

UNIVERSIDAD SAN FRANCISCO DE QUITO
COLEGIO DE CIENCIAS E INGENIERÍA

Programación de software y protocolo de transferencia de datos de una mesa de corte CNC por plasma a ser usada en el corte de planchas de acero inoxidable.

Pavel Alba Arbakova

Fausto Pasmay MA, Director de Tesis

Tesis de grado presentada como requisito para la obtención del título de ingeniero en sistemas

Quito
Mayo 2014

Universidad San Francisco de Quito

Colegio de ciencias e ingeniería

HOJA DE APROBACION DE TESIS

**Programación de software y protocolo de transferencia de datos de una mesa
de corte CNC por plasma a ser usada en el corte de planchas de acero
inoxidable.**

Pavel Alba Arbakova

Fausto Pasmay. MS Computer Sciences

Director de tesis

Miembro del Comité de Tesis

Daniel Fellig

Miembro del Comité de Tesis

Mauricio Iturralde

Miembro del Comité de Tesis

Ximena Cordova, Ph.D

Decano de ciencias e ingeniería

Quito, Mayo 2014

© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído la Política de Propiedad Intelectual de la Universidad San Francisco de Quito y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo de investigación quedan sujetos a lo dispuesto en la Política.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo de investigación en el repositorio virtual, de conformidad a lo dispuesto en el Art.144 de la Ley Orgánica de Educación Superior.

Firma: _____

Nombre: Pavel Alba Arbakova

C. I.: 1719406827

Fecha: Mayo 2014

RESUMEN

En la actualidad no se puede imaginar un día de trabajo en el que no se utilice una computadora, en especial si se trabaja en ingeniería. Se complica la situación ya que todo lo que se diseñamos en la computadora es exacto, pero al momento de pasarlo a la realidad aparecen varias diferencias, es por esto que se crea el CNC (Control Numérico por Computadora), para poder pasar de la computadora a la realidad de una forma más precisa, rápida y simple.

La importancia de esta tesis se basa en la optimización de tiempo para la construcción de equipos industriales, esto influye en los precios de venta de los equipos y costos de los mismos, lo que permite que la industria alimenticia Ecuatoriana crezca dando a toda la población precios más bajos con una mayor calidad en los productos. También al contar con un equipo CNC se reduce los riesgos de trabajo ya que una máquina que corta metal puede causar muchos daños al operario si existe algún accidente.

Se realizó la construcción de la máquina como el software de control y un protocolo de comunicación. Se puede ver la razón detrás de todas las decisiones tomadas y un manual de cómo fue desarrollado el software y protocolo en este documento.

ABSTRACT

Today we can not imagine a day in which we do not use a computer, especially if you work in engineering. But the situation is difficult because everything we design on the computer is correct, but when passing it to the real world we see several differences, this is why we need a CNC (Computer Numerical Control), to move from the computer to reality in a more accurate, fast and simple way.

The importance of this project is based on the optimization of time for construction of industrial equipment, this influences the sales prices and costs, allowing the Ecuadorian food industry to grow giving lower prices with higher quality products. By having a CNC we reduce the risks of accidents.

We performed the construction of the machine, the control software and a communication protocol. You can see the reason behind all decisions made and a detailed manual on how the software and the protocol were developed.

TABLA DE CONTENIDO

RESUMEN	iii
ABSTRACT	iv
REFERENCIA DE FIGURAS	vii
CAPITULO I	1
Antecedentes y Justificación	1
Objetivo Final	1
Objetivos Específicos.....	2
Metas.....	3
Requerimientos del Cliente.....	3
Metodología.....	5
CAPITULO II	6
Marco Teórico	6
Definición CNC (Control numérico por computadora).....	6
Formato DWG y DXF	7
Bulge.....	8
Comunicación serial	9
Matemática utilizada para transformar la información.....	10
Trigonometría:.....	11
Línea en base a dos puntos:.....	12
Protocolo.....	12
CAPITULO III	15
Implementación	15
Interfaz usuario	15
Facilidades de uso para usuario	15
¿Por qué usar c#?	15
Organización Archivos DXF.....	17
Explicación del código	20
Organizar Grupos	26
Ordenar	27
Generar instrucciones	27
Comunicación Serial.....	27
DXF a imagen.....	28
Seleccionar figura.....	29
Orden y eliminación de figuras.....	30

Zoom in y Out	30
Diseño de Protocolo	31
Objetivos a cumplir con el protocolo	35
Pasos en X y pasos en Y	39
Secuencia.....	39
Suma.....	40
Pedir Mensaje	41
Tiempo de llegada	41
Mensaje ejecución rápida	41
Aviso de recepción	42
Cero	42
Orden del Protocolo	43
CAPITULO IV	45
Calibración y Pruebas	45
Metodología a ser utilizada	45
Pruebas de dibujo	45
Datos Tiempo	47
Correcciones Realizadas.....	49
Recomendaciones para pruebas.....	51
CONCLUSIONES	52
BIBLIOGRAFÍA.....	56

REFERENCIA DE FIGURAS

Tipo	Número	Descripción
Figura	1	Definición Bulge
Figura	2	Puntos iguales con diferente Bulge
Figura	3	Definiciones Trigonométricas
Figura	4	Programa DXFViewer
Figura	5	Programa proyecto tesis
Figura	6	MÓDULOS PROGRAMA
Figura	7	Módulos Abrir Archivo
Figura	8	Polígonos en círculo
Figura	9	Módulos Modificación Grafico
Figura	10	Módulos Generar Instrucción
Figura	11	Diferencia entre imágenes con y sin grupo
Figura	12	Representación de Antorcha
Figura	13	Figura seleccionada
Figura	14	Izquierda figura vista por el usuario, derecha, figura con códigos
Figura	15	Orden antes de eliminar figuras
Figura	16	Orden después de eliminar figuras
Figura	17	Simulación de corte
Figura	18	Flujo del Protocolo
Figura	19	Diagrama de Secuencia del Protocolo
Figura	20	Estructura del Mensaje
Figura	21	Gráficos de Pruebas de Líneas
Figura	22	Gráficos de Pruebas de Círculo y Semicírculo
Figura	23	Máquina Completa
Figura	24	Prueba de Corte 1
Figura	25	Prueba de Corte 2
Figura	26	Eje X
Figura	27	Eje Y
Figura	28	Tarjeta de Control
Tabla	1	Línea DXF
Tabla	2	Arco DXF
Tabla	3	Elipse DXF
Tabla	4	Círculo DXF
Tabla	5	Polilínea DXF
Tabla	6	LW-Polilínea DXF
Tabla	7	ArrayList usados
Tabla	8	Orden usado en ArrayList

Tabla	9	Objetivos del Protocolo
Tabla	10	Conversión ASCII a números
Tabla	11	Dirección y corte antorcha
Tabla	12	Datos pruebas Línea
Tabla	13	Datos pruebas Arco y Semicírculo
Tabla	14	Datos pruebas Línea 500mm
Tabla	15	Datos pruebas Línea 1000mm
Tabla	16	Datos pruebas Círculo
Tabla	17	Conclusiones pruebas Tiempo
Tabla	18	Velocidades de Corte
Tabla	19	Pruebas calidad de corte
Tabla	20	Códigos Velocidades

CAPITULO I

Antecedentes y Justificación

Existen varios software disponibles para hacer lo que se propone este proyecto, pero en la mayoría de casos son hechos por el mismo proveedor de las maquinas en cuestión y no se pueden aplicar correctamente a una maquina armada bajo necesidades específicas. Es por esto que se propuso hacer un software a medida para la máquina que está siendo desarrollada en el proyecto complementario.

En la actualidad no podemos imaginarnos un día de trabajo en el que no utilicemos una computadora, en especial si trabajamos en ingeniería. Pero se complica la situación ya que todo lo que diseñamos en la computadora es exacto, pero al momento de pasarlo a la realidad aparecen varias diferencias, es por esto que se crea el CNC (Control Numérico por Computadora), para poder pasar de la computadora a la realidad de una forma más precisa, rápida y simple.

Una CNC es básicamente una impresora, que interpreta mediante un programa en el computador información y la trasmite a algún elemento externo. Puede ser a un papel, metal, madera, señales de luz, etc... todo depende de la herramienta que va a controlar el computador. Si el computador controla un esfero podría escribir documentos, pero si tiene un taladro puede hacer huecos sobre superficies. El límite de una CNC depende de la herramienta que permitimos que el computador utilice, y en el caso de esta Tesis la herramienta es una mesa de corte mediante plasma.

La importancia de esta tesis se basa en la optimización de tiempo para la construcción de equipos industriales, esto influye en los precios de venta de los equipos y costos de los mismos, el desarrollo del software planteado permite usar con facilidad la máquina y reduce tiempos de obra y posibles accidentes ya que se elimina la necesidad de manejarla manualmente.

Objetivo Final

Desarrollar un software de control para una mesa CNC de corte por plasma que permite leer archivos de AutoCAD en formato DXF y transmitir la información a la máquina para que esta pueda hacer los cortes pertinentes.

Objetivos Específicos

Programación de software que pueda realizar las siguientes tareas:

- Leer archivos DXF de AutoCAD, interpretando este formato para dibujar en la pantalla del usuario las piezas diseñadas.
- Establecer una comunicación por medio de puerto serial entre la computadora y la tarjeta.
- Permitir la agrupación de elementos de la misma pieza que no están marcados como grupo en el archivo leído.
- Permitir cambiar el orden en el cual las piezas se cortan y decidir cuales se cortan y cuáles no.
- Permitir agrandar o achicar la imagen para visualizar detalles.
- Simular los cortes a hacerse sobre el dibujo en la pantalla del usuario
- Visualización en pantalla de las instrucciones generadas y progreso de las mismas.
- Generar instrucciones que puedan ser entendidas por la tarjeta de la maquina e interpretadas como movimientos en los ejes X y Y a cierta velocidad.
- Creación de protocolo de comunicación entre tarjeta y software que sea capaz de retransmitir mensajes perdidos o incorrectos y mantener un flujo constante de instrucciones.
- Comprobación por suma de la integridad del mensaje.
- Facilitar por medio del software el uso de la maquina reduciendo así el tiempo y los riesgos asociados.
- Obtener un software hecho a medida para la máquina.

Metas

- Diseñar un software de uso simple pero con la capacidad de transformar planos de AutoCAD en piezas cortadas. El Software deberá correr en cualquier computadora con Windows XP y comunicación serial o USB.
- Entregar a Interinox S.A un software que se adapte a la máquina desarrollada en el proyecto relacionado, que sea confiable y que permita reducir el costo y tiempo para la fabricación de equipos en acero inoxidable.
- Aprender sobre la correcta comunicación entre software y hardware para lograr minimizar los errores y aprovechar al máximo la tecnología moderna.

Requerimientos del Cliente

Para el desarrollo de este proyecto se tenía que desarrollar un software de control para una mesa CNC de corte por plasma que permite leer archivos de AutoCAD en formato DXF y transmitir la información a la máquina para que esta pueda hacer los cortes pertinentes. Y se tenían los siguientes requerimientos específicos del cliente:

- Abrir archivos de AutoCad 2010 en formato DXF.
- Mostrar en pantalla las figuras definidas en el archivo de AutoCad con la adición de un eje de coordenadas para referencia del usuario.
- Permitir funcionalidades con el uso del mouse para revisar el archivo completo:
 - Zoom con rueda del mouse centrado sobre el cursor.
 - Click en botón central para movimiento.
 - Doble click del botón izquierdo para regresar a la escala original.
 - Cambio de color de figura cuando el mouse se encuentra encima.
 - Focus automático de las diferentes secciones de la interfaz dependiendo de la ubicación del cursor.

- Coordenadas del cursor.

- Mostrar al usuario la posición de la antorcha en coordenadas de los ejes X, Y y Z del área de corte.
- Agrupar automáticamente las figuras que se encuentren separadas, en caso de que varias partes de una misma figura estén presentadas como figuras distintas.
- Permitir el ordenamiento y la eliminación de figuras que no se desee cortar.
- Simular el proceso de corte de las figuras seleccionadas o de todas las figuras en su defecto.
- Generar instrucciones de corte y movimiento en base a la posición actual de la antorcha.
- Mostrar al usuario una tabla de instrucciones generadas en base a las figuras que se van a cortar.
- Permitir iniciar, pausar o detener un proceso de corte.

Se cumplieron las siguientes metas:

Diseñar un software de uso simple pero con la capacidad de transformar planos de AutoCAD en piezas cortadas. El Software corre en cualquier computadora con Windows XP y comunicación serial o USB.

Entregar a Interinox S.A un software que se adapte a la máquina desarrollada para el proyecto relacionado que sea confiable y que permita reducir el costo y tiempo para la fabricación de equipos en acero inoxidable.

Aprender sobre la correcta comunicación entre software y hardware para lograr minimizar los errores y aprovechar al máximo la tecnología moderna.

Metodología

Este fue un proyecto conjunto con Martin Reinoso, estudiante de Ingeniería Electrónica quien diseño la máquina y programó la tarjeta de control que con la cual se utiliza el software.

Para el desarrollo del software se usó C# como lenguaje de programación. La razón para esto es la mayor disponibilidad de bibliografía del mismo en lo referente a este tipo de software.

Se desarrolló un software que permite leer archivos de AutoCAD y transformar su información para poder ser usada en la mesa, el software permite observar el plano en pantalla así como la verificación de datos sobre el mismo. Permite cambiar el orden de las piezas a cortar así como la variación de velocidad durante el corte. Incluye una forma de control directo sobre la mesa para poder definir un cero y permite parar un proceso para continuarlo después. El software es de fácil uso y utiliza recursos mínimos.

Se diseñó un protocolo apto para la transmisión de datos, que garantiza que el dibujo no tenga errores. El protocolo debe garantizar que los datos son correctos y que se ejecutan todas las instrucciones.

Una vez que la máquina estuvo terminada se procedió a la etapa de pruebas y calibración, primero sólo dibujando sobre papel y verificando que sean correctas las imágenes impresas. Una vez terminado esto se procedió a hacer pruebas con el plasma montado para verificar la velocidad óptima de corte así mismo como la altura.

CAPITULO II

Marco Teórico

Definición CNC (Control numérico por computadora)

Para entender fácilmente lo que es y en que nos ayuda el control numérico por computadora les invito a imaginar la vida actual sin impresoras. Todo documento que escribimos en la PC se quedaría en eso, para verlo digitalmente, pero no podríamos tener una copia en papel. Seguramente un trabajo muy importante sería transcribir lo que está en el monitor mediante máquinas de escribir, pero esto hace que el documento sea muy diferente ya que puede contener errores, así como va a perder el formato y la posición de los elementos, imágenes en un documento serían aparte y necesitaríamos de un experto en dibujo para poder pasar del monitor al papel si es solo en negro y un verdadero artista para ponerlo en colores.

Este gran cambio se dio entre 1940-1950 (Control numérico 2012) cuando aparecieron las primeras CNC, con esto comenzó a cambiar la industria ya que las herramientas pasaron a ser controladas por computadoras y no operarios.

“El control numérico (CN) es un sistema de automatización de máquinas herramienta que son operadas mediante comandos programados en un medio de almacenamiento, en comparación con el mando manual mediante volantes o palancas.” [1]

Una mesa CNC es un soporte, el cual mueve a una velocidad dada y en una dirección en 3 dimensiones cualquier herramienta a utilizar. Las aplicaciones de una mesa CNC son muchas pero la única relevante a esta tesis es el corte de metales. Dentro del corte de metales existen varias opciones de materiales que se pueden usar para cortar. En este caso se usa el plasma, este proceso permite cortar todo tipo de metales a una velocidad media pero con costos bajos haciéndola ideal para la industria, en especial para acero inoxidable. [2]

Ventajas de usar una CNC sobre trabajo manual según el manual de Beneficios de Hypertehrm [3]:

- Calidad y precisión en corte tienen una mejora sustancial.

- Se logra un aumento de mínimo 100% en la cantidad de piezas cortadas.
- Los consumibles duran mucho más y el tiempo de vida de las maquinas es mayor.
- No se necesita que los operarios tengan experiencia y en poco minutos pueden ser expertos.
- Las maquinas no se cansan y no necesitan tiempo de descanso.
- El operario no tiene que estar cargando la antorcha.
- El operario no necesita estar con equipo especial de trabajo, gafas, guantes, etc...
- La posición sería más cómoda ya que estará frente a un computador y no agachado.
- No se necesita de herramientas aparte como un flexómetro para marcar el corte.
- Las chispas inferiores se concentrarían en un lugar seguro.

Formato DWG y DXF

“DWG es un formato de archivo informático de dibujo computarizado, utilizado principalmente por el programa AutoCAD; producto de la compañía AutoDesk.

Los archivos DWG no son siempre compatibles entre sí, existiendo numerosas versiones de este tipo de archivo, aparejadas a muchas de las distintas versiones del programa AutoCAD. Debido a la elevada cuota de mercado del programa AutoCAD en la industria y el diseño, se ha cubierto la necesidad de lectura de este tipo de archivos por parte de otros programas mediante un archivo de intercambio, importación-exportación, conocido como DXF (Drawing eXchange File).” [4]

DXF (Drawing Exchange Format) es un formato de archivos en texto, esto quiere decir que se lo puede visualizar como texto en cualquier editor. Este tipo de

DEFINICIÓN BULGE

([HTTP://WWW.AFRALISP.NET/ARCHIVE/GIFS/ANGLE_1.JPG](http://www.afralisp.net/archive/gifs/angle_1.jpg))

Como se puede observar en la siguiente imagen, los puntos iniciales y finales son iguales, pero cambia el bulge.

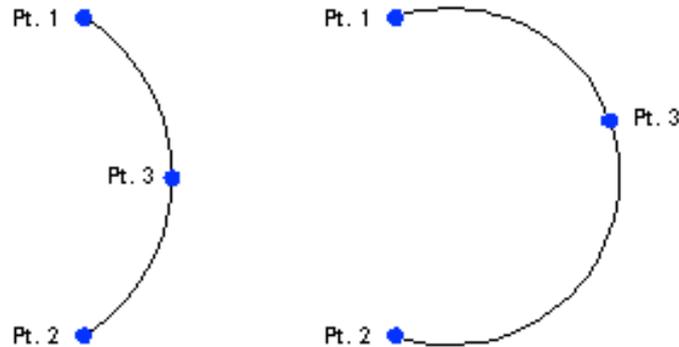


FIG.2

PUNTOS IGUALES CON DIFERENTE BULGE

([HTTP://WWW.PROFANTASY.COM/CCHELP/IMAGES/FCW1033_WMF.GIF](http://www.profantasy.com/cchelp/images/FCW1033_wmf.gif))

La figura de la izquierda tiene un bulge menor a 1, mientras que la figura de la derecha tiene uno mayor a 1. El signo del bulge define la dirección, si se tiene un bulge positivo el arco se genera en sentido de manecillas de reloj, mientras que se invierte si el bulge es negativo.

La forma específica que usamos para decodificar los archivos en formato DXF se puede observar en detalle en el siguiente capítulo.

Comunicación serial

Se va a utilizar comunicación RS-232 (Recommended Standard 232, también conocido como Electronic Industries Alliance RS-232C)

“es una interfaz que designa una norma para el intercambio de una serie de datos binarios entre un DTE (Equipo terminal de datos) y un DCE (Data Communication Equipment, Equipo de Comunicación de datos), la cual utiliza el puerto serial para intercambiar datos.

La interfaz RS-232 fue diseñada para imprimir documentos para distancias cortas, de hasta 15 metros según la norma, y para velocidades de comunicación bajas, de no más de 20 kbps. A pesar de esto, muchas veces se utiliza a mayores velocidades con un resultado aceptable.

Si un dispositivo de los que están conectados a una interfaz RS-232 procesa los datos a una velocidad menor de la que los recibe deben de conectarse las líneas handshaking que permiten realizar un control de flujo tal que al dispositivo más lento le dé tiempo de procesar la información.” [5]

Esta comunicación envía señales en voltios los cuales se interpretan como símbolos digitales (0 y 1). Utilizamos la comunicación por RS-232 como capa 1 (Nivel Físico), para poder crear en este proyecto el protocolo de comunicación.

Matemática utilizada para transformar la información

Para la generación del código que se imprime en pantalla, la información obtenida del archivo DXF, se necesita conocimiento básico de matemáticas, trigonometría y aritmética, es por esto que se presentan a continuación conceptos básicos:

Trigonometría [6]:

$$\sin \alpha = \frac{\overline{CB}}{\overline{AB}} = \frac{a}{c}$$

$$\cos \alpha = \frac{\overline{AC}}{\overline{AB}} = \frac{b}{c}$$

$$\tan \alpha = \frac{\overline{CB}}{\overline{AC}} = \frac{a}{b}$$

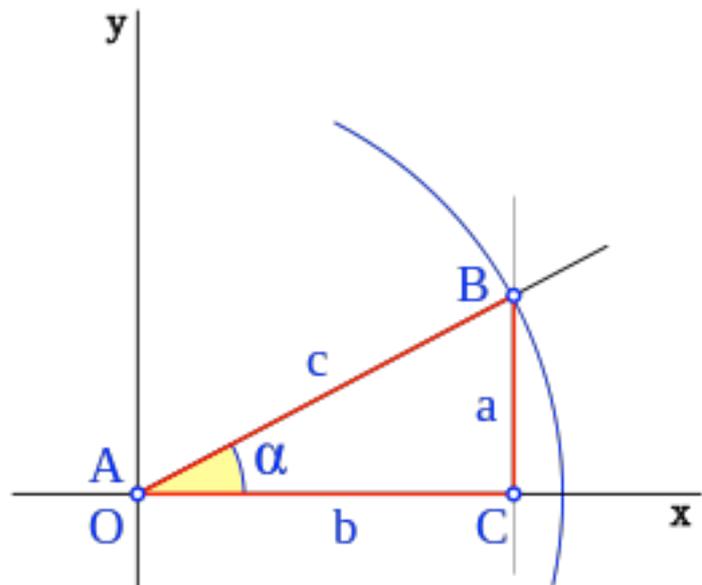


FIG.3

DEFINICIONES TRIGONOMÉTRICAS

Línea en base a dos puntos:

Ya que la máquina interpretara rectas es muy importante conocer la ecuación de la recta si se conoce dos puntos que la conforman.

$$y = \frac{y_1 - y_2}{x_1 - x_2} x + y_1 - \frac{y_1 - y_2}{x_1 - x_2} x_1$$

Protocolo

“En informática y telecomunicación, un protocolo de comunicaciones es un conjunto de reglas y normas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellos para transmitir información por medio de cualquier tipo de variación de una magnitud física. Se trata de las reglas o el estándar que define la sintaxis, semántica y sincronización de la comunicación, así como posibles métodos de recuperación de errores. Los protocolos pueden ser implementados por hardware, software, o una combinación de ambos.

Por ejemplo, el protocolo sobre palomas mensajeras permite definir la forma en la que una paloma mensajera transmite información de una ubicación a otra, definiendo todos los aspectos que intervienen en la comunicación: tipo de paloma, cifrado del mensaje, tiempos de espera antes de dar la paloma por 'perdida'... y cualquier regla que ordene y mejore la comunicación.” [7]

Si bien los protocolos pueden variar mucho en propósito y sofisticación, la mayoría especifica una o más de las siguientes propiedades:

- Detección de la conexión física subyacente (con cable o inalámbrica), o la existencia de otro punto final o nodo.
- Handshaking (ambas partes se ponen de acuerdo en cuando inicia y termina la conexión).
- Negociación de varias características de la conexión.
- Cómo iniciar y finalizar un mensaje.

- Procedimientos en el formateo de un mensaje.
- Qué hacer con mensajes corruptos o formateados incorrectamente (corrección de errores).
- Cómo detectar una pérdida inesperada de la conexión, y qué hacer entonces.
- Terminación de la sesión y/o conexión.

Los protocolos de comunicación permiten el flujo información entre equipos que manejan lenguajes distintos, por ejemplo, dos computadores conectados en la misma red pero con protocolos diferentes no podrían comunicarse jamás, para ello, es necesario que ambas "hablen" el mismo idioma. El protocolo TCP/IP fue creado para las comunicaciones en Internet. Para que cualquier computador se conecte a Internet es necesario que tenga instalado este protocolo de comunicación.

Un protocolo son reglas básicas para cualquier interacción. Por ejemplo, cuando se utiliza un teléfono seguimos las siguientes reglas básicas:

- Levantar el teléfono y esperar a la señal de tono
- Marcar el número con quien se desea comunicar
- Esperar a recibir la palabra "Aló"
- Responder con la palabra "Aló"
- Comunicarnos enviando un mensaje y escuchando a la otra persona
- Uno de los usuarios empieza una despedida
- El otro usuario responde la despedida
- Cerrar el teléfono

Como se puede observar en el ejemplo anterior, se necesita de ciertas reglas básicas para poder mantener una comunicación. Si se salta cualquier paso la comunicación no será exitosa.

En las comunicaciones entre computadoras las reglas son más estrictas, para garantizar que la comunicación sea correcta dependiendo el caso, no todas las comunicaciones deben ser perfectas, por ejemplo al conversar con alguien por una video llamada no importa que no se vea todo, es más importante que la

comunicación sea fluida, si por algún motivo un paquete desaparece no es importante volverlo a enviar ya que es un momento que ya pasó, es más importante tener los paquetes actuales. En otros casos si es muy importante recibir la información correcta y ordenada, por ejemplo en un chat, si llegan los mensajes en otro orden puede causar cambios en la idea que se desea transmitir, en este caso es preferible esperar que verlos al instante en otro orden.

Todas estas reglas se determinan previamente, es por esto la importancia de los protocolos, para que las dos partes que se comunican sepan cómo deben actuar en cada caso. El diseño del protocolo usado en esta tesis será explicado en detalle en el capítulo 3.

CAPITULO III

Implementación

Interfaz usuario

El usuario deberá poder abrir un archivo en DXF, revisarlo dentro del programa y simular un corte, una vez revisado todo, deberá el programa permitir el inicio de corte con variaciones de velocidad instantánea, esto ayuda a tener mejor calidad de corte ya que si el operario observa que la máquina va muy rápido o lento, puede modificar la velocidad en la primera pieza. A diferencia de otras CNC en el mercado, las cuales necesariamente deben terminar la pieza para poder modificar estos parámetros.

Facilidades de uso para usuario

Ya que el usuario que generará los planos para el corte está acostumbrado al uso de AutoCad, se incorporará facilidades de uso dentro del programa parecidas a las incorporadas en AutoCad 2010. Estas ayudas serán principalmente en la parte visual, como movimientos con el mouse, zoom con rueda del mouse en dirección al cursor, etc...

¿Por qué usar c#?

Se eligió utilizar C# debido a la facilidad que presenta para integrar interfaz gráfica, comunicación serial y facilidades para el usuario final. Un punto muy importante fue encontrar (<http://www.codeproject.com/Articles/11570/A-DXF-Reader-Solution-and-a-Simple-DXF-Viewer>) un programa que podía leer DXF. Aunque el programa fue completamente modificado, saber que el lenguaje C# permitía llegar a la generación de las imágenes fue la base para elegir este lenguaje de programación.

El programa DXFViewer tiene muchas fallas, no puede leer muchos archivos, ni dibujar líneas con Bulge (definición en matemáticas), se presenta a continuación

capturas de pantallas de los 2 programas abriendo el mismo archivo para que se observen las mejoras realizadas a este programa.

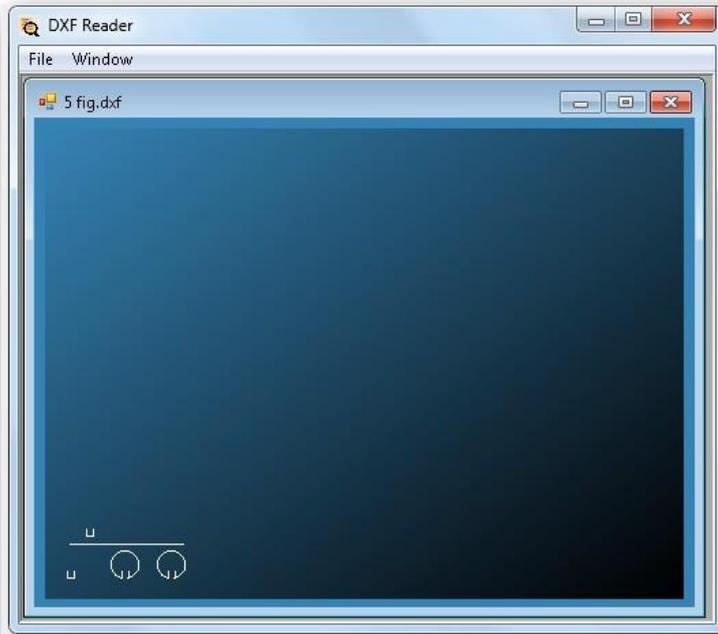


FIG. 4
PROGRAMA DXFVIEWER

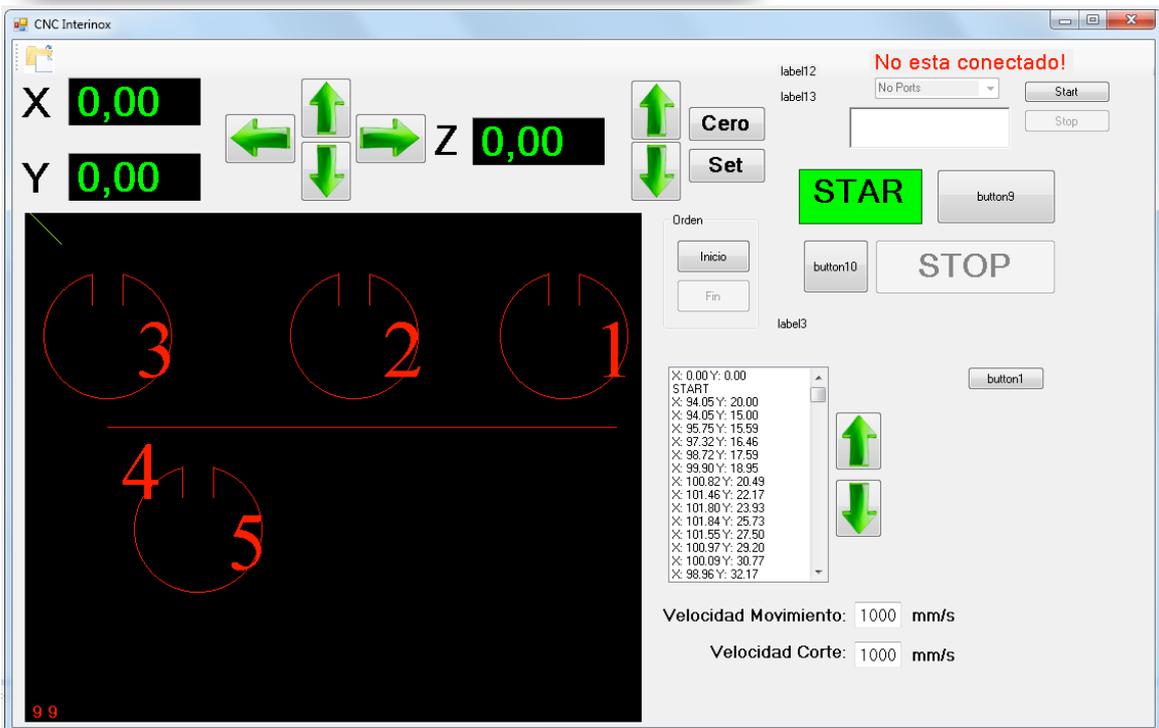


FIG. 5
PROGRAMA PROYECTO TESIS

Organización Archivos DXF

DXF tiene un formato especial, el cual cuenta de 2 líneas, la primera envía un código el cual se interpreta como la información siguiente, y la segunda línea es la información en sí. A continuación se presentan tablas con los códigos y sus significados para cada una de las diferentes figuras que se presenta en el software. En las siguientes tablas se especifican las combinaciones de líneas que buscamos en un archivo DXF para identificar la figura correcta.

Línea

Línea 1	Línea 2	Explicación
0	LINE	Inicia una figura (línea)
10	X1	Punto inicio X
20	Y1	Punto inicio Y
11	X2	Punto final X
21	Y2	Punto final Y

TABLA 1
LÍNEA DXF

Arco

Línea 1	Línea 2	Explicación
0	ARC	Inicia una figura (Arco)
10	XC	Punto centro C
20	YC	Punto centro Y
40	R	Radio
50	Ang inicio	Ángulo de inicio
51	Ang fin	Ángulo de fin

TABLA 2
ARCO DXF

Elipse

Línea 1	Línea 2	Explicación
0	ELLIPSE	Inicia una figura (línea)
10	X1	Punto inicio X
20	Y1	Punto inicio Y
11	X2	Punto final X
21	Y2	Punto final Y

TABLA 3
ELIPSE DXF

Círculo

Línea 1	Línea 2	Explicación
0	CIRCLE	Inicia una figura (círculo)
10	XC	Punto centro X
20	YC	Punto centro Y
40	R	Radio

TABLA 4
CÍRCULO DXF

Polilínea

Línea 1	Línea 2	Explicación
0	POLYLINE	Inicia una figura (línea)
10	X1	Punto inicio X
20	Y1	Punto inicio Y
42	B	Bulge
70	Open or Close	Figura abierta o cerrada
90	Num	Número de vértices

TABLA 5
POLILÍNEA DXF

Línea 1	Línea 2	Explicación
0	LWPOLYLINE	Inicia una figura (línea)
10	X1	Punto inicio X
20	Y1	Punto inicio Y
42	B	Bulge
70	Open or Close	Figura abierta o cerrada
VERTEX o SEQEND	Fin de lista	Fin de puntos en polyline

LW-Polilínea

TABLA 6
LW-POLILÍNEA DXF

Para aclarar como se ve un archivo DXF podemos ver el siguiente ejemplo:

```

0
LINE
10
205.3045814328873
20
137.3316310653253
11
205.3045814328873
21
171.4984661896244

```

En base a las tablas presentadas anteriormente (Tablas 1 – 6) podemos ver que el ejemplo presenta una Línea con punto de inicio X = 205.3045814328873, punto de inicio Y = 137.3316310653253, punto final X = 205.3045814328873 y punto final Y = 171.4984661896244.

Explicación del código

El código del programa se lo puede encontrar en el Anexo 1, a continuación se expondrán las ideas detrás de las funciones más importantes del programa y como fueron aplicadas. La Figura 6 representa los principales módulos dentro del programa:

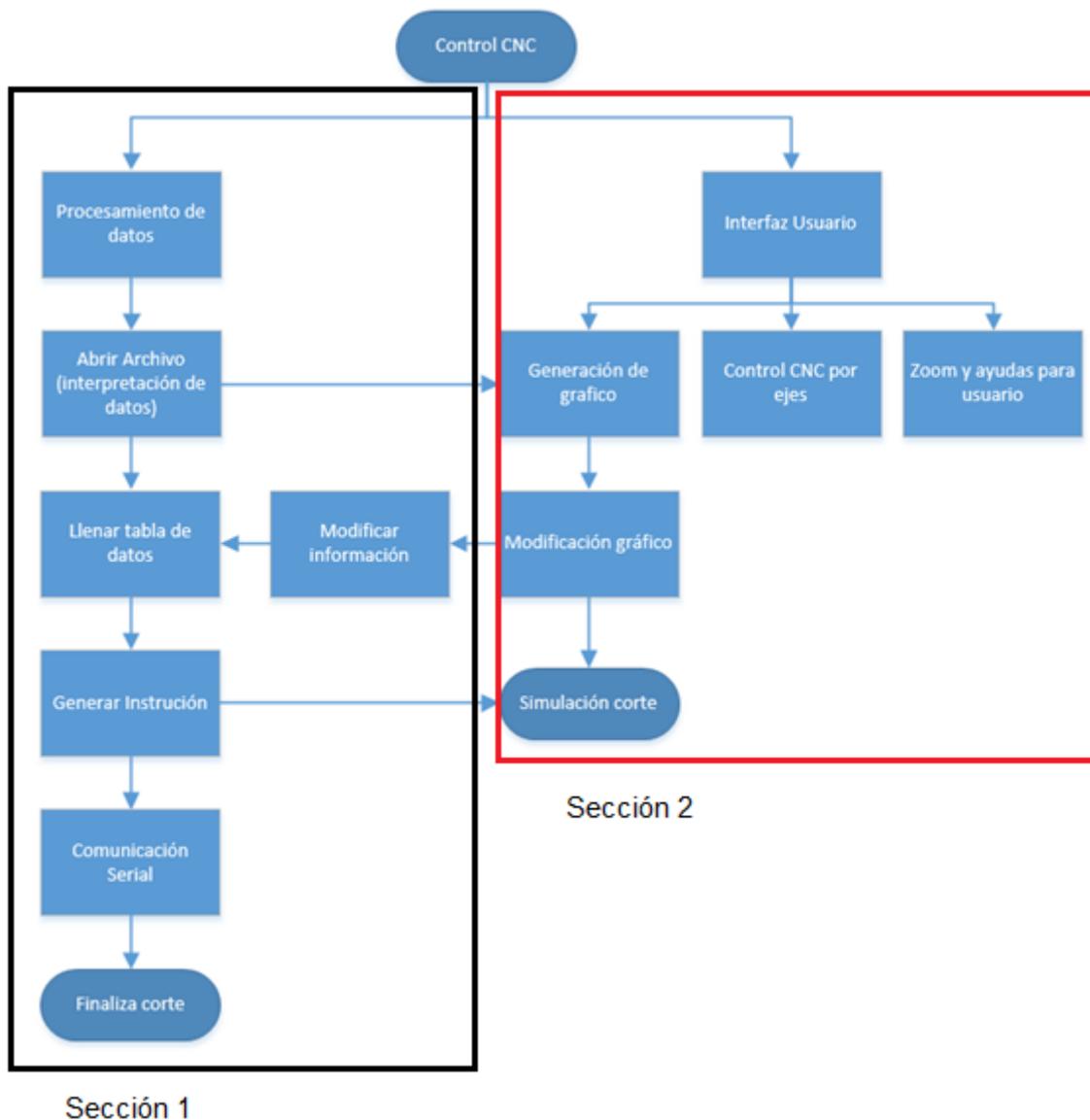


FIG. 6
MÓDULOS PROGRAMA

El software generado se divide en 2 secciones, la primera se encarga de obtener, modificar y organizar la información de un archivo DXF, mientras que la segunda parte muestra la información al usuario y permite que la pueda modificar.

Primero abrimos el archivo en DXF el cual tiene el formato en base a códigos como se puede observar en el capítulo anterior. Esta información se la transforma a un arreglo de datos los cuales se dividen dependiendo del tipo de figura. Esta información se transforma para generar instrucciones, las cuales indican la dirección de movimiento y la cantidad de pasos, básicamente transformamos coordenadas de una figura en instrucciones de movimiento para la cortadora CNC.

La segunda sección se encarga de generar la parte gráfica para el usuario. En base a la tabla de datos obtenida y transformada ya del DXF se genera un gráfico con las figuras a cortar dentro del software, este gráfico puede ser utilizado para comprobar que lo que deseamos cortar es correcto. Podemos hacer pequeñas modificaciones, como el orden de corte y que figuras cortar, las cuales transmitimos antes de generar las instrucciones.

Es muy importante simular cómo será el corte en base a las instrucciones generadas, es por esta razón que la simulación dentro del software es exactamente lo que producirá la máquina ya que se utilizan las mismas instrucciones. No debemos olvidar que en esta sección encontramos un control básico de los elementos de la máquina, para poder poner en cero y moverla a la posición donde deseamos que empiece a cortar, además de las ayudas para el usuario como movimientos con el mouse, zoom, etc...

El software de control para la mesa de corte CNC se divide en la información visual para el usuario y el procesamiento interno de datos. Las facilidades de uso como zoom, selección de elementos y control manual de la antorcha se encuentran como ítems visuales en pantalla, mientras que la parte de decodificación del archivo DXF en ítems para generar instrucciones de corte y luego poder enviarlas mediante un protocolo de comunicación, se encuentran en el procesamiento interno de los datos. Existe una interconexión entre estas partes del programa cuando se necesita

modificar los items o el orden de corte, el en cual la parte gráfica modifica los datos antes de ser convertidos en instrucciones.

Para la decodificación del archivo DXF debemos conocer claramente las reglas que lo componen, las cuales se pueden observar en las tablas al inicio de este capítulo. En este proyecto se utilizaron las tres figuras más comunes que se emplean en AutoCad para cortar en una mesa CNC; Líneas, círculos y poli líneas.

“Una poli línea es un objeto en AutoCAD que consiste de uno o más segmentos de línea (rectas o arcos).” [8]

Como se explicó anteriormente, el formato de DXF consta de la lectura de dos líneas simultáneas, con un código en la línea 1, el cual explica el valor que aparece en la línea 2. El usuario elige un archivo a abrir, a continuación se abre el archivo DXF, que contiene la información de las figuras, para almacenar estos datos se utilizan tres arreglos dinámicos llamados arrayList (matrices de datos que aumentan su tamaño al necesitar guardar más información, no hay necesidad de declararlas con un tamaño definido.)

ArrayList	Datos guardados
IDtipo	Se guarda el tipo de figura, ejemplo, linea, polyline, arc, etc...
IDinicio	Guardamos la posición donde se encuentra el primer valor dentro del arreglo Datos
Datos	Se guardan los datos de todas las figuras.

TABLA 7
ARRAYLIST USADOS

Para obtener la información de las figuras se va a utilizar su “Código”, por ejemplo, para la figura “1”, se ve en IDtipo[1], luego se va a IDinicio[1] el cual da un código, con este código se tiene la posición en Datos donde empiezan los puntos.

Dtipo[1] = Linea

IDinicio[1] = 20

Datos[20]= 10

Datos[21]= 20

Datos[22]= 15

Datos[23]= 25

En base a los datos anteriores se define una línea recta entre los puntos (10,20) - (15,25).

En base al tipo en IDtipo se va a determinar cuántos valores se debe leer del archivo datos en base a la Tabla 8.

IDtipo	Cantidad de datos	Valores
Line	4	Ini X, Ini Y, Fin X, Fin Y
Arc	5	Centro X, Centro Y, Radio, Angulo Ini, Angulo Fin
circulo	3	Centro X, Centro Y, Radio
polyline	No definido	X, Y, B, X, Y, B ... seguimos recibiendo datos hasta obtener un END
Ellipse	7	Centro X, Centro Y, Relación de eje menor a eje mayor, Punto final del eje mayor X, Punto final del eje mayor Y, Angulo Ini, Angulo Fin

TABLA 8
ORDEN USADO EN ARRAYLIST

Al momento de generar una tabla de datos debemos transformar la información recibida del archivo por los siguientes 2 motivos:

- Todas las figuras deben tener un inicio y fin para el corte
- Cuando se enciende el plasma, el punto de inicio genera un daño mayor que el corte, por eso, este punto debe estar en un área que no afecte a la figura deseada.

Por lo expuesto en los puntos anteriores, se puede deducir que si una figura está formada por varias líneas, primero debemos encontrar cual es la figura final, para después poder definir su punto de inicio y final. Por esta razón debemos convertir

todas las líneas que tengan un punto de inicio igual al final de otra en un polígono continuo. En el caso de un círculo, lo importante es conocer si el material que deseamos obtener es una circunferencia o un hueco dentro de otra figura para definir el punto correcto de inicio y final.

Una vez obtenidos estos valores, todas las figuras a cortar serán polígonos. Este proceso se lo puede observar en la siguiente figura.

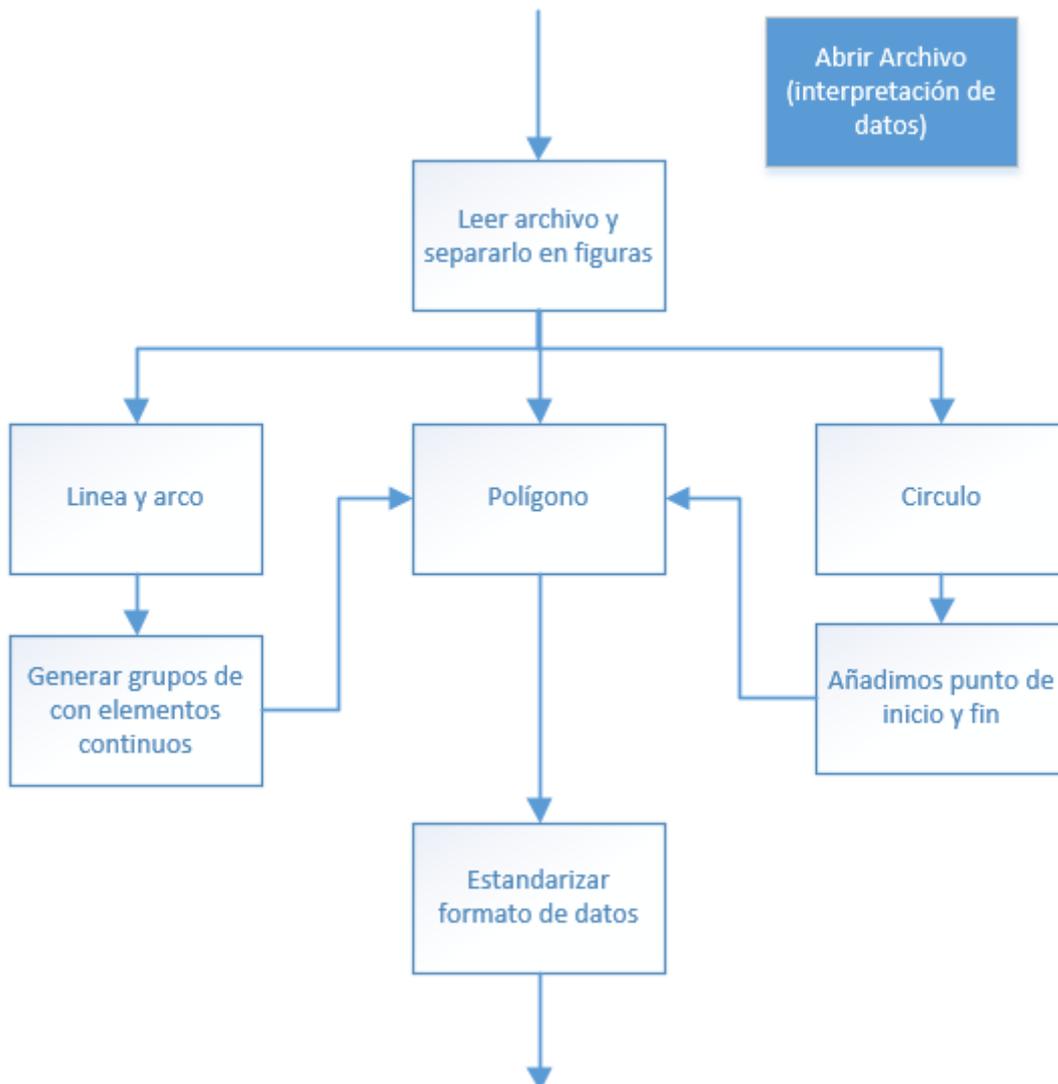


FIG. 7
MÓDULOS ABRIR ARCHIVO

Un punto extra que se debe aclarar es en las líneas, ya que no se especificó que sean siempre rectas. Esto quiere decir que al hablar de líneas estamos hablando también de arcos. Para diferencias entre una línea recta y un arco entra el concepto de un bulge, que fue explicado en el capítulo 2 en la sección Bulge.

Entonces con la idea del bulge, una figura a cortar no necesariamente tiene todos los lados rectos, por lo tanto estos polígonos mediante el bulge se transforman en poli líneas. Para poder dividir los datos a instrucciones necesitamos enviar coordenadas en X y Y de los cambios, es por esta razón necesario siempre tener líneas rectas, por lo tanto debemos diferenciar entre líneas y curvas de la poli línea. Para facilitar esto se va a definir un ángulo de paso, esto es un ángulo en el cual se dividen las figuras.

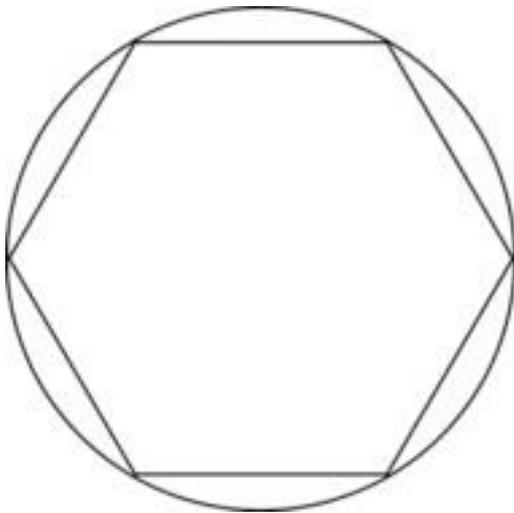
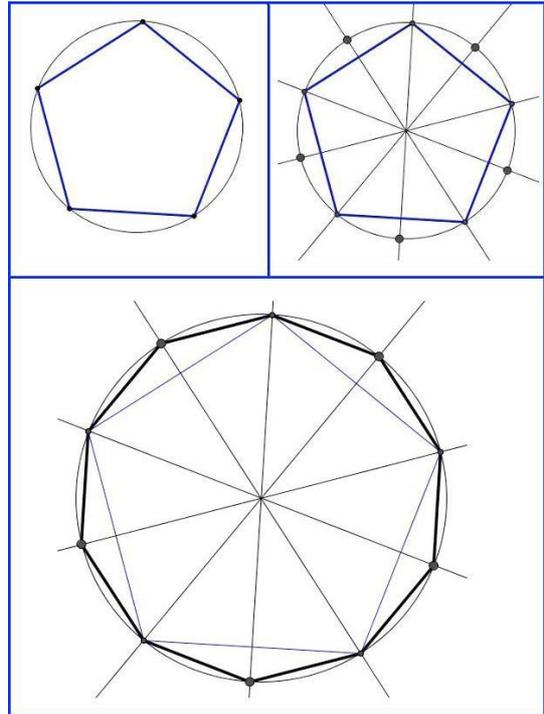


FIG. 8
POLÍGONOS EN CÍRCULO
[HTTP://GAUSSIANOS.COM/IMAGES/
PENTAGONO-DECAGONO.JPG](http://gaussianos.com/images/pentagono-decagono.jpg)

Como se puede observar en la figura anterior se tiene un ángulo de separación de 60 grados, pero si el ángulo de paso se lo divide a la mitad se tiene una figura en

líneas más parecida al círculo original. Entonces solo se necesita las ecuaciones de los arcos, círculos y elipses para poder determinar líneas que los conforman.

Mientras menor sea el ángulo de paso, la curva tendrá más rectas que la conforman, por esta razón será más parecida a la curva original. El diagrama a continuación toma en cuenta la teoría expresada anteriormente para poder transformar las poli líneas en instrucciones donde cada una es una cantidad de pasos en X y Y que forman una línea recta

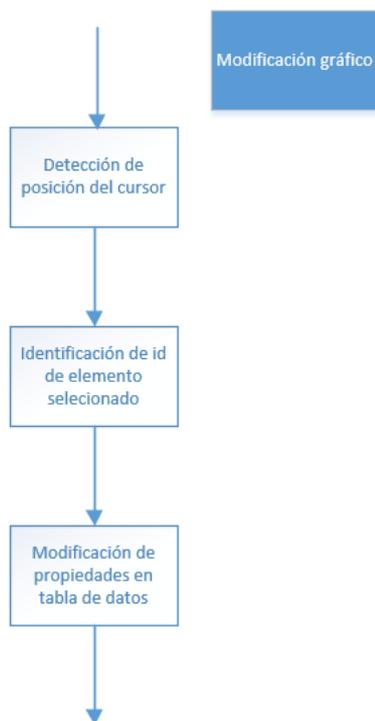


FIG. 9
MÓDULOS MODIFICACIÓN GRÁFICO

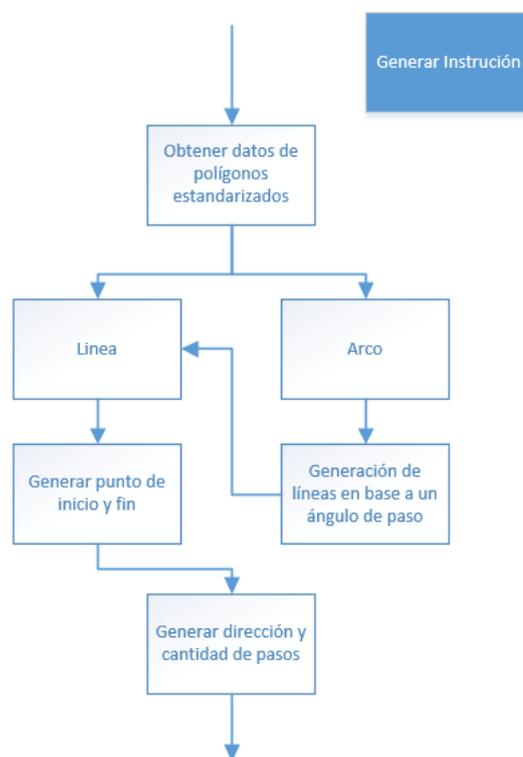


FIG. 10
MÓDULOS GENERAR INSTRUCCIÓN

Organizar Grupos

Este es uno de los módulos representados en la Figura 6. Con los datos guardados en variables se dispone a realizar un conexión de las figuras, ya que pueden ser partes de un mismo grupo, para esto analizamos los puntos extremos de las figuras y se busca puntos en común, así poder ponerlas a continuación. Por ejemplo, una

figura generada por 2 líneas y un arco después de procesarla será definida como un polyline con cuatro puntos y bulge entre los puntos intermedios.

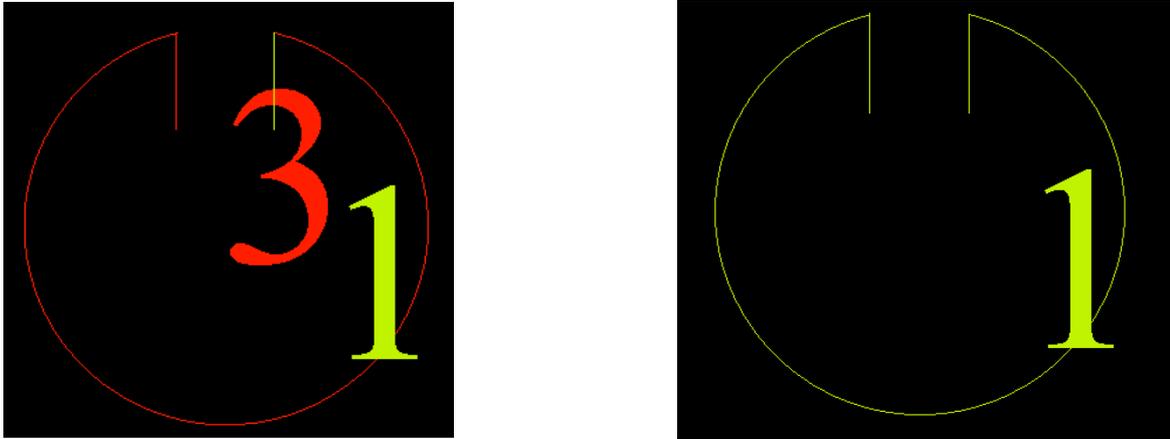


FIG. 11
DIFERENCIA ENTRE IMÁGENES CON Y SIN GRUPO

Ordenar

Antes de proceder con una simulación de corte o un corte es posible cambiar el orden en el cual se procederá. Por ejemplo si se desea realizar un hueco dentro de un cuadrado, primero se debe cortar el círculo interno y después el cuadrado externo. El usuario puede mediante clicks realizar esta operación. Dentro de esta funcionalidad de orden también se permite eliminar figuras que no se desea cortar, dando un orden a las que se vaa a cortar y el fin de la operación.

Generar instrucciones

Una vez listos con las figuras a cortar y en orden correcto, se van a generar instrucciones, las cuales serán la cantidad de pasos que se debe dirigir a los motores en cierta dirección. Es muy importante tomar en cuenta que se tiene un límite máximo de pasos, por esta razón se debe analizar las distancias para conocer cuántas instrucciones pequeñas se debe generar.

Comunicación Serial

La comunicación se refiere a los parámetros básicos siguientes:

```
serialPort.PortName = comboBox1.SelectedItem.ToString();  
  
serialPort.BaudRate = 9600;  
  
serialPort.Parity = Parity.None;  
  
serialPort.DataBits = 8;  
  
serialPort.StopBits = StopBits.One;
```

Estos mismos parámetros se deben configurar en la tarjeta eZdsp F2808 para que exista comunicación.

Código de objetos que interactúan con el usuario

DXF a imagen

Al abrir el archivo seleccionado, este se debe dibujar en pantalla para que el usuario pueda revisar lo que desea cortar. En base a los datos se van a dibujar todos elementos en referencia a su código. Se añaden dentro de la imagen bordes de color blanco para definir los ejes y una cruz con círculo que representa la antorcha.

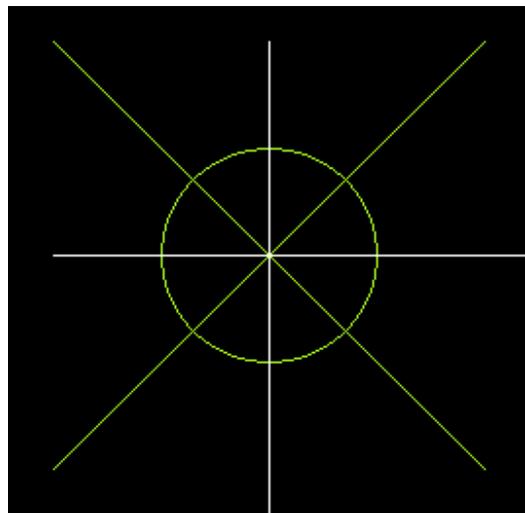


FIG. 12
REPRESENTACIÓN DE ANTORCHA

Seleccionar figura

Al decir seleccionar, esto se refiere a mover el cursor por encima de la figura y que esta cambie de color. Para esto se genera dos figuras exactamente iguales, la primera con todas las figuras en color rojo, la que verá el usuario, mientras que se genera una segunda imagen con color igual al código de la figura. La segunda imagen tiene anchos de línea mayores para que sea más fácil seleccionarla. Una vez desplegada la figura buscamos las coordenadas del mouse para saber qué color se observa y con esto definimos el código de la figura resaltada, para poder dibujarla.

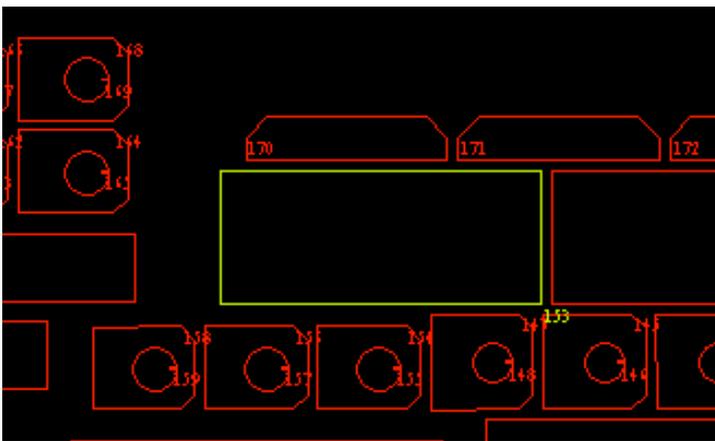


FIG. 13
FIGURA SELECCIONADA

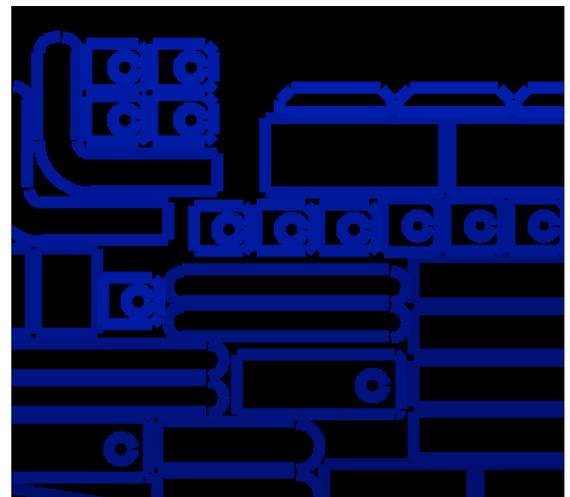
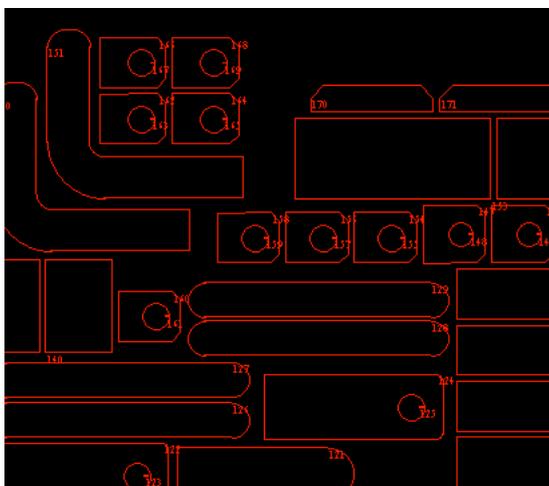


FIG. 14
IZQUIERDA FIGURA VISTA POR EL USUARIO,
DERECHA, FIGURA CON CÓDIGOS

Orden y eliminación de figuras

Es importante que el orden de las figuras sea muy gráfico es por esto que se añade un número el cual además de indicar el código de figura, también define su orden en el corte. A continuación se puede observar un proceso de orden eliminando figuras que no se desea cortar.

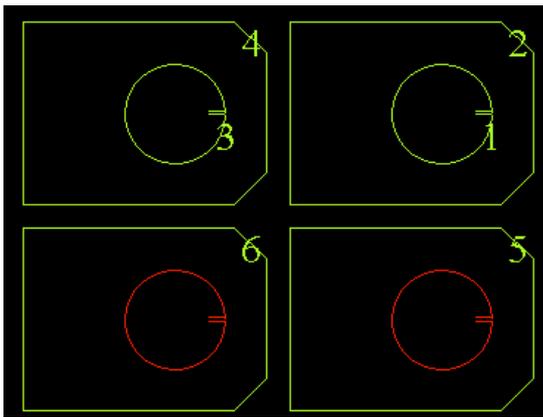


FIG. 15
ORDEN ANTES DE ELIMINAR FIGURAS

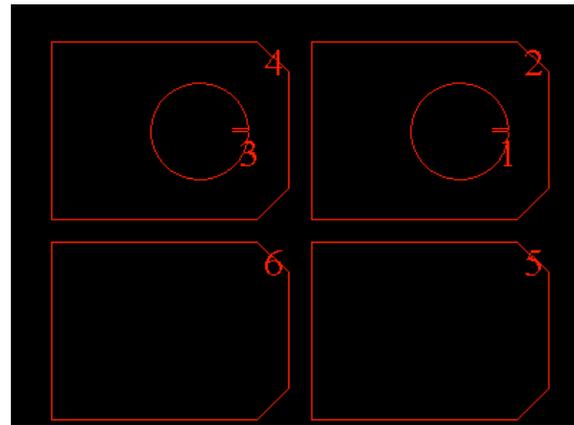


FIG. 16
ORDEN DESPUÉS DE ELIMINAR FIGURAS

Zoom in y Out

Para mantener la calidad en la imagen, se la genera nuevamente cada vez que se requiera ver más detalles, pero solo se genera el área que se observa, con esto el usuario no notará un retraso en el desempeño del programa. Se elige el zoom al mover la rueda media del mouse, para que sea igual al que el usuario está acostumbrado a usar en AutoCad. Se utilizó el cursor como punto centro del zoom, así a donde este apuntando se va a acercar.

Además del zoom también se va a permitir otros controles con el movimiento del mouse, al realizar doble click sobre la imagen se regresa al zoom original. Al aplastar con el botón centro del mouse y desplazarlo se va a mover la imagen y así se puede observar más detalles en la imagen.

Para revisar que todas las instrucciones están bien, se permite simular el orden y la posición del corte con colores como se muestra en la figura a continuación. Cuando

se tiene un verde oscuro se refiere a un movimiento sin corte de antorcha, mientras que el amarillo es corte de antorcha. Ya que todas las figuras se forman con líneas se puede ver si la resolución usada para los arcos y elipses es correcta.

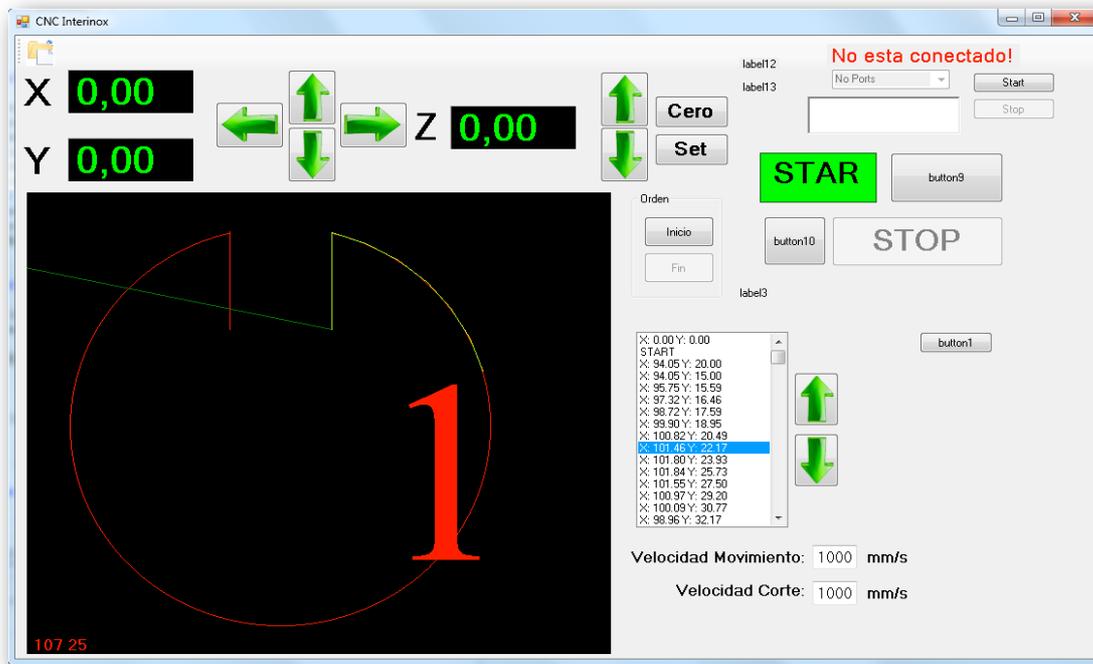


FIG. 17
SIMULACIÓN DE CORTE

Diseño de Protocolo

El protocolo son las reglas de comunicación, y se las define de la manera siguiente:

- Borrar cualquier información anterior de la maquina
- Enviar las velocidades para corte y movimiento de la antorcha cuando pasa de una figura a otra sin cortar.
- Se añade una suma de comprobación, que funciona de la siguiente forma:
 - En el software se suman todos los valores ascii del mensaje y este número se lo añade al final.
 - La tarjeta que recibe el mensaje nuevamente vuelve a sumar los valores del mensaje y compara la suma obtenida con la que llega al final del mensaje, si este valor es el mismo se tomó como mensaje correcto, caso contrario se lo descarta.

- Se añade un orden de mensaje, esto es para poder informar el mensaje que estamos enviando, si el receptor nos pide el mismo mensaje que enviamos anteriormente, asumimos que existió un error y lo reenviamos.
- Una vez finalizado el envío de todos los mensajes terminamos el proceso exitosamente.

A continuación se encuentra la Figura 18 que representa los módulos que se usan en la comunicación y en los cuales se implementa el protocolo:

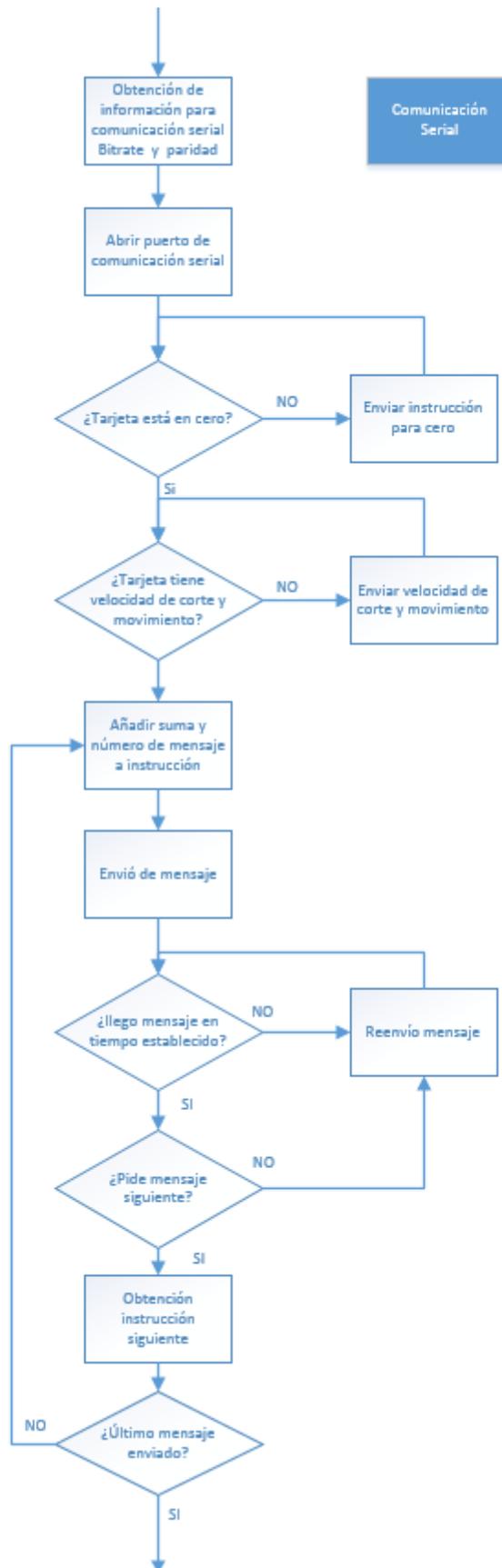


FIG. 18
FLUJO DEL PROTOCOLO

En la Figura 19 podemos observar el diagrama de secuencia que sigue el protocolo:

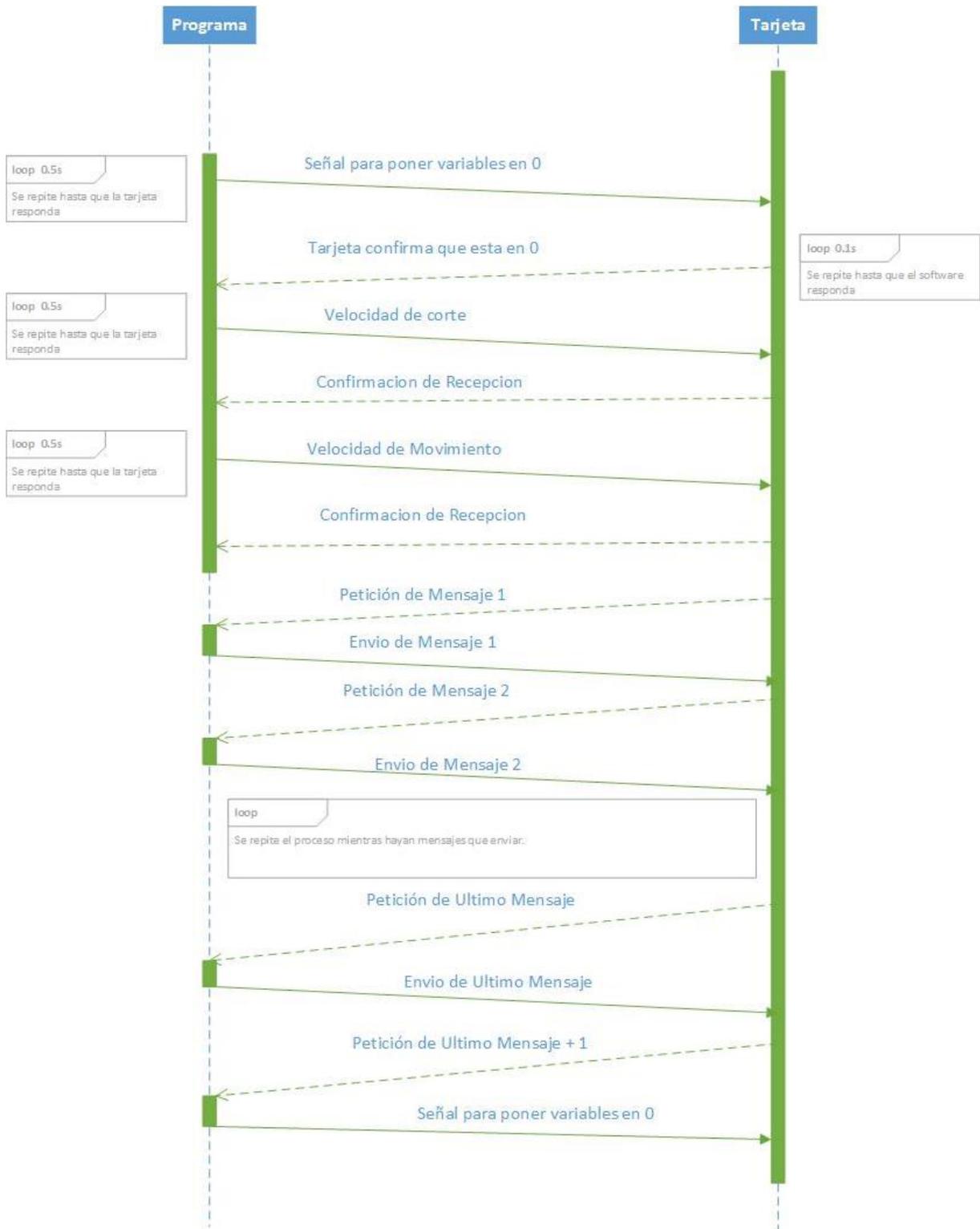
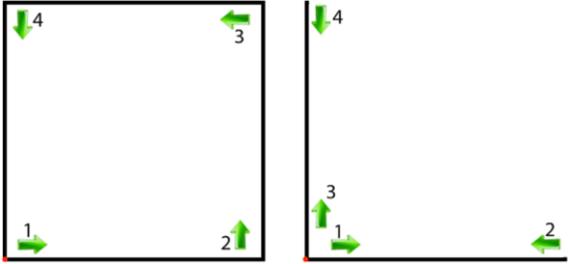


FIG. 19
DIAGRAMA DE SECUENCIA DEL PROTOCOLO

Para diseñar un protocolo se tiene que identificar las características principales que debe cumplir nuestra comunicación. Para esto se presenta la siguiente tabla donde se muestra la necesidad y su explicación.

Objetivos a cumplir con el protocolo

Necesidad	Explicación
Mensaje completo	Se necesita obtener un mensaje completo, no sirve de nada mover el motor X si no se conoce cuanto debe moverse el motor Y
Mensaje correcto	Si el mensaje llega cambiado, se puede pasar de movernos 10 pasos a movernos 1000 dañando todo el corte, la mesa y con esto provocar un accidente.
Mensaje Ordenado	<p>Si se quiere formar un cuadrado (+X, +Y, -X, -Y) es importante seguir un orden, no sirve movernos en el eje X regresar por el eje X y moverse en el eje Y, si no se cumple el orden, la pieza no va a ser la misma.</p>  <p style="text-align: center;">FIG. 20 CORTE CON Y SIN ORDEN</p>
Pedir mensaje	El software no sabe cuánto se demora en ejecutar una instrucción, por esto se debe pedir el siguiente mensaje cuando se lo necesitamos.
Tiempo de llegada	No se puede tener tiempo de espera, apenas termina una instrucción debe comenzar con la siguiente, por eso se debe tener listo por lo menos un mensaje
Mensaje ejecución rápida	Se necesita también poder enviar mensajes que se ejecuten instantáneamente, sin esperar.
Aviso de recepción	Se necesita estar seguros que el mensaje llego, que la tarjeta informe que la comunicación fue correcta.

Cero	Se debe poder enviar una instrucción para eliminar cualquier instrucción y empezar con todo desde cero.
------	---

TABLA 9
OBJETIVOS DEL PROTOCOLO

Mensaje completo y correcto

Para garantizar que el mensaje es correcto, se va a utilizar una estructura, esto es unas reglas que debe cumplir el mensaje para que sea aceptado como correcto.

- El mensaje estará conformado por 12 caracteres ASCII.
- El primer símbolo debe ser una H.
- Los siguientes 8 caracteres conforman la sección de datos. Estos están divididos de la manera siguiente:
 - Primer carácter representa la dirección del corte o el movimiento, explicado en la Tabla 11.
 - Siguiendo tres caracteres representan el número de pasos en X.
 - Siguiendo tres caracteres representan el número de pasos en Y.
 - El último carácter se mantuvo en caso de que fuera necesario a futuro pero actualmente no realiza ninguna función, es llenado con un símbolo ASCII aleatorio.
- El décimo carácter representa el número de mensaje para mantener la secuencia, limitado a valores del 0 al 9 o en el caso de mensajes rápidos está limitado a ser una F, indicando que el mensaje tiene prioridad. (Los mensajes rápidos serán explicados más adelante.)
- Los dos últimos caracteres que se encuentran al finalizar el mensaje se usan para comprobar la integridad del mensaje siendo una suma de los 10 anteriores caracteres.

Los caracteres ASCII son interpretados por la tarjeta de la manera que se muestra en la Tabla 10.

Ascii a número

0 = 48	= = 61	J = 74	W = 87	d = 100	q = 113
1 = 49	> = 62	K = 75	X = 88	e = 101	r = 114
2 = 50	? = 63	L = 76	Y = 89	f = 102	s = 115
3 = 51	@ = 64	M = 77	Z = 90	g = 103	t = 116
4 = 52	A = 65	N = 78	[= 91	h = 104	u = 117
5 = 53	B = 66	O = 79	\ = 92	i = 105	v = 118
6 = 54	C = 67	P = 80] = 93	j = 106	w = 119
7 = 55	D = 68	Q = 81	^ = 94	k = 107	x = 120
8 = 56	E = 69	R = 82	_ = 95	l = 108	y = 121
9 = 57	F = 70	S = 83	` = 96	m = 109	z = 122
: = 58	G = 71	T = 84	a = 97	n = 110	{ = 123
; = 59	H = 72	U = 85	b = 98	o = 111	= 124
< = 60	I = 73	V = 86	c = 99	p = 112	} = 125

TABLA 10
CONVERSIÓN ASCII A NÚMEROS

Esta tabla se determinó en base a los símbolos que la tarjeta F2808 puede enviar y recibir.

La tarjeta está programada para restar 48 a cualquier número que recibe, así el 0 (equivalente a 48 en la Tabla 10), se convierte en 0 también para la tarjeta. Ya que el símbolo más grande está representado por el número 125, al restarle 48 obtenemos 77. Esto significa que cada carácter enviado puede ir del 0 al 77.

La estructura del mensaje es la siguiente:

	Header	Direccion	Pasos en X			Pasos en Y			Padding	Orden	Suma	
			X	X	X	Y	Y	Y			S	S
Ejemplo	H	A	:	3	4	<	2	3	:	2	8	8
Interpretacion tarjeta	24	17	10	3	4	12	2	3	11	2	8	8

FIG. 20
ESTRUCTURA DEL MENSAJE

En la Figura 20 podemos ver un ejemplo de mensaje y como lo interpreta la tarjeta, primero haciendo la conversión según la Tabla 10 y después restando 48 a cada número.

Header

El primer elemento de cada mensaje va a ser una H, la tarjeta convierte esto en el número 72 y al restar 48 obtiene 24. Este campo se usa para indicarle a la tarjeta que es el comienzo de un mensaje y debe tomar los siguientes 11 caracteres como parte de un mismo mensaje.

Dirección y corte antorcha

El siguiente campo hace referencia a la dirección del corte o el movimiento, los valores específicos se muestran en la Tabla 11.

Código #	Instrucción
A	Traslado X positivo, Y positivo, antorcha apagada
B	Traslado X positivo, Y negativo, antorcha apagada
C	Traslado X negativo, Y positivo, antorcha apagada
D	Traslado X negativo, Y negativo, antorcha apagada
E	Traslado X positivo, Y positivo, antorcha prendida
F	Traslado X positivo, Y negativo, antorcha prendida
G	Traslado X negativo, Y positivo, antorcha prendida
H	Traslado X negativo, Y negativo, antorcha prendida

TABLA 11
DIRECCIÓN Y CORTE ANTORCHA

Pasos en X y pasos en Y

Para enviar la información del número de pasos en X y en Y se usan 6 caracteres, 3 para cada eje. Este límite se definió porque se lo considero suficiente, de ser necesario podría aumentarse, esto genera un límite máximo de pasos por instrucción.

Se tiene 78 diferentes símbolos como se puede observar en la tabla Ascii (Tabla 10), ya que solo podríamos expresar números del 0 al 77 se podría enviar como número máximo:

77 77 77

Para simplificar y tener los números en una base decimal y poder contar así desde el primer número hasta el último de manera continua, se van a restringir los dos últimos símbolos solo a números del 0 - 9 (esto se realiza dentro del software a la hora de generar los mensajes) y el primer símbolo con todos los ASCII, con esto queda como máximo:

77 9 9 = 7799

Esto nos permite pasar desde el 0 hasta el 7799 continuamente, de la otra forma solo podíamos expresar números desde el 0 hasta el 77, esto nos obligaría a pasar del 77 al 100 ya los números entre el 78 y el 99 no estarían definidos.

Esto quiere decir, que máximo se puede enviar instrucciones que contengan 7799 pasos, así como la velocidad máxima no podrá superar los 7799 mm/s. Si se tiene instrucciones más grandes, el software deberá dividir las en varias instrucciones pequeñas. No existe ningún límite para la cantidad de instrucciones.

Padding

El siguiente campo se usa como relleno, no tiene un uso en el protocolo actual, pero decidió dejarse en caso de necesitar un carácter extra en el futuro. En el ejemplo de la Figura 20, este campo tiene el carácter “;”, como se explicó este carácter no cumple ninguna función en protocolo.

Secuencia

Como se necesita enviar un mensaje con orden se va a añadir un código a cada mensaje, este será el último símbolo antes de la suma. Cada mensaje tendrá un orden y al momento de enviarlos se mantiene la siguiente secuencia 1-2-3-4-5-6-7-8-9-0, al recibir los mensajes se comprobará que el orden sea correcto, es decir que se esté recibiendo el mensaje en secuencia y no haya mensajes desordenados. En el ejemplo de la Figura 20 el campo Orden tiene como dato el número 2. Esto nos indica que el anterior mensaje tenía el número 1 y el siguiente tendrá el 3.

Suma

Los últimos 2 campos están diseñados para asegurar la integridad del mensaje. Representan una suma de los anteriores 10 caracteres. La suma se calcula después de restar 48 al valor numérico de cada carácter como lo haría la tarjeta. Cuando el mensaje es recibido completamente, la tarjeta suma los 10 primeros valores y compara su resultado con el valor recibido en los últimos 2. Si son iguales asume que el mensaje llegó correctamente. En el ejemplo de la Figura 20 se puede ver que el mensaje completo tiene la forma: HA:34<23;289 siendo los dos últimos caracteres los representantes de la suma. Haciendo la conversión basados en la Tabla 10 y restando 48 al número correspondiente (explicado en la Sección: Ascii a número) obtenemos el siguiente resultado:

24 17 10 3 4 12 2 3 11 2 8 8

A continuación la conversión por pasos de cada carácter:

H -> 72-48 -> 24

A -> 65-48 -> 17

: -> 58-48 -> 10

3 -> 51-48 -> 3

4 -> 52-48 -> 4

< -> 60-48 -> 12

2 -> 50-48 -> 2

3 -> 51-48 -> 3

; -> 59-48 -> 11

2 -> 50-48 -> 2

8 -> 56-48 -> 8

8 -> 56-48 -> 8

Para obtener la suma, sumamos los diez primeros valores numéricos:

$$24+17+10+3+4+12+2+3+11 = 88$$

Así obtenemos el valor de los dos últimos campos, 88. Esto comprueba que el mensaje llegó sin errores.

Pedir Mensaje

En este punto no existe una comprobación de mensaje recibido, por lo que se va a utilizar un pedido de mensaje, esto funciona de la siguiente forma:

El software envía un mensaje, si el mensaje cumple con las reglas para un mensaje completo, correcto y en el orden adecuado, la tarjeta pedirá el siguiente mensaje, si el mensaje que llega no cumple con los parámetros esperados por la tarjeta o no llega nada, pasado 0.1 segundos la tarjeta pedirá el mismo mensaje nuevamente hasta que llegue el mensaje correcto. Esto se realiza para asegurarse que el software siga el flujo de información, ya que el software no envía mensajes a menos que la tarjeta lo pida. El formato del mensaje enviado por la tarjeta al software será "M" + # mensaje a pedir.

Tiempo de Llegada

Para garantizar que la información de la instrucción siguiente esté lista se debe tener un buffer de instrucciones, el buffer almacena una instrucción, esto quiere decir que se va a pedir una instrucción antes de necesitarla, mientras estamos ejecutando la anterior.

Mensaje ejecución rápida

Existen ciertos casos donde se debe ejecutar una instrucción rápidamente, sin pasar por el buffer, en estos casos se va a utilizar un código especial el cual se pondrá en lugar del valor de secuencia, este código será la letra 'F'. Un mensaje de ejecución rápida puede ser enviado para cambiar la velocidad de corte o la velocidad de movimiento. En caso de que sea un cambio en velocidad de corte en el campo de Dirección se envía el carácter 'C' y si es un cambio en la velocidad de movimiento en ese campo se envía el carácter 'M'.

En este caso los valores usados para pasos en X serán cambiados por valores correspondientes a velocidad. Los campos usados para pasos en Y no se utilizan.

La estructura del mensaje rápido es igual al de los mensajes normales con la excepción de tener una F en lugar de un valor entre 0 y 9 en el campo de secuencia y usar el campo de dirección para expresar la diferencia entre velocidad de corte y de movimiento. Estos mensajes son ejecutados instantáneamente sin pasar por el buffer para que el cambio de velocidad se haga efectivo antes de ejecutar la siguiente instrucción de movimiento o de corte.

Aviso de recepción

Los mensajes normales no necesitan confirmación pero los mensajes rápidos si, para enviarlos nuevamente en caso de no llegar, por esta razón la tarjeta enviara una 'F' cuando llegue el mensaje rápido, si el software no recibe la respuesta, después de un tiempo determinado reenvía el mensaje rápido, hasta tener el mensaje rápido confirmado. Si se pierde la confirmación no habrá problema porque la tarjeta reenviará la misma información. Si pasa el tiempo definido se reenvía el mensaje. Cuando la tarjeta confirma el software deja de enviarlo.

Cero

Se necesita un método para un reset de la tarjeta, que elimine las velocidades guardadas y cualquier instrucción en memoria. Esto también funciona al prender la tarjeta por que podría quedar alguna basura de una operación anterior.

El formato será el siguiente:

Sf RRRRRRRRRRRR
T z

En este caso no se comprueba la suma, si llega un mensaje completo solo con R se llevan todos los valores a cero en la tarjeta y se elimina cualquier instrucción almacenada en el buffer.

La tarjeta continuara enviando z cada cierto tiempo avisando que está en blanco.

Orden del Protocolo

Con el diseño del protocolo listo se debe generar una orden, para esto se va a seguir los siguientes pasos que usan todas las instrucciones diseñadas anteriormente.

- Sf comienza enviando RRRRRRRRRRRR y repite cada 0,5 s
- Tarjeta realiza proceso de cero y envía respuesta “z”, repite cada 0,1 s
- Cuando tarjeta recibe z envía mensaje para guardar velocidad de corte con el siguiente formato:

H C ### ???? F SS

H = Header

C = Código identificación velocidad de corte

= Valor a actualizar de velocidad de corte

? = Valor sin importancia no se lo toma en cuenta

F = Código mensaje rápido

S = Comprobación por suma

- Sf repite mensaje hasta recibir confirmación
- T después de guardar el nuevo valor de velocidad envía “F”
- Sf después de recibir F envía velocidad de movimiento con el siguiente

formato

H M ### ???? F SS

H = Header

M = Código identificación velocidad de movimiento

= Valor a actualizar de velocidad de movimiento

? = Valor sin importancia no se lo toma encuentra

F = Código mensaje rápido

S = Comprobación por suma

- Sf repite mensaje hasta recibir confirmación
- T pide el primer mensaje enviando "M1", y repite cada 0.1s
- Sf envía mensaje solicitado asumiendo que lo recibió correctamente sin pedido de confirmación
- T pide el mensaje siguiente con "M2", repite cada 0.1s
- Sf envía mensaje solicitado asumiendo que lo recibió correctamente sin pedido de confirmación
- (Se repite el proceso desde M3 hasta M8)
- T pide el mensaje siguiente enviando "M9", y repite cada 0.1s
- Sf envía mensaje solicitado asumiendo que lo recibió correctamente sin pedido de confirmación
- T pide el mensaje siguiente con "M0", repite cada 0.1s
- Sf envía mensaje solicitado asumiendo que lo recibió correctamente sin pedido de confirmación
- T pide el mensaje siguiente enviando "M1", y repite cada 0.1s
- Sf envía mensaje solicitado asumiendo que lo recibió correctamente sin pedido de confirmación
- (Se repite el proceso mientras existan mensajes)
- Para finalizar Sf envía RRRRRRRRRRRR
- T se mantiene enviando "z" para mostrar un estado de espera.

CAPITULO IV

Calibración y Pruebas

Metodología a ser utilizada

Para realizar correctamente las pruebas de la máquina se va a comprobar el correcto funcionamiento de la misma en los movimientos y las velocidades.

Primer se van a realizar movimientos separados en los ejes y movimientos combinados, se utilizará un marcador, para dibujar sobre una superficie líneas, arcos y polígonos. Mediante reglas y calibradores se comprobarán los tamaños obtenidos y se compararán con los planos originales, luego de esto se calibrará el software para corregir cualquier error y se comprueban nuevamente los dibujos hasta que estén dentro del rango de error adecuado. El rango de error aceptable de 1mm fue definido por los futuros usuarios de la máquina en base a su experiencia en el corte de acero. Mientras se realizan estas pruebas también se debe tomar datos del tiempo que se demora el corte, para poder de la misma forma corregir el software a tiempos de corte adecuados.

Pruebas de dibujo

Datos pruebas de dibujo

Tipo	Dibujo	Longitud (mm)	Grados de Inclinación	Distancia X (mm)	Distancia Y (mm)	Distancia XY (mm)
Teórica	línea	1000	0	1000	0	1.000
Real	línea	1030	0	1030	0	1.030
Teórica	línea	1000	30	866	500	1.000
Real	línea	1029	30	890	516	1.029
Teórica	línea	1000	45	707	707	1.000
Real	línea	1031	45	725	733	1.031
Teórica	línea	1000	60	500	866	1.000
Real	línea	1032	60	517	889	1.029
Teórica	línea	1000	90	0	1000	1.000
Real	línea	1030	90	0	1.030	1.030
Teórica	línea	500	0	0	500	500
Real	línea	514	0	0	514	514
Teórica	línea	500	30	250	433	500
Real	línea	515	30	255	447	515
Teórica	línea	500	45	354	354	500
Real	línea	515	45	362	366	515
Teórica	línea	500	60	433	250	500
Real	línea	516	60	448	256	516

Tipo	Dibujo	Longitud (mm)	Grados de Inclinación	Distancia X (mm)	Distancia Y (mm)	Distancia XY (mm)
Teórica	línea	500	90	500	0	500
Real	línea	514	90	514	0	514

TABLA 12
DATOS PRUEBAS LÍNEA

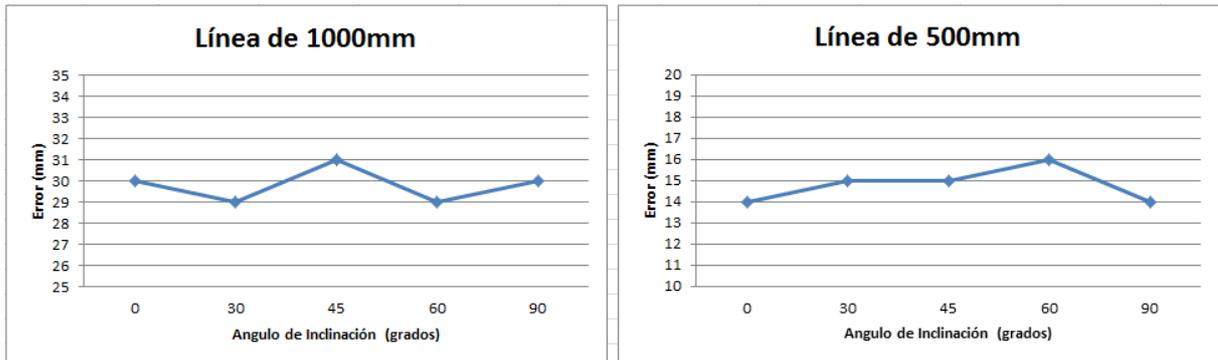


FIG. 21
GRÁFICOS DE PRUEBAS DE LÍNEAS

Datos pruebas de dibujo Arco y Semicírculo

Tipo	Dibujo	Radio (mm)	Perímetro (mm)	Distancia max X (mm)	Distancia min Y (mm)	Diferencia entrada y salida (mm)
Teórica	Semicírculo	500	1570,795	1000	500	1000
Real	Semicírculo	515	1617,91885	1030	515	1030
Teórica	Semicírculo	100	314,159	200	100	200
Real	Semicírculo	103	323,58377	206	103	206
Teórica	Círculo	500	3141,59	1000	1000	0
Real	Círculo	515	3235,8377	1030	1030	5
Teórica	Círculo	100	628,318	200	200	0
Real	Círculo	103	647,16754	206	206	3

TABLA 13
DATOS PRUEBAS ARCO Y SEMICÍRCULO

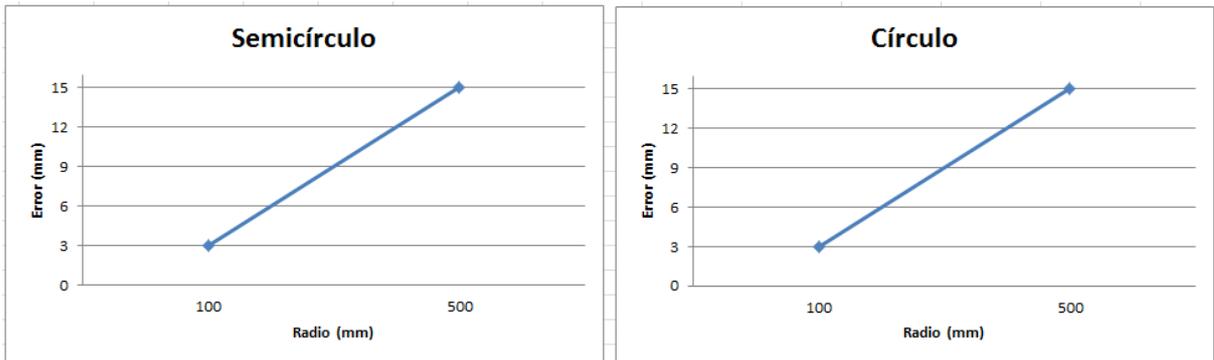


FIG. 22
GRÁFICOS DE PRUEBAS DE CÍRCULO Y SEMICÍRCULO

Datos Tiempo

Se van a realizar las siguientes pruebas:

- Línea 500 mm en diferentes ángulos.

Datos prueba de dibujo Línea 500 mm

Grados de inclinación	Distancia recorrida (mm)	Número de pasos	Tiempo (s) Prueba 1	Tiempo (s) Prueba 2	Promedio (s)	Velocidad (mm/s)	Velocidad (Pasos/s)
0	500	12.500	18,09	18,17	18,13	27,58	689
1	500	12.500	19,21	19,29	19,25	25,97	649
5	500	12.500	19,33	19,37	19,35	25,84	646
30	500	12.500	19,50	19,64	19,57	25,55	639
45	500	12.500	19,31	19,46	19,385	25,79	645
60	500	12.500	19,68	19,47	19,58	25,54	639
89	500	12.500	19,41	19,16	19,29	25,93	648
90	500	12.500	18,09	18,17	18,13	27,58	689

TABLA 14
DATOS PRUEBAS LÍNEA 500MM

- Línea 1000 mm en diferentes ángulos

Datos prueba de dibujo Línea 1000 mm

Grados de inclinación	Distancia recorrida (mm)	Número de pasos	Tiempo (s) Prueba 1	Tiempo (s) Prueba 2	Promedio (s)	Velocidad (mm/s)	Velocidad (Pasos/s)
0	1000	25.000	36,10	36,12	36,11	27,69	692

1	1000	25.000	38,44	38,43	38,44	26,02	650
5	1000	25.000	38,81	38,80	38,81	25,77	644
30	1000	25.000	38,74	38,81	38,78	25,79	645
45	1000	25.000	38,90	38,82	38,86	25,73	643
60	1000	25.000	38,91	38,98	38,95	25,68	642
89	1000	25.000	38,44	38,48	38,46	26,00	650
90	1000	25.000	36,07	36,12	36,10	27,70	693

TABLA 15
DATOS PRUEBAS LÍNEA 1000MM

- Círculo con diferentes Radios

Datos prueba de dibujo Círculo

Prueba	Número de pasos	Distancia recorrida (mm)	Tiempo (s) Prueba 1	Tiempo (s) Prueba 2	Promedio (s)	Velocidad (mm/s)	Velocidad (Pasos/s)
R 500 mm	78.540	3.142	120,66	120,57	120,62	26,05	651
R 100 mm	15.708	628	24,16	24,17	24,17	26,00	650

TABLA 16
DATOS PRUEBAS CÍRCULO

Análisis Datos Tiempo

La prueba comienza con un número 1000, este número se envía al motor como frecuencia a la cual debe girar, en base a los datos tomados, se va a determinar la velocidad real en mm/s.

Conclusiones prueba de tiempo

Velocidad Máxima (mm/s)	Velocidad Mínima (mm/s)	Promedio de velocidad (mm/s)	Desviación con Max (mm/s)	Desviación con Min (mm/s)
27,70	25,54	26,23	1,5	0,69

TABLA 17
CONCLUSIONES PRUEBAS TIEMPO

Como se puede observar en la Tabla 17, cuando el motor recibe una frecuencia de 1000, se mueve a una velocidad promedio de 26.23 mm/s.

A continuación la Tabla 18 muestra las velocidades transformadas a mm/s. (Datos tomados del manual Hypertherm para equipos plasma).

Velocidad de corte

Tipo de Acero	Espesor(mm)	Velocidad de Corte Máxima(mm/s)	Velocidad de Corte Óptima (mm/s)
Acero al Carbono	0,8	211,67	137,58
	6,4	31,33	20,32
	12,7	25,40	16,52
	25,4	6,767	4,233
Acero Inoxidable	0,8	209,97	136,32
	6,4	19,90	13,12
	12,7	21,17	13,97
	25,4	5,933	3,817
Aluminio	0,8	258,23	168,07
	6,4	32,17	20,75
	12,7	31,75	15,67
	Max	258,23	168,07
	Min	5,93	3,82

TABLA 18
VELOCIDADES DE CORTE

Pruebas de corte

Pruebas calidad de corte

Plancha acero inoxidable	Figura	Distancia / Radio (mm)	Calidad	Velocidad (mm/ s)
2 mm	Línea	100	Media	100
	Círculo	100	Buena	90
3mm	Línea	100	Buena	75
	Círculo	100	Media	80
5mm	Línea	100	Buena	25
	Círculo	100	Buena	25
6mm	Línea	100	Media	13
	Círculo	100	Buena	15

TABLA 19
PRUEBAS CALIDAD DE CORTE

Correcciones Realizadas

En base a los cálculos de las tablas presentadas anteriormente se pudo calcular la longitud del paso de la siguiente forma:

Distancia recorrida = Distancia de paso x Número de pasos

Distancia de paso = Distancia recorrida / Número de pasos

Como se puede observar en las Tablas 12 y 13, y en las Figuras 21 y 22 se tiene un error aproximado de 30 mm por cada 1000 mm, esto es debido a un incorrecto valor de cada paso, por lo tanto se debe determinar el valor correcto de paso y así arreglar este problema.

Teóricamente el software está enviando una instrucción de $1000/0,04 = 25000$ pasos, pero se ve que la distancia recorrida en cada paso es en realidad es $1030/25000 = 0,0412$ mm, con este valor corregido en el software se tienen resultados precisos.

Como se puede observar en la Tabla 18 la velocidad máxima será 258.23 mm/s mientras que la mínima es 3.82 mm/s. Para no modificar el programa en la tarjeta y simplemente calibrar el software se va a calibrar de la siguiente forma:

Si al enviar 1000 como frecuencia al motor se tenía una velocidad de 26.23 mm/s entonces:

Constante a multiplicar = $\frac{1000}{26.23} = 38.12$. Se usara esta constante para multiplicar por ella los valores de velocidad que queramos hacer llegar a la tarjeta. En la Tabla 20 a continuación se puede observar la conversión de los valores máximos y mínimos de velocidad.

Código velocidades

Velocidad (mm/s)	Número a enviar por protocolo
26,23	1000
258,23	9844,83415935951
3,82	145,634769348075

TABLA 20
CÓDIGOS VELOCIDADES

Como se puede observar en la tabla anterior, se debe enviar como máximo el 9845 por el protocolo, pero puesto que se tiene un límite físico de 7799, se va a modificar al programa en la tarjeta para que acepte cuatro caracteres como velocidad, ya que la velocidad se envía mediante mensajes rápidos y estos tenían 4 caracteres que no se usaban, es un cambio bastante sencillo. En esencia se marcó uno de estos caracteres no usados como parte de lo que la tarjeta toma como el dato de velocidad.

Recomendaciones para pruebas.

- Organizar correctamente las pruebas y tener los archivos listos antes de comenzar.
- Utilizar un método preciso para la medida de tiempos en especial cuando deben ser muy cortos ya que mientras más corto es el tiempo, más impacto tiene el error humano.
- Identificar las velocidades máximas y mínimas que la máquina puede ejecutar sin problemas mecánicos de inercia o precisión. Esto generalmente se puede encontrar en los manuales de los motores o realizando pruebas tomando en cuenta el margen de error que se considera aceptable. En nuestro caso los futuros usuarios definieron el error aceptable en 1mm.

CONCLUSIONES

- Se cumplió con el diseño e implementación de un protocolo de comunicación y un software de control para una mesa CNC de corte por plasma, el cual lee archivos de AutoCAD en formato DXF.
- Al diseñar por completo el software permite una completa manipulación para adaptarlo a cualquier trabajo fácilmente. El conocimiento obtenido sobre, protocolos de comunicación en base a puerto serial, programación en C#, programación de eZdsp F2808 son conocimientos obtenidos en base a la resolución de problemas.
- Gracias a la conclusión de este proyecto de tesis, INTERINOX queda muy satisfecho con una máquina diseñada y construida para sus necesidades, la cual facilitara de gran manera el trabajo dentro de la planta al momento de cortar figuras en acero inoxidable. Aun no tenemos datos puntuales sobre la mejora de procesos ya que la maquina aún no se encuentra en completo funcionamiento.
- En relación a las metas propuestas se logró desarrollar un software de control funcional sin incurrir en el precio comercial de los mismos, tomando en cuenta que el software para este tipo de máquinas puede variar en su costo entre 1000 y 4000 dólares [9] y es difícil encontrar proveedores que lo vendan sin la maquina respectiva. El software no fue cobrado al cliente y por lo tanto se generó un ahorro importante de dinero. Cambiando la programación del software se puede ajustar el funcionamiento para varias necesidades distintas. En el futuro se planifica cargar varios programas para diferentes tipos de trabajos.
- Después de un largo tiempo en este proyecto, la conclusión es muy satisfactoria, y se encuentra listo para continuar con una segunda etapa, donde se incorporará un control automático de altura de la antorcha.
- A continuación tenemos algunas fotografías del producto actual.



FIG. 23
MÁQUINA COMPLETA

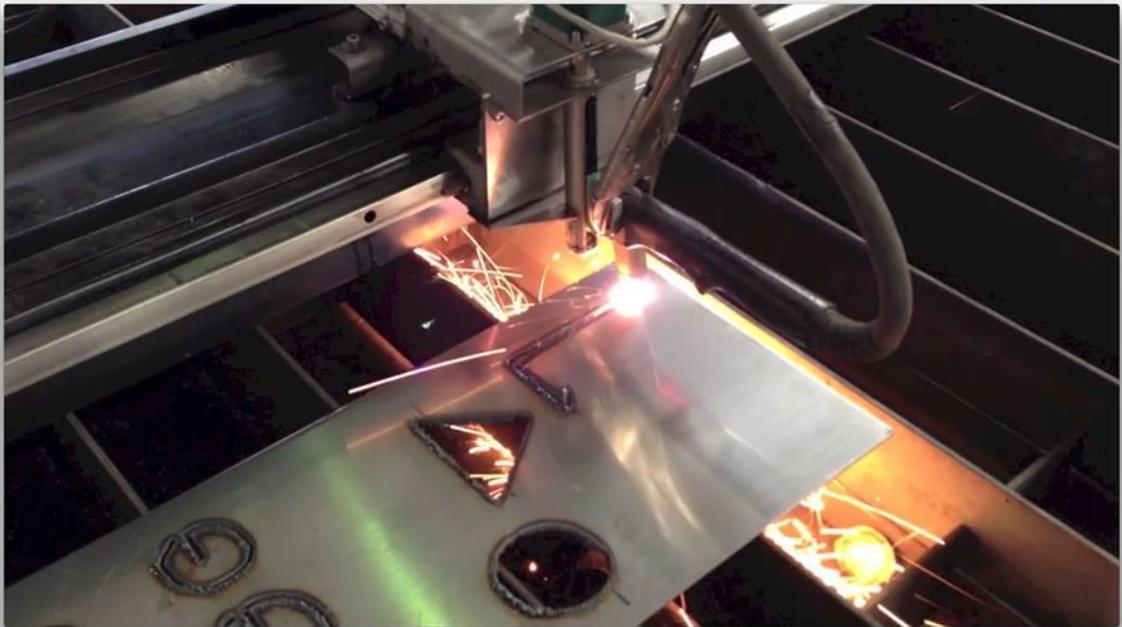


FIG. 24
PRUEBA DE CORTE 1



FIG. 25
PRUEBA DE CORTE 2

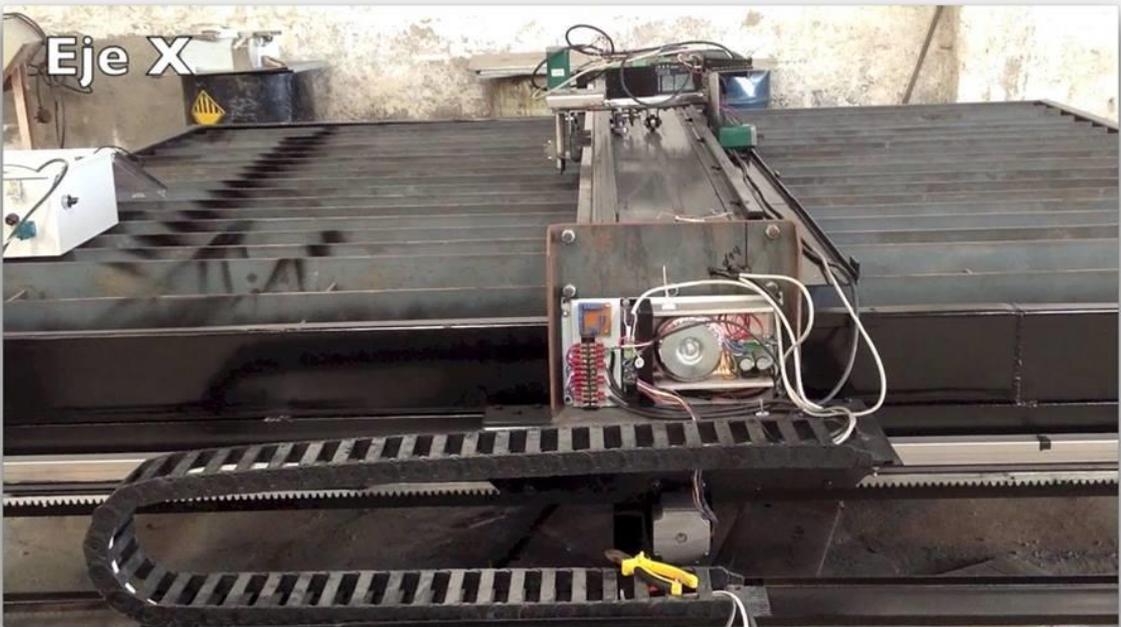


FIG. 26
EJE X

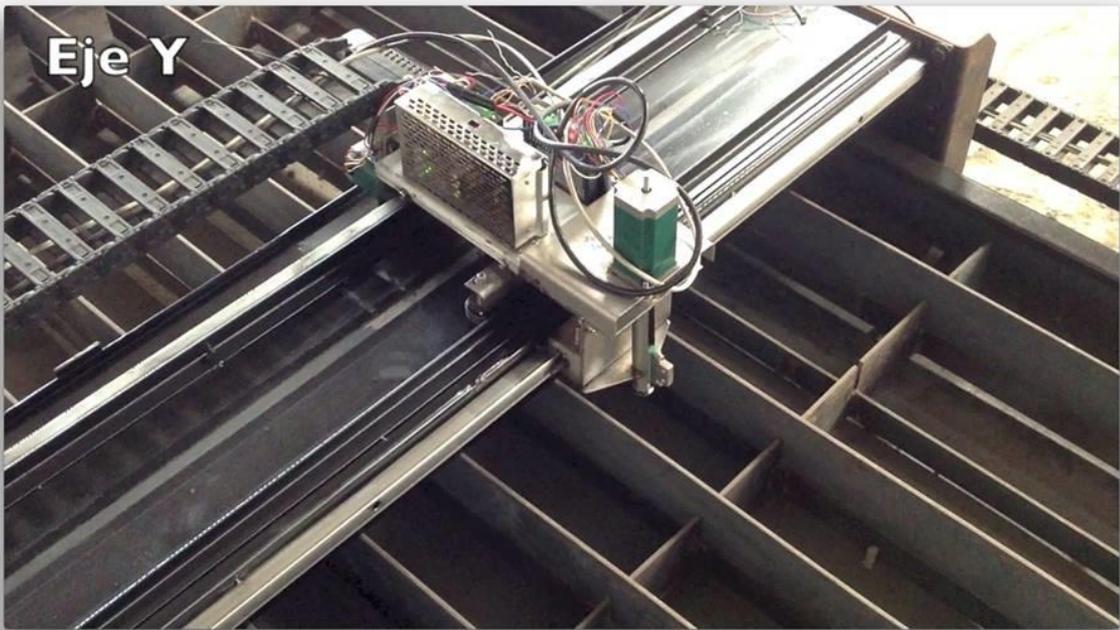


FIG. 27
EJE Y



FIG. 28
TARJETA DE CONTROL

BIBLIOGRAFÍA

- 1. Wikipedia. Control numérico.
<http://es.wikipedia.org/w/index.php?title=Control_num%C3%A9rico&oldid=56838068>
- 2. Hypertherm. Aplicaciones;
- 3. Hypertherm. Beneficios;
- 4. Wikipedia, DWG. Wikipedia, La enciclopedia libre:
< <http://es.wikipedia.org/wiki/DWG>>
- 5. Wikipedia, RS-232. Wikipedia, La enciclopedia libre:
< <http://es.wikipedia.org/wiki/RS-232>>
- 6. Wikipedia, C.d.e.w.o.w.i.p.t.T.C.A.o.e. Trigonometría. Wikipedia, La enciclopedia libre:
<<http://es.wikipedia.org/w/index.php?title=Trigonometr%C3%ADa&oldid=61741923>>
- 7. Wikipedia, Protocolo de comunicaciones. Wikipedia, La enciclopedia libre:
< http://es.wikipedia.org/wiki/Protocolo_de_comunicaciones>
- 8. Polilíneas < <http://www.scribd.com/doc/17593519/POLILINEA>>
- 9. Minitex Machinery Corporation < <http://www.minitex.com/> >
- 10. Overby, A., CNC machining handbook: building, programming, and implementation. 1st ed 2011, New York, NY: McGraw-Hill/TAB Electronics. xii, 260p.
- 11. AMS. Step101. 2010;
<<http://www.ams2000.com/pdf/step101.pdf>>
- 12. Source, C.R. Stepper vs Servo Motors.
< <http://www.cncroutersource.com/stepper-vs-servo.html> >
- 13. Torchmate. Stepper Motors vs. Servo Motors vs. Intelligent Motors -The Facts.
<http://torchmate.com/resource_center/editorials/stepper_motors_vs._servo_motors_vs._intelligent_motors_-_the_facts/ >
- 14. Probotix. CNC Routers, Plasma Tables, & CNC Control Systems. 2012 < <http://www.probotix.com> >