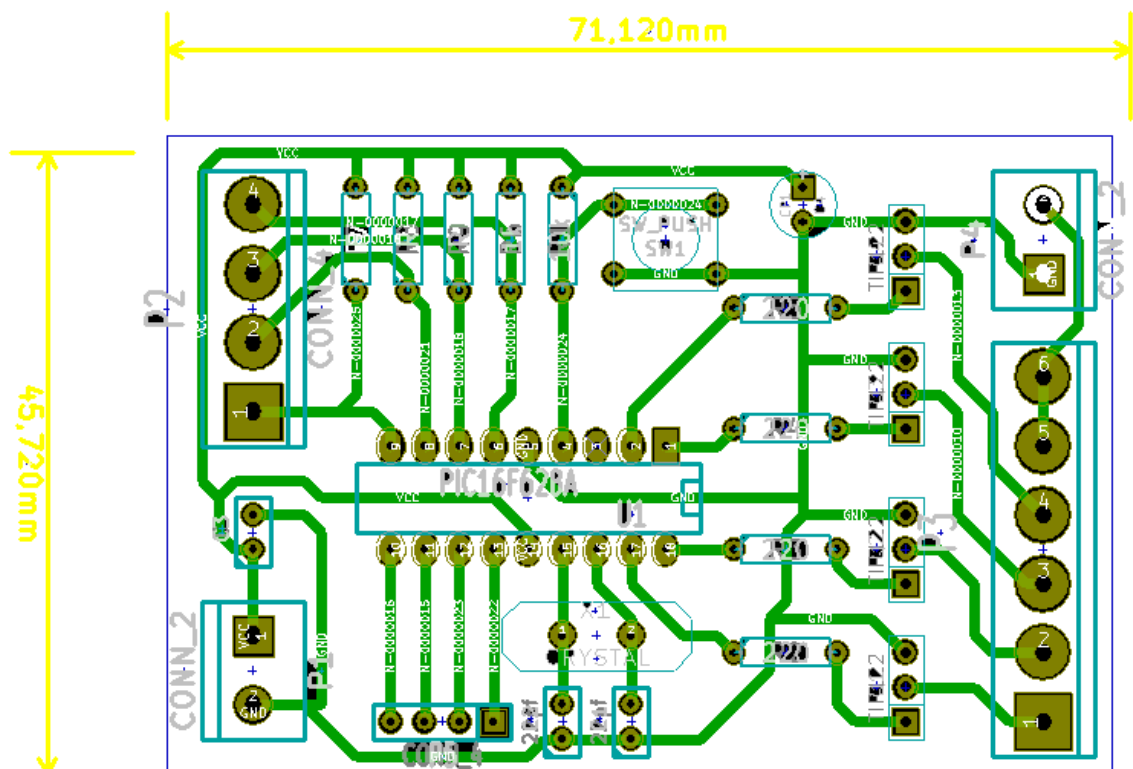


Figura 4.9: Circuito impreso de la etapa de potencia

4.5 Sistema mecánico

El diseño del sistema mecánico consiste en una plancha que soporta al sistema completo, sobre esta plataforma se montaran los tres ejes del CNC. Existen varias propuestas de diseño y se ha buscado un sistema simple por lo cual se han integrado las partes más importantes en el diseño, los planos de la mecánica y los isométricos se muestran en el apéndice E.

El conjunto de piezas mecánicas se diseñaron en Solid Works y se muestran a continuación

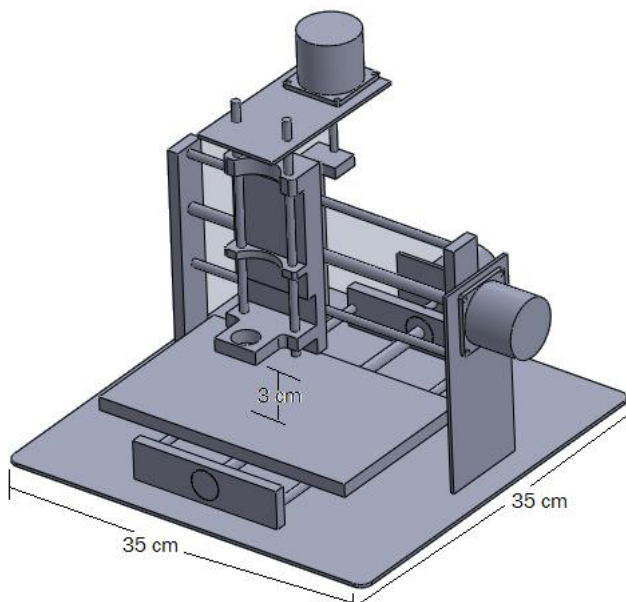


Figura 4.10: Sistema mecánico de la fresadora

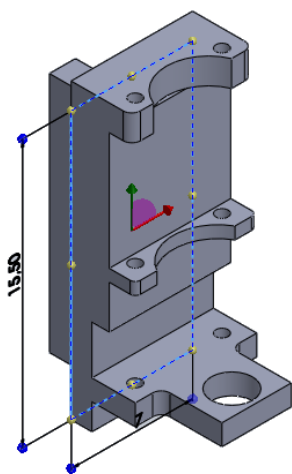


Figura 4.11: Soporte para roto tool

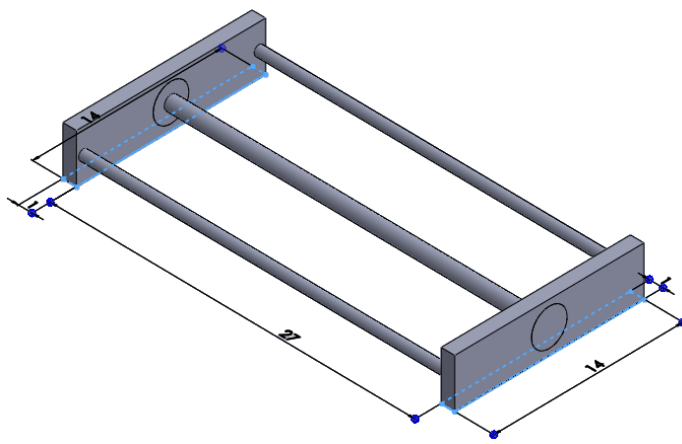


Figura 4.12: Soporte para el eje Y

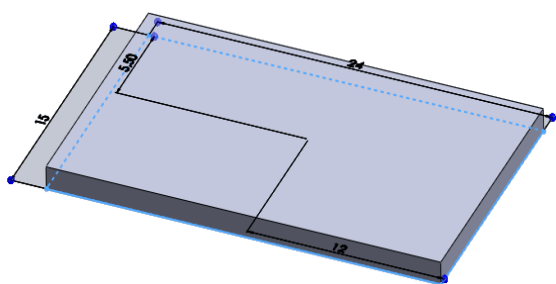


Figura 4.13: Área de trabajo del CNC

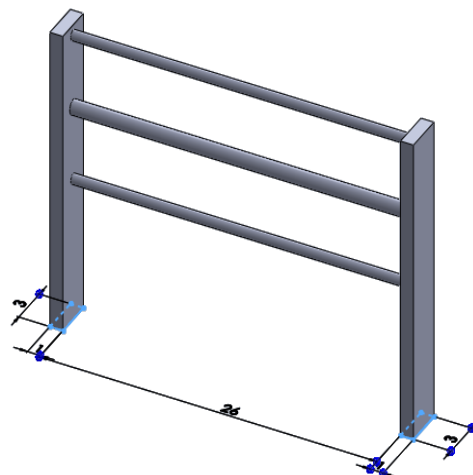


Figura 4.14: Soporte para el eje X

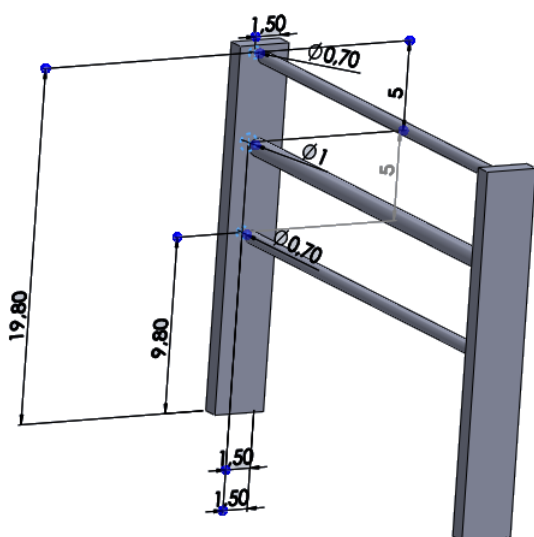


Figura 4.15: Medidas de la separación de guías del eje X

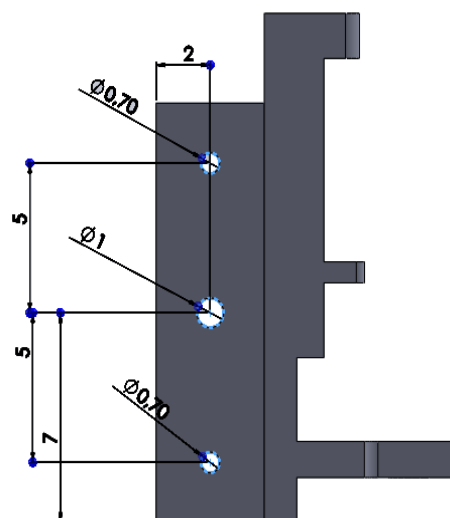


Figura 4.16: Medidas de la separación de guías del soporte

De las figuras 4.10 a 4.16 se muestran los diseños en Solid Works de la máquina CNC. El tamaño total es de 35 cm de ancho por 35 cm de largo y 24 cm de altura. El área efectiva de trabajo se muestra en la figura 4.13 y es de 24 cm por 15 cm para el eje X y Y respectivamente.

El espacio de trabajo en el eje Z es de 3 cm como se muestra en la figura (4.10), a este espacio se le reduce en tamaño que tiene el Roto Tool que se utiliza para que atraviese la boquilla de la base y sujete la broca que se utiliza como herramienta de corte.

CAPITULO V: RESULTADOS Y CONCLUSIONES.

5.1 Resultados.

En este apartado se muestran las evidencias de la ejecución del programa y las pruebas que se han hecho al sistema CNC.

Los costos totales de la placa de control y la etapa de potencia se muestran en la tabla 8

Tabla 8: Costos del sistema

Función	Descripción	Costo (Pesos mexicanos)
Control	Circuito del control implementado con PIC18F2550	\$250.00
Potencia	Etapa de potencia para motores unipolares más fuentes	\$600.00 Circuitos \$800.00 Fuentes
Mecánica	Sistema mecánico que cuenta con tres motores a pasos, un roto tool, nylamid y partes metálicas	\$7,150.00
	Taladro	\$1,200
Total		\$10,000.0

Como se puede observar en la tabla 8 el costo total es de aproximadamente \$10,000. El precio de las tarjetas controladoras mostradas en la tabla 2 y los precios mostrados en la tabla 8 de la etapa de control más la etapa de potencia son comparables.

También el costo total de la parte mecánica depende del tamaño y el material utilizado, si el material para la construcción fuera madera los costos se reducirían, pero se la dureza sería diferente y afectaría en el tiempo de vida del sistema mecánico.

Se realizó un primer prototipo del sistema mecánico el cual se muestra.



Figura 5.1: Primer prototipo

En este sistema se tienen varios problemas uno de ellos es la vibración, debido a que la base es de madera y la guía que sostiene al roto tool también existe vibración que hace que los maquinados realizados no tengan un buen acabado, además de que existe cabeceado debido principalmente a que en el momento de que la herramienta empieza a trabajar produce vibración que se transmite en la parte inferior donde se hace el devastado. En la parte posterior el sistema de guía del eje X solo está sostenido por una barra de nylon que sujeta todo el eje Z, al momento de ir trasladándose tiene una pequeña inclinación que hace que la posición establecida para el eje X siempre mantenga un error de 2 a 3mm.

A continuación se muestran las imágenes de las partes construidas:



Figura 5.2 Prueba líneas



Figura 5.3 Prueba círculos

Como se puede observar en la figura 5.2 y 5.3 los acabados no son muy buenos y la precisión de la máquina varía en varias zonas cambia la profundidad de corte.

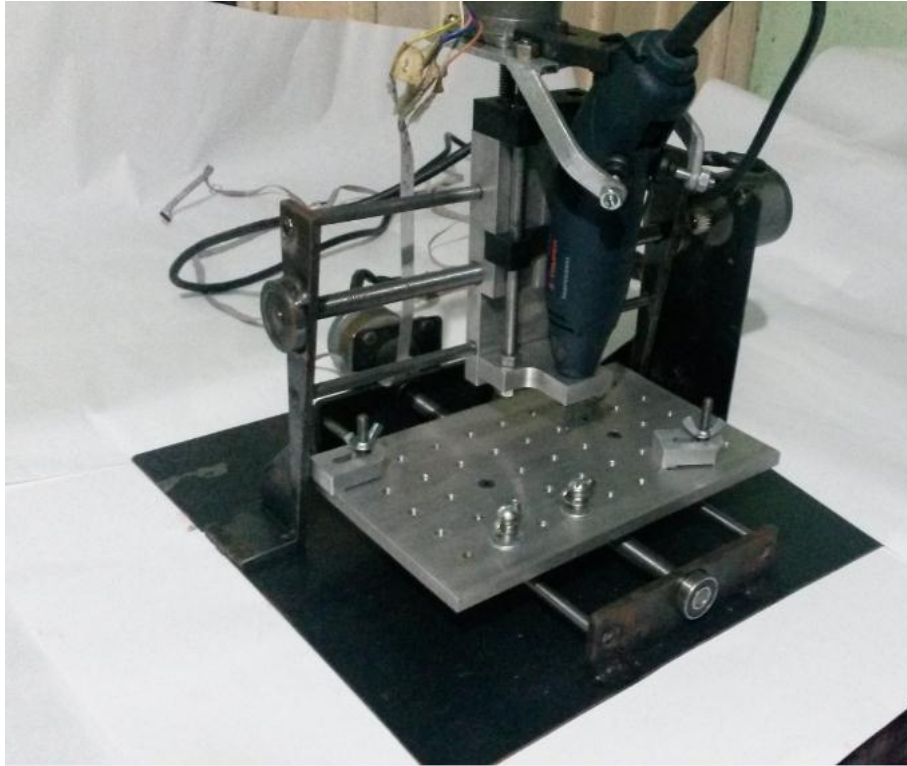


Figura 5.3.1: Prototipo final.

En la figura 5.3.1 se muestra el sistema mecánico final, en este sistema se cambió la base, el sistema guía del eje x y se añadieron soporte para el roto tool y así se evitara la vibración. Se definió el sistema de coordenadas a utilizar las cuales se muestran en la figura 5.4.

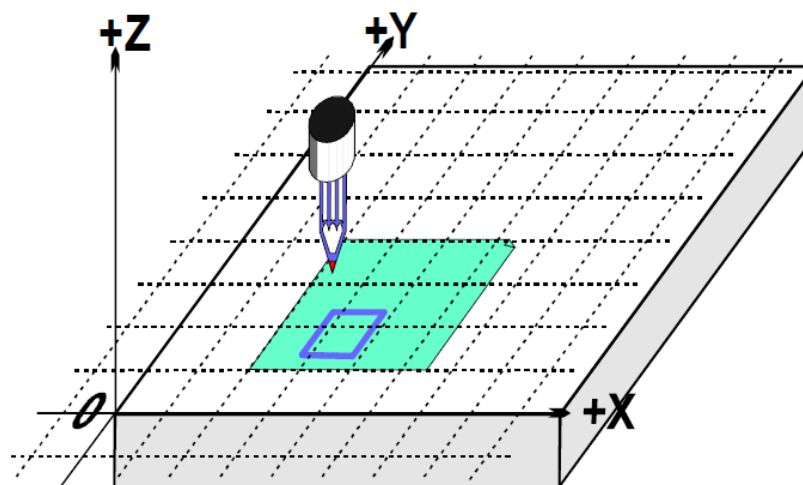


Figura 5.4: Sistema de coordenadas para la maquina CNC.

En la figura 5.4 se muestra el roto tool que se traslada de izquierda a derecha de acuerdo al valor de X asignado. La relación de numero de pasos con respecto al desplazamiento lineal

es de 0.2 cm por cada 200 pasos para paso simple y 0.2cm por cada 400 pasos en la configuración de medio paso.

El número de pasos para dar una vuelta completa se calculó sabiendo que el motor especifica que por cada paso se desplaza 1.8 grados por lo tanto para realizar una vuelta completa o desplazarse 360 grados se divide el ángulo total por el desplazamiento por paso. La relación de desplazamiento lineal se calculó haciendo girar el motor una vuelta completa y midiendo la distancia lineal. Los demás motores conservan la misma relación de desplazamiento.

Las tarjetas de control y de potencia se muestran a continuación:



Figura 5.5: Tarjeta de control

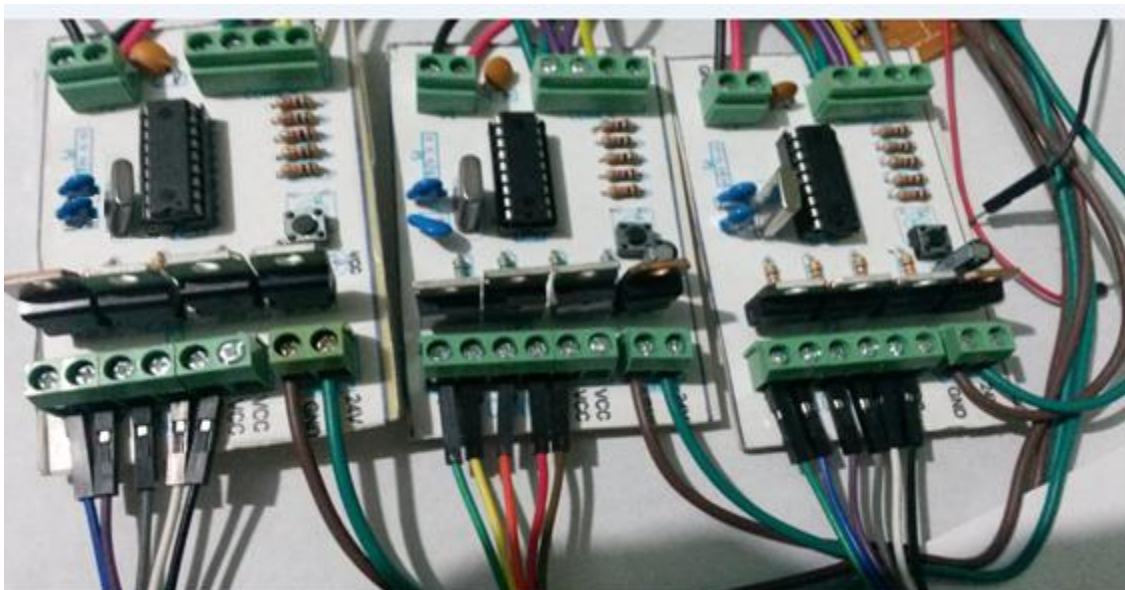


Figura 5.6: Tarjetas de potencia

La tarjeta de control de la figura 5.5 se conecta con la etapa de potencia haciendo uso de bornes de conexión, esta etapa funciona correctamente y la etapa de potencia no sufre calentamiento, esto se debe principalmente a la corriente que consumen los motores, la cual es de 0.18 A y consumen un voltaje de 24 Volts.

Con lo que respecta a la velocidad de desplazamiento para cada paso se programó un retardo de 5 ms por lo que para una vuelta completa o 200 pasos se tienen un tiempo de 1000 ms o 1 segundo por vuelta completa. Para un recorrido de 1 cm se necesitan 1000 pasos y 5 segundos ya que cada 0.2 cm da una vuelta completa el motor. El tiempo calculado puede variar ya que el retardo por paso puede variar de 1 ms a 5ms. Para tiempos más pequeños de 1ms el motor tiene problemas en su funcionamiento. Para las pruebas hechas en el CNC se utiliza un tiempo de 5ms por paso.

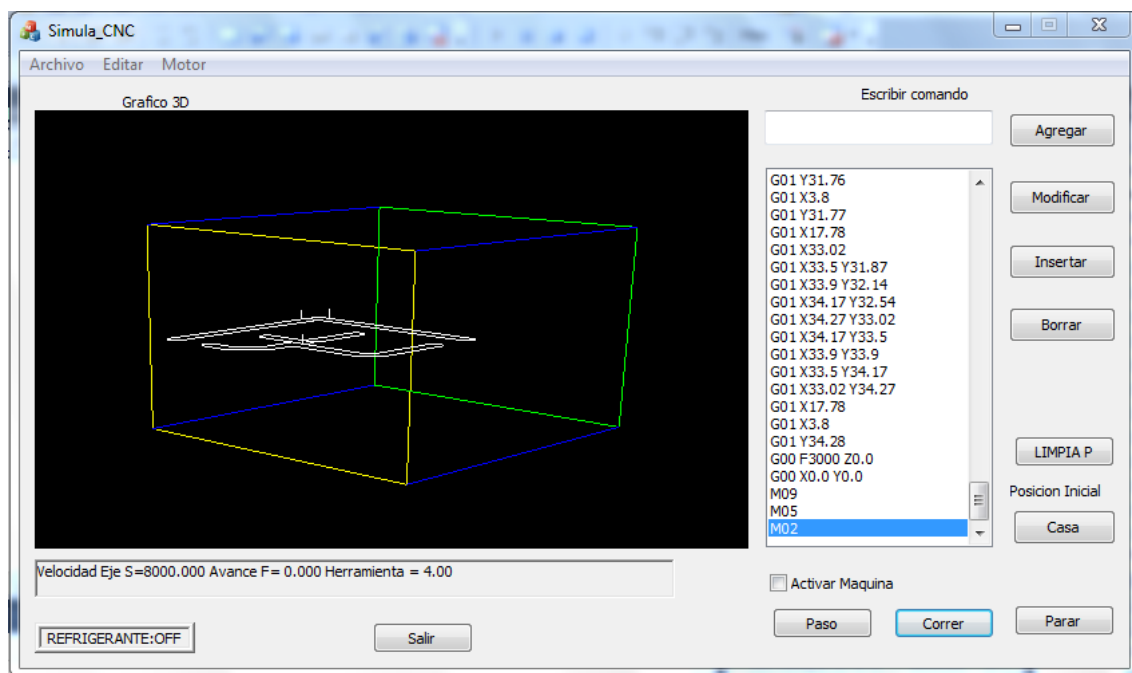


Figura 5.7: Pantalla de programación de códigos G

Con lo que respecta al software de programación que se muestra en la figura 5.7 funciona de manera adecuada comparada con el software KCAM. Sin embargo hay muchas mejoras que hacer, una de esta es poner botones de movimiento de los tres ejes como lo manejan los programas del Mach 3 y KCAM. El software funciona bajo el sistema operativo Windows 7 y demás versiones.

Al programa se le ha añadido un botón Check Box para poder simular el código G y una vez probado el código se puede activar la máquina para tener un producto terminado.

Los programas como Mach 3 y KCAM tiene la opción de simulación implementada de otra forma.

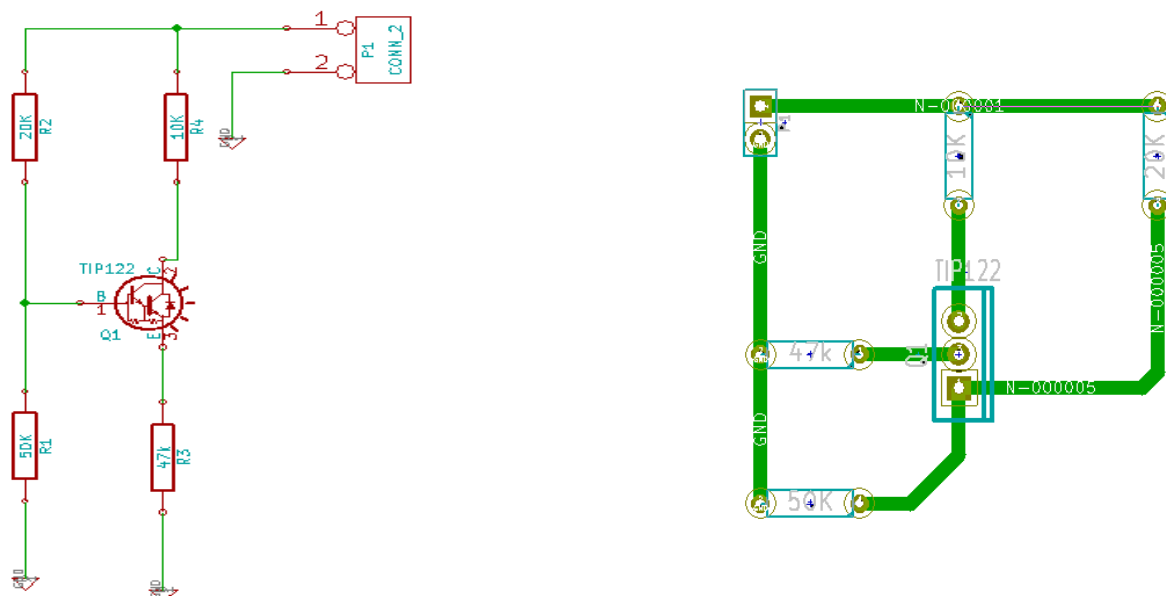


Figura 5.8: Circuito electrónico de prueba

En la figura 5.8 se muestra el circuito de prueba que se utilizó para maquinar el circuito impreso. Se trata de un amplificador basado a transistor que se utiliza en etapas de audio. La placa de circuito impreso se muestra a continuación.



Figura 5.9: Placa de circuito impreso maquinado con el CNC

En la figura 5.9 se muestra el producto final del sistema se eligió este tipo de maquinado debido a que se pueden observar la profundidad y calidad el maquinado. El impreso se desbaste usando una broca en V de 30 grados que originalmente tiene la capacidad de solo atacar pequeñas pistas del tamaño de 1 mm de la placa impresa.

También se hicieron varias pruebas con otro tipo de figuras. A continuación se muestra las imágenes de las pruebas hechas con el sistema.



Figura 5.10: Circulo en madera

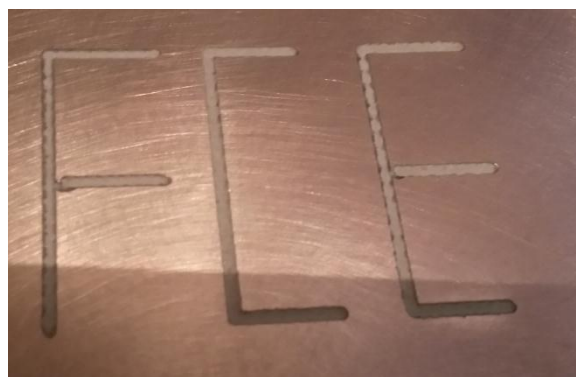


Figura 5.11: Letras en circuito impreso



Figura 5.12: Figura logo fce



Figura 5.13: Mikey Mouse

5.2 Conclusiones

- 1.- La aportación de la propuesta de tesis consiste en diseñar un software propio que no necesita del pago de licencia. El cual funciona correctamente comparado con el software KCAM y Match 3. La comunicación por puerto USB representa una mejora con respecto de los programas que usan puerto paralelo. Sin embargo, se deben implementar más funciones que ayuden al usuario final
- 2.- Las ventajas del programa con respecto a otros comerciales:
 - a) No necesitamos una licencia para usarlo es de diseño propio
 - b) Pueden incorporarse mejoras en la interfaz gráfica como incluir el uso de asistentes para facilitar la programación de códigos G.
 - c) Se pueden implementar nuevas instrucciones de código G y M.
- 3.- Las desventajas del programa con respecto a otro comercial:
 - a) El número de funciones G es limitado con respecto a los comerciales
 - b) La calidad de los productos es inferior debido principalmente a la construcción del sistema mecánico.
- 4.- El sistema mecánico cumple con la aplicación de maquinar circuitos impresos, sin embargo, tiene pequeños errores de construcción que pueden mejorarse. El error de posicionamiento del sistema es de 0.1mm, sin embargo, debido a los problemas de un motor en el eje X se midieron errores de 1mm en ese eje.
- 5.- El tiempo de desplazamiento por vuelta completa del motor puede mejorarse a un tiempo de 1segundo por cada centímetro de desplazamiento lineal.
- 6.- El sistema mecánico se mejoró del primer prototipo al segundo como se ve en las pruebas hechas en ambas etapas. La razón principal es la mejora en el diseño.
- 7.- El costo total del sistema es de \$10,000 pesos mexicanos que representa un costo reducido debido al área de trabajo que es de 24cm x 15cm x3 cm. El precio de los CNC de la tabla 2b puede aumentar debido a la electrónica, el tamaño y al software de control. En el caso presente el software es propio y no se paga por la licencia.

Bibliografía

Referencias

Software CNC USB controller

[1] <http://www.planet-cnc.com/index.php?page=shop>

Historia USB

[2] http://es.wikipedia.org/wiki/Universal_Serial_Bus

Tarjetas de control CNC

[3] http://listado.mercadolibre.com.mx/industrias-manufacturera-maquinas/controladora-cnc_DisplayType_LF

[4] Mecatrónica, W. Bolton. Ed. Alfaomega

[5] Enciclopedia Microsoft Visual Basic 6, Francisco Javier Ceballos. Ed. Alfaomega

[6] Microcontrolador PIC16F84. Desarrollo de proyectos. Enrique palacios, Fernando Remiro y Lucas J. López. Ed. Alfaomega

Software Mach 3

[7] <http://www.machsupport.com/>

Software KCam

[8] <http://www.kellyware.com/kcam/>

Router CNC y Electronica

[9] <http://listado.mercadolibre.com.mx/router-cnc>

Apéndice A

Códigos de programación para el control de motores a PASOS en PIC-C del PIC16f84A

```
#include <motor_int0.h>
void secuencia(void);
#int_EXT
void EXT_isr(void)
{ secuencia();}

int tipo=0;
int conta=1;
int conta2=1;
int conta_medio=1;
int dir=0;
int ban=2;

int paso1[5]={0,8,4,2,1};
int medio_paso1[9]={0,8,12,4,6,2,3,1,9};
int doble_paso1[5]={0,12,6,3,9};

void main()
{int pin2,pin3;
 enable_interrupts(INT_EXT);
 enable_interrupts(GLOBAL);
 set_tris_a(0x00);
 set_tris_b(0x0f);
 output_a(0x00);
 while(TRUE)
 {
  pin2=input(PIN_B2);
  pin3=input(PIN_B3);

  if((pin3==1)&&(pin2==1)){
   tipo=0; //Paso simple
  }
  if((pin3==0)&&(pin2==1)){
   tipo=1; //Medio paso
  }
  if((pin3==1)&&(pin2==0)){
   tipo=2; //Paso doble
  }
  if((pin3==0)&&(pin2==0)){
   tipo=3;
  }
 }
}

void secuencia(void)
{ dir=input(PIN_B1);
 switch(tipo)
 {
  case 0: //Paso simple
   while(!input(PIN_B0));

   if(dir==0)
   {
    if(ban==1) conta++;
    if(conta==5) conta=1;
    output_a(paso1[conta]); conta++;ban=0;
   }
   if(dir==1)
```

```

    {
        if(ban==0) conta--;
        conta--;
        if(conta==0) conta=4;
        output_a(paso1[conta]); ban=1;
    }
break;
case 1: //Medio paso
while(!input(PIN_B0));
    if(dir==0)
    {
        if(ban==1) conta_medio++;

        if(conta_medio==9) conta_medio=1;
        output_a(medio_paso1[conta_medio]);
        conta_medio++; ban=0;
    }
    if(dir==1)
    {
        if(ban==0) conta_medio--;
        conta_medio--;
        if(conta_medio==0) conta_medio=8;
        output_a(medio_paso1[conta_medio]); ban=1;
    }
break;
case 2: //Paso doble

    while(!input(PIN_B0));
    if(dir==0)
    {
        if(ban==1) conta2++;
        if(conta2==5) conta2=1;
        output_a(doble_paso1[conta2]);
        conta2++; ban=0;
    }
    if(dir==1)
    {
        if(ban==0) conta2--;
        conta2--;
        if(conta2==0) conta2=4;
        output_a(doble_paso1[conta2]); ban=1;
    }
break;

case 3:
    while(!input(PIN_B0));
    tipo=0; conta=1; conta2=1; conta_medio=1;
    dir=0; ban=2; output_a(0x00);
break;
default:
break;
}
}

```

Apéndice B

Códigos de programación de la implementación de código G

```
////////////////////////////////////

void CSimula_CNCDlg::OnClickedButtonPaso()
{
    //Codigo para ejecutar un comando CNC
    //se comenzara por los codigos de comentarios y lineas basicas
    CString str,dato,valor,a,b,cmd_prev;
    char m_dato,cad_dato[30];
    int tcad;
    int tam,tam_linea,i,j,codigo;
    cmd_prev="G0";
    tam=m_listDatos.GetCount();           //Obtener tamaño de linea
    m_listDatos.SetCurSel(indice);       //Seleccionar actual
    m_listDatos.GetText( indice, str );
    tam_linea=str.GetLength();

    for(i=0;i<tam_linea;i++)
    {
        //Leer una linea y dibujar el dato
        dato=str.GetAt(i);
        if(dato=='(') break;
        if(dato[0] >='A' & dato[0]<'Z')           //Es decir solo letras
        {                                           //0 Del if
            donde chequea datos correctos
                m_dato=dato[0];
                switch(m_dato){                   //1
                    case 'G':                     //dentro del codigo G hay
                        varias opciones
                            a="";

                            i++;
                            valor=str.GetAt(i);

                            while(valor[0]>='0' & valor[0] <='9')
                            {
                                a=a+valor;           //concatena cadena

                                j=i+1;
                                if(j>=tam_linea) break;
                                valor=str.GetAt(j);   //Leer el siguiente
                                dato debe ser un numero
                                                                    //
                                sino terminar
                                    i++;
                                }
                                i--;                     //Regresar
                                al indice correcto

                                //Tomar el entero del string a
                                for(j=0;j<=a.GetLength();j++)
                                    cad_dato[j]=a[j];

                                codigo=atoi(cad_dato);
                                //Obtiene el valor entero del codigo G

```

```

a="";

switch(codigo){      //Del codigo G

////////////////////////////////////////
case 0:
//Solo moverse sin ranurar

////////////////////////////////////////

do{
    a="";
    i++;
    if(i>=tam_linea) break;
    dato=str.GetAt(i);

    if(dato=='X' || dato == 'Y' || dato=='Z')
    {
        i++;
        valor=str.GetAt(i);

        while(valor[0]>='0' & valor[0] <='9' ||
valor=='-' || valor=='.') //Obtener el numero
        {
            a=a+valor;
            j=i+1;
            if(j>=tam_linea) {i++; break;}
            valor=str.GetAt(j);

            i++;
        }

        for(j=0;j<=a.GetLength();j++)
            cad_dato[j]=a[j];

            if(dato=='X') xp=atof(cad_dato);
            if(dato=='Y') yp=atof(cad_dato);
            if(dato=='Z') zp=atof(cad_dato);

            i--;
        }

    }while(dato[0]>='X' & dato[0]<='Z' ||dato==" ");

    //m_Show.Format("Anterior:xp = %.4lf yp = %.4lf zp
= %.4lf Actual:x = %.4lf y = %.4lf z = %.4lf",xp,yp,zp,x,y,z);
    //UpdateData(FALSE);
    i--;          //Checa siguiente codigo o sale si ya
se termino la linea

////////////////////////////////////////
//Apilar los datos para dar el efecto correcto
x=xp;
y=yp;

```

```

        z=zp;

////////////////////////////////////
        //Ejecutar codigo
        if(activa_cnc)

{Puerto.EjecutarCodigo((x/10.0),(y/10.0),(z/10.0),0);}

        break;
////////////////////////////////////
        case 1:
//Moverse ranurando/////
////////////////////////////////////

        do{
                a="";
                i++;
                if(i>=tam_linea) break;
                dato=str.GetAt(i);

                if(dato=='X' || dato == 'Y' || dato=='Z')
                {
                        i++;
                        valor=str.GetAt(i);

                        while(valor[0]>='0' & valor[0] <='9' ||
valor=='-' || valor=='.') //Obtener el numero
                        {
                                a=a+valor;
                                j=i+1;
                                if(j>=tam_linea) {i++; break;}
                                valor=str.GetAt(j);

                                i++;
                        }

                        for(j=0;j<=a.GetLength();j++)
                                cad_dato[j]=a[j];

                        if(dato=='X') x=atof(cad_dato);
                        if(dato=='Y') y=atof(cad_dato);
                        if(dato=='Z') z=atof(cad_dato);

                        i--;
                }

        }while(dato[0]>='X' & dato[0]<='Z' ||dato==" ");

        //m_Show.Format("Anterior:xp = %.4lf yp = %.4lf zp
= %.4lf Actual:x = %.4lf y = %.4lf z = %.4lf",xp,yp,zp,x,y,z);
        //UpdateData(FALSE);
        i--; //Checa siguiente codigo o sale si ya
se termino la linea

////////////////////////////////////

```

```

////////////////////////////////////
// Proceso de dibujado los datos estan
encapsulados en codigos de la Clase DrawCode
    codigos[m_cuenta].xp=xp;
    codigos[m_cuenta].yp=yp;
    codigos[m_cuenta].zp=zp;

    codigos[m_cuenta].x=x;
    codigos[m_cuenta].y=y;
    codigos[m_cuenta].z=z;

    codigos[m_cuenta].tipo=1; //Dibujar lineas

    m_cuenta++;
    Dibuja_Linea();

////////////////////////////////////
////////////////////////////////////
    xp=x;
    yp=y;
    zp=z;

////////////////////////////////////
    //Ejecutar codigo
    if(activa_cnc)

{Puerto.EjecutarCodigo((x/10.0),(y/10.0),(z/10.0),0);}

    break;

////////////////////////////////////
    // Creacion de circulos
////////////////////////////////////

////////////////////////////////////
case 2: //Hacer un circulo G02
derecha/////
    ///// Se necesitan dos puntos. Iniciar en P1
    ///// y terminar en P2 a travez de
un radio R
    ///// G02 X Y R
forma 1 o
    ///// G02 x y I J
    ///// I y J son los desplazamientos
para el centro del circulo
    ///// A partir del punto 1
    /////
    r=0.0;
    ri=0.0;
    rj=0.0;
    rk=0.0;

```

```

do{
    a="";
    i++;
    if(i>=tam_linea) break;
    dato=str.GetAt(i);

    if(dato=='X' || dato == 'Y' || dato=='Z' ||
dato=='R' || dato=='I' || dato=='J' || dato=='K')
    {
        i++;
        valor=str.GetAt(i);

        while(valor[0]>='0' & valor[0] <='9' ||
valor=='-' || valor=='.') //Obtener el numero
        {
            a=a+valor;
            j=i+1;
            if(j>=tam_linea) {i++; break;}
            valor=str.GetAt(j);

            i++;
        }

        for(j=0;j<=a.GetLength();j++)
            cad_dato[j]=a[j];

            if(dato=='X') x=atof(cad_dato);
            if(dato=='Y') y=atof(cad_dato);
            if(dato=='Z') z=atof(cad_dato);
            if(dato=='R') r=atof(cad_dato);
            if(dato=='I') ri=atof(cad_dato);
            if(dato=='J') rj=atof(cad_dato);
            if(dato=='K') rk=atof(cad_dato);

            i--;
        }

        }while(dato[0]>='X' & dato[0]<='Z' ||dato==" " ||
dato=='R' ||dato=='I' ||dato=='J' ||dato=='K');

        //m_Show.Format("Anterior:xp = %.4lf yp = %.4lf zp
= %.4lf Actual:x = %.4lf y = %.4lf z = %.4lf",xp,yp,zp,x,y,z);
        //UpdateData(FALSE);
        i--; //Checa siguiente codigo o sale si ya
se termino la linea

////////////////////////////////////
//Extraer valores i,j,k y R

////////////////////////////////////
////////////////////////////////////

```

```

// Proceso de dibujado los datos estan
encapsulados en codigos de la Clase DrawCode
codigos[m_cuenta].xp=xp; //Punto
inicial
codigos[m_cuenta].yp=yp;
codigos[m_cuenta].zp=zp;

//Punto final
codigos[m_cuenta].x=x;
codigos[m_cuenta].y=y;
codigos[m_cuenta].z=z;

codigos[m_cuenta].R=r;
codigos[m_cuenta].i=ri;
codigos[m_cuenta].j=rj;
codigos[m_cuenta].k=rk;

circulos
codigos[m_cuenta].tipo=2; //Dibujar
codigos[m_cuenta].ban=0;
codigos[m_cuenta].id_circle=id_circle;

m_cuenta++;
id_circle++; //Indice o
contador de circulos
Dibuja_Linea();
//Dibujar un circulo

////////////////////////////////////
////////////////////////////////////
xp=x;
yp=y;
zp=z;

break;

////////////////////////////////////
case 3: //Hacer un circulo G02
derecha/////
///// Se necesitan dos puntos. Iniciar en P1
///// y terminar en P2 a travez de
un radio R
///// G02 X Y R
forma 1 o
///// G02 x y I J
///// I y J son los desplazamientos
para el centro del circulo
///// A partir del punto 1
////////////////////////////////////
r=0.0;
ri=0.0;
rj=0.0;
rk=0.0;

/*x=0.0;

```

```

y=0.0;
z=0.0;*/

do{
    a="";
    i++;
        if(i>=tam_linea) break;
    dato=str.GetAt(i);

    if(dato=='X' || dato == 'Y' || dato=='Z' ||
dato=='R' || dato=='I' || dato=='J' || dato=='K')
    {
        i++;
        valor=str.GetAt(i);

        while(valor[0]>='0' & valor[0] <='9' ||
valor=='-' || valor=='.') //Obtener el numero
        {
            a=a+valor;
            j=i+1;
            if(j>=tam_linea) {i++; break;}
            valor=str.GetAt(j);

            i++;
        }

        for(j=0;j<=a.GetLength();j++)
            cad_dato[j]=a[j];

            if(dato=='X') x=atof(cad_dato);
            if(dato=='Y') y=atof(cad_dato);
            if(dato=='Z') z=atof(cad_dato);
            if(dato=='R') r=atof(cad_dato);
            if(dato=='I') ri=atof(cad_dato);
            if(dato=='J') rj=atof(cad_dato);
            if(dato=='K') rk=atof(cad_dato);

            i--;
        }

        }while(dato[0]>='X' & dato[0]<='Z' || dato==" " ||
dato=='R' || dato=='I' || dato=='J' || dato=='K');

        //m_Show.Format("Anterior: xp = %.4lf yp = %.4lf zp
= %.4lf Actual: x = %.4lf y = %.4lf z = %.4lf",xp,yp,zp,x,y,z);
        //UpdateData(FALSE);
        i--; //Checa siguiente codigo o sale si ya
se termino la linea

////////////////////////////////////
//Extraer valores i,j,k y R

////////////////////////////////////

```

```

////////////////////////////////////
// Proceso de dibujado los datos estan
encapsulados en codigos de la Clase DrawCode
codigos[m_cuenta].xp=xp; //Punto
inicial
codigos[m_cuenta].yp=yp;
codigos[m_cuenta].zp=zp;

//Punto final
codigos[m_cuenta].x=x;
codigos[m_cuenta].y=y;
codigos[m_cuenta].z=z;

codigos[m_cuenta].R=r;
codigos[m_cuenta].i=ri;
codigos[m_cuenta].j=rj;
codigos[m_cuenta].k=rk;

codigos[m_cuenta].tipo=3; //Dibujar
circulos
codigos[m_cuenta].ban=0;
codigos[m_cuenta].id_circle=id_circle;

m_cuenta++;
id_circle++; //Indice o
contador de circulos
Dibuja_Linea();
//Dibujar un circulo

////////////////////////////////////
////////////////////////////////////
xp=x;
yp=y;
zp=z;

break;

} //Del codigo G

break;

case 'S':

m_S=Extraer_Valor(tam_linea,&i,str);

m_Show.Format(_T("Velocidad Eje S=%.31f
Avance F= %.31f Herramienta = %.21f"),m_S,m_F,m_T);
UpdateData(FALSE);

```

```

        break;

        case 'F':    //Avance en pulgadas o milímetros por minuto
                    //generalmente aparece en el código

G01

                    m_F=Extraer_Valor(tam_linea,&i,str);

                    m_Show.Format(_T("Velocidad Eje S=%.3lf
Avance F= %.3lf Herramienta = %.2lf"),m_S,m_F,m_T);
                    UpdateData(FALSE);

        break;

        case 'T':

                    m_T=Extraer_Valor(tam_linea,&i,str);

                    m_Show.Format(_T("Velocidad Eje S=%.3lf
Avance F= %.3lf Herramienta = %.2lf"),m_S,m_F,m_T);
                    UpdateData(FALSE);
        break;

        case 'M':    //Códigos M prender o apagar refrigerantes

para el programa

        a="";
        i++;
        valor=str.GetAt(i);

        while(valor[0]>='0' & valor[0] <='9')
        {
            a=a+valor;
            j=i+1;
            if(j>=tam_linea) break;
            valor=str.GetAt(j);    //Leer el siguiente
                                // sino terminar

                                dato debe ser un número

            i++;
        }
        i--;    //Regresar

al índice correcto

        //Tomar el entero
        for(j=0;j<=a.GetLength();j++)
        cad_dato[j]=a[j];

        codigo=atoi(cad_dato);
        //Obtiene el valor entero del código M
        a="";
        switch(codigo){    //Del código M    //2
        //////////////////////////////////////
        case 8:
        //Encender refrigerante
        //////////////////////////////////////

```

```

        m_Coolant="REFRIGERANTE:ON";
        UpdateData(FALSE);
    break;
    case 9:
//Apagar refrigerante
////////////////////////////////////
        m_Coolant="REFRIGERANTE:OFF";
        UpdateData(FALSE);
    break;
    }//2 Para el case del codigo M

    }//1 De las opciones de la linea general del primer case

    }//0 Del primer if donde checa letras
}
//Siguiente linea sino empezar
if(indice>=tam-1)
{
    indice=0; m_cuenta=0;id_circle=0;}
else indice++;

}

```

Apéndice C

Códigos de programación para la comunicación por medio del puerto USB y el PIC18F2550

```
#include <F:\Program Files\PICC\Devices\18F2550.h>
#device adc=8 //Usa resolución de 8 bits
#fuses HSPLL,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL3,CPUDIV1,VREGEN
#use delay(clock=48000000)

#define USB_HID_DEVICE FALSE //deshabilitamos el uso de las directivas HID
#define USB_EP1_TX_ENABLE USB_ENABLE_BULK //turn on EP1(EndPoint1) for IN bulk/interrupt
transfers
#define USB_EP1_RX_ENABLE USB_ENABLE_BULK //turn on EP1(EndPoint1) for OUT bulk/interrupt
transfers
#define USB_EP1_TX_SIZE 1 //size to allocate for the tx endpoint 1 buffer
#define USB_EP1_RX_SIZE 31 //size to allocate for the rx endpoint 1 buffer

#include <F:\Program Files\PICC\Drivers\pic18_usb.h> //Microchip PIC18Fxx5x Hardware layer for CCS's PIC
USB driver
#include "usb_desc_scope.h" //Configuración del USB por Enumerador Palitroquez
#include <usb.c> //handles usb setup tokens and get descriptor reports

#use fast_io(b) //Hay que configurar manualmente el trisb

int8 recibe[31]; //declaramos variables
int8 puerto[31];
int8 envia[1];
int j,i=0;
int ban=0;
int retardo=5;

#int_TIMER0
void TIMER0_isr(void)
{
    //output_toggle(PIN_B0);

    if(usb_enumerated()) //si el PicUSB está configurado
    {
        if (usb_kbhit(1)) //si el endpoint de salida contiene datos del host
        {
            usb_get_packet(1, recibe, 31);

            if(recibe[0] == 0xff) // Debe activar BCD
            {
                output_b(recibe[1]);
            }

            if(recibe[0]==0x00)
            {
                envia[0]=read_adc (); //Hace conversión AD
                usb_put_packet(1, envia, 1, USB_DTS_TOGGLE);
            }

            if(recibe[0]==0x02)
            {

```

```

    puerto[0]=0x00;
    for(i=1;i<31;i++)
    puerto[i]=recibe[i];
    ban=1;
}
if(recibe[0]==0x03)
{
    output_bit(PIN_A3,recibe[1]); //1 Simple 0 Medio 1 Doble 0 Reset
    output_bit(PIN_A5,recibe[2]); //1      1      0      0
}
if(recibe[0]==0x04)
{
    retardo=recibe[1];
}

if(recibe[0]==0x05)
{
    if(ban==2){
        envia[0]=0x01;
        ban=0;
    }
    else{
        envia[0]=0x00;
    }

    usb_put_packet(1, envia, 1, USB_DTS_TOGGLE);
}

}

}

set_timer0(64790); //65170=1ms // 2ms = 64790
}

```

```

void main(void) {

// int8 recibe[10]; //declaramos variables
// int8 envia[1];
int8 bitX;

set_tris_b(0x00); //Todo salidas
set_tris_a(0xC7); //Todo entradas 1100 0111 0x7f
output_b(0x40);

output_high(PIN_A4); //Apaga led para indicar que no ha sido detectado por pc

setup_adc (adc_clock_div_32); //Enciende ADC
setup_adc_ports (AN0_ANALOG); //Elige terminales analógicas
set_adc_channel (0); //Elige canal a medir

//////////
//Para la interrupcion del TIMER 0 a 16 bits ///
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_16);
set_timer0(64790);
enable_interrupts(INT_TIMER0);
enable_interrupts(GLOBAL);
//////////

```

```
////////////////////////////////////
```

```
usb_init();           //inicializamos el USB
usb_task();           //habilita periférico usb e interrupciones
usb_wait_for_enumeration(); //esperamos hasta que el PicUSB sea configurado por el host

output_low(PIN_A4); //Ya fue detectado por PC pin_b7
delay_ms(500);
output_high(PIN_A4); //Ya fue detectado por PC
delay_ms(500);
output_low(PIN_A4); //Ya fue detectado por PC
delay_ms(500);
output_high(PIN_A4); //Ya fue detectado por PC
delay_ms(500);
output_low(PIN_A4); //Ya fue detectado por PC
delay_ms(500);
output_high(PIN_A4); //Ya fue detectado por PC
delay_ms(500);
j=1;
while (1)
{

    if(ban==1)
    {
        bitX=puerto[j]; //Mandar datos como llegan del usb
        output_b(bitX);
        delay_ms(retardo);

        bitX= puerto[j]&0x55; //0101_0101 14= 0000 c1 1 c1 0
        output_b(bitX);      //0000_1100 12
        delay_ms(retardo);   //0000_1110 14

                                //0000_0100
                                //0000_0010

        if(j==30) {ban=2;j=1; }
        else {j++; }
    }
}
}
```

Apéndice D

Códigos G de programación para los ejemplos de prueba

CODIGO DE CIRCUITO IMPRESO

```
( 3 - 06/10/2014 / ISO-Mill Output )  
( J:\CopperCAM\Ejercicios_coopercam\Amplifica_1.iso created 24/11/2014 at 07:13 )  
( Workpiece dimensions: 35.56 x 35.56 x 1 mm )  
G00 G90 G94 G40 G54 G80  
G00 F3000 Z0.0  
T4 M06 ( Drilling tool )  
M03 S8000  
M07  
G00 F3000 X18.56 Y12.82  
G00 F1500 Z0  
G01 F60 Z-5.0  
G01 F600 X18.76 Y12.99  
G01 X18.93 Y13.2  
G01 X19.06 Y13.44  
G01 X19.14 Y13.7  
G01 X19.17 Y13.97  
G01 X19.14 Y14.24  
G01 X19.06 Y14.5  
G01 X18.93 Y14.74  
G01 X18.76 Y14.95  
G01 X18.55 Y15.12  
G01 X18.34 Y15.24  
G01 X16.8 Y15.24  
G01 X18.55 Y15.36  
G01 X18.76 Y15.53  
G01 X18.93 Y15.74  
G01 X19.06 Y15.98  
G01 X19.14 Y16.24  
G01 X19.17 Y16.51  
G01 X19.14 Y16.78  
G01 X19.06 Y17.04  
G01 X19.03 Y17.1  
G01 Y25.4  
G01 X18.93 Y25.88  
G01 X18.66 Y26.28  
G01 X18.26 Y26.55  
G01 X17.78 Y26.65  
G01 X17.3 Y26.55  
G01 X16.9 Y26.28  
G01 X16.63 Y25.88  
G01 X16.53 Y25.4  
G01 Y17.1  
G01 X16.5 Y17.04  
G01 X16.42 Y16.78  
G01 X16.39 Y16.51  
G01 X16.42 Y16.24  
G01 X16.5 Y15.98  
G01 X16.63 Y15.74  
G01 X16.8 Y15.53  
G01 X17.01 Y15.36  
G01 X17.22 Y15.24  
G01 X17.19 Y15.22
```

G01 X10.16
G01 X9.68 Y15.12
G01 X9.28 Y14.85
G01 X9.01 Y14.45
G01 X8.91 Y13.97
G01 X9.01 Y13.49
G01 X9.28 Y13.09
G01 X9.68 Y12.82
G01 X10.16 Y12.72
G01 X16.39
G01 Y10.04
G01 X16.53
G01 Y6.87
G01 X13.45 Y3.79
G01 X10.16
G01 X9.68 Y3.69
G01 X9.28 Y3.42
G01 X9.01 Y3.02
G01 X8.91 Y2.54
G01 X9.01 Y2.06
G01 X9.28 Y1.66
G01 X9.68 Y1.39
G01 X10.16 Y1.29
G01 X13.97
G01 X14.45 Y1.39
G01 X14.85 Y1.66
G01 X18.66 Y5.47
G01 X18.93 Y5.87
G01 X19.03 Y6.35
G01 Y10.04
G01 X19.17
G01 Y10.18
G01 X30.48
G01 X30.96 Y10.28
G01 X31.36 Y10.55
G01 X33.9 Y13.09
G01 X34.17 Y13.49
G01 X34.27 Y13.97
G01 Y25.4
G01 X34.17 Y25.88
G01 X33.9 Y26.28
G01 X33.5 Y26.55
G01 X33.02 Y26.65
G01 X32.54 Y26.55
G01 X32.14 Y26.28
G01 X31.87 Y25.88
G01 X31.77 Y25.4
G01 Y14.49
G01 X29.96 Y12.68
G01 X19.17
G01 Y12.82
G01 X16.2
G00 F3000 Z0.0
G00 F3000 X1.3 Y30.23
G00 F1500 Z0
G01 F60 Z-5.0
G01 F600 X1.29 Y30.48
G01 Y30.36
G01 Y13.97

G01 Y2.54
G01 X1.39 Y2.06
G01 X1.66 Y1.66
G01 X2.06 Y1.39
G01 X2.54 Y1.29
G01 X3.02 Y1.39
G01 X3.42 Y1.66
G01 X3.69 Y2.06
G01 X3.79 Y2.54
G01 Y13.97
G01 Y30.36
G01 X3.8 Y30.48
G01 X3.78 Y30.73
G01 X3.71 Y30.96
G01 X3.59 Y31.18
G01 X3.43 Y31.37
G01 X3.24 Y31.53
G01 X3.02 Y31.65
G01 X2.79 Y31.72
G01 X2.54 Y31.74
G01 X2.29 Y31.72
G01 X2.06 Y31.65
G01 X1.84 Y31.53
G01 X1.65 Y31.37
G01 X1.49 Y31.18
G01 X1.37 Y30.96
G01 X1.3 Y30.73
G01 X1.28 Y30.48
G01 X1.29 Y30.36
G01 X1.3 Y30.23
G00 F3000 Z0
G00 F3000 X3.8 Y34.28
G00 F1500 Z0
G01 F60 Z-5.0
G01 F600 X1.28
G01 Y31.76
G01 X3.8
G01 Y31.77
G01 X17.78
G01 X33.02
G01 X33.5 Y31.87
G01 X33.9 Y32.14
G01 X34.17 Y32.54
G01 X34.27 Y33.02
G01 X34.17 Y33.5
G01 X33.9 Y33.9
G01 X33.5 Y34.17
G01 X33.02 Y34.27
G01 X17.78
G01 X3.8
G01 Y34.28
G00 F3000 Z0.0
G00 X0.0 Y0.0
M09
M05
M02

CODIGO DE CÍRCULO

%
 (Encabezamiento)
 (Generado por gcodetools desde Inkscape.)
 (Usando encabezamiento por defecto. Para generar tu propio encabezamiento crea un archivo "header" en el directorio de salida.)
 M3
 (Fin Encabezamiento.)
 G21 (Todas las unidades en mm)

G00 Z0.000000
 G00 X18.679583 Y36.015000

G01 Z-5.000000 F100.0
 G03 X10.394168 Y34.094728 Z-5.000000 I-0.080783 J-18.486157 F100.000000
 G03 X3.501319 Y28.042222 Z-5.000000 I7.896860 J-15.944464
 G03 X1.493123 Y23.704808 Z-5.000000 I12.793973 J-8.557131
 G03 X0.749653 Y18.384931 Z-5.000000 I18.661378 J-5.319878
 G03 X0.957451 Y15.034719 Z-5.000000 I27.110713 J0.000000
 G03 X1.437569 Y12.890417 Z-5.000000 I11.329859 J1.410904
 G03 X3.414412 Y8.802015 Z-5.000000 I16.183091 J5.302795
 G03 X6.473472 Y5.261583 Z-5.000000 I15.232353 J10.069515
 G03 X10.217055 Y2.690064 Z-5.000000 I11.874243 J13.275677
 G03 X14.428611 Y1.187028 Z-5.000000 I7.798644 J15.200076
 G03 X23.629195 Y1.482962 Z-5.000000 I4.005229 J18.648470
 G03 X31.220834 Y5.940694 Z-5.000000 I-4.751805 J16.785700
 G03 X33.849979 Y9.254588 Z-5.000000 I-13.145511 J13.129154
 G03 X35.630556 Y13.022694 Z-5.000000 I-15.398017 J9.580909
 G03 X36.091569 Y15.032573 Z-5.000000 I-9.706437 J3.284212
 G03 X36.300834 Y18.384931 Z-5.000000 I-26.747262 J3.352358
 G03 X36.091568 Y21.737337 Z-5.000000 I-26.957239 J0.000000
 G03 X35.630556 Y23.747153 Z-5.000000 I-10.166342 J-1.274179
 G03 X33.850339 Y27.519882 Z-5.000000 I-17.277001 J-5.846028
 G03 X31.220834 Y30.837986 Z-5.000000 I-15.723492 J-9.759472
 G03 X25.221960 Y34.764982 Z-5.000000 I-12.376691 J-12.361181
 G03 X18.679583 Y36.015000 Z-5.000000 I-6.621712 J-16.911055
 G01 X18.679583 Y36.015000 Z-5.000000
 G00 Z0.000000
 M30

CODIGO DE LETRAS

(PROGRAMA 1)
 G00 Z0.0
 G00 X5.0 Y10.0
 G01 Z-5.0
 G01 X5.0 Y40.0
 G01 X15.0 Y40.0
 G00 Z0.0
 G00 X15.0 Y25.0
 G01 Z-5.0
 G01 X5.0 Y25.0
 G00 Z0.0
 G00 X30.0 Y40.0
 G01 Z-5.0
 G01 X20.0 Y40.0
 G01 X20.0 Y10.0
 G01 X30.0 Y10.0

G00 Z0.0
 G00 X45.0 Y10.0
 G01 Z-5.0
 G01 X35.0 Y10.0
 G01 X35.0 Y40.0
 G01 X45 Y40.0
 G00 Z0.0
 G00 X45.0 Y25.0
 G01 Z-5.0
 G01 X35.0 Y25.0
 G00 Z0.0
 G00 X0 Y0
 M30

CODIGO DE FIGURA CIRCULAR LOGO FCE

(Fin Encabezamiento.)
 G21 (Todas las unidades en mm)
 G00 Z0.000000
 G00 X24.748408 Y41.993458
 G01 Z-5.000000 F100.0
 G03 X22.206177 Y41.929787 Z-5.000000 I0.007330 J-51.077402 F150.000000
 G03 X20.735560 Y41.772971 Z-5.000000 I0.656029 J-13.126379
 G03 X18.501263 Y41.335818 Z-5.000000 I6.002898 J-36.609279
 G03 X16.687435 Y40.846930 Z-5.000000 I5.298179 J-23.265977
 G03 X7.809383 Y35.971323 Z-5.000000 I7.614418 J-24.386068
 G03 X2.479310 Y28.834846 Z-5.000000 I12.745947 J-15.078353
 G03 X1.315601 Y25.199119 Z-5.000000 I16.801147 J-7.381744
 G03 X0.918269 Y21.347138 Z-5.000000 I18.473031 J-3.851981
 G03 X1.316040 Y17.499691 Z-5.000000 I18.806111 J0.000000
 G03 X2.479310 Y13.868249 Z-5.000000 I17.983366 J3.758625
 G03 X9.320097 Y5.489102 Z-5.000000 I18.517639 J8.135910
 G03 X20.735560 Y0.956582 Z-5.000000 I15.070786 J21.315237
 G03 X22.025295 Y0.779157 Z-5.000000 I24.127026 J170.607091
 G03 X22.561185 Y0.718457 Z-5.000000 I1.774449 J13.269890
 G03 X25.602048 Y0.633319 Z-5.000000 I2.177990 J23.443249
 G03 X28.928824 Y0.947763 Z-5.000000 I-1.063999 J29.012621
 G03 X33.244776 Y1.969280 Z-5.000000 I-4.166551 J27.232155
 G03 X37.298477 Y3.664152 Z-5.000000 I-8.156655 J25.203778
 G03 X40.204190 Y5.396082 Z-5.000000 I-13.342201 J25.688083
 G03 X40.394102 Y5.754360 Z-5.000000 I-0.242997 J0.358278
 G01 X40.274551 Y5.844973 Z-5.000000
 G03 X38.639033 Y5.260471 Z-5.000000 I3.376265 J-12.027709
 G02 X34.461173 Y3.938617 Z-5.000000 I-8.278605 J18.902192
 G02 X29.793130 Y3.470124 Z-5.000000 I-4.668044 J23.021845
 G02 X25.208868 Y3.914760 Z-5.000000 I-0.000000 J23.854532
 G02 X21.079519 Y5.181096 Z-5.000000 I3.990747 J20.379069
 G02 X13.811234 Y10.239589 Z-5.000000 I8.453380 J19.897179
 G02 X9.817088 Y17.104985 Z-5.000000 I12.690123 J11.977405
 G01 X9.552505 Y17.942832 Z-5.000000
 G01 X18.883477 Y17.969291 Z-5.000000
 G01 X28.205630 Y18.004568 Z-5.000000
 G01 X28.717158 Y17.598874 Z-5.000000
 G03 X33.775547 Y16.443483 Z-5.000000 I3.719929 J4.635431
 G03 X37.527783 Y19.636166 Z-5.000000 I-1.206286 J5.218971
 G03 X37.864748 Y21.351548 Z-5.000000 I-4.197743 J1.715382
 G03 X37.527783 Y23.066930 Z-5.000000 I-4.534709 J0.000000
 G03 X33.735794 Y26.265705 Z-5.000000 I-4.952349 J-2.023768
 G03 X28.637783 Y25.042485 Z-5.000000 I-1.296084 J-5.833408

G01 X28.082158 Y24.583874 Z-5.000000
 G01 X18.874658 Y24.610332 Z-5.000000
 G01 X9.667158 Y24.645610 Z-5.000000
 G01 X9.922921 Y25.483457 Z-5.000000
 G02 X10.208984 Y26.306777 Z-5.000000 I11.237842 J-3.443239
 G02 X10.716671 Y27.511930 Z-5.000000 I22.391271 J-8.723119
 G02 X16.641298 Y34.781279 Z-5.000000 I16.740824 J-7.595013
 G02 X26.362366 Y38.897832 Z-5.000000 I13.711883 J-18.843814
 G02 X29.898966 Y39.202899 Z-5.000000 I3.536600 J-20.347156
 G02 X33.435560 Y38.897832 Z-5.000000 I-0.000000 J-20.652156
 G02 X36.232081 Y38.243770 Z-5.000000 I-4.267104 J-24.549948
 G02 X38.744866 Y37.327971 Z-5.000000 I-6.110200 J-20.670499
 G03 X40.270171 Y36.792129 Z-5.000000 I4.389372 J10.055744
 G01 X40.394102 Y36.886999 Z-5.000000
 G03 X40.219760 Y37.223065 Z-5.000000 I-0.411076 J-0.000000
 G03 X37.554241 Y38.880194 Z-5.000000 I-15.732700 J-22.333989
 G03 X33.476783 Y40.648311 Z-5.000000 I-11.881197 J-21.813674
 G03 X28.822991 Y41.772971 Z-5.000000 I-9.130874 J-27.592207
 G03 X27.284035 Y41.930713 Z-5.000000 I-2.269873 J-14.559279
 G03 X24.748408 Y41.993458 Z-5.000000 I-2.528313 J-50.907627
 G01 X24.748408 Y41.993458 Z-5.000000
 G00 Z0.000000
 G00 X0.0 Y0.0
 M30

CODIGO DE MIKEY MOUSE

%
 (Encabezamiento)
 (Generado por gcodetools desde Inkscape.)
 (Usando encabezamiento por defecto. Para generar tu propio encabezamiento crea un archivo "header" en el directorio de salida.)
 M3
 (Fin Encabezamiento.)
 G21 (Todas las unidades en mm)

G00 Z0.000000
 G00 X10.909653 Y49.217709

G01 Z-5.000000 F100.0
 G03 X8.738622 Y49.136604 Z-5.000000 I-0.000574 J-29.082657 F400.000000
 G03 X8.131528 Y49.006042 Z-5.000000 I0.169503 J-2.264886
 G03 X3.734076 Y46.118103 Z-5.000000 I3.907300 J-10.741587
 G03 X0.917222 Y41.377222 Z-5.000000 I9.124053 J-8.628446
 G03 X0.487409 Y39.534300 Z-5.000000 I10.521319 J-3.425405
 G03 X0.335139 Y37.373195 Z-5.000000 I14.665684 J-2.119250
 G03 X0.413820 Y35.913902 Z-5.000000 I14.323021 J0.040489
 G03 X0.582083 Y35.062500 Z-5.000000 I4.790136 J0.504352
 G03 X2.608120 Y31.879285 Z-5.000000 I6.517358 J1.911767
 G03 X6.129514 Y30.176528 Z-5.000000 I4.811871 J5.458616
 G03 X8.161596 Y29.979243 Z-5.000000 I2.223597 J12.339497
 G03 X9.622014 Y30.123611 Z-5.000000 I0.098693 J6.460564
 G01 X9.983611 Y30.202986 Z-5.000000
 G01 X9.366250 Y29.435695 Z-5.000000
 G03 X7.375694 Y26.280190 Z-5.000000 I12.959300 J-10.380583
 G03 X6.103056 Y22.697639 Z-5.000000 I15.753505 J-7.613472
 G03 X5.863767 Y21.350286 Z-5.000000 I11.748045 J-2.781363
 G03 X5.723819 Y19.469722 Z-5.000000 I20.913760 J-2.501851
 G02 X5.618971 Y18.099961 Z-5.000000 I-14.646248 J0.432209

G02 X5.503333 Y17.741111 Z-5.000000 I-0.998651 J0.123753
 G03 X5.121193 Y16.702074 Z-5.000000 I3.299416 J-1.803261
 G03 X4.956528 Y15.104097 Z-5.000000 I7.724356 J-1.603436
 G03 X5.022040 Y13.977978 Z-5.000000 I9.589076 J-0.007119
 G03 X5.177014 Y13.252028 Z-5.000000 I3.954152 J0.464605
 G03 X6.319570 Y10.812090 Z-5.000000 I9.602774 J3.009231
 G03 X8.643056 Y7.872139 Z-5.000000 I16.332211 J10.519466
 G03 X10.136769 Y6.461702 Z-5.000000 I15.230637 J14.633729
 G03 X12.082639 Y4.926472 Z-5.000000 I20.104571 J23.481319
 G02 X12.903747 Y4.254155 Z-5.000000 I-5.361743 J-7.385923
 G02 X14.358056 Y2.827416 Z-5.000000 I-21.867070 J-23.744201
 G03 X16.779915 Y0.702655 Z-5.000000 I13.358955 J12.784260
 G03 X17.921111 Y0.216861 Z-5.000000 I1.605833 J2.189008
 G03 X18.274209 Y0.183302 Z-5.000000 I0.391337 J2.243190
 G03 X19.014722 Y0.199363 Z-5.000000 I0.161666 J9.625482
 G03 X20.229607 Y0.443950 Z-5.000000 I-0.265644 J4.458985
 G03 X21.978056 Y1.240029 Z-5.000000 I-3.565107 J10.148256
 G02 X22.895838 Y1.758915 Z-5.000000 I121.720370 J-214.222466
 G02 X23.794861 Y2.263085 Z-5.000000 I189.709875 J-337.232462
 G03 X24.695063 Y2.776918 Z-5.000000 I-28.533623 J51.034536
 G03 X24.862014 Y2.880474 Z-5.000000 I-1.787644 J3.068341
 G02 X25.033068 Y2.974573 Z-5.000000 I0.674461 J-1.023519
 G02 X26.590625 Y3.665391 Z-5.000000 I60.820459 J-135.028056
 G03 X33.245637 Y7.651532 Z-5.000000 I-12.160696 J27.851204
 G03 X35.321875 Y10.544557 Z-5.000000 I-3.979637 J5.047606
 G03 X35.549271 Y11.749952 Z-5.000000 I-3.734041 J1.328568
 G03 X35.401250 Y13.137474 Z-5.000000 I-5.092831 J0.158352
 G02 X35.374519 Y13.563329 Z-5.000000 I1.150526 J0.285987
 G02 X35.568820 Y14.522141 Z-5.000000 I5.952992 J-0.707266
 G03 X35.913990 Y15.898888 Z-5.000000 I-18.947412 J5.482029
 G03 X36.142084 Y17.238516 Z-5.000000 I-16.208020 J3.448916
 G03 X36.250690 Y19.681311 Z-5.000000 I-14.591034 J1.872529
 G03 X35.939236 Y22.494905 Z-5.000000 I-19.805349 J-0.768339
 G03 X35.564677 Y24.224088 Z-5.000000 I-27.543510 J-5.061054
 G03 X35.233681 Y25.299488 Z-5.000000 I-10.173024 J-2.542507
 G01 X35.074931 Y25.705182 Z-5.000000
 G01 X35.374792 Y25.669905 Z-5.000000
 G03 X37.762467 Y25.418153 Z-5.000000 I4.836626 J34.423162
 G03 X38.823195 Y25.431780 Z-5.000000 I0.448812 J6.354766
 G03 X41.669286 Y26.235445 Z-5.000000 I-0.797698 J8.266351
 G03 X44.070764 Y27.998238 Z-5.000000 I-3.670740 J7.517882
 G03 X45.707953 Y30.288903 Z-5.000000 I-7.424987 J7.037200
 G03 X46.654861 Y32.919488 Z-5.000000 I-9.069595 J4.750421
 G03 X46.934282 Y35.419505 Z-5.000000 I-12.672988 J2.682052
 G03 X46.716597 Y37.602613 Z-5.000000 I-9.690022 J0.136185
 G03 X44.099245 Y42.883483 Z-5.000000 I-11.766233 J-2.542634
 G03 X39.978542 Y45.487196 Z-5.000000 I-6.271705 J-5.363138
 G03 X38.359360 Y45.703843 Z-5.000000 I-1.633691 J-6.050865
 G03 X36.521320 Y45.496002 Z-5.000000 I-0.018672 J-8.066130
 G03 X32.928074 Y43.453695 Z-5.000000 I1.633563 J-7.056239
 G03 X30.462361 Y39.172460 Z-5.000000 I7.436434 J-7.133565
 G03 X30.031972 Y36.616829 Z-5.000000 I10.983078 J-3.163697
 G03 X30.180139 Y33.968988 Z-5.000000 I12.158014 J-0.647732
 G03 X30.453834 Y32.479558 Z-5.000000 I44.706405 J7.445303
 G03 X30.532917 Y32.249196 Z-5.000000 I0.905366 J0.182055
 G01 X30.505212 Y32.176260 Z-5.000000
 G01 X30.400625 Y32.196280 Z-5.000000
 G03 X29.156584 Y33.149003 Z-5.000000 I-5.076664 J-5.340394
 G03 X26.969861 Y34.286488 Z-5.000000 I-9.523321 J-15.637170

G03 X25.912973 Y34.687552 Z-5.000000 I-4.567629 J-10.443581
 G03 X24.623889 Y35.044960 Z-5.000000 I-4.835053 J-14.935431
 G03 X23.174502 Y35.281122 Z-5.000000 I-2.502326 J-10.791694
 G03 X21.175486 Y35.371280 Z-5.000000 I-1.898147 J-19.880073
 G03 X19.452861 Y35.299359 Z-5.000000 I0.118958 J-23.515365
 G03 X18.406181 Y35.159613 Z-5.000000 I0.761346 J-9.691967
 G02 X17.382669 Y34.976157 Z-5.000000 I-9.962908 J52.636889
 G01 X17.374306 Y34.983224 Z-5.000000
 G01 X17.377682 Y34.999690 Z-5.000000
 G02 X17.691806 Y35.732877 Z-5.000000 I1116.401931 J-477.873603
 G03 X18.361702 Y37.810199 Z-5.000000 I-10.782642 J4.623870
 G03 X18.679583 Y40.160238 Z-5.000000 I-13.738840 J3.054922
 G03 X18.378940 Y43.350606 Z-5.000000 I-11.146586 J0.558953
 G03 X17.268472 Y45.981071 Z-5.000000 I-8.631162 J-2.094071
 G03 X16.027303 Y47.479270 Z-5.000000 I-6.308642 J-3.963120
 G03 X14.402153 Y48.635724 Z-5.000000 I-5.251224 J-5.659351
 G03 X13.250097 Y49.025663 Z-5.000000 I-1.764814 J-3.317240
 G03 X10.909653 Y49.217807 Z-5.000000 I-2.340714 J-14.161294
 G01 X10.909653 Y49.217709 Z-5.000000
 G00 Z0.000000

 G00 Z0.000000
 G00 X10.971389 Y49.129514

 G00 Z0.000000
 G00 X20.743333 Y34.374584

 G01 Z-5.000000 F100.0
 G03 X19.851383 Y34.300965 Z-5.000000 I0.053867 J-6.092856 F400.000000
 G03 X19.288125 Y34.145278 Z-5.000000 I0.399370 J-2.541616
 G03 X18.149271 Y33.550152 Z-5.000000 I2.455604 J-6.086368
 G03 X16.756945 Y32.496042 Z-5.000000 I6.993691 J-10.684237
 G03 X13.817871 Y28.986499 Z-5.000000 I9.602442 J-11.027021
 G03 X12.400139 Y25.228820 Z-5.000000 I9.662815 J-5.791964
 G03 X12.193655 Y22.296232 Z-5.000000 I13.019774 J-2.390288
 G03 X12.638264 Y19.407986 Z-5.000000 I13.090138 J0.536718
 G01 X12.596925 Y19.320311 Z-5.000000
 G02 X12.426597 Y19.310972 Z-5.000000 I-0.097190 J0.214677
 G03 X11.605890 Y19.547026 Z-5.000000 I-3.732169 J-11.431230
 G03 X10.592153 Y19.769583 Z-5.000000 I-4.419577 J-17.710854
 G03 X9.808223 Y19.855576 Z-5.000000 I-0.939095 J-4.944759
 G03 X9.101667 Y19.813681 Z-5.000000 I-0.121308 J-3.933034
 G03 X6.890580 Y18.693273 Z-5.000000 I0.586507 J-3.899408
 G03 X5.653264 Y16.347639 Z-5.000000 I3.218071 J-3.196682
 G03 X5.726908 Y13.838937 Z-5.000000 I5.696844 J-1.088198
 G03 X7.073194 Y10.747306 Z-5.000000 I10.057303 J2.540625
 G03 X8.370207 Y9.069638 Z-5.000000 I10.588916 J6.846140
 G03 X10.398125 Y7.113695 Z-5.000000 I16.986646 J15.582480
 G03 X13.111396 Y4.949955 Z-5.000000 I34.147341 J40.036775
 G03 X13.493750 Y4.811806 Z-5.000000 I0.419731 J0.563493
 G01 X13.642929 Y4.961584 Z-5.000000
 G03 X13.449653 Y5.649639 Z-5.000000 I-1.662194 J-0.095739
 G02 X13.235773 Y6.091396 Z-5.000000 I5.515022 J2.942789
 G01 X13.246806 Y6.108250 Z-5.000000
 G01 X13.275955 Y6.091679 Z-5.000000
 G02 X13.537847 Y5.623194 Z-5.000000 I-8.476331 J-5.045880
 G03 X13.808655 Y5.151526 Z-5.000000 I6.226122 J3.261141
 G03 X14.305139 Y4.388472 Z-5.000000 I22.768151 J14.271145

G03 X16.415974 Y2.058285 Z-5.000000 I9.189436 J6.203229
G03 X18.115139 Y1.231111 Z-5.000000 I2.543762 J3.066563
G03 X19.347771 Y1.130904 Z-5.000000 I0.992372 J4.575714
G03 X20.522847 Y1.354583 Z-5.000000 I-0.225106 J4.380990
G03 X22.148661 Y2.159705 Z-5.000000 I-2.151666 J6.389042
G03 X23.759583 Y3.541806 Z-5.000000 I-5.213023 J7.705959
G02 X24.012113 Y3.809184 Z-5.000000 I14.235902 J-13.192378
G01 X24.015347 Y3.806389 Z-5.000000
G03 X23.938796 Y3.546805 Z-5.000000 I0.294388 J-0.227894
G01 X24.050625 Y3.400694 Z-5.000000
G03 X24.270125 Y3.393231 Z-5.000000 I0.121177 J0.332328
G03 X25.250070 Y3.753472 Z-5.000000 I-2.056749 J7.107828
G02 X25.409092 Y3.820073 Z-5.000000 I1.342872 J-2.983207
G02 X26.035000 Y4.053333 Z-5.000000 I8.927489 J-22.998808
G03 X28.900248 Y5.290021 Z-5.000000 I-8.179090 J22.887457
G03 X31.176736 Y6.672694 Z-5.000000 I-8.172367 J16.020701
G03 X32.870469 Y8.103864 Z-5.000000 I-7.904108 J11.072012
G03 X34.034236 Y9.521389 Z-5.000000 I-6.466028 J6.494998
G03 X34.355286 Y10.059621 Z-5.000000 I-5.285778 J3.517773
G03 X34.651597 Y10.676750 Z-5.000000 I-7.406287 J3.935796
G03 X34.852485 Y11.323099 Z-5.000000 I-2.950714 J1.271485
G03 X34.925000 Y12.061389 Z-5.000000 I-3.722063 J0.738290
G03 X34.859448 Y12.801546 Z-5.000000 I-4.211353 J0.000000
G03 X34.713334 Y13.252028 Z-5.000000 I-1.707167 J-0.304783
G03 X33.576622 Y14.510130 Z-5.000000 I-2.722679 J-1.317407
G03 X31.820556 Y15.033542 Z-5.000000 I-1.852888 J-3.008973
G02 X31.360488 Y15.191313 Z-5.000000 I0.022344 J0.814828
G01 X31.326667 Y15.395139 Z-5.000000
G03 X31.581895 Y15.883379 Z-5.000000 I-1.785272 J1.244081
G03 X32.199792 Y17.749931 Z-5.000000 I-26.294621 J9.740038
G03 X32.793503 Y20.807753 Z-5.000000 I-16.123437 J4.717097
G03 X32.817153 Y23.967639 Z-5.000000 I-17.217044 J1.708893
G03 X32.633626 Y25.614472 Z-5.000000 I-30.556490 J-2.571651
G03 X32.508472 Y26.207778 Z-5.000000 I-4.300365 J-0.597275
G03 X31.974125 Y27.917861 Z-5.000000 I-34.357533 J-9.797139
G03 X31.635347 Y28.756597 Z-5.000000 I-7.657972 J-2.605372
G03 X30.862712 Y30.127315 Z-5.000000 I-9.699501 J-4.564223
G03 X30.047847 Y31.146667 Z-5.000000 I-6.208447 J-4.127627
G03 X28.497825 Y32.335606 Z-5.000000 I-4.592959 J-4.382998
G03 X27.163889 Y32.725347 Z-5.000000 I-1.556790 J-2.850645
G03 X26.592273 Y32.693407 Z-5.000000 I-0.158178 J-2.300105
G03 X26.070278 Y32.522500 Z-5.000000 I0.362939 J-1.991120
G03 X25.268571 Y31.986681 Z-5.000000 I1.547821 J-3.183567
G03 X24.720903 Y31.323056 Z-5.000000 I1.849098 J-2.083799
G02 X24.531534 Y31.064248 Z-5.000000 I-1.427746 J0.845994
G01 X24.500417 Y31.076111 Z-5.000000
G03 X24.423052 Y31.426561 Z-5.000000 I-0.832423 J-0.000000
G03 X23.794861 Y32.584236 Z-5.000000 I-9.545447 J-4.430383
G03 X23.029617 Y33.517893 Z-5.000000 I-4.044478 J-2.534502
G03 X22.172083 Y34.118820 Z-5.000000 I-2.469689 J-2.611974
G03 X21.627057 Y34.300078 Z-5.000000 I-0.897524 J-1.788720
G03 X20.743333 Y34.374584 Z-5.000000 I-0.841485 J-4.702750
G01 X20.743333 Y34.374584 Z-5.000000
G00 Z0.000000

G00 Z0.000000
G00 X21.246042 Y29.215209

G01 Z-5.000000 F100.0

G03 X20.193269 Y28.692249 Z-5.000000 I0.000000 J-1.321151 F400.000000
G03 X18.979445 Y26.225417 Z-5.000000 I5.756950 J-4.364804
G03 X18.595388 Y24.355237 Z-5.000000 I14.563235 J-3.965201
G03 X18.467917 Y22.538889 Z-5.000000 I13.041438 J-1.827896
G03 X18.558797 Y20.989413 Z-5.000000 I13.057517 J-0.011548
G03 X18.829514 Y19.469722 Z-5.000000 I13.168503 J1.561870
G02 X18.936453 Y18.957316 Z-5.000000 I-8.846673 J-2.113655
G01 X18.926528 Y18.931736 Z-5.000000
G01 X18.898782 Y18.916009 Z-5.000000
G02 X18.512014 Y18.764167 Z-5.000000 I-20.938067 J52.764322
G03 X17.653648 Y18.229065 Z-5.000000 I0.984862 J-2.535845
G01 X17.727083 Y18.040972 Z-5.000000
G01 X17.793565 Y18.008750 Z-5.000000
G01 X17.797639 Y17.952778 Z-5.000000
G01 X17.826553 Y17.909478 Z-5.000000
G03 X18.467917 Y18.040972 Z-5.000000 I-0.583136 J4.474135
G02 X19.665438 Y18.294012 Z-5.000000 I2.495857 J-8.851572
G02 X20.822708 Y18.376111 Z-5.000000 I1.157270 J-8.115388
G02 X23.741914 Y17.962321 Z-5.000000 I0.000000 J-10.504094
G02 X25.885070 Y16.912083 Z-5.000000 I-1.915285 J-6.620226
G03 X26.191304 Y16.736692 Z-5.000000 I0.950462 J1.304475
G01 X26.290764 Y16.735695 Z-5.000000
G02 X26.399164 Y16.774813 Z-5.000000 I0.496332 J-1.205630
G01 X26.431875 Y16.779792 Z-5.000000
G01 X26.524008 Y16.902159 Z-5.000000
G03 X26.255486 Y17.388333 Z-5.000000 I-1.198479 J-0.344698
G01 X26.043820 Y17.617639 Z-5.000000
G01 X26.237847 Y17.855764 Z-5.000000
G03 X27.427379 Y20.034301 Z-5.000000 I-4.484428 J3.862626
G03 X27.948820 Y23.623681 Z-5.000000 I-12.093181 J3.589379
G03 X27.651390 Y25.934059 Z-5.000000 I-9.122013 J-0.000000
G03 X27.084514 Y26.904514 Z-5.000000 I-2.019074 J-0.528617
G03 X26.747282 Y27.127850 Z-5.000000 I-0.766749 J-0.791497
G03 X26.370139 Y27.204375 Z-5.000000 I-0.377143 J-0.891088
G03 X26.006325 Y27.110831 Z-5.000000 I0.000000 J-0.754249
G03 X25.567570 Y26.763403 Z-5.000000 I0.814840 J-1.479794
G03 X24.961106 Y25.820301 Z-5.000000 I2.561210 J-2.313537
G03 X24.306389 Y23.844167 Z-5.000000 I10.822434 J-4.682125
G03 X23.760280 Y21.018954 Z-5.000000 I34.500914 J-8.134354
G03 X23.644931 Y19.487361 Z-5.000000 I10.110443 J-1.531593
G01 X23.644931 Y18.861181 Z-5.000000
G01 X23.389167 Y18.914097 Z-5.000000
G03 X23.132144 Y18.961090 Z-5.000000 I-1.335020 J-6.575338
G03 X22.507222 Y19.055208 Z-5.000000 I-4.272562 J-26.247110
G02 X21.890023 Y19.146878 Z-5.000000 I4.342716 J31.362522
G01 X21.863403 Y19.161042 Z-5.000000
G01 X21.858328 Y19.187470 Z-5.000000
G02 X22.110347 Y19.787222 Z-5.000000 I11.203714 J-4.355034
G03 X22.455762 Y20.726409 Z-5.000000 I-6.298320 J2.849510
G03 X22.833542 Y22.327222 Z-5.000000 I-16.393140 J4.713643
G03 X23.100093 Y25.128963 Z-5.000000 I-15.624297 J2.900011
G03 X22.868820 Y27.036806 Z-5.000000 I-7.597243 J0.046982
G03 X22.016513 Y28.828556 Z-5.000000 I-4.925218 J-1.244257
G03 X21.246042 Y29.215209 Z-5.000000 I-0.770471 J-0.574321
G01 X21.246042 Y29.215209 Z-5.000000
G00 Z0.000000

G00 Z0.000000

G00 X21.246042 Y28.280347
G00 Z0.000000

G00 Z0.000000
G00 X21.254861 Y17.829306

G01 Z-5.000000 F100.0
G03 X18.763819 Y17.275870 Z-5.000000 I0.036805 J-6.048534 F400.000000
G03 X17.630070 Y16.162431 Z-5.000000 I0.981265 J-2.133100
G03 X17.481139 Y15.720963 Z-5.000000 I1.698454 J-0.818836
G03 X17.418403 Y15.104097 Z-5.000000 I3.038122 J-0.620603
G03 X17.594541 Y14.183851 Z-5.000000 I2.476897 J-0.002895
G03 X18.097500 Y13.410750 Z-5.000000 I2.184395 J0.870955
G03 X19.207668 Y12.605028 Z-5.000000 I2.918517 J2.853607
G03 X21.016736 Y11.964361 Z-5.000000 I3.592429 J7.269544
G03 X22.633877 Y11.799534 Z-5.000000 I1.491445 J6.617369
G03 X24.218195 Y12.026111 Z-5.000000 I-0.120353 J6.493934
G03 X25.644284 Y12.737217 Z-5.000000 I-1.068458 J3.928271
G03 X26.290764 Y13.631250 Z-5.000000 I-1.201528 J1.549585
G03 X26.378979 Y14.897392 Z-5.000000 I-2.231630 J0.791626
G03 X25.788056 Y16.100695 Z-5.000000 I-2.606103 J-0.533069
G03 X24.052961 Y17.320444 Z-5.000000 I-3.287762 J-2.832881
G03 X21.254861 Y17.829306 Z-5.000000 I-2.751280 J-7.181146
G01 X21.254861 Y17.829306 Z-5.000000
G00 Z0.000000
G00 Z0.000000
G00 X11.271250 Y16.188889

G01 Z-5.000000 F100.0
G03 X10.454742 Y16.169056 Z-5.000000 I-0.237132 J-7.054899 F400.000000
G03 X9.930695 Y16.065417 Z-5.000000 I0.194288 J-2.359139
G03 X8.464399 Y15.211063 Z-5.000000 I1.184110 J-3.717695
G03 X8.484306 Y14.724861 Z-5.000000 I0.217169 J-0.234617
G01 X8.632884 Y14.684282 Z-5.000000
G03 X8.942917 Y14.786597 Z-5.000000 I-0.115359 J0.870443
G02 X10.051546 Y15.259545 Z-5.000000 I2.853378 J-5.152720
G02 X10.336389 Y15.218750 Z-5.000000 I0.099428 J-0.320590
G01 X10.376769 Y15.166043 Z-5.000000
G02 X10.556875 Y14.574917 Z-5.000000 I-11.082813 J-3.699753
G03 X11.114769 Y13.066917 Z-5.000000 I9.126622 J2.519254
G03 X12.082639 Y11.320556 Z-5.000000 I13.551397 J6.369079
G02 X13.056645 Y9.606442 Z-5.000000 I-14.776454 J-9.530168
G02 X13.608403 Y8.260194 Z-5.000000 I-9.100051 J-4.515836
G03 X15.615543 Y4.536810 Z-5.000000 I11.405463 J3.745586
G03 X17.859375 Y2.906806 Z-5.000000 I4.054891 J3.222475
G03 X18.519511 Y2.734514 Z-5.000000 I1.172031 J3.139849
G03 X19.376320 Y2.668695 Z-5.000000 I0.856809 J5.543844
G03 X20.553536 Y2.825077 Z-5.000000 I-0.000000 J4.509119
G03 X21.634097 Y3.286028 Z-5.000000 I-1.106203 J4.090169
G03 X22.316200 Y3.772091 Z-5.000000 I-2.817671 J4.675734
G03 X24.209375 Y5.437972 Z-5.000000 I-31.452061 J37.652078
G02 X25.399289 Y6.400520 Z-5.000000 I7.747449 J-8.360720
G02 X26.467153 Y7.025472 Z-5.000000 I4.268375 J-6.068622
G03 X27.592795 Y7.578323 Z-5.000000 I-25.722411 J53.794948
G03 X27.719514 Y7.660472 Z-5.000000 I-0.315090 J0.624852
G02 X27.840651 Y7.750389 Z-5.000000 I1.059404 J-1.300694
G02 X28.380972 Y8.101472 Z-5.000000 I10.391891 J-15.401880
G03 X29.410991 Y8.844859 Z-5.000000 I-5.086837 J8.133481

G03 X30.753403 Y10.059361 Z-5.000000 I-13.119704 J15.850601
 G01 X31.361945 Y10.650278 Z-5.000000
 G01 X31.661806 Y10.288695 Z-5.000000
 G03 X31.960125 Y9.932019 Z-5.000000 I46.996266 J39.003954
 G03 X32.076320 Y9.794806 Z-5.000000 I29.864318 J25.171950
 G03 X32.206780 Y9.703656 Z-5.000000 I0.239515 J0.203873
 G03 X32.349722 Y9.697778 Z-5.000000 I0.080553 J0.217918
 G01 X32.437983 Y9.789043 Z-5.000000
 G03 X32.455556 Y10.129917 Z-5.000000 I-0.884517 J0.216489
 G03 X31.703858 Y11.370947 Z-5.000000 I-1.813041 J-0.249999
 G03 X29.968472 Y11.893805 Z-5.000000 I-1.641082 J-2.305477
 G03 X29.309901 Y11.794989 Z-5.000000 I0.095758 J-2.882158
 G01 X29.218820 Y11.690972 Z-5.000000
 G01 X29.278851 Y11.567347 Z-5.000000
 G03 X29.792084 Y11.355833 Z-5.000000 I0.988311 J1.669685
 G02 X30.297165 Y10.978052 Z-5.000000 I-0.200695 J-0.794853
 G02 X30.259514 Y10.562083 Z-5.000000 I-0.307707 J-0.181836
 G02 X29.032805 Y9.347603 Z-5.000000 I-8.566529 J7.426010
 G02 X28.689653 Y9.221528 Z-5.000000 I-0.343152 J0.403961
 G01 X28.643498 Y9.207534 Z-5.000000
 G03 X28.416250 Y9.045139 Z-5.000000 I2.446223 J-3.663319
 G02 X28.168284 Y8.890578 Z-5.000000 I-0.937558 J1.227955
 G02 X26.581806 Y8.127917 Z-5.000000 I-26.957944 J54.046179
 G02 X24.526822 Y7.470104 Z-5.000000 I-3.671965 J7.932333
 G02 X21.651736 Y7.201861 Z-5.000000 I-2.875086 J15.273772
 G02 X18.903964 Y7.567674 Z-5.000000 I-0.000000 J10.502737
 G02 X16.121945 Y8.701195 Z-5.000000 I3.223725 J11.892780
 G02 X13.192250 Y11.170233 Z-5.000000 I4.913787 J8.803235
 G02 X11.412361 Y14.539639 Z-5.000000 I7.714378 J6.229937
 G02 X11.523922 Y15.117098 Z-5.000000 I0.605700 J0.182489
 G02 X12.073820 Y15.368681 Z-5.000000 I0.524191 J-0.418991
 G01 X12.841111 Y15.333403 Z-5.000000
 G01 X12.814653 Y15.571528 Z-5.000000
 G03 X12.728996 Y15.774727 Z-5.000000 I-0.380039 J-0.040549
 G03 X12.470695 Y15.986042 Z-5.000000 I-0.668240 J-0.553300
 G03 X12.088148 Y16.113409 Z-5.000000 I-0.560632 J-1.045685
 G03 X11.271250 Y16.188889 Z-5.000000 I-1.052555 J-6.933205
 G01 X11.271250 Y16.188889 Z-5.000000
 G00 Z0.000000

(Pie de pagina)

M5

G00 X0.0000 Y0.0000

M2

(Usando pie por defecto. Para añadir tu propio pie crea un archivo "footer" en el directorio de salida.)

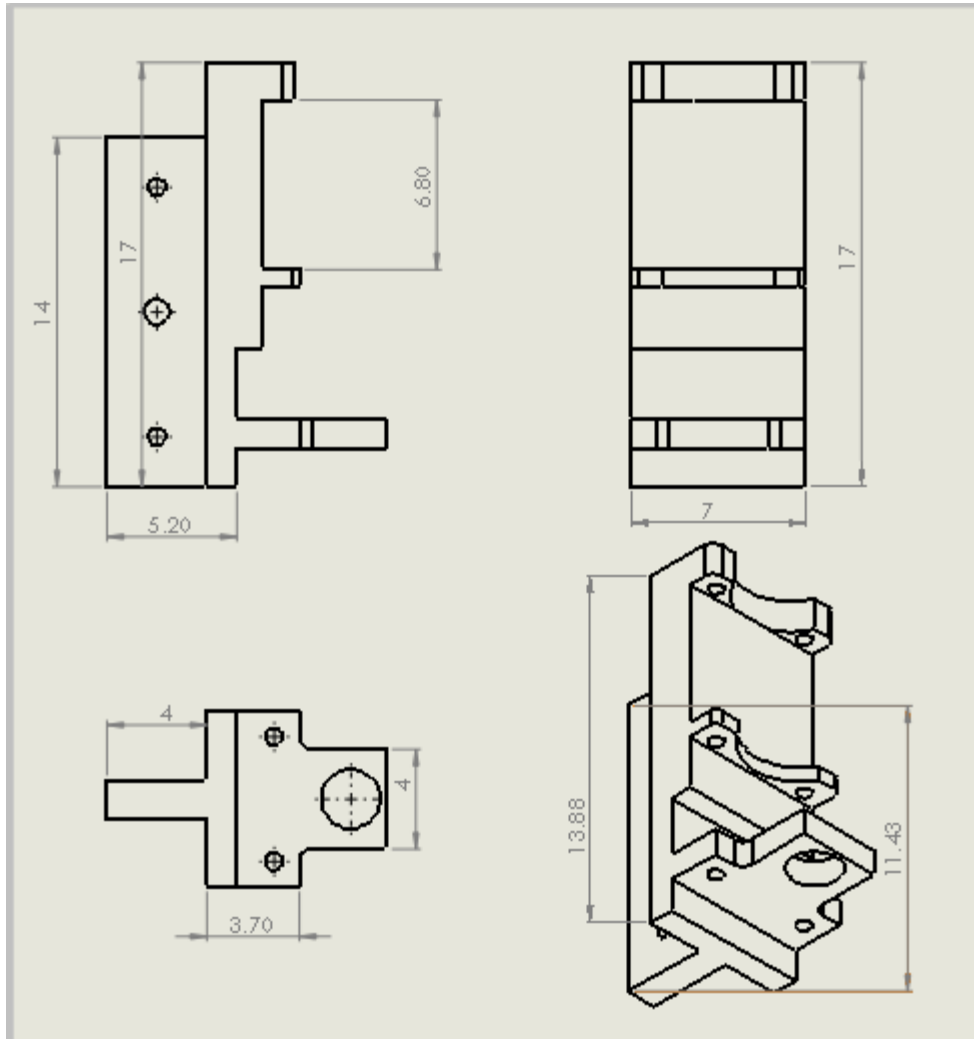
(end)

%

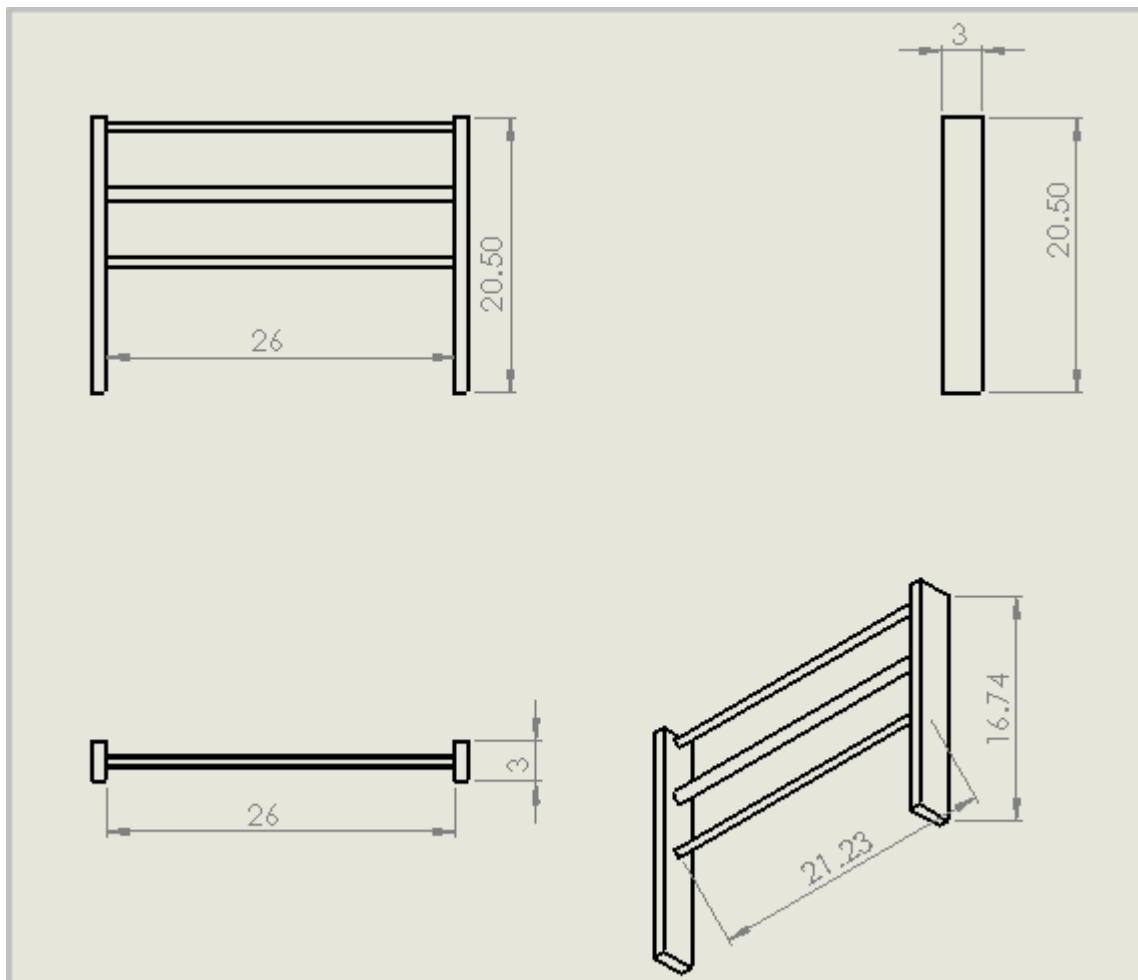
Apéndice E

Isométricos de la parte mecánica

Eje Z



Eje X



Eje Y

