



**UNIVERSIDAD SAN FRANCISCO DE QUITO**  
**COLEGIO POLITÉCNICO**

**Diseño, construcción, 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.**

**Martín Nicolás Reinoso Naranjo**

**Omar Aguirre, M.Sc, Director de Tesis**

Tesis de grado presentada como requisito para la obtención del título de ingeniero eléctrico electrónico

Quito

Diciembre 2012

**Universidad San Francisco de Quito**

**Colegio Politécnico**

## **HOJA DE APROBACION DE TESIS**

**Diseño, construcción, programación de software y protocolos de transferencia de datos de una mesa de corte CNC por plasma a ser usada en el corte de planchas de acero inoxidable.**

**Martín Nicolás Reinoso Naranjo**

Omar Aguirre, M.Sc. Ingeniería Eléctrica

Director de tesis .....

Miembro del Comité de Tesis

Nelson Herrera, Ingeniero de Sistemas

Miembro del Comité de Tesis .....

René Játiva, Ph.D

Miembro del Comité de Tesis .....

Santiago Gangotena, Ph.D

Decano colegio politécnico .....

Quito, Diciembre 2012

## © 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: Martín Nicolás Reinoso Naranjo

C. I.: 1712462025

Fecha: Diciembre 2012

## 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. Pero 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 detallado de como construirla 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 to build it in this document.

## TABLA DE CONTENIDO

<b>Lista de figuras</b>	<b>x</b>
<b>Objetivos</b>	<b>xv</b>
Objetivo final	xv
Objetivos personales	xv
Objetivos específicos	xv
Metas	xvi
<b>Capitulo I ¿QUÉ ES UNA MESA CNC PARA CORTE POR PLASMA?</b>	<b>1</b>
Definición CNC (Control numérico por computadora)	1
Mesa CNC y sus aplicaciones (Corte de metales)	2
Ventajas de usar una CNC sobre trabajo manual	3
Partes de una mesa CNC (Introducción)	5
<i>Partes mecánicas</i>	5
<i>Partes Electrónicas</i>	6
<i>Software</i>	8
<i>Comunicación entre Software y Hardware</i>	9
Corte por plasma	9
<b>Capitulo II Diseño mecánico</b>	<b>11</b>
Requerimientos del cliente	11
Diseño de la máquina	13
<i>Parte Estática</i>	13
<i>Parte Móvil</i>	17

<b>Construcción parte mecánica</b>	<b>23</b>
<b>Recomendaciones construcción parte mecánica</b>	<b>23</b>
<b>Capitulo III Diseño Electrónico</b>	<b>25</b>
<b>Requerimientos del cliente</b>	<b>25</b>
<i>Análisis requerimientos del cliente</i>	<i>26</i>
<b>Diseño Electrónico de la máquina</b>	<b>29</b>
<i>Motores</i>	<i>29</i>
<i>Control Motores</i>	<i>37</i>
<i>Análisis de potencia</i>	<i>53</i>
<i>Determinación de cables a utilizar.</i>	<i>54</i>
<i>Tablero de control</i>	<i>59</i>
<i>Armado del tablero.</i>	<i>63</i>
<i>Armado parte eléctrica en máquina</i>	<i>66</i>
<b>Instalación de software y paso de información a la tarjeta.</b>	<b>68</b>
<i>Instalación de software</i>	<i>68</i>
<i>Generar código para correr en tarjeta</i>	<i>69</i>
<i>Probar programa en tarjeta</i>	<i>70</i>
<b>Recomendaciones parte electrónica</b>	<b>73</b>
<b>Capitulo IV Diseño Software de control</b>	<b>75</b>
<b>Requerimientos del cliente</b>	<b>75</b>
<b>Diseño de la aplicación</b>	<b>76</b>
<i>Leer archivos DXF</i>	<i>76</i>
<i>Comunicación serial</i>	<i>76</i>
<i>Interfaz usuario</i>	<i>76</i>
<i>Facilidades de uso para usuario</i>	<i>77</i>

<i>¿Por que usar c#?</i>	77
<b>Organización archivos DXF</b>	<b>78</b>
<i>Linea</i>	78
<i>Arco</i>	79
<i>Elipse</i>	79
<i>Circulo</i>	79
<i>Polyline</i>	79
<i>LwPolyline</i>	80
<b>Matemática utilizada para transformar la información</b>	<b>80</b>
<i>Trigonometría [9]</i>	80
<i>Linea en base a dos puntos</i>	81
<i>Arco, Circulo y Elipse a líneas</i>	81
<i>Arco definido como Bulge</i>	82
<b>Explicación del código</b>	<b>83</b>
<i>Código objetos que no interactua con el usuario</i>	83
<i>Código de objetos que interactuan con el usuario</i>	86
<b>Capitulo V Protocolo de comunicación</b>	<b>90</b>
<b>Que es un protocolo?</b>	<b>90</b>
<b>Diseño de protocolo</b>	<b>91</b>
<i>Mensaje completo y correcto</i>	92
<i>Mensaje Ordenado</i>	93
<i>Pedir Mensaje</i>	94
<i>Tiempo de llegada</i>	95
<i>Mensaje ejecución rápida</i>	95
<i>Aviso de recepción</i>	95
<i>Cero</i>	96

<i>Orden del Protocolo</i>	96
<i>Información en mensaje normal</i>	98
<i>Tiempo de llegada de la información</i>	99
<b>Limites y restricciones</b>	<b>99</b>
<b>Programación protocolo en tarjeta F2808</b>	<b>100</b>
<i>Funciones utilizadas</i>	100
<b>Programación protocolo en software</b>	<b>107</b>
<i>Funciones utilizadas</i>	107
<b>Recomendaciones para desarrollo de protocolo de comunicación</b>	<b>111</b>
<b>Capitulo VI Calibración y pruebas</b>	<b>112</b>
<b>Metodología a ser utilizada</b>	<b>112</b>
<b>Pruebas de dibujo</b>	<b>112</b>
<b>Tiempo</b>	<b>113</b>
<i>Datos Tiempo</i>	113
<i>Análisis Datos Tiempo</i>	114
<i>Pruebas de corte</i>	115
<b>Correcciones realizadas</b>	<b>115</b>
<b>Recomendaciones para calibración y pruebas</b>	<b>117</b>
<b>Conclusiones</b>	<b>118</b>
<b>Bibliografía</b>	<b>119</b>
<b>Anexos</b>	<b>120</b>
<b>Anexo A</b>	<b>120</b>
<i>Planos Construcción Mecánica</i>	120
<i>Conexión por Colores Cables de Control</i>	135

<b>Anexo B</b>	<b>137</b>
<i>Referencia técnica de tarjeta EzDSP F2808</i>	<b>137</b>
<i>DataSheet motores</i>	<b>163</b>
<i>Datasheet Drivers</i>	<b>164</b>
<i>Datasheet fuentes</i>	<b>177</b>
<i>Datasheet transistor 2N2222</i>	<b>178</b>
<i>Tarjeta acople EzDSP F2808</i>	<b>179</b>

## LISTA DE FIGURAS

Tipo	Número	Descripción
Imagen	1	Mesa CNC con los 3 ejes
Imagen	2	Trabajador usando corte por plasma manual
Imagen	3	Detalle de corte manual
Imagen	4	Corte por plasma
Imagen	5	Dos planchas tamaño estándar en área de trabajo.
Imagen	6	Una plancha grande en área de trabajo.
Imagen	7	Pata con pie roscado
Imagen	8	Viga tipo UPN
Imagen	9	Ángulo Laminado
Imagen	10	Planchas y platinas de Acero
Imagen	11	Tuerca de seguridad
Imagen	12	Tornillo tuerca y rodela
Imagen	13	Base de la mesa CNC
Imagen	14	Pata con pie roscado
Imagen	15	Pie Roscado
Imagen	16	Tanque con cuello de limpieza
Imagen	17	Rejilla de material consumible
Imagen	18	Engranaje 14 dientes y cremallera
Imagen	19	Distancia al dar una vuelta Perímetro
Imagen	20	Tubo Cuadrado
Imagen	21	Rodamiento
Imagen	22	Eje X, vista parte lateral
Imagen	23	Detalle carros eje X
Imagen	24	Eje Y, vista frontal
Imagen	25	Eje Z
Imagen	26	Base Mesa
Imagen	27	Eje X
Imagen	28	Eje Y
Imagen	29	Mesa con tanque y consumible
Imagen	30	Area segura de corte según el cliente
Imagen	31	Funcionamiento motor a pasos
Imagen	32	Motor en Y

Tipo	Número	Descripción
Imagen	33	Motor en X
Imagen	34	Driver Y y Z
Imagen	35	Driver X
Imagen	36	eZdsp F2808
Imagen	37	eZdsp F2808 otras vistas
Imagen	38	eZdsp F2808 esquema
Imagen	39	Esquema tarjeta acople
Imagen	40	Esquema y tarjeta de acople final
Imagen	41	Señal sumatoria
Imagen	42	Señal cuadrada
Imagen	43	Señal cuadrada con cambio de frecuencia
Imagen	44	Bloque generador de señal a motor final
Imagen	45	Prueba Bloque generador de señal a motor final
Imagen	46	Prueba real con osciloscopio Bloque generador de señal a motor
Imagen	47	Bloque contador de pasos
Imagen	48	Bloque pulso en bajada
Imagen	49	Prueba Bloque pulso en bajada
Imagen	50	Triángulo Pitágoras
Imagen	51	Calculo de velocidades para mantener velocidad lineal
Imagen	52	Controlador de motor con contador de pasos
Imagen	53	Prueba Controlador de motor con contador de pasos
Imagen	54	Controladores de motores X y Y
Imagen	55	Prueba con osciloscopio Controladores de motores X y Y
Imagen	56	Interpretación de instrucciones dirección .
Imagen	57	Fuentes de poder utilizadas
Imagen	58	Cable 2x16
Imagen	59	Esquema cables de potencia
Imagen	60	Pruebas de señal en osciloscopio
Imagen	61	Cable apantallado
Imagen	62	Cadena Portacables
Imagen	63	Selector 2 posiciones
Imagen	64	Funcionamiento de Relay y Contactor
Imagen	65	Conexión cable serial cruzado
Imagen	66	Cable Serial a USB
Imagen	67	Canaleta Ranurada

Tipo	Número	Descripción
Imagen	68	Riel DIN
Imagen	69	Terminal U
Imagen	70	Tablero Metálico
Imagen	71	Braker, contactor y relay
Imagen	72	Conector puerto serial
Imagen	73	Selector dos posiciones instalado
Imagen	74	Fuentes para eZdsp y control
Imagen	75	Tablero finalizado
Imagen	76	Fuente y Driver motor X2
Imagen	77	Conexión Driver y motor
Imagen	78	Fuente y Driver motor X1
Imagen	79	Ventanas instalación software
Imagen	80	Generar código para tarjeta
Imagen	81	Generar código para tarjeta, Ram o Flash
Imagen	82	Crear configuración de tarjeta
Imagen	83	Final Crear configuración de tarjeta
Imagen	84	Selección tarjeta eZdsp f2808
Imagen	85	Código actual en tarjeta
Imagen	86	Cargar código a tarjeta
Imagen	87	Pins en SW1 para que cargue código de la memoria Flash
Imagen	88	Programa DXFViewer
Imagen	89	Programa proyecto tesis
Imagen	90	Trigonometría
Imagen	91	Hexágono en círculo
Imagen	92	Polígonos en círculo
Imagen	93	Definición Bulge
Imagen	94	Puntos iguales con diferente Bulge
Imagen	95	Diferencia entre imágenes con y sin grupo
Imagen	96	Representación de antorcha
Imagen	97	Figura seleccionada
Imagen	98	Izquierda figura vista por el usuario, derecha, figura con
Imagen	99	Orden antes de eliminar figuras
Imagen	100	Orden después de eliminar figuras
Imagen	101	Zoom bajo
Imagen	102	Simulación de corte

<b>Tipo</b>	<b>Número</b>	<b>Descripción</b>
Imagen	103	Corte con y sin orden
Imagen	104	Configuración puerto serial
Imagen	105	Buffer de datos
Imagen	106	Comprobación por suma y H
Imagen	107	Guardar variables por código
Imagen	108	Ascii a int
Imagen	109	Orden de mensaje
Imagen	110	Mantener último mensaje correcto
Imagen	111	Comprobar mensaje correcto
Imagen	112	Pedir mensaje siguiente
Imagen	113	Buffer de instrucción
Imagen	114	Mensaje rápido
Imagen	115	Dirección y corte antorcha
Imagen	116	Encontrar distancia a recorrer
Imagen	117	Encontrar frecuencias para X y Y
Imagen	118	Comprobar instrucción ZERO
Imagen	119	Comprobación que todo es ZERO
Imagen	120	Programa final
Tabla	1	Precisión en base a pasos del motor
Tabla	2	Velocidades optimas de corte
Tabla	3	Instrucciones dirección motores.
Tabla	4	Amperajes máximos de los componentes
Tabla	5	Amperaje con fuente para cada motor
Tabla	6	Consumo fuentes.
Tabla	7	Longitud cables
Tabla	8	Características tablero
Tabla	9	Consumo amperaje diferentes elementos
Tabla	10	Formato Linea en DXF
Tabla	11	Formato Arco en DXF
Tabla	12	Formato Ellipse en DXF
Tabla	13	Formato Círculo en DXF
Tabla	14	Formato Polyline en DXF
Tabla	15	Formato LwPolyline en DXF
Tabla	16	ArrayList usados
Tabla	17	Orden usado en ArrayList

<b>Tipo</b>	<b>Número</b>	<b>Descripción</b>
Tabla	18	Objetivos a cumplir con el protocolo
Tabla	19	Asci a número
Tabla	20	Dirección y corte antorcha
Tabla	21	Datos pruebas de dibujo
Tabla	22	Datos pruebas de dibujo Arco y Semicírculo
Tabla	23	Datos prueba de dibujo Linea 500 mm
Tabla	24	Datos prueba de dibujo Linea 1000 mm
Tabla	25	Datos prueba de dibujo Circulo
Tabla	26	Conclusiones prueba de tiempo
Tabla	27	Velocidad de corte
Tabla	28	Pruebas calidad de corte
Tabla	29	Código velocidades

## OBJETIVOS

### OBJETIVO FINAL

- Construir y dejar en completo funcionamiento una mesa CNC de corte por plasma que incluye el software de control de la máquina y que lee archivos de AutoCAD en formato DXF.

### OBJETIVOS PERSONALES

- Construir una mesa CNC con diseño completo.
- Tener control total sobre todos los aspectos.
- La parte electrónica como el software incorporarán comunicación diseñada desde la base.
- El software será completamente diseñado para esta máquina y no adaptado.
- Aprender a generar tecnología, no simplemente usarla.

### OBJETIVOS ESPECÍFICOS

- Diseño y construcción física de una mesa CNC en acero ASTM A-36, que pueda mantener 2 planchas de acero inoxidable tamaño estándar al mismo tiempo.
- Utilización de una tarjeta FPGA eZdsp F2808 para el control de motores.
- Creación de protocolo de comunicación entre tarjeta y software.
- Programación de software para leer archivos DXF de AutoCAD.
- Obtener mayor precisión, reducción de tiempo y costo comparado con el método actual de corte (manual).

- Obtener una máquina con características similares a las del mercado a un precio mucho menor.

## METAS

- Construir una mesa CNC que puede intercambiar su cabezal para realizar diferentes procesos, como pueden ser corte, taladrado, dibujo, pintura, etc... de diseños en 2 dimensiones.
- Entregar a Interinox S.A una máquina confiable que permita reducir el costo y tiempo para la fabricación de equipos en acero inoxidable.
- Diseñar un software de uso simple pero con gran capacidad, que permita fácilmente transformar planos de AutoCAD en piezas cortadas. El Software deberá correr en cualquier computadora con Windows XP y comunicación serial o USB.
- Aprender sobre la correcta comunicación entre software y hardware para lograr minimizar los errores y aprovechar al máximo la tecnología moderna.
- Construir una máquina con un precio 5 veces más bajo que una máquina similar en el mercado.

# CAPITULO I ¿QUÉ ES UNA MESA CNC PARA CORTE POR PLASMA?

## DEFINICIÓN CNC (CONTROL NUMÉRICO POR COMPUTADORA)

Para entender fácilmente lo que es y en que ayuda el control numérico por computadora les invito a imaginar la vida actual sin impresoras. Todo documento que se escribe en la PC se quedaría en eso, para verlo digitalmente, pero no se podría 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 al transcribir, así como va a perder el formato y la posición de los elementos, para añadir imágenes en un documento se necesitaría de un experto en dibujo e ilustración si se desea pasar del monitor al papel, si es solo en negro y un verdadero artista para ponerlo en colores.

Este gran cambio apareció 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 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 bolígrafo podría escribir documentos, pero si tiene un taladro puede hacer huecos sobre superficies. El límite de una CNC depende de la herramienta que se utilice que el

computador utilice, y en el caso de esta tesis, la herramienta es una mesa de corte mediante plasma.

## MESA CNC Y SUS APLICACIONES (CORTE DE METALES)

Como se ve en la definición de CNC, las aplicaciones son enormes, es por esto que este documento se centrará en una mesa para corte de metales. Con esto se verá en un solo campo la versatilidad de una CNC.



**Fig 1.** Mesa CNC con los 3 ejes  
<http://img.diytrade.com/cdimg/1518475/22729931/0/1312249543/XYZ->

Una mesa CNC es un soporte, el cual se mueve a una velocidad dada y en una dirección a cualquier herramienta que se desee utilizar. Al igual que en matemáticas

los ejes son X, Y y Z. En la imagen obtenida, se puede observar una mesa pequeña CNC con sus ejes básicos.

Puesto que la herramienta es lo que cambia el corte, aquí se presenta una lista de métodos de corte de metales, esta lista se obtiene en base al fabricante de herramientas llamado Hypertherm [2]

- Corte por Plasma

Este proceso permite cortar todo tipo de metales a una velocidad media pero con costos bajos convirtiéndola en el método ideal para la industria, en especial para acero inoxidable.

- Corte por Láser

Al ser una tecnología nueva que corta metales a alta velocidad, el precio es alto, es por esto que en la actualidad este corte sea poco usado.

- Corte por Oxido

Este proceso es el más antiguo, y pese a que es muy lento, por su bajo precio y por su facilidad para cortar planchas gruesas se utiliza mucho en la industria.

- Corte por Chorro de agua

Este proceso se desarrolla a pasos grandes en la actualidad ya que permite cortar cualquier tipo de material sin importar el grosor, sin daño en las zonas laterales. El alto precio en la actualidad limita su uso, aunque se pronostica que será el proceso en el futuro.

- Corte por fresa

Este proceso es muy antiguo y se basa en pasar una cuchilla que va comiendo el metal. Se necesita de mucho tiempo y varias cuchillas.

- Corte por Sierra

Proceso que se utiliza principalmente en el corte de tubos el cual utiliza una sierra con movimiento recto para cortar.

Todos estos procesos se describen ya que fácilmente pueden ser adaptados a una mesa de corte CNC. Desde el método más antiguo, hasta el más moderno pueden ser utilizados en metales. Es muy probable que muchos métodos de corte del futuro se puedan acoplar fácilmente a una CNC.

## VENTAJAS DE USAR UNA CNC SOBRE TRABAJO MANUAL

En base a los resultados observados por Hypertherm en sus experimentos [3], se puede describir las siguientes ventajas cuando se utilizan sistemas CNC.

- Calidad y precisión en corte tienen una mejora sustancial.
- Se logra un aumento de mínimo 100% en la cantidad de piezas cortadas en un mismo



**Fig 2.** Trabajador usando corte por plasma manual

tiempo.

- Los consumibles duran más y el tiempo de vida de las máquinas es mayor.
- No se necesita que los operarios tengan experiencia y en poco minutos pueden ser expertos.
- Las máquinas no se cansan y no necesitan tiempo de descanso.

En base a la fotografía (véase figura 2) a continuación se va a describir grandes mejoras que tendrá una mesa CNC.

- El operario no tiene que estar cargando la antorcha.
- El operario no necesita estar con equipo especial de trabajo, gafas, guantes, etc...
- La posición será 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.



**Fig 3.** Detalle de corte manual

## PARTES DE UNA MESA CNC (INTRODUCCIÓN)

Después de entender las utilidades de una mesa CNC se va a analizar las partes principales ya que es un elemento mecánico operado por un computador.

### **Partes mecánicas**

La parte mecánica es físicamente la mesa y sus componentes. En este capítulo se hace una descripción poco detallada a manera de introducción, para analizar detalladamente todos los componentes al realizar el diseño mecánico.

- **Soporte mesa**

La mesa necesita un lugar de apoyo, generalmente son 4 patas pero pueden ser más dependiendo de las dimensiones de la mesa y la altura. Las mesas pequeñas pueden ser de madera, plástico o metal, generalmente aluminio. Las mesas grandes son Metálicas o Hierro fundido ya que cargan enormes pesos. Es muy importante que la mesa sea estable, para esto se ponen patas regulables y así se logra nivelar correctamente la mesa. La altura cambia dependiendo de la aplicación pero en general no es muy importante.

- **Rieles movimiento eje X**

Existen 2 tipos básicos para mesas CNC, la primera se trata en mover la pieza, la segunda es mover la herramienta. En los 2 casos el eje X tiene la misma función, mantener 2 soportes paralelos que pueden avanzar o retroceder. Dependiendo del tamaño de la mesa se puede utilizar un solo motor en el centro y mover sobre los soportes que no son más que guías. Para mesas grandes se necesitan dos motores sincronizados para que el movimiento se genere sobre las dos guías. Es muy importante que sean paralelas ya que esto asegura que operaciones sean correctas.

- **Puente soporte eje Y**

Se Necesita tener un puente transversal al eje X que conecte las guías. Este puente contiene un carro el cual se desliza para general el movimiento en el eje Y. Para mesas pequeñas, el motor se lo sujeta en uno de los extremos y mediante

un tornillo se mueve el carro sin tener que mover el motor. Cuando las mesas son grandes, el motor debe estar incorporado en el carro para mantener una buena precisión. Por lo general el carro no contiene un gran peso por lo que en la mayoría de casos 1 motor será suficiente.

- **Dispositivo altura eje Z**

Sobre el carro se debe colocar un soporte para la herramienta, este soporte debe tener la capacidad de levantar o bajar la herramienta en una dirección perpendicular al plano XY. Puesto que el peso aquí es bastante bajo no se necesita más que un solo motor. El eje Z es el único que se modificaría cuando se cambia una herramienta ya que es el eje que ejerce fuerza sobre la pieza.

- **Material desechable**

Toda mesa de corte debe tener un material desechable, este es el material sobre el cual se asienta la pieza. Este material sufre cortes, daños y se lo reemplazará cuando no pueda cumplir más su función.

## **Partes Electrónicas**

Al igual que las partes mecánicas aquí se va a describir de manera introductoria lo básico de la electrónica en una mesas CNC. En la parte de diseño se discutirán diferentes motores y un análisis más completo de toda la electrónica involucrada.

- **Motores**

Los motores son la base del movimiento. Es el elemento que transforma la electrónica en mecánica y en nuestro caso es el método de comunicación entre la computadora y la herramienta. Es importante que los motores permitan controlar bien su posición, que puedan realizar cambios rápidos de dirección así como cambios rápidos de velocidad.

- **Control de motores**

Los controladores son circuitos electrónicos que se encargan de dar las instrucciones a los motores, a qué lado moverse y la velocidad con la que se debe generar el movimiento. Los controles pueden o no tener sensores que indican

topes al final de la mesa así como contadores para conocer cuánto se movió el motor. Los controladores, también denominados drivers transforman las señales digitales a señales para movimiento del motor. Por lo general se conectan señales de datos y de potencia para poder alimentar un motor con varios amperios pero controlado por señales bajas.

- **Potencia a los motores**

Los motores necesitan ser alimentados por líneas de potencia, esto quiere decir que son cables que llevan voltaje y corriente necesaria. Por lo general también incluyen fuentes que convierte señal AC en DC. Esto depende de los motores que se usen, pero por lo general se debe tomar en cuenta el grosor de los cables ya que en motores fuertes pueden alcanzar amperajes que quemarían los cables. También de debe tomar en cuenta que cables pesados y rígidos dificultarán el movimiento.

- **Cableado**

Para el cableado se encuentran tres diferentes líneas. La primera es la línea de datos, la cual se comunica con el computador y envía las instrucciones que deben ejecutar los motores. La segunda línea es de potencia que lleva la energía que usar los motores para su movimiento. La tercera línea sería de instrumentos la cual lleva información de sensores y demás datos que ayudan a verificar que todo esté correcto. Es muy importante tener en cuenta el ruido generado por el ambiente y la herramienta, ya que un pequeño error puede causar grandes problemas. Utilizando cables apantallados se va a reducir el ruido y con esto los errores.

Al ser una máquina grande con movimiento se necesita una forma de llevar los cables con seguridad, para esto se utilizan cadenas portacables, las cuales permiten el movimiento.

## Software

### Leer archivos de AutoCad (DXF)

Puesto que una mesa CNC es una impresora, se necesita la información de la forma que se va a cortar. AutoCAD tiene un formato de exportación denominado DXF (Drawing Exchange Format) el cual, se puede leer como texto, y de esta forma se entiende el dibujo por un software fuera de AutoCAD. Este formato es diseñado por Autodesk (dueños de AutoCAD) y tiene un formato especial para cada figura.

### Interfaz gráfica y ayudas para el usuario

Es muy importante la forma en la que se muestra la información en la pantalla ya que esto permitirá manejar correctamente la máquina. Según el análisis que se ha realizado, los software comerciales cuentan con los siguientes elementos:

- Ventana donde muestra los objetos.
- Tres labels que muestran la posición de la antorcha en los ejes.
- Botón para empezar, pausar y detener la operación de corte.
- Orden en que se van a cortar los elementos.
- Ventana de información de las instrucciones, generalmente código-G.
- Casilleros de velocidad de corte y velocidad en movimiento.

Dentro de las ayudas para el usuario se necesita una forma de fácil revisión de los elementos a cortar los cuales se resumen de la siguiente forma:

- Zoom para observar con detalle los elementos.
- Movimiento de la imagen con botón medio del mouse (estilo AutoCAD).
- Ver gráficamente el orden de corte.

## **Comunicación entre Software y Hardware**

El software interpreta los objetos y sabe crear instrucciones para cortar los objetos, el problema está en enviar estas instrucciones y que sean interpretadas correctamente por el sistema móvil. A esto se lo denomina la comunicación entre software y hardware. El software es el programa que corre en la computadora, mientras que el hardware son todas las partes físicas de la máquina. Es por esta razón que se necesita comunicar el computador con el controlador de los motores, los que generan el movimiento del cabezal de la máquina, transformando las instrucciones del programa en cortes en plancha.

La información, que son las instrucciones o pasos, se traslada mediante un protocolo de comunicación el cual asegura una correcta transmisión e interpretación. La base del protocolo debe asegurar la correcta transmisión de mensajes, sin errores, así como el orden de las instrucciones.

Una dificultad especial en esta mesa CNC viene con la importancia de la velocidad y el trayecto. No solo se necesita llegar de un punto a otro, es importante la velocidad con la que viaja y el camino entre los puntos.

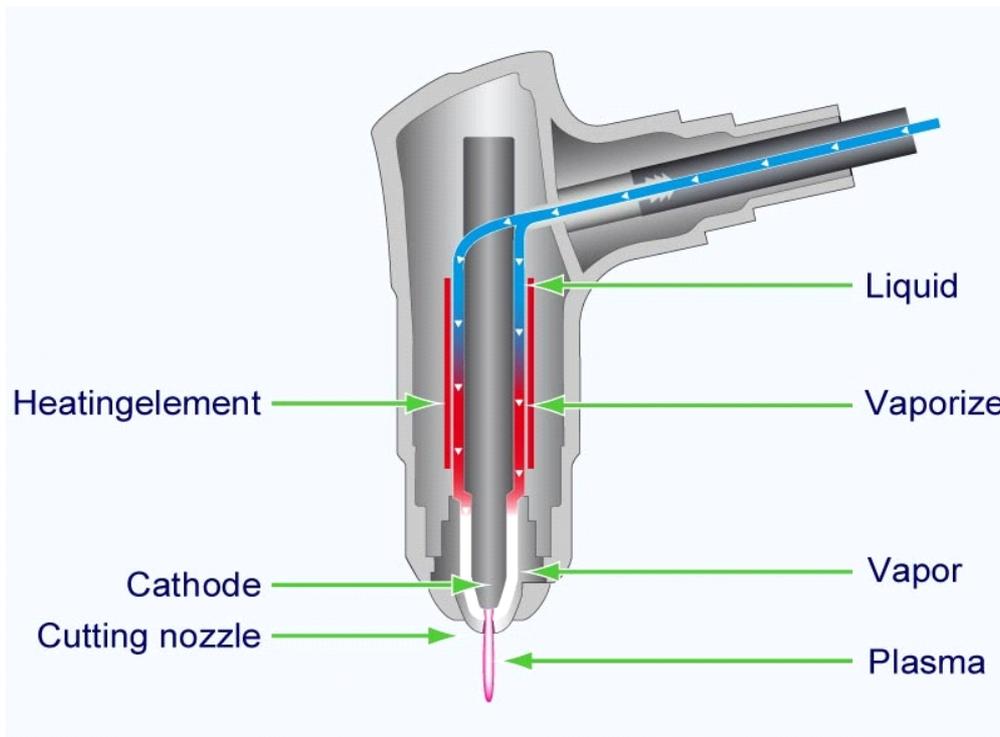
## **CORTE POR PLASMA**

Existen tres estados de la materia, sólido, líquido y gaseoso. En un material se puede generar los tres estados con cambios de temperatura, en agua por ejemplo, si se la enfría bajo los 0°C resulta hielo y al calentarla sobre los 100°C se encuentra vapor. Estos cambios se dan ya que el calor da energía a los electrones dentro del material separando a los átomos y así generando diferentes estructuras. El cuarto estado de la materia, denominado plasma, aparece cuando la energía dada, a los electrones mediante calor, es tan grande que logran separarse de su núcleo, permitiendo la formación de iones. El plasma es un conductor eléctrico, ya que tiene electrones libres, en forma de gas.

El procedimiento de corte por plasma consiste en calentar un gas hasta hacerlo llegar al estado de plasma. Este gas aplicado sobre cualquier superficie la calienta a muy altas temperaturas en instantes y puesto que el calor estará centralizado

podrá fundir y expulsar cualquier material antes de permitir que se expanda el calor sobre toda la superficie del mismo, y así es como corta.

Para calentar el gas, llevarlo al cuarto estado de la materia, se genera un arco eléctrico sobre una boquilla muy pequeña, concentrando la energía cinética propia del gas con el arco eléctrico, dando el calor necesario para obtener un chorro de plasma.



**Fig 4.** Corte por plasma

[http://www.fronius.com.mx/internet/img\\_mx/ST/Grundlagen\\_ST\\_TransCut\\_abb01\\_en\\_rdax\\_100\\_rdax\\_100.jpg](http://www.fronius.com.mx/internet/img_mx/ST/Grundlagen_ST_TransCut_abb01_en_rdax_100_rdax_100.jpg)

## CAPITULO II DISEÑO MECÁNICO

Este capítulo muestra toda la parte física de la estructura de la mesa CNC y su movimiento incorporado. Diseño de la parte mecánica así como información de otras posibles soluciones, se da las razones para las soluciones tomadas y su construcción.

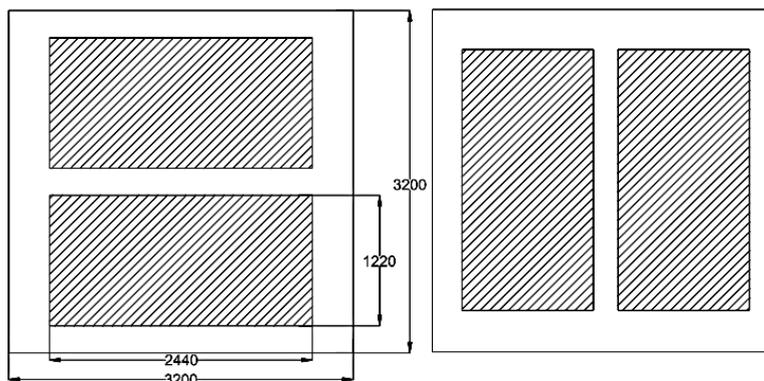
### REQUERIMIENTOS DEL CLIENTE

Puesto que la mesa tiene un objetivo final y con esto un cliente, se va a enumerar aquí las condiciones impuestas. Esta lista sale en base a conversaciones y visita de otras mesas CNC.

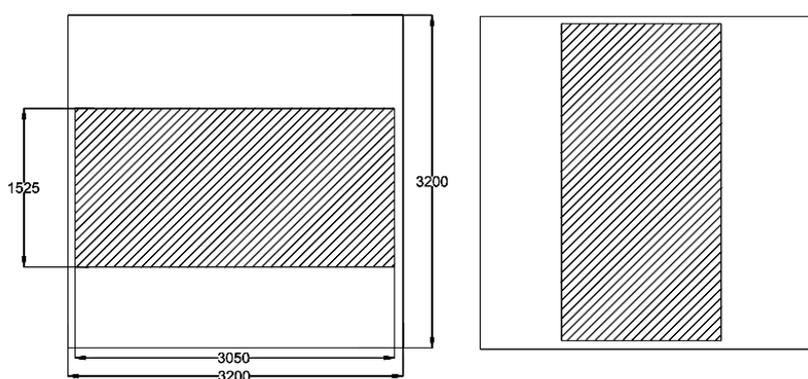
- La mesa debe poder cortar dos planchas tamaño estándar (8x4 pies o 2440x1220 mm) al mismo tiempo, para poder tener un corte continuo. Mientras corta la segunda plancha, la primera puede ser removida con seguridad y colocada una tercera para lograr tener un corte continuo de las piezas.
- La mesa debe poder cortar planchas más grandes (10x5 pies o 3050x1525 mm).
- Se necesita que la mesa tenga una altura buena para el trabajo, por seguridad de los operarios, para esto se necesita que la base donde se coloca la plancha de acero tenga una distancia de 730 mm desde el piso.
- La Mesa debe tener un color amarillo ya que así se representa maquinaria con movimiento.
- Se necesita un sistema en las patas de la mesa que permitan regular la altura para nivelar la mesa ya que el piso donde va a ser colocada no es perfectamente plano.

## Análisis

Para poder cumplir con los requerimientos del cliente se va a hacer un análisis de las medias para determinar el tamaño de la máquina y sus partes.



**Fig 5.** Dos planchas tamaño estándar en área de trabajo.

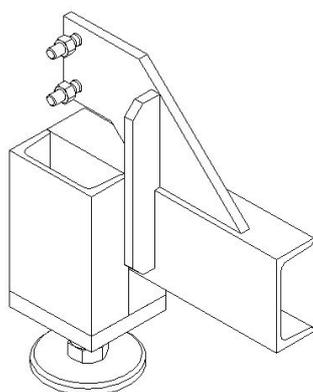


**Fig 6.** Una plancha grande en área de trabajo.

Utilizando dos planchas juntas se necesita un área mínima de 2440x2440 mm pero las planchas grandes requieren un mayor espacio, por lo tanto la distancia mínima del área de corte debería ser 3050

x 2440 mm.

Puesto que con esta área quedan las planchas justas se va a ampliar unos rangos laterales (75 mm a cada lado) para maniobrar las planchas, así como transformar en un cuadrado y con eso evitar cualquier problema en la distribución de las planchas.



**Fig 7.** Pata con pie roscado

Las medidas para el área de corte quedaran en 3200 X 3200 mm.

Para la altura, se va a disponer de 6 patas con altura regulable mediante pernos, estas patas deberán ser calculadas en base a la altura de las vigas de la mesa, para que la altura sea la deseada.

Las patas se forman con un pie roscado, el cual va a dar una buena precisión en regulación de altura permitiendo corregir el nivel para tener una mesa plana.

## DISEÑO DE LA MÁQUINA

En esta sección se va a realizar un análisis de los elementos usados para la construcción mecánica de la mesa.

### Parte Estática

#### Materiales a utilizar



**Fig 8.** Viga tipo UPN

[http://images03.olx.pt/ui/11/69/02/1342114765\\_412226702\\_6-Vigas-Perfis-Chapas-em-Ferro-e-Aco-inox-Corte-a-medida-Portugal.jpg](http://images03.olx.pt/ui/11/69/02/1342114765_412226702_6-Vigas-Perfis-Chapas-em-Ferro-e-Aco-inox-Corte-a-medida-Portugal.jpg)

#### •Vigas UPN

Este tipo de vigas son acero extruidos en forma de U. Son muy resistente a las torsiones en cualquier sentido y son abundantes en el mercado. Todas las UPN tienen un código numérico el cual se relaciona con su capacidad de resistencias y sus medidas, las cuales se pueden ver en tablas.

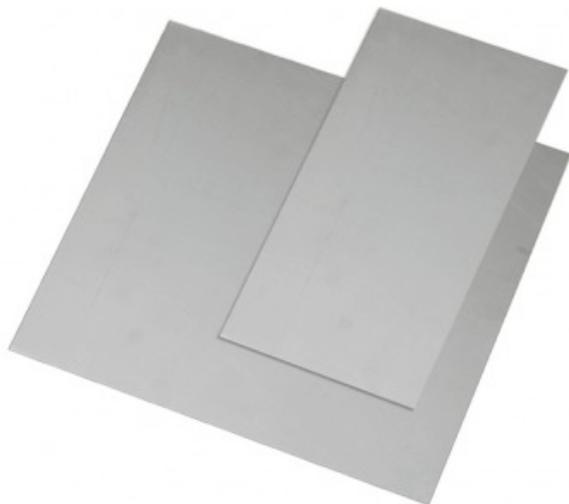
#### • Angulo Laminado

Un ángulo es una pieza de metal que va a garantizar un ángulo de 90 grados. Son resistentes y tienen muy poco error comparado con placas de metal dobladas. A diferencia de los UPN son laminados y están definidos por tamaños estándar mediante códigos.



**Fig 9.** Ángulo Laminado

<http://www.ua.all.biz/img/ua/catalog/521153.jpeg>



**Fig 10.** Planchas y platinas de Acero

[http://www.rdz.it/file/prodotti/242/scheda\\_1201040\\_DB.jpg](http://www.rdz.it/file/prodotti/242/scheda_1201040_DB.jpg)

- Plancha y platinas

Las planchas o platinas son laminas de acero. las planchas viene en dimensiones estándar de 2440 x 1220 mm mientras que las platinas cambian su tamaño dependiendo de la necesidad. Las planchas y platinas se las identifica por el grosor, materia y acabado dándonos diferentes planchas como aceros en el mercado.

- Tornillo perno tuerca de seguridad



**Fig 11.** Tuerca de seguridad

[http://www.multimateriales.com/images/p045\\_1\\_01.jpg](http://www.multimateriales.com/images/p045_1_01.jpg)

Los tornillos son barras metálicas con cabeza que permiten unir 2 superficies mediante un hueco en ambas. En nuestro caso es muy importante la tuerca de seguridad ya que esta no se afloja con las vibraciones ni el movimiento. La tuerca de seguridad es igual que una tuerca norma pero tiene un aro de algún material que se deforma como un caucho o nylon.



**Fig 12.** Tornillo tuerca y rodela

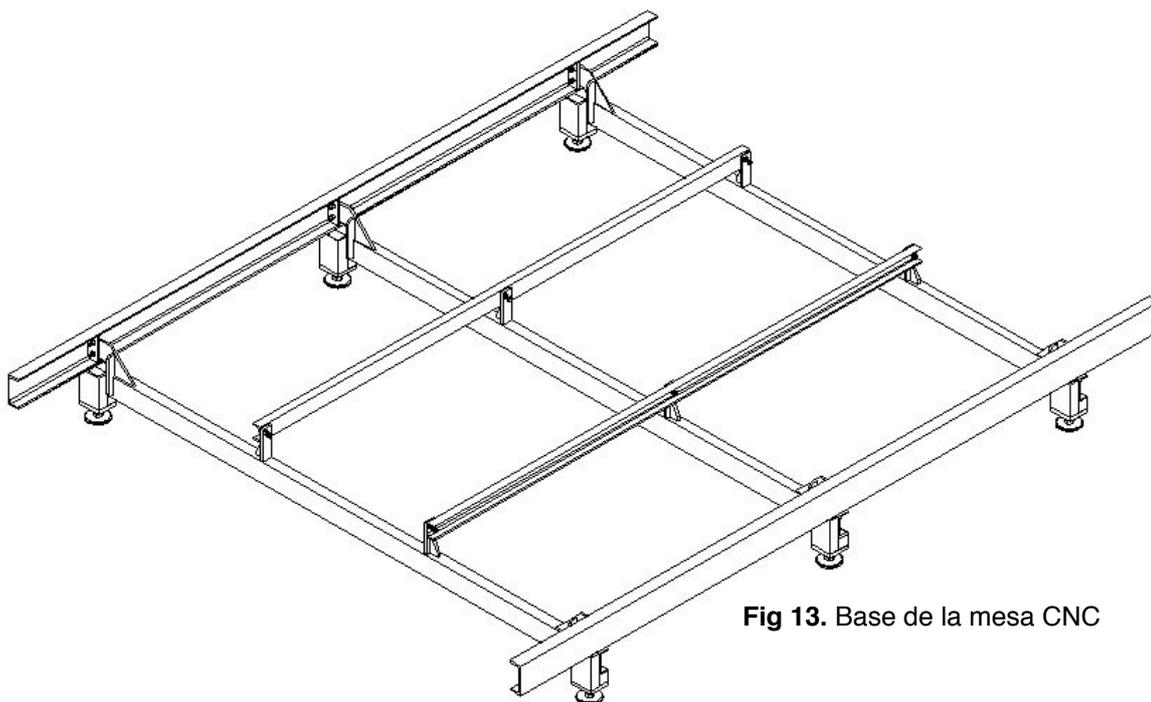
[http://brainstomping.files.wordpress.com/2011/03/tornillo\\_cabeza\\_hexagonal\\_tuerca\\_y\\_arandela-gran.jpg](http://brainstomping.files.wordpress.com/2011/03/tornillo_cabeza_hexagonal_tuerca_y_arandela-gran.jpg)

## Partes de la mesa CNC

### Base

Como su nombre lo indica este elemento se encarga de soportar las demás partes de la mesa CNC. Las medidas se basan en los requerimientos del cliente. El área de corte debe ser de 3200 x 3200 mm por lo tanto la mesa debe sobrepasar estas medidas.

Las medidas de la base son 4000x3202 mm, estas medidas se determinan para que pueda entrar el cajón que será el área de corte. Los 4000 mm están diseñados para un puente que cruza estas vigas paralelas. Esta distancia extra se selecciona para que la antorcha de plasma pueda llegar a toda el área de corte.



**Fig 13.** Base de la mesa CNC

Como se puede observar en la imagen, el armado se lo realiza en base a vigas UPN. Para las vigas laterales se utilizaron UPN160, las que se ve en el centro paralelas son UPN80 y las perpendiculares a estas con UPN120. Se encuentran unidas mediante estructuras soldadas y pernos  $\frac{1}{2}$ "UNC x 2". Para más detalles se pueden referir a la sesión de planos en Anexos.

## Patas

Existen 6 patas las cuales levantan la altura de la mesa. Las patas están armadas con una viga UPN120 Soldadas en las vigas laterales como se muestra en la

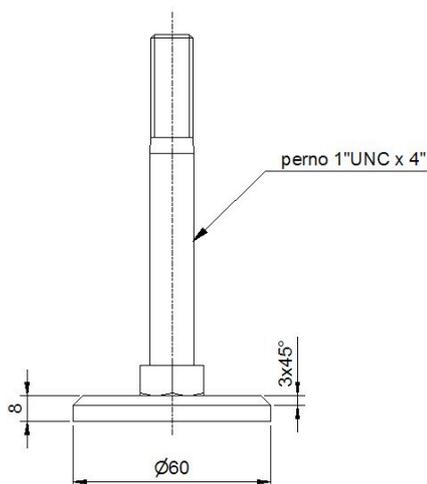


Fig 15. Pie Roscado

figura. Una placa de 120x120x20 está debajo de las 6 bases, en esta placa se encuentra un hueco en el

medio soldado a una tuerca, la cual sirve de conexión con los pies roscados. Ese último es la base que asienta con el piso, se forma mediante una base y un perno el cual entra en las patas de la mesa. Al girar el pie roscado se

puede determinar de una forma muy precisa la altura y con esto regular el nivel, para que la mesa este plana sin importar la superficie donde fue asentada

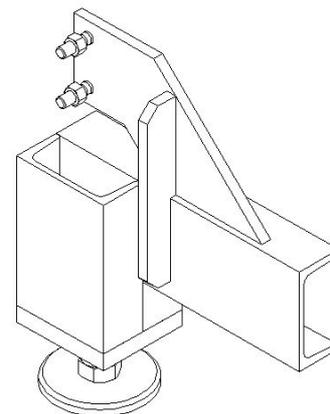


Fig 14. Pata con pie roscado

## Tanque

Sobre esta estructura se encuentra un tanque el cual va a estar lleno de agua. La función del tanque será retener todos los desperdicios

después del corte así como el humo generado por el plasma. Es importante que el tanque tenga una forma de llenado y un cuello de limpieza para poder drenar el agua y limpiar el tanque.

El tanque es exactamente igual al área de corte ya que

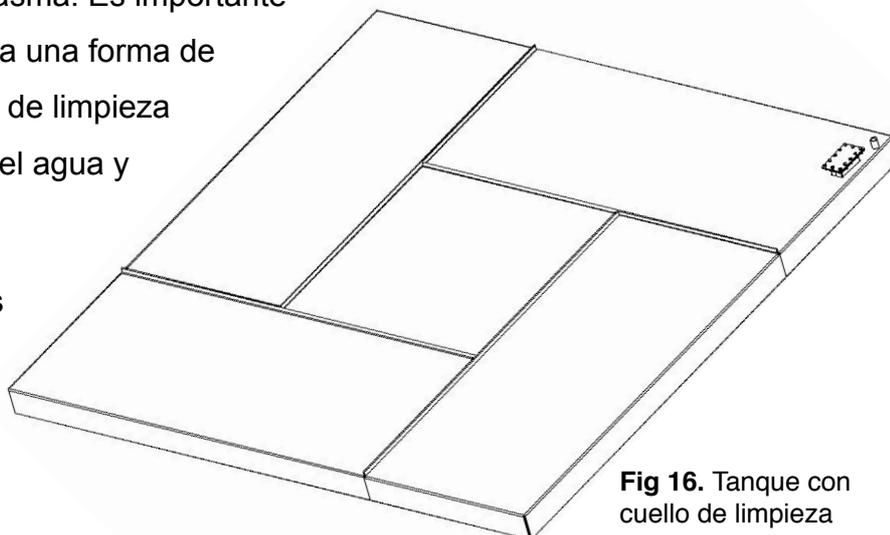
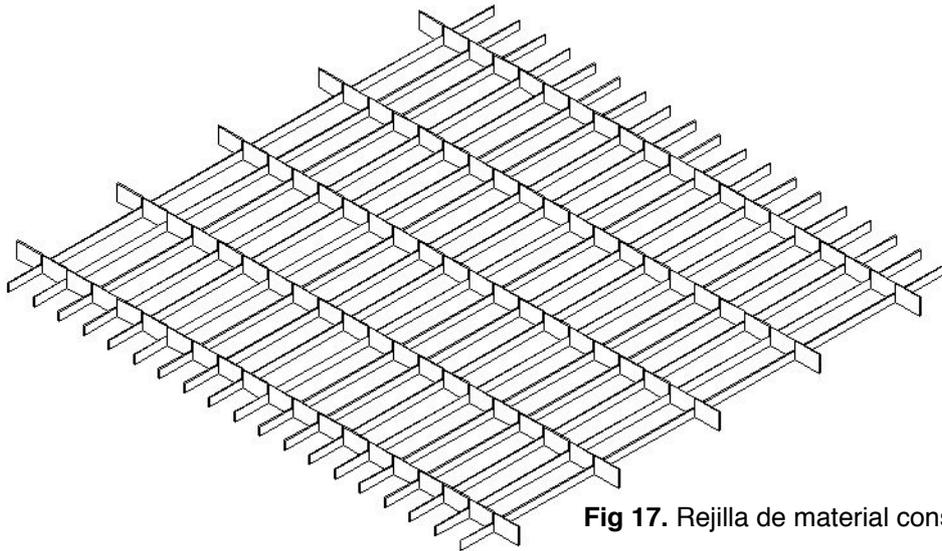


Fig 16. Tanque con cuello de limpieza

sobre él se va a realizar todas las operaciones. Se determina un tamaño de 3202 x 3202 mm para dar un posible error. La altura de este tanque es de 148 mm.

### **Rejilla**

Esta rejilla es el material que sostiene a las planchas, no importa si llega a dañarse ya que este es el objetivo, con el tiempo esta rejilla deberá ser remplazada. La altura es 150mm logrando que sobresalga del tanque y dando la superficie necesaria para cortar.



**Fig 17.** Rejilla de material consumible

### **Parte Móvil**

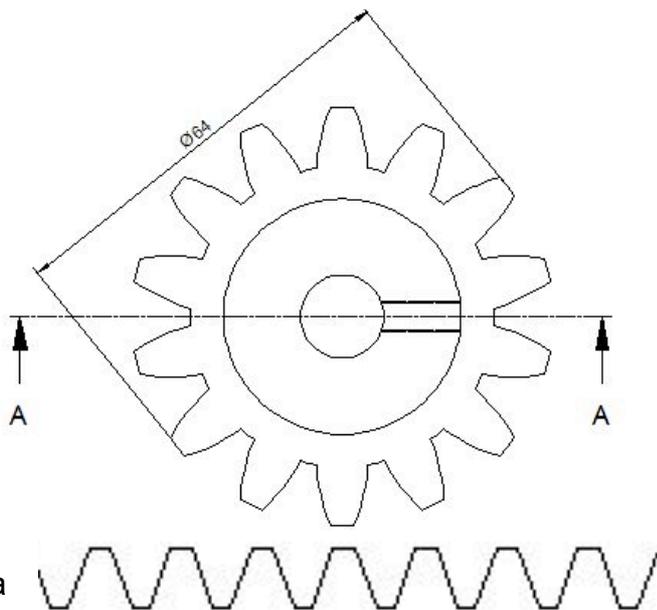
Esta es la parte principal que transforma una mesa cualquiera en una CNC. Para moverse alrededor de la mesa se va a utilizar carros laterales, los cuales sostienen a un puente que cruza la mesa. Este puente tendrá un carro propio para poder desplazarse. El carro también tendrá incorporado un dispositivo para subir o bajar un soporte y con esto regular la altura. En esta sección se va a analizar los métodos utilizados para que los movimientos de los motores lleven a una posición deseada.

## Materiales a utilizar

- Engranaje y cremallera

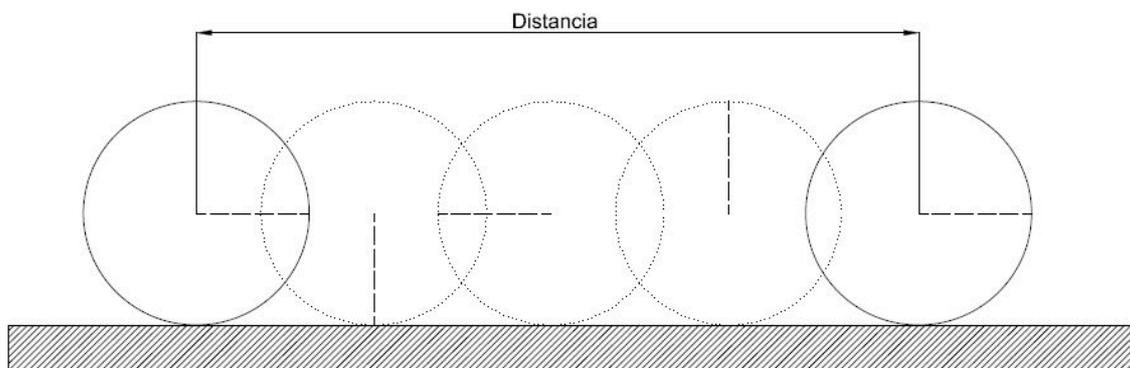
El sistema descrito a continuación permite cambiar un movimiento circular a uno lineal, ya que al girar el piñón sobre la cremallera se va a tener un desplazamiento.

Un punto muy importante es la distancia que recorreremos si se gira una vuelta el engranaje.



**Fig 18.** Engranaje 14 dientes y cremallera

Analizando un círculo que gira sobre una superficie plana, la distancia recorrida será igual al perímetro del círculo, como se indica en la figura.



**Fig 19.** Distancia al dar una vuelta  
Perímetro

$$P = 2\pi R$$

f.1

P = perímetro de un círculo.

R = radio del círculo

Una idea común sería calcular el perímetro del diámetro exterior, pero esto no funciona de esta manera en los engranajes, ya que el punto de giro cambia en los dientes internos, es por esto que se recurre al libro “manual del ingeniero

mecánico”, el cual explica como calcular un radio que equivaldría a un círculo perfecto en una superficie plana. Se determina exactamente la distancia que se recorre cuando se gira una vuelta del engranaje con las siguientes formulas:

$$D = mN \quad \text{f.2}$$

$$D_0 = m(N + 2) \quad \text{f.3}$$

D = Diámetro de paso

m = Modulo

N = Número de dientes

Do = Diámetro exterior

Datos base:

En el mercado ecuatoriano lo más difícil de conseguir son las cremalleras y estas tiene una especificación módulo 4, por lo tanto el engranaje debe mantener el mismo modulo. Lo que se necesita ahora es calcular el número de dientes de los engranajes. Cuando se reduce el tamaño de los engranajes se reduce el torque también, pero se tiene más precisión ya que va a tener una menor distancia por vuelta. Esto es algo complicado de decidir, por esta razón se recurre a CNC que funcionan, las cuales tiene un diámetro exterior (Do) de 64 mm.

$$64 = m(N+2) \quad \text{Aplicando f.3 con los datos base}$$

$$N = 14 \quad \text{Dientes}$$

Con estos datos se puede calcular el diámetro de paso

$$D = 4 \times 14 \quad \text{Aplicando f.2}$$

$$D = 56$$

Con el diámetro de paso se puede calcular la distancia con la fórmula básica de perímetro de un círculo.

Sabiendo que  $D = 2 \times R$

$$P = 2 \times \pi \times \frac{D}{2} \quad \text{Aplicando f.1}$$

$$P = 175.9 \text{ mm}$$

Después de todos estos cálculos se puede definir que cada vuelta que el motor llegue a completar significara un desplazamiento de 175.9mm.

- Tubo cuadrado

EL tubo cuadrado es un elemento muy importante en la industria ya que se los utiliza para soportar pesos así como para mantener la guía en superficies, al ser cuadrados permiten desplazar elementos móviles sobre ellos



**Fig 20.** Tubo Cuadrado  
<http://interlanco.com/shop/images/categories/squaretube.gif>

- Rodamiento



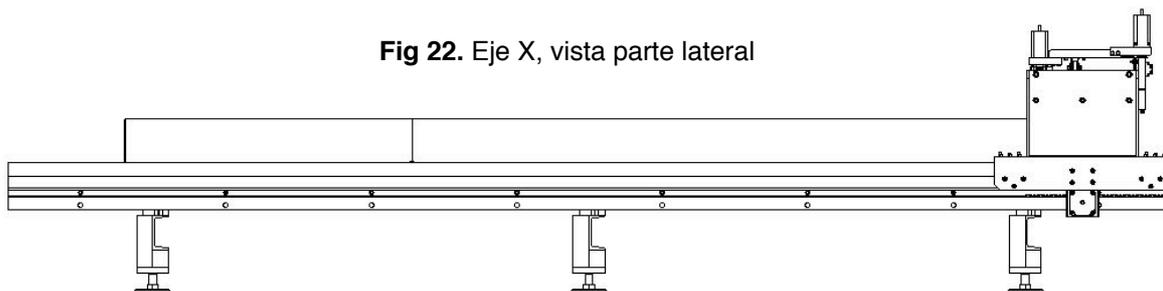
**Fig 21.** Rodamiento  
[http://3.bp.blogspot.com/-19\\_v8rQ3Ub8/T7Ti7xfuJQI/AAAAAAAAAwU/O9M5UJimY7U/s1600/caja+de+bolas.jpg](http://3.bp.blogspot.com/-19_v8rQ3Ub8/T7Ti7xfuJQI/AAAAAAAAAwU/O9M5UJimY7U/s1600/caja+de+bolas.jpg)

Un rodamiento es un elemento mecánico que permite unir dos piezas que giran de forma diferente, este elemento evita el desgaste de mantener piezas girando. Se los utiliza mucho para mantener en una posición a ejes que giran, pero también pueden servir como ruedas para soportar pesos grandes y deslizarlos sobre superficies planas.

## Eje X

Este eje consta de dos carros que llevan un puente. Se necesita soportan un peso relativamente grande y moverlo mediante el sistema engranaje cremallera. Existen aquí dos posibilidades ya que se puede mantener el engranaje estático y mover la cremallera, o mover el engranaje y dejar la cremallera estática.

Fig 22. Eje X, vista parte lateral



Se decide ir por el engranaje en movimiento ya que es mucho más simple el sistema y se puede acoplar para no salir del área de trabajo.

No se va a cargar el peso completo del puente sobre el engranaje y la cremallera, lo que se va a hacer es soportar mediante rodamientos sobre el tubo cuadrado.

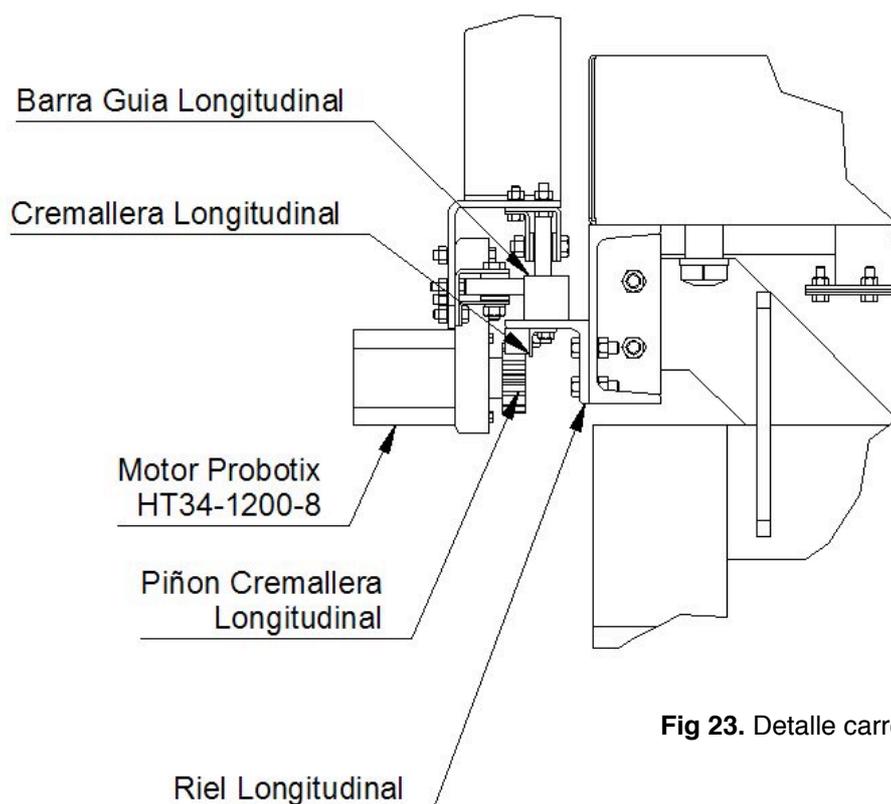


Fig 23. Detalle carros eje X

Como se puede ver en la imagen, la barra guía longitudinal es el tubo cuadrado donde se asienta todo el peso. Cada carro consta de un sistema de cuatro rodamientos los cuales permiten un movimiento suave que mantiene al carro sobre la barra guía. Se coloca el motor por el exterior y debajo de la barra de guía para que la cremallera se sostenga invertida, esto se realiza para que cualquier basura no se quede en la cremallera sino que por gravedad caiga, protegiendo

todo el sistema de movimiento. El sistema se asienta sobre un ángulo. Detalles se los podrá observar en la sección de Planos en Anexos.

## Eje Y

El eje Y consta de un puente que sirve de riel para un carro el cual consta del mismo sistema engranaje-cremallera.

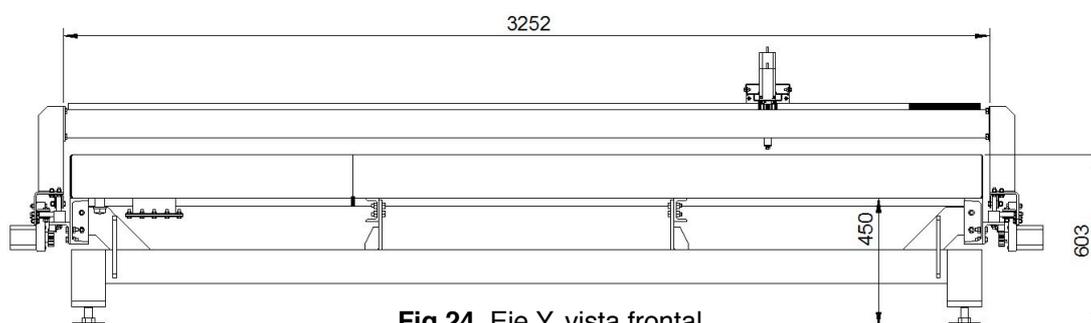


Fig 24. Eje Y, vista frontal

## Eje Z

El Eje Z controla la altura del soporte. Este sistema de movimiento es diferente ya que se utiliza un tornillo y tuerca. Puesto que no se permite girar la tuerca, el tornillo al girar le obliga a subir o bajar. Dependiendo de la rotación y la velocidad del motor se va a tener el movimiento en el eje Z. Puesto que esto no es parte del proyecto actual solo se explica mediante los planos el funcionamiento, pero no va a ser utilizado dentro de este documento.

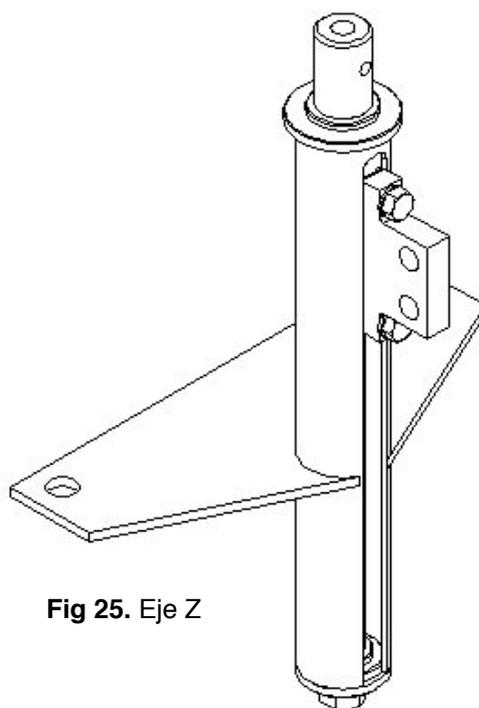


Fig 25. Eje Z

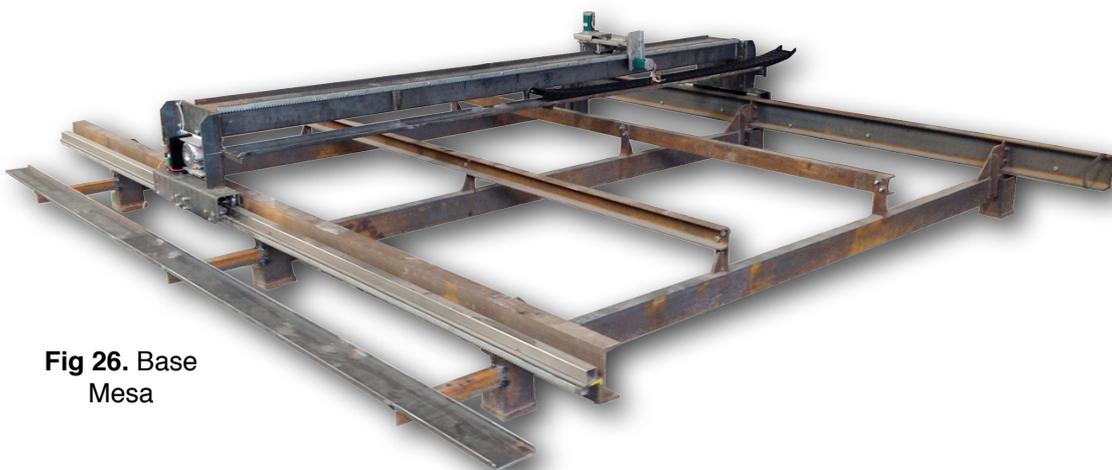
## Soportes para canaletas

Dentro de la estructura metálica se debe colocar 2 canaletas, las cuales sostendrán

a los cables que energizan la máquina. Estas canaletas deben tener la longitud total de la mesa en X y debe recorrer con el puente del eje Y.

## CONSTRUCCIÓN PARTE MECÁNICA

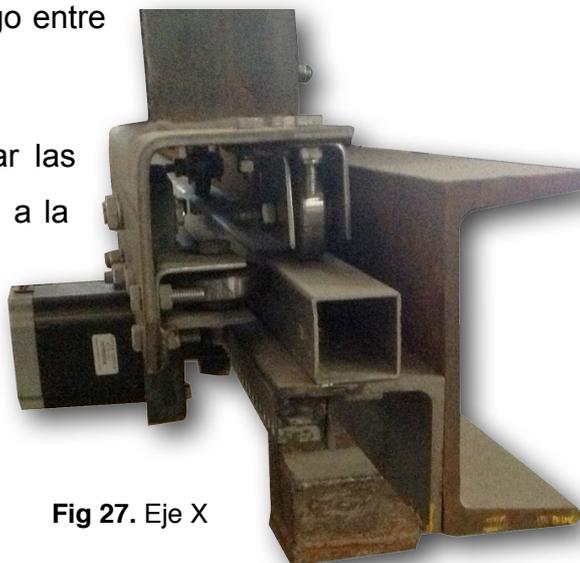
A continuación se muestra imágenes del armado mecánico de la mesa.



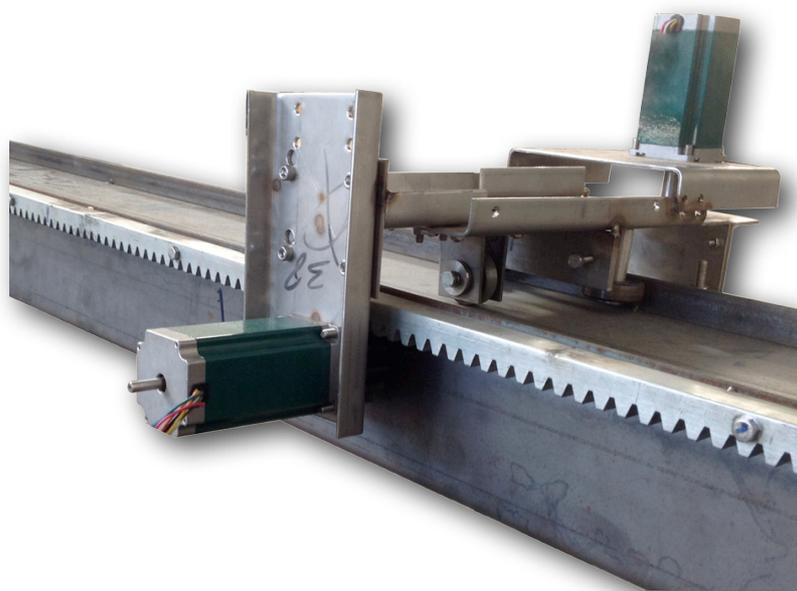
**Fig 26.** Base  
Mesa

## RECOMENDACIONES CONSTRUCCIÓN PARTE MECÁNICA

- Revisar cuidadosamente que las vigas estén paralelas entre si.
- Permitir calibrar la altura de los motores para disminuir al mínimo el juego entre piñón y cremallera
- Tener mucho cuidado al montar las cremalleras y revisar que estén a la misma altura, sin dobleces.
- Armar sobre una superficie plana para evitar problemas de nivel.



**Fig 27.** Eje X



**Fig 28.** Eje Y



**Fig 29.** Mesa con tanque y consumible

## CAPITULO III DISEÑO ELECTRÓNICO

Este capítulo trata sobre toda la parte electrónica dentro del proyecto, el control de los motores, cableado, sensores, etc... Esta sección asume tener una instrucción correcta y la ejecuta. EL capítulo trata también de problemas de ruido así como los aislamientos utilizados.

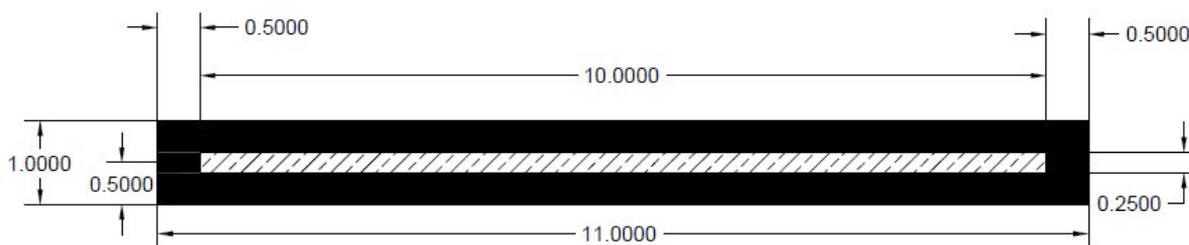
### REQUERIMIENTOS DEL CLIENTE

Dentro de la parte electrónica también se tiene requerimientos específicos del cliente, los cuales son bases para todo el diseño de la mesa. La lista a continuación se basa en requerimientos específicos del cliente y visitas e investigación de otras mesas CNC.

- La precisión en el corte no debe tener más de 1 mm de error.
- El acero a utilizar es Inoxidable.
- La velocidad de la máquina debe poder variar para cortar planchas desde 0.8 mm a 25.4 mm.
- La velocidad durante el corte debe ser constante en todo momento.
- La alimentación de la máquina debe ser 110 V o 220 V
- Los cables deben estar correctamente sujetos en canaletas y no sueltos.
- La mesa debe incorporar un paro de emergencia.
- Toda la parte electrónica debe tener protecciones contra sobrecarga o cortocircuito.

## Análisis requerimientos del cliente

Retomando la información sobre el engranaje y cremallera, se llega a concluir que el motor al girar una vuelta completa se traslada 175.9 mm. El error máximo que se debe tener es de 1 milímetro. En la figura se puede observar un corte de 10mm, y en este se muestra el rango de error permitido. Si el corte está dentro del área negra entonces es aceptable según el cliente.



**Fig 30.** Área segura de corte según el cliente

Analizado de la imagen se encuentra que el error real que se puede tener es máximo de 0,5mm ya que con un error mayor se sale del área segura. Entonces lo que se va a hacer es determinar la precisión que se debe tener en el motor para que pueda cumplir con esta tasa de error.

Número de pasos por vuelta = distancia recorrida en una vuelta / error máximo

$$\frac{175,9}{0,5} = 351,86 = 352 \text{ pasos}$$

Como se ve en la fórmula, se debe poder detener el motor 352 veces en una sola vuelta y con esto se consigue cumplir con los requerimientos mínimos del cliente. En la siguiente tabla se muestra el error deseado y cuanto pasos se deben tener en el motor, y para entender fácilmente, los grados que el motor debe poder girar como mínimo.

Tabla 1 Precisión en base a pasos del motor

Distancia 1 paso (mm)	Cantidad de pasos en 1 vuelta	Distancia en grados (°) 1 paso
0.500	352	1.023
0.400	440	0.819
0.300	586	0.614
0.200	880	0.409
0.100	1759	0.205
0.090	1955	0.184
0.080	2199	0.164
0.070	2513	0.143
0.060	2932	0.123
0.050	3519	0.102
0.040	4398	0.082
0.030	5864	0.061
0.020	8796	0.041
0.010	17593	0.020
0.009	19548	0.018
0.008	21991	0.016
0.007	25133	0.014
0.006	29322	0.012
0.005	35186	0.010
0.004	43982	0.008
0.003	58643	0.006
0.002	87965	0.004
0.001	175929	0.002

Pasando al análisis de velocidades, se va a tomar información del fabricante Hypertherm del equipo plasma a utilizar, Powermax 1250. En Anexos se puede encontrar el manual de operación del equipo plasma y en base a la velocidad en estas tablas determinaremos las necesidades del cliente.

Tabla 2 Velocidades optimas de corte

Tipo de Acero	Espesor (mm)	Velocidad máxima de corte (mm/min)	Velocidad de corte optima (mm/min)
Acero al carbono	0.8	12700	8255
	6.4	1880	1219
	12.7	1524	991
	25.4	406	254
Acero Inoxidable	0.8	12598	8179
	6.4	1194	787
	12.7	1270	838
	25.4	356	229
Aluminio	0.8	15494	10084
	6.4	1930	1245
	12.7	1905	940
	25.4	No Dato	No Dato

Para poder cumplir los requerimientos mínimos se va a hacer un análisis solo del acero inoxidable que es el requerido.

En acero inoxidable la velocidad más alta es 12598 mm/min para cortar rápidamente un plancha de 0.8 mm. La velocidad más baja es 229 mm/min para cortar 25.4 mm.

Esta información sirve para conocer y definir el rango de velocidad que los motores deben mantener. Conociendo que una vuelta del engranaje es 175.9 mm y la velocidad, se va a calcular el tiempo, con la formula básica.

Espacio = velocidad x tiempo

$$t = \frac{e}{v}$$

Puesto el que espacio es exactamente una vuelta, se tiene el tiempo que debe durar una vuelta y esto significa que al invertirlo se encuentra el número de vueltas en 1 minuto o llamadas revoluciones por minuto (RPM).

$$RPM = \frac{1}{t}$$

t = 0.01396 min ( RPM 71.62 ) para la máxima velocidad.

t = 0.76812 min ( RPM 1.301 ) para la mínima velocidad.

Para realizar un análisis completo se va a encontrar estos valores para el caso extremo que es aluminio con espesor 0.8mm que tiene velocidad de 15494 mm / min.

t = 0.011352 min ( RPM 88.0841)

Determinado esto, se tiene ya transformada la información del cliente en nuestras necesidades básicas para los motores. Todo el resto de requerimientos los iremos analizando con el diseño ya que se necesita datos reales.

## DISEÑO ELECTRÓNICO DE LA MÁQUINA

En esta sección se va a encontrar información sobre las decisiones tomadas así como el armado final.

### **Motores**

Antes de continuar se necesita conocer el termino NEMA el cual aparece constantemente al buscar información sobre motores.

*NEMA es una asociación con sede en EE.UU. que intenta establecer normas para la fabricación. Para los motores, han elegido una etiqueta que indica el tamaño en primer lugar. Así que un "NEMA 17" tiene una medida de 1,7 por 1,7 pulgadas de panel frontal. Cualquier motor NEMA 17 se monta en los orificios de montaje de otro NEMA 17. Esto le da flexibilidad a los fabricantes más con sus diseños, ya que no tiene que rediseñar el método de montaje*

*del motor sólo porque cambiaron proveedor del motor. ([http://reprap.org/wiki/NEMA\\_Motor](http://reprap.org/wiki/NEMA_Motor))*

El sistema NEMA ayuda a encontrar repuestos en otros fabricantes y en nuestro caso a comprar motores equivalentes ya que la mejor forma de comprar 2 motores es si calzan perfectamente en el mismo lugar sin tener que hacer cambios.

## Servo vs Stepper (A pasos)

Para entender la diferencias que existen se va primero a analizar las similitudes y la razón por la que estos dos tipos de motores son los más utilizados en máquinas CNC.

Estos dos tipos de motores son elementos electrónicos que tiene la capacidad de ser controlados en su velocidad como en la posición, esto quiere decir que se puede hacerlos girar a altas o bajar revoluciones sin necesidad de elementos como vareadores de velocidad. Controlar la posición toma en cuenta dos puntos importantes, el primero es mantenerla, esto quiere decir que se puede dejar un motor quieto en la posición deseada. El segundo punto es su capacidad de colocarse en cualquier posición (dentro de su rango). Para que se entienda bien la importancia de estos puntos se puede analizar el siguiente ejemplo:

En la industria se utilizan mucho bandas transportadoras, estas siempre llevan un motor y un vareador de frecuencia (velocidad). Lo que permiten estos elementos es girar mientras estén prendidos, y depende de si les da energía o no para que giren. Teóricamente se podría hacer que un motor de una vuelta completa si se lo paga en el tiempo justo, pero aquí entra la gran diferencia entre los motores normales y estos (servo o stepper), este motor por inercia se movería un poco más, incluso por error un operario podría hacerlo girar de más porque no hay nada que evita que gire, mientras que los motores que se utiliza en CNC se quedarían estáticos en su posición, estos si se bloquean en su posición.

Los dos tipos de motores necesitan drivers, estos son circuitos electrónicos que se encargan de alimentar correctamente a los motores para que puedan girar en

el sentido adecuado y a la velocidad deseada. Dependiendo de la calidad del driver se puede tener mayor precisión, durabilidad y ahorro de energía.

Analizado esto se va a entrar a entender las diferencias que existe, no se va a explicar el funcionamiento ya que la importancia aquí es su aplicación más que sus componentes internos.

Los motores a pasos (Stepper) por lo general tiene pasos de  $1.8^\circ$ , esto quiere decir que pueden dividir una vuelta completa en 200 pasos. No se puede hablar de pasos en los motores servo, ya que su presión depende de sus componentes electrónicos, pero según el libro CNC machine handbook [4] se puede afirmar que la precisión que se consigue con ambos motores es muy cercana. El precio de los motores a pasos es más bajo comparado con motores servo de las mismas características. Los motores servo tiene respuestas más rápidas a un cambio de velocidad y un mayor torque con velocidades altas, mientras que los motores a pasos tienden a vibrar más y perder fuerza con la velocidad. Un beneficio muy importante de los motores a pasos es su capacidad de funcionar en "open loop" (circuito abierto) a diferencia de los motores a servo que necesitan siempre estar conectados a un encoder el cual funciona como un "closed loop" (circuito cerrado). Esto quiere decir que los motores servo necesitan una retroalimentación mediante un sensor para mantener su posición correcta, mientras que los stepper pueden o no utilizarla, pero no es necesario para su correcto funcionamiento. Si un motor a pasos llega a trabarse este simplemente se salta los pasos sin afectar al motor, en un caso similar, el motor servo se quedaría intentando avanzar hasta que algún componente se rompa, del motor o externo. Un beneficio significativo de los motores servo está en la seguridad de cumplir la instrucción, esto quiere decir que estás seguro que llego a donde debe estar, mientras que un stepper pudo no avanzar y no se entera del error. El último punto a comprar es en su calibración, ya que los motores servo siempre deberán ser calibrados antes de entrar a operar, mientras que un stepper nunca va a cambiar su comportamiento, si es de 200 pasos siempre va a ser así.

Aunque los servo son mucho más comunes en el mercado por su gran variedad, se decidió traer los motores del extranjero para no tener que acoplarse a un motor

sino elegir uno deseado y esa razón permitió elegir el uso de Stepper por los siguientes motivos:

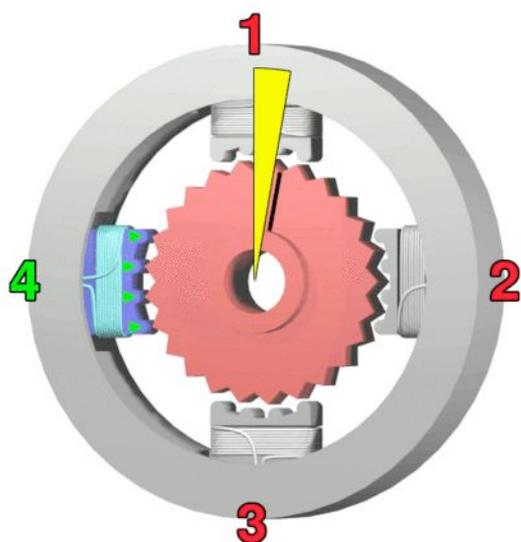
- EL precio es considerablemente mas bajo.
- Puesto que la precisión no es un factor de suma importancia en este equipo, no se necesita error  $< 0.00001$ , se prefiere evitar el uso de un elemento externo como el sensor de los servo (encoder), ya que con más componentes sensibles, que uno llegara a dañarse es más fácil.
- La aplicación nuestra es muy lenta, comprada con otras opciones de CNC, es por esta razón que tener un mayor torque a revoluciones bajas es mucho mejor.
- Ya que la mesa va a trabajar dentro de una metalmecánica es probable que existan elementos que traben a la mesa. Con motores a pasos se va a poder continuar el trabajo sin afectación a ningún componente.
- El funcionamiento de los motores a pasos es muy simple, y puesto que se desea dar una solución completa con software y hardware, se facilita para el proceso total.
- Los motores a pasos son algo novedoso para quien escribe esta tesis y el factor de aprender a usar elementos nuevos también influyó en la decisión.

Toda la información de esta sección aparece en muchas paginas web además de el libro antes mencionado, estas son las paginas con mayor información. [5-7]

## Motor a pasos y Driver de control

Los motores a pasos no tienen tarjeta de control incorporada, esto quiere decir que los motores solo tienen cables para alimentarlos con energía, pero se tiene que conocer en qué orden se los debe alimentar para que dé correctamente un paso. Por lo general se encuentran motores de 8 cables, pero también existen de 6 y de 4.

Para entender la forma de conectarlos y luego el control primero se debe analizar su funcionamiento:



**Fig 31** Funcionamiento motor a pasos

<http://upload.wikimedia.org/wikipedia/commons/6/67/StepperMotor.gif>

Como se puede ver en la imagen de ejemplo, este motor consta de 4 electroimanes, los cuales se alimentan en el orden indicado. Al empezar, el electroimán 1 se energiza forzando al motor a cuadrar y mantener con fuerza magnética en una posición, se desactiva el electroimán 1 y se activa el 2, generando un pequeño paso y lo mantiene en esta posición, luego se energiza el tercer electroimán y así se va dando los pasos. Si se cambia el orden y se regresa, se va a generar un giro en sentido contrario. Como se puede ver, todo depende del orden en

que se energizan los electroimanes. Es por esta razón que se necesita un driver, esta tarjeta conecta una fuente de energía externa al motor en el orden indicado para generar los pasos.

Un driver tiene varios objetivos, primero son los elementos de seguridad, para evitar que se dañe el motor por fallas eléctricas. El segundo objetivo muy importante es limitar la corriente del motor, esto controla directamente la potencia que el motor consume dándole un mayor torque, pero haciendo un movimiento menos fluido y calentando al motor. El tercer elemento es el control de los pasos y para esto se tiene 3 alternativas diferentes:

### **Full step drive**

Aquí se conecta como el ejemplo anterior, dando el número de pasos que el motor indica en sus características. Por lo general son 200 pasos por revolución.

### **Half Step drive**

En este caso el driver de control conecta 2 electroimanes al mismo tiempo. Energiza el electroimán 1, pero sin dejar de darle energía energiza el electroimán 2, luego desconecta el 1 y energiza el 3.

Esto permite que generemos un paso intermedio dando el doble de los pasos normales del motor permitiendo tener por lo general 400 pasos por revolución. Este efecto causa una pérdida de torque que puede notarse a velocidades altas.

### **Microstepping**

Entendido el funcionamiento anterior se puede generar casos especiales usando alimentaciones senoidales a los electroimanes para generar diferentes posiciones dentro de cada paso generando pasos intermedios. Dependiendo de la calidad del controlador, se puede generar 800, 1600, 3200, 6400, 12800, 25600 incluso 51200 pasos por revolución. Esto daría una increíble resolución, pero afectando el torque del motor. El torque es importante ya que además de mover el motor también frena e invierte su rotación, cuando el torque es muy bajo no alcanza a frenarlo e invertir perdiendo gran cantidad de pasos, por esta razón se debe balancear bien el número de pasos con el torque necesario y en esto también entra la corriente que aplicamos.

### **Funcionamiento del Driver.**

Puesto que el driver es un elemento que transforma señales de control (bajo voltaje) a potencia para el motor se necesita conocer los elementos necesarios para que funcione como se desea.

- El driver consta de una entrada de alto voltaje para conectar la energía que va a los motores.
- Salida para el motor, donde se conectan los cables del motor a controlar.
- Entrada de paso
- Entrada de dirección

Dependiendo del fabricante se controlan de forma diferente, pero básicamente al alimentar la señal de paso con una onda cuadrada de 5 voltios se va a generar movimiento. La velocidad y cantidad de pasos está dada por la frecuencia de esta señal cuadrada.

Para elegir la dirección de giro se hace exactamente lo mismo, una señal cuadrada que al estar en 5 voltios gira en un sentido y 0 el sentido contrario.

El enabler es una entrada para activar o desactivar el motor, por lo general no se la utiliza, pero está diseñada para dejar de alimentar al motor si no es necesario. Funciona de la misma forma con una onda cuadra pero cuando esta con 5 voltios se apaga el motor y al estar desconectado el motor tiene energía.

Los voltajes pueden cambiar entre driver y se debe leer al manual del fabricante para hacerlos funcionar correctamente.

## Motores y drivers de control seleccionados

Seleccionamos los siguientes motores: (Toda la información se obtiene de [8])

### **Green Monster 400 OzIn 8-Wire Stepper Motor**



Fig 32. Motor en Y

- 400 Oz In. Hybrid
- 1.8° /200 Steps Per Rev.
- 3 Amps Current Per Phase
- 8-wire Uni-polar or Bi-polar
- NEMA 23 Frame
- Factory Enameled Green Body
- Dual Shaft

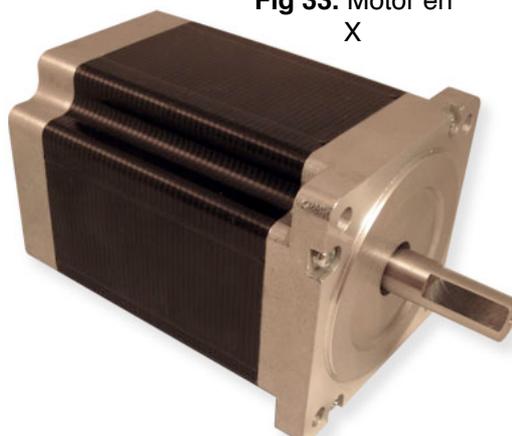
### **1150 OzIn 8-Wire Stepper**

**Motor**

- 1150 Oz In. Hybrid

- 1.8° /200 Steps Per Rev.
- 8 Amps Current Per Phase (Bi-polar)
- 8-wire Uni-polar or Bi-polar
- NEMA 34 Frame
- Single Shaft

Fig 33. Motor en X



### MondoStep 4.2 Bi-Polar Stepper Motor Driver

- High performance, cost-effective
- Supply voltage up to +50VDC



Fig 34. Driver Y y Z

- Output current up to 4.2A
- Pulse input frequency up to 300 KHz
- Automatic idle-current reduction
- 15 selectable resolutions, up to 25600 steps/rev
- Step/Direction and CW/CCW modes
- Short-circuit, over-voltage, over-current and over temperature protection

### MondoStep 7.8 Bi-Polar Stepper Motor Driver

- High performance, cost-

effective

- Supply voltage up to +80VDC
- Output current up to 7.8A
- Pulse input frequency up to 300 KHz
- Automatic idle-current reduction
- 16 selectable resolutions, up to 51,200 steps/rev
- Step/Direction and CW/CCW modes
- Short-circuit, over-voltage, over-current and over temperature protection

Fig 35. Driver X



## Control Motores

Ahora que se conoce las partes electrónicas se necesita general las señales correctas para transformar información que saldrá de la computadora a pasos y frecuencias que drivers van a utilizar para alimentar los motores. Para este capítulo se va a asumir que se tiene la velocidad con la que se va a mover los motores, el número de pasos y la dirección para los motores X, Y y Z.

### Tarjeta eZdsp F2808

La tarjeta de control seleccionada en base a su precio y por su uso en clases dentro de la universidad fue una eZdsp F2808. El chip que usa esta tarjeta es fabricado por Texas Instrumens y es el TMS320F2808, la tarjeta fue fabricada por Spectrum Digital.

Las características principales son las siguientes:

- TMS320F2808 Digital Signal Processor.
- 100 MIPS operating speed.
- 18K words on-chip zero wait state SARAM.
- 64K words on-chip Flash memory.
- 256K bits serial I2C EEPROM memory.
- 20 MHz. clock.
- Expansion Connectors (analog, I/O).
- Onboard IEEE 1149.1 JTAG Controller.
- 5-volt only operation with supplied AC adapter.
- TI F28xx Code Composer Studio tools driver.
- On board USB JTAG emulation connector.

**Fig 36.** eZdsp F2808



Para nuestra aplicación lo más importante que tiene esta tarjeta son las siguientes características.

- Varias salidas de control 3.2 V.
- Puerto serial para comunicación.
- Programación vía USB.
- Generar señales cuadradas hasta 10 KHz.
- Mantener programa en memoria que no se pierde por falta de energía.

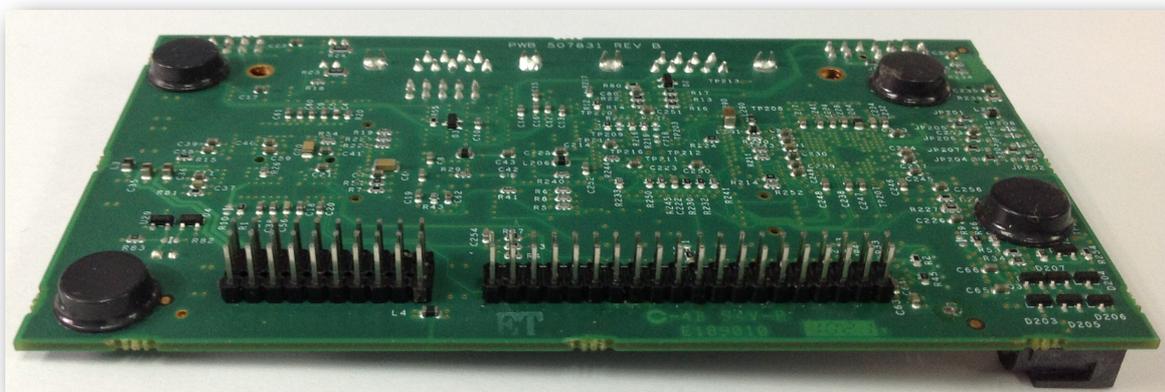
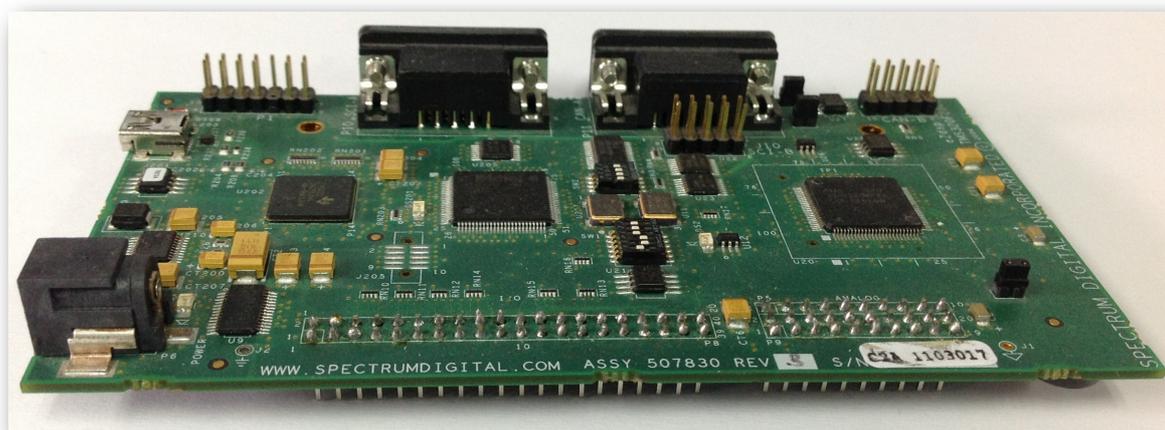
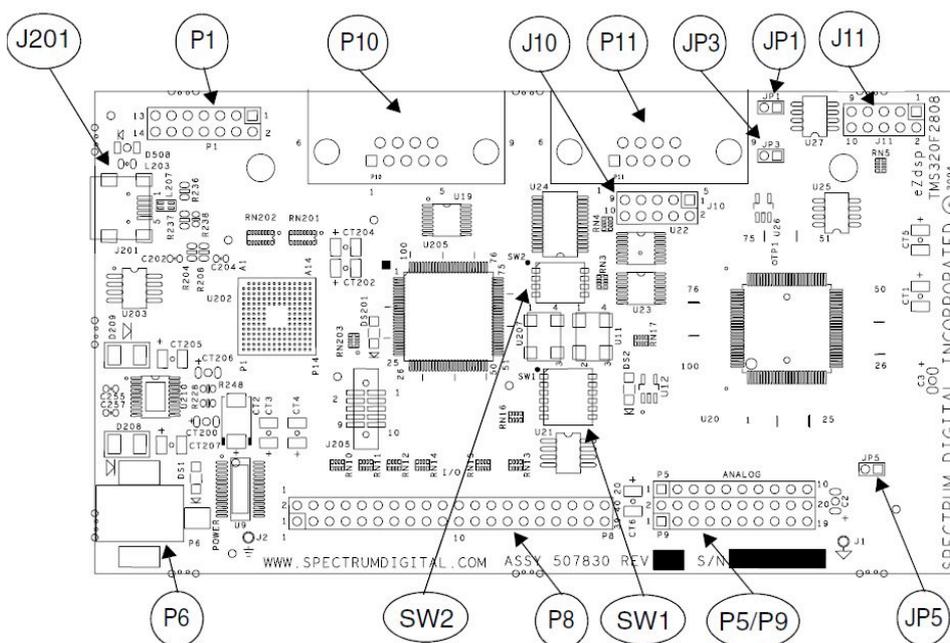


Fig 37. eZdsp F2808 otras vistas



La tarjeta consta de entradas y salidas análogas y digitales, además se tiene interfaz de conexión por puerto serial, lo que permitirá enviar mensajes entre el software en la computadora y esta tarjeta. Para la programación se utiliza USB JTAG y no se necesita más que el software que viene en un CD con la tarjeta para programarla.



**Fig 38. eZdsp F2808 esquema**

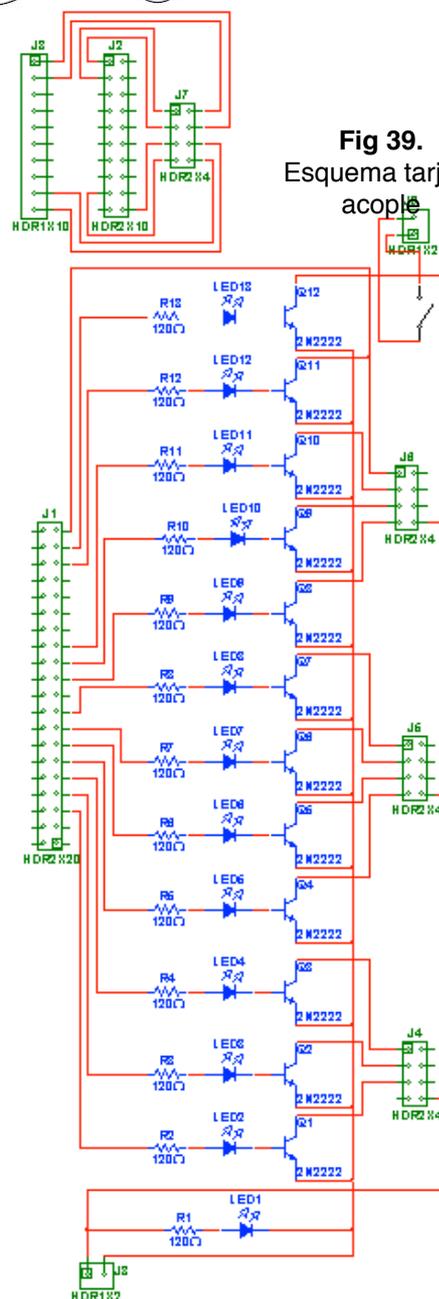
Connector	Function
P1	JTAG Interface
P8	I/O Interface
P5/P9	Analog Interface
P6	Power Connector
P10	DB-9, RS-232
P11	DB9, eCAN-A
J10	2x5 Header, SCIB
J11	2x5 header, eCAN-B
J201	USB Controller Interface

### Acople Salida de Tarjeta a control de motores

La tarjeta eZdspF2808 fue acoplada a una tarjeta diseñada y construida para facilitar las pruebas de funcionamiento y conexión con los drivers.

Un punto muy importante a considerar es la confiabilidad de las conexiones, para evitar falsos contactos y que en el futuro los cables se desconecten, es por esta razón que se utilizaron conectores de Ethernet, y los cables utilizados tendrán estos terminales, esto también facilita las conexiones puesto que no se necesitara herramientas para separar el tablero de control.

El diseño incorpora salidas de la tarjeta con salidas de tierra y energía 5 Voltios, también permite el acceso a entradas análogas para su uso si fuera necesario. Se añadió leds a las salidas para facilitar la programación ya que se puede comprobar que está enviando



**Fig 39. Esquema tarjeta acople**

la señal deseada a la salida adecuada solo viendo los leds sin necesidad de voltímetros u otros elementos. Se cálculo el valor de la resistencia  $120\Omega$  para que la corriente en el led pueda encenderlo sin problema. El voltaje de control de la tarjeta es 3.2 V pero los drivers están diseñados para mínimo 5V, es por esto que se tiene una fuente de poder adicional. Los transistores 2N2222 fueron seleccionados por su rápida respuesta a bajas frecuencias y por su bajo consumo. Se Utiliza un relé (bloque Azul) de 5V- 110v el cual prendera el plasma cuando sea necesario cortar.

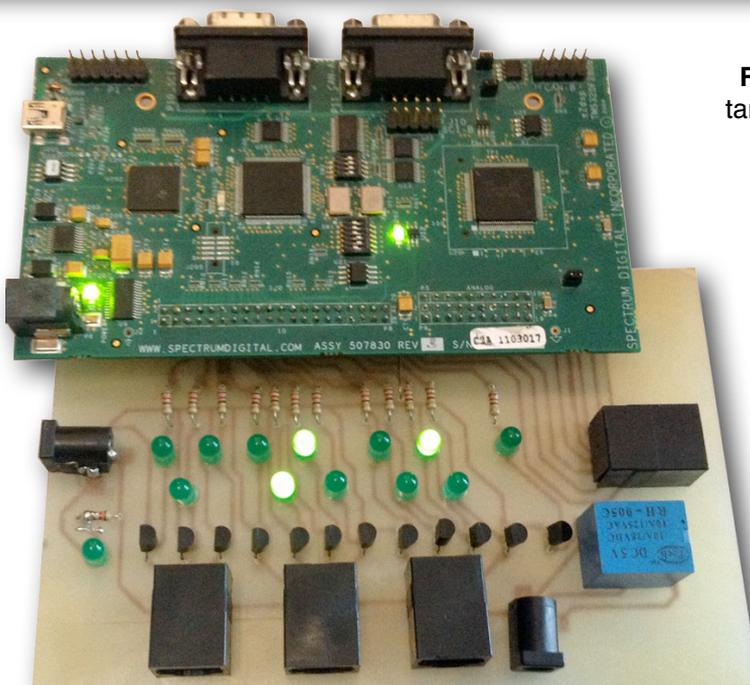
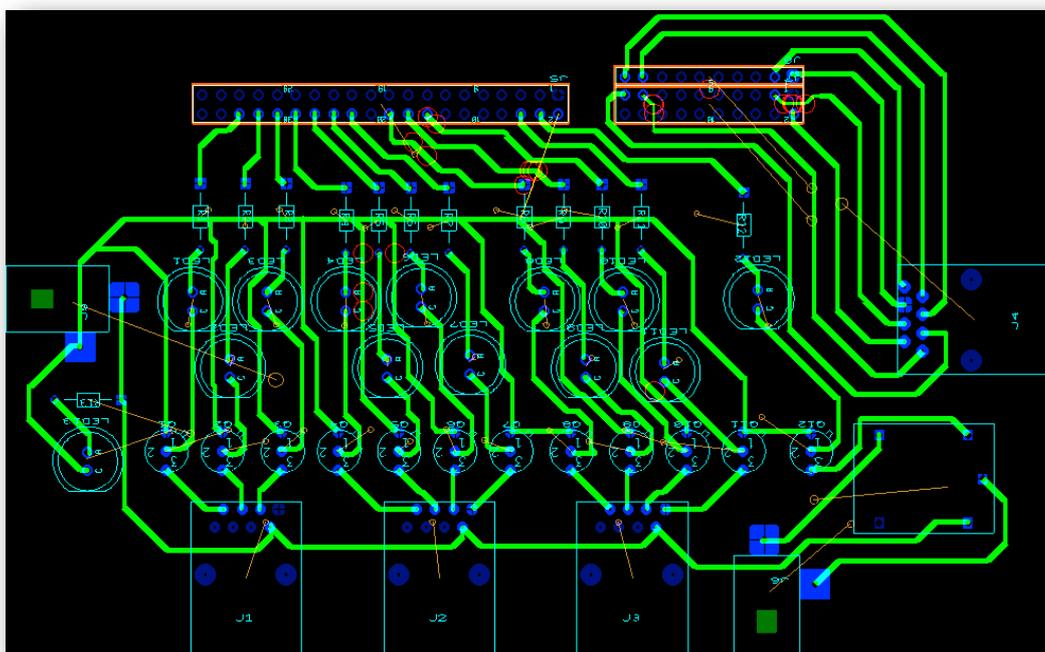


Fig 40. Esquema y tarjeta de acople final

## Programación

Para la instalación de todo el software y grabado de código en la tarjeta referirse al final de este capítulo. Para la programación de la tarjeta se va utilizar un programa denominado VisSim, el cual permite mediante bloques programar código y ejecutarlo. El programa genera un archivo compilado, y mediante Code Composer Studio V4 se va a grabarlo a la tarjeta por interfaz USB. Para esto se necesita instalar drivers y demás software que viene con la tarjeta.

Mientras se explica la programación se va a ir introduciendo algunos bloques utilizados y el porqué de las decisiones tomadas.

### Generador de onda cuadrada

Este bloque es el principal para control de los motores, el objetivo es el siguiente

- Dada una frecuencia generar una señal cuadrada
- Se necesita poder iniciar y detener la señal mediante una entrada on/off.
- Cuando se detenga la señal esta debe reiniciar desde cero y no en medio de un pulso

Este bloque tiene un funcionamiento en base a las características limitadas en la realidad. Esto lo digo ya que si funcionamiento se basa en un límite de números, para explicar pongo el siguiente ejemplo:

Se empieza con una lista de números:

1    2    3    4    5    6    7    8    9    10    11    12

Si nuestra capacidad de ver un número fuera limitada a ver un solo carácter, entonces la lista anterior la veríamos de la siguiente forma:

1    2    3    4    5    6    7    8    9    0    1    2

Como se puede ver, se repite, este es el mismo concepto que se lo usa al programar, se tiene una sumatoria que aumenta y cuando se llega al máximo más

un dígito se regresa al límite bajo. Esto se entiende al ver números binarios. Si se tiene el siguiente ejemplo:

$$-2 = 00$$

$$-1 = 01$$

$$0 = 10$$

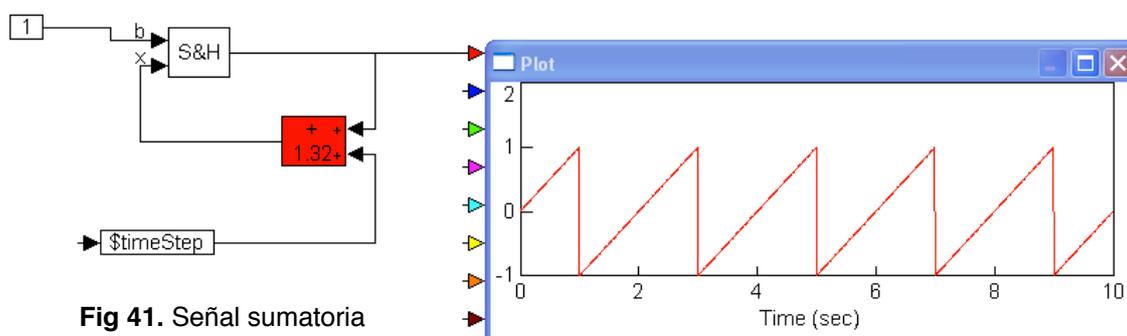
$$1 = 11$$

Cuando se pasa al siguiente número que por secuencia debería ser:

$$2 = 100 = ?00$$

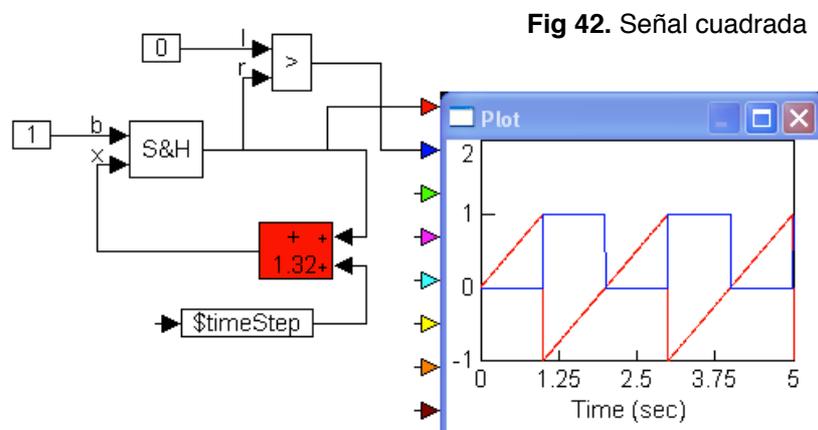
No se puede ver el tercer dígito por tanto se regresa al 00 que es -2.

Entendido lo explicado anteriormente se va a explicar la programación. Comenzamos con esta sumatoria, este circuito simplemente añade un valor al valor anterior, se puede ver el bloque (+ 1.32) el cual suma lo que tiene en sus entradas y da un resultado, este bloque tiene el límite en -1.00000 a 0.99999 seguido se va a tener un bloque (S & H) el cual funciona con una señal de reloj, este circuito al estar con un 1 envía lo mismo que llega, y con un 0 se mantiene enviando lo anterior. En este caso siempre está conectado al 1 por que solo lo se lo usa para retroceder un paso, sin él la sumatoria no se podría dar. La variable \$TimeStep es parte del sistema y es el valor que depende de la velocidad de reloj que se lo usa. Este valor es constante.

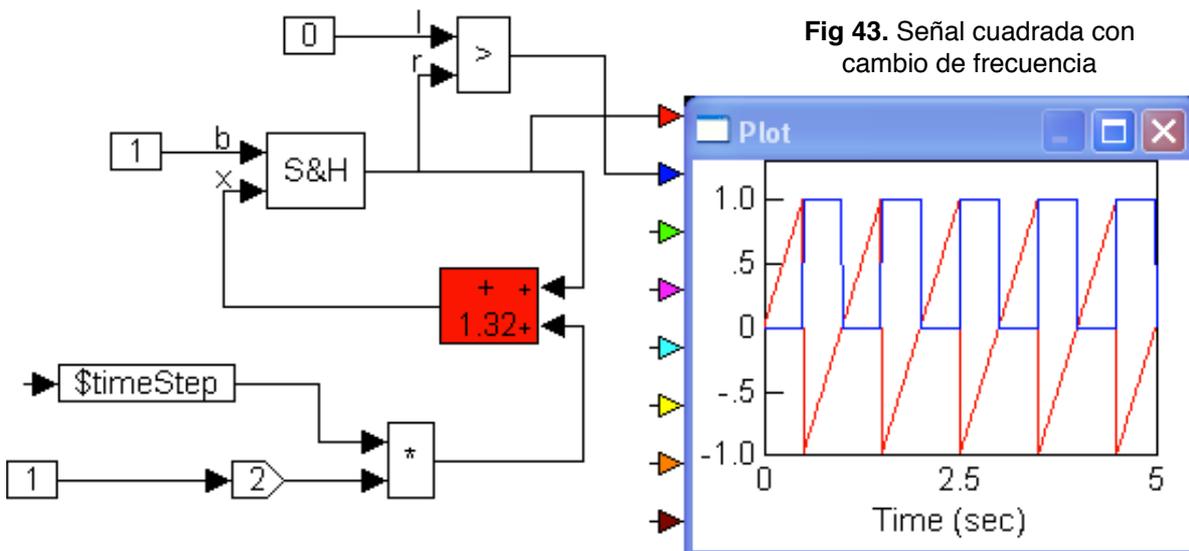


Una vez generada esta señal sierra se necesita pasarla a una onda cuadrada, con

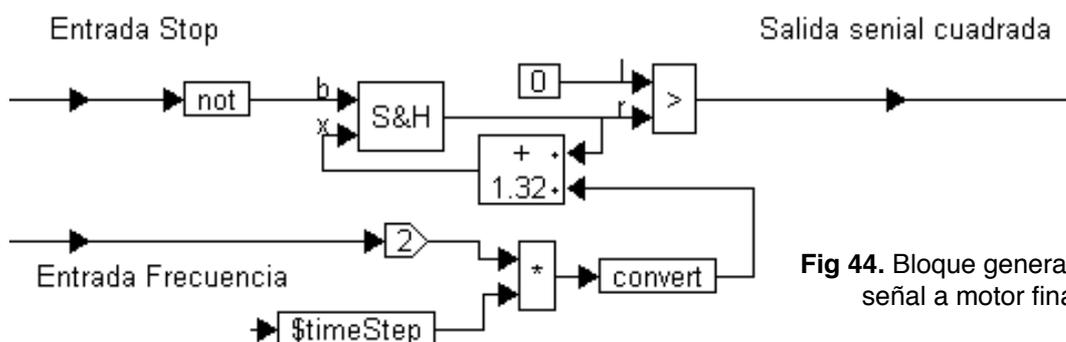
un bloque comparador se consigue esto ( $>$ ) ya que al ser la señal menor a 0 se va a tener un 1 y al tener un valor mayor a 0 se tiene un cero.



Con esto ya se tiene la señal cuadrada pero no se puede cambiar la frecuencia ni deshabilitarla. Para poder cambiar la frecuencia lo que se necesita es aumentar o disminuir el valor de  $\$TimeStep$ , ya que al ser una valor más grande la suma llegara al límite más rápido generando una señal de mayor frecuencia. Es por esto que multiplicamos el valor  $\$TimeStep$  por el de la frecuencia deseada, se multiplica todo esto por dos ya que el  $\$TimeStep$  es la mitad del periodo que tiene el ciclo de reloj.



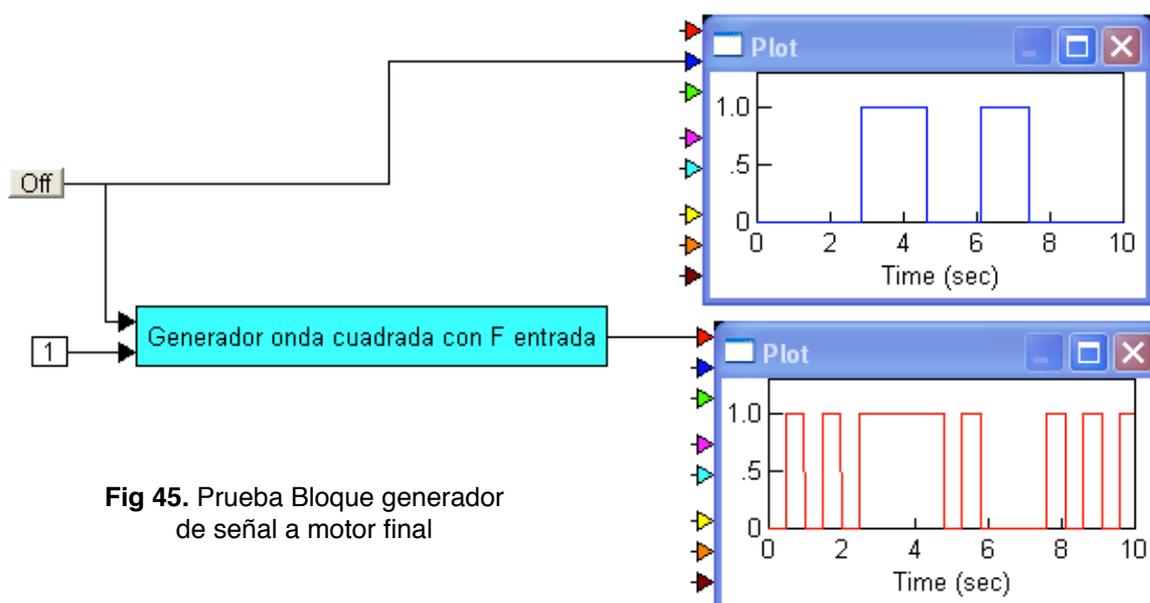
Se aumenta un bloque (Convert) para evitar errores si por equivocación no llegara un número. EL circuito final con todos estos cambios se muestra continuación.



**Fig 44.** Bloque generador de señal a motor final

Para probar que todo este funcionando correctamente se va a definir una frecuencia de 1Hz y se activa el generador, como se ve en la imagen, cuando se activa la entrada Stop se detiene el generador, cuando se desactiva inicia.

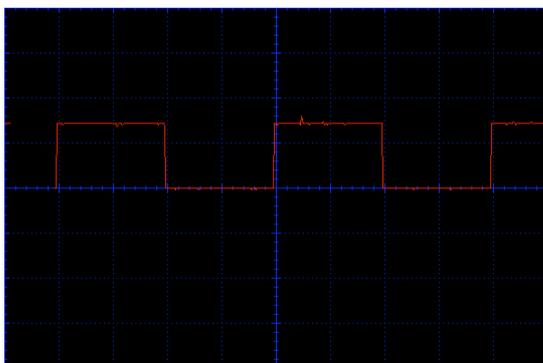
Todo lo anterior queda en teoría hasta que se demuestra mediante un osciloscopio el correcto funcionamiento del bloque generador de onda cuadrada. las siguientes imágenes:



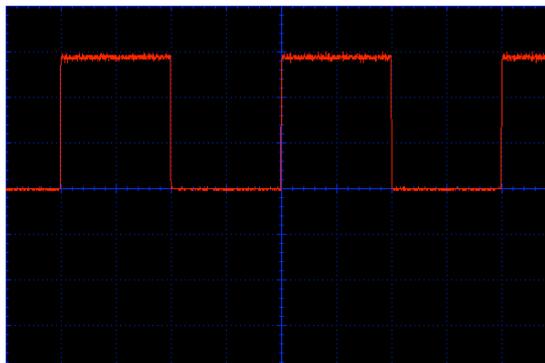
**Fig 45.** Prueba Bloque generador de señal a motor final

**Fig 46.** Prueba real con osciloscopio Bloque generador de señal a motor final

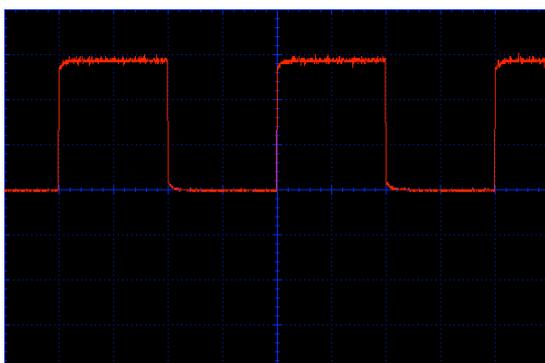
Señal 1 Hz



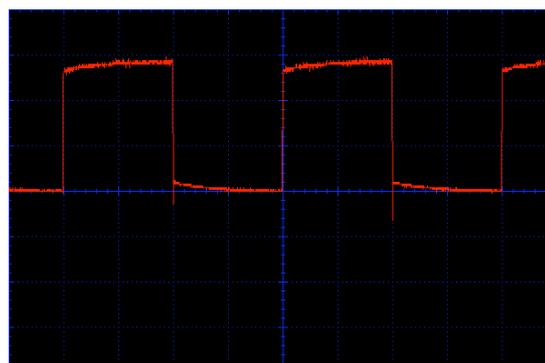
Señal 10 Hz



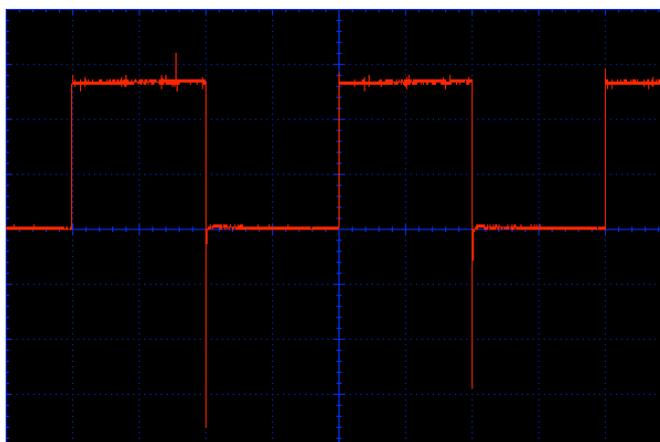
Señal 100 Hz



Señal 1 KHz



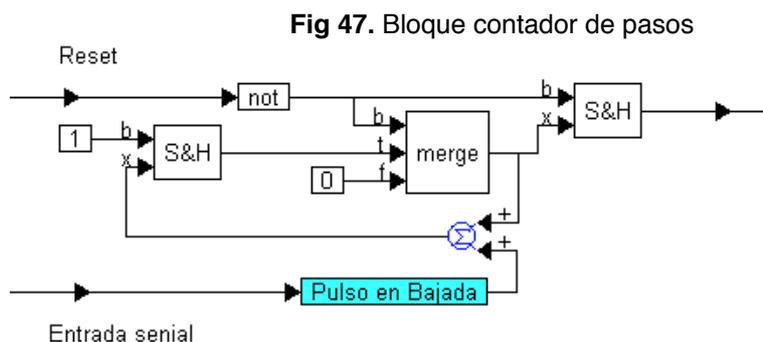
Señal 10 KHz



## Contador de pasos

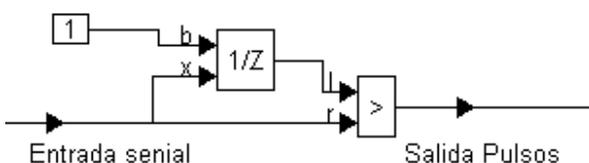
El circuito para contar pasos va a tener varias características similares al anterior pero su función es muy diferente ya que va a contar el número de veces que la señal dio un alto.

- Entrada de señal cuadrada a contar
- Entrada para reiniciar el conteo desde 0
- Salida con el número de pasos, si la entrada de reinicio esta en 0 debe permanecer con el último valor.



Este circuito tiene como base un sumador al igual que el anterior, pero en este caso se tiene límites muy amplios, el bloque SummingJunction es el encargado de sumar. Se tiene el mismo (S&H) y el merge del generador de onda, pero la gran diferencia está en el circuito Pulso en bajada y el (S&H) de la salida.

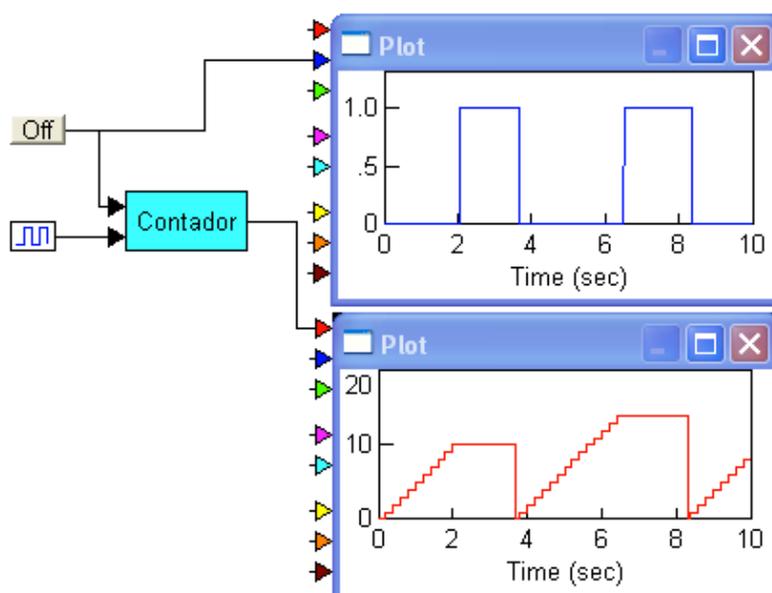
**Fig 48. Bloque pulso en bajada**



Como su nombre lo indica, este circuito pulso en bajada genera un pulso cada vez que la señal pasa de estar en alto a estar en bajo. Esto se utiliza para contar el paso cuando termina no cuando comienza. EL circuito funciona retrasando la señal y comparándola, es por esto que se utiliza un  $(1/Z)$ . Se

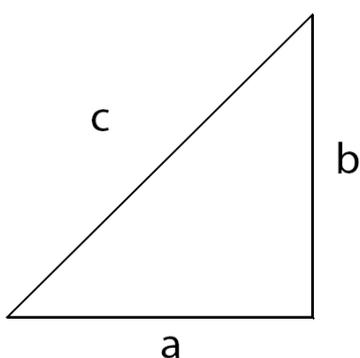
compara la señal con un instante anterior, solo el caso en que la señal estaba alta y después baja saldría un 1, en los demás casos saldría 0, por esto se logra generar los pulsos.

El (S&H) de salida tiene la función de mantener el último valor. El reset se ejecuta cuando se pasa la señal de bajo a alto, pero mientras esta en bajo va a mostrar el último valor.



**Fig 49.** Prueba Bloque pulso en bajada

### Calculo de frecuencias X y Y



**Fig 50.** Triángulo Pitágoras

Una parte trascendental de este proyecto es la velocidad de corte. No solo es importante que llegue al lugar adecuado sino que tenga una velocidad lineal siempre constante.

Usando el teorema de Pitágoras se puede calcular la distancia  $c$  en base a los lados  $a$  y  $b$  de la siguiente forma:

$$c = \sqrt{a^2 + b^2}$$

Al imaginar que las líneas  $a$ ,  $b$  y  $c$  son caminos a recorrer, el eje X será la línea  $a$ , y el eje Y la línea  $b$ . Si " $a$ " es una distancia recorrida se puede calcularla en base a la velocidad y al tiempo.

$$a = v_a \times t$$

Lo importante es mantener constante la velocidad lineal, esta se representa por  $v_c$  ya que es la velocidad que tendría un objeto en movimiento en c. Si el tiempo que toma en trasladarnos “a” es igual al tiempo que toma trasladarnos “b”, esto va a ser igual al tiempo si se va por “c”, por lo tanto al tener los tiempos constantes se puede calcular la velocidad en a y b, en función de la velocidad de c.

$$t = \frac{e}{v}$$

$$\frac{c}{v_c} = \frac{a}{v_a} = \frac{b}{v_b}$$

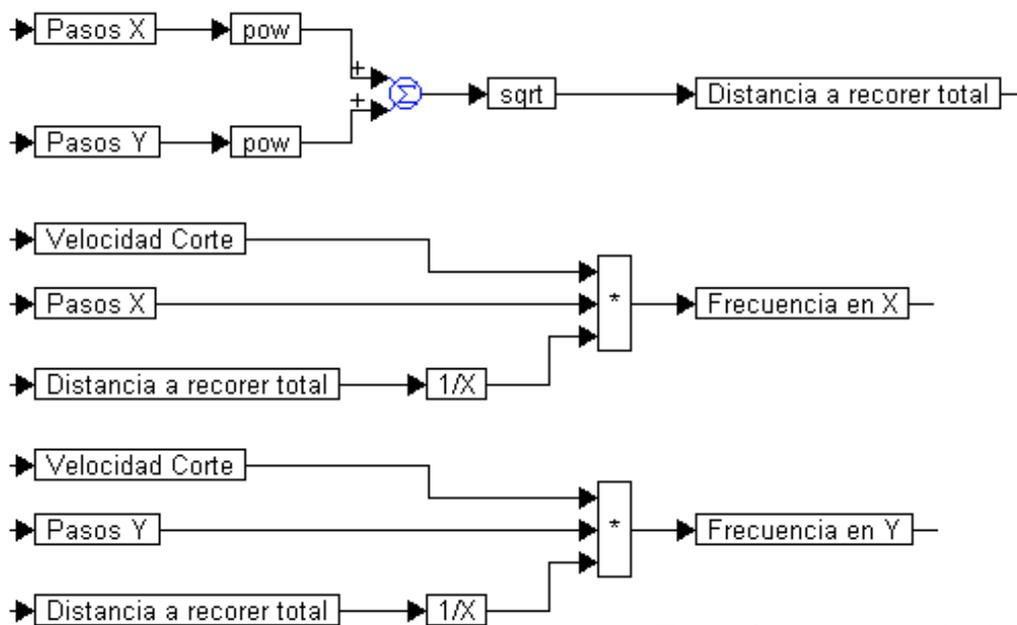
$$v_a = v_c \frac{a}{c} \quad v_b = v_c \frac{b}{c}$$

$$v_a = v_c \frac{a}{\sqrt{a^2 + b^2}} \quad v_b = v_c \frac{b}{\sqrt{a^2 + b^2}}$$

Esto implica que conociendo la distancia que se va a mover el eje X, el eje Y y la velocidad lineal que se tiene que mantener se puede calcular las velocidades de cada eje.

- Este bloque debe permitir la entrada de distancia para X y Y.
- Entrada de velocidad lineal
- Debe entregar las velocidades o frecuencias para los motores X y Y

Puesto que un paso va a ser una unidad de distancia, se puede alimentar la función con el número de pasos y debe entregar la velocidad de los motores.



**Fig 51.** Cálculo de velocidades para mantener velocidad lineal constante

### Generador de onda cuadrada con contador de pasos

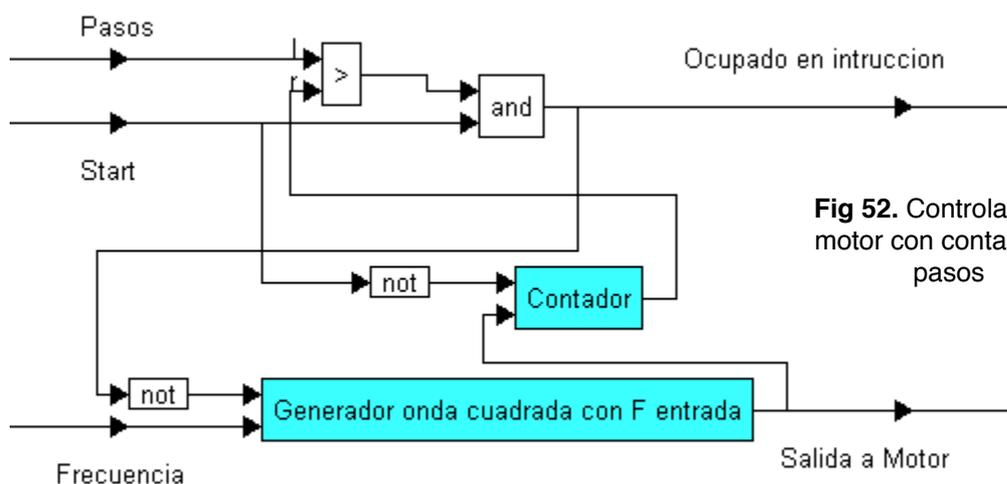
Este bloque se encargará de dar las señales en pasos para el motor, por lo tanto se necesitan las siguientes características:

- Debe tener una entrada para el valor de la frecuencia que se desea utilizar, esto controla la velocidad del motor.
- Debe tener una entrada para el número de pasos que se desea dar, esto controlará la posición del motor.
- Se necesita una entrada para detener y resetear el conteo de pasos, para que empiece una nueva instrucción.
- Salida de señal cuadrada para el motor
- Salida de señal de información que indica que está ejecutando una instrucción o no.

Una vez entendido el funcionamiento del generador de onda cuadrada y el contador se puede agruparlos en un solo bloque, el cual recibe las instrucciones de frecuencia y pasos para luego ejecutarlos, al terminar se necesita que avise

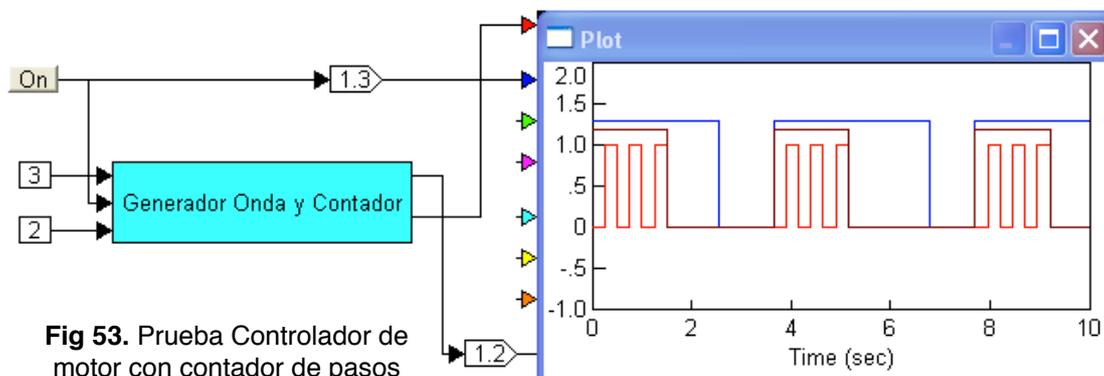
para enviar una nueva instrucción, al realizar esto debe reiniciar el contador y comenzar nuevamente.

Para esto se necesita un bloque de comparación  $>$  el cual compara el valor de pasos ingresados con el del contador, cuando el valor se iguala este bloque pasa a 0 deteniendo el generador de onda cuadrada. Existe también el Bloque AND el cual une las señales de inicio y de comparación para indicar si el motor esta ocupado en una instrucción o ya termino y espera la siguiente. Cuando el Start pasa de bajo a alto, reinicia el contador comenzando nuevamente un conteo.



**Fig 52.** Controlador de motor con contador de pasos

Como se puede observar en la imagen a continuación dado una cantidad de pasos 3, una frecuencia de 2 se puede ver que al completar los pasos el generador de onda se detiene y cambia la señal a no ocupado. Cuando se cambia el botón de bajo a alto, comienza nuevamente el proceso y vuelve a generar los pasos.



**Fig 53.** Prueba Controlador de motor con contador de pasos

Ya que nuestro sistema cuenta con dos ejes se debe repetir todos estos bloques para generar las señales de control de los ejes X y Y, en este momento se asume tener la información de pasos en X, pasos en Y, y frecuencias de cada uno, se conoce cuando se tiene un nuevo mensaje y avisa mediante la variable pedir mensaje cuando se completa el último mensaje.

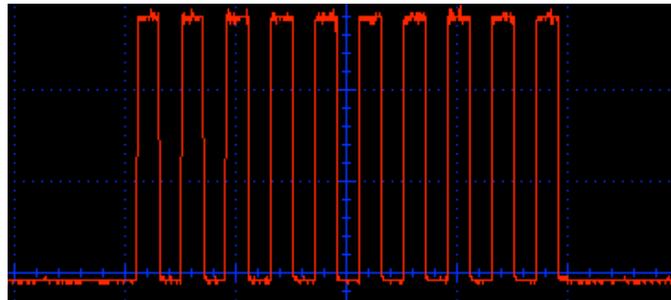


Fig 54. Controladores de motores X y Y

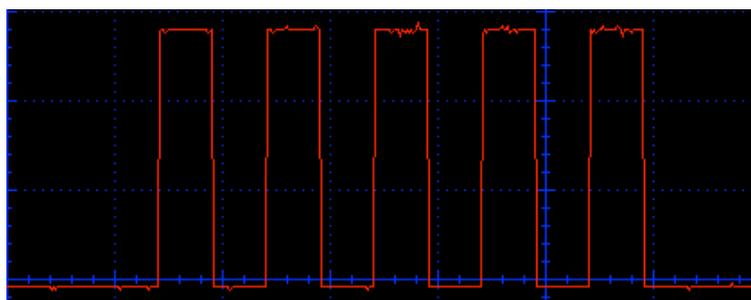
A continuación se tiene imágenes obtenidas del osciloscopio las cuales demuestran el correcto funcionamiento con frecuencia y número de pasos correctos.

Fig 55. Prueba con osciloscopio  
Controladores de motores X y Y

10 Hz 10 Pasos



10 KHz 5 Pasos

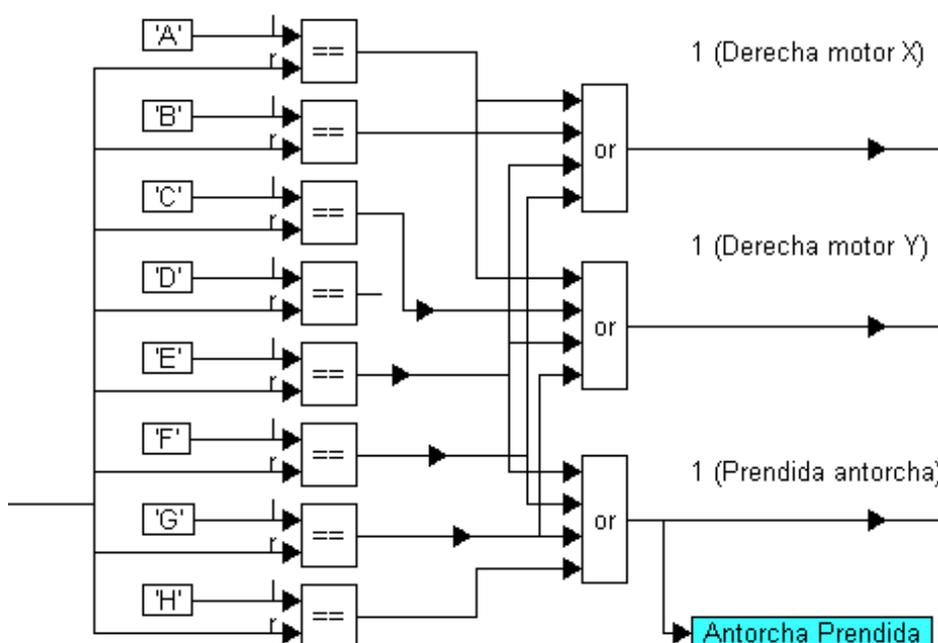


Con el control de los motores completo lo que falta por realizar es la selección de direcciones correcta para saber a que dirección los motores deben moverse también esta incluido si la antorcha del plasma debe estar encendida, para diferenciar un movimiento para posicionar la antorcha con uno de corte. Para esto se asume tener la información en forma de una letra, como lo indica la siguiente tabla.

**Tabla 3.** Instrucciones dirección motores.

LETRA	INFORMACIÓN
A	X Derecha, Y Derecha, Antorcha Apagada
B	X Derecha, Y Izquierda, Antorcha Apagada
C	X Izquierda, Y Derecha, Antorcha Apagada
D	X Izquierda, Y Izquierda, Antorcha Apagada
E	X Derecha, Y Derecha, Antorcha Prendida
F	X Derecha, Y Izquierda, Antorcha Prendida
G	X Izquierda, Y Derecha, Antorcha Prendida
H	X Izquierda, Y Izquierda, Antorcha Prendida

Con esta información se genera el siguiente código:



**Fig 56.** Interpretación de instrucciones dirección .

Para este momento todo esta funcionando si se conoce los valores de pasos en X, Y y la velocidad linea, en el capítulo de protocolo se encargará de obtener estos datos.

## Análisis de potencia

Para definir la cantidad de potencia que se va a necesitar para tener un movimiento deseado de la mesa se procede a analizar las características de los motores y drivers seleccionados.

**Tabla 4.** Amperajes máximos de los componentes

Elemento	Amperaje máximo
Motor Negro 1150 OzIn	8Amp
Motor Verde 400 OzIn	3 Amp
Driver MondoStep 7.8	4.2 Amp
Driver MondoStep 4.2	7.8 Amp

Como se puede observar, para el combo Motor verde MondoStep 4.2 el límite máximo da el motor, con un máximo de 3 Amp. El caso Motor Negro MondoStep 7.8 se tiene que el máximo da el Driver con 7.8 Amp.

Puesto que se tiene 2 motores para el eje X se necesita 2 fuentes de por lo menos 7.8 Amp. Para el Eje Y se necesita una fuente de 3 Amp, pero puesto que el motor Z esta junto a Y se puede usar la misma fuente para dar energía a los



24Volt 6.5Amp Switching Power Supply



40VDC 10Amp Linear Power Supply

**Fig 57.** Fuentes de poder utilizadas

dos motores, es por esta razón que necesitaremos una fuente de por lo menos 6 Amp. Con los datos anteriores se determinaron las fuentes de la figura 57, del proveedor Probotix( [www.probotix.com](http://www.probotix.com))

Con los datos de estas fuentes se puede determinar los amperajes máximos que se va a necesitar para cada una de las líneas.

**Tabla 5.** Amperaje con fuente para cada motor

Línea	Amperaje Necesario	Fuente
1 Motor X con Driver	7.8 Amp	10 Amp
1 Motor Y con Driver	3 Amp	Fuente 6.5 (/ 2) 3.25 Amp
1 Motor Z con Driver	3 Amp	Fuente 6.5 (/ 2) 3.25 Amp

Los datos anteriores muestran que las fuentes cumplen las especificaciones necesarias, pero no se debe confundir esto con la potencia que se necesita alimentar a las fuentes ya que estos valores son en DC, para determinar el consumo de entrada en AC presentamos la tabla 6:

**Tabla 6.** Consumo fuentes.

Fuente	Consumo Voltios	Consumo Amperios
24 Volt 6.5Amp Switching Power Supply	100-120V	4 A
40VDC 10 Amp Linear Power Supply	100-120V	5 A

Determinados los valores se puede observar que se necesita llevar un total de 10 A para alimentar el eje X y 4 A para los ejes Y y Z.

## **Determinación de cables a utilizar.**

### **Cables de potencia**

Una vez conocida la potencia que se va a utilizar, se determina el cable necesario para transmitir la corriente a las fuentes de poder. Puesto que las fuentes fueron seleccionadas en voltaje 110, todo los cables deberán poder llevar este voltaje. El

voltaje en conjunto para poder llevar la potencia total debería soportar dos fuentes para la línea X y una para YZ por un total de 14 A.

En base a la información del fabricante (General Cable) se determina que para una distancia baja como es en nuestro caso, un cable AWG 16 será suficiente para llevar la corriente necesaria sin dañar el cable. Es

por esta razón que se compro el cable “concéntrico 2x16 AWG” o también llamado “cable sucre 2x16”.

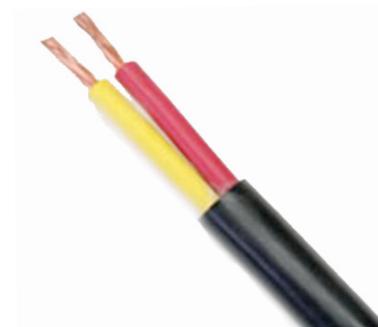


Fig 58. Cable 2x16

<http://www.refermat.com.ve/imagenes/productos/760582.JPG>

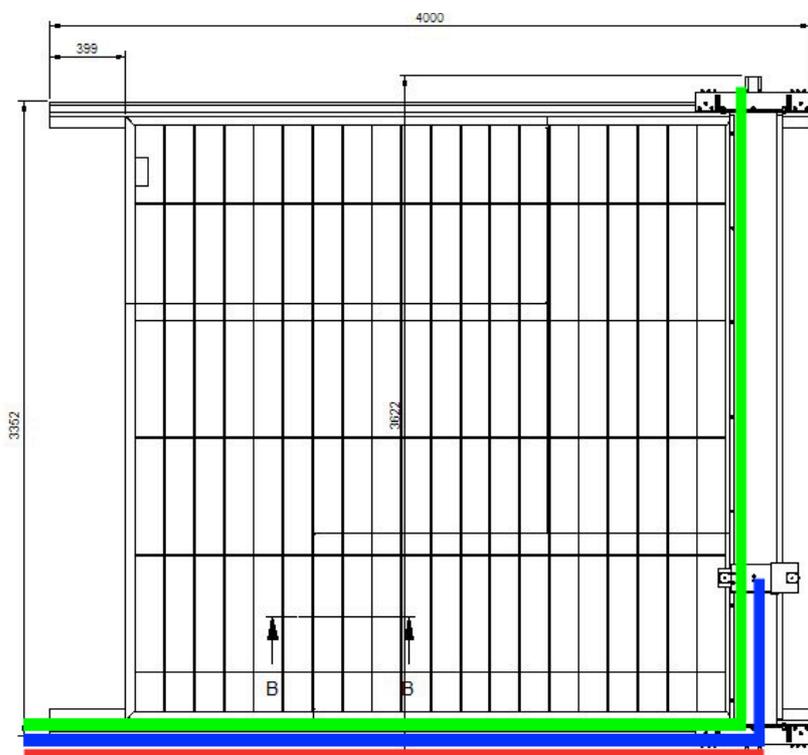


Fig 59.  
Esquema  
cables de  
potencia

Como se puede observar en la Fig 59, se tiene 3 líneas de potencia, las cuales tienen las siguientes longitudes:

Tabla 7. Longitud cables

Línea	Color	Longitud
Motor X Izquierdo	Rojo	4000mm
Motor X Derecho	Verde	$4000 + 3352 = 7352$ mm
Motor Y y Z	Azul	$4000 + 3352 = 7352$ mm

La línea Azul y Verde tiene la misma distancia, la diferencia está en que la línea azul tiene va cambiando su longitud dependiendo de la posición del carro en Y.

El total de cable necesario será la suma de las tres líneas (18.704 m). Para evitar cualquier problema en la forma y posición de cable y que no quede muy estirado se va a solicitar un 30% extra de cable. Por lo tanto el pedido es para 25 m de cable 2x16 AWG.

## Señal de control

El principal problema que existe en una mesa CNC de plasma es la gran cantidad de ruido electromagnético que produce el plasma, es por esta razón que se debe tener mucho cuidado con el cable que controla los motores.

En base a lo explicado previamente en este proyecto se conoce que la velocidad máxima a la que un motor se necesita mover es 88.0841 RMP. Para poder tener un error permisible se necesita determinar la cantidad de pasos por vuelta necesarios, pero esto aumenta la frecuencia de control a los motores, esto quiere decir que mientras mayor resolución se tiene, mayor frecuencia y esto es un problema puesto que alta frecuencia dentro de un cable tendrá mucha atenuación y distorsión generando problemas y falla en la cantidad de pasos.

Puesto que el efecto de distorsión depende de la alta frecuencia se va a poner como límite 3KHz que en la banda de frecuencias es ELF (frecuencia extremadamente baja), la cual se puede corregir deformando la señal que se envía y que el efecto la reconstruya al final. Con estos datos se va a determinar el valor máximo de número de pasos por vuelta.

Si se tiene el máximo valor de giro de un motor es 88.0841 RMP

$$88.0841 \text{ RPM} \times 1 \text{ min}/60\text{s} = 1.47 \text{ revoluciones por segundo}$$

Si Frecuencia = RPS x pasos en una vuelta

$$\text{entonces pasos por vuelta} = 3\text{KHz} / 1.47 = 2040$$

En base a las especificaciones de los drivers se puede seleccionar 1600 pasos/rev o 3200 pasos/rev, estos en frecuencia serían los siguientes valores:

$$F_{1600} = 1.47 \times 1600 = 2352 = 2.3 \text{ KHz}$$

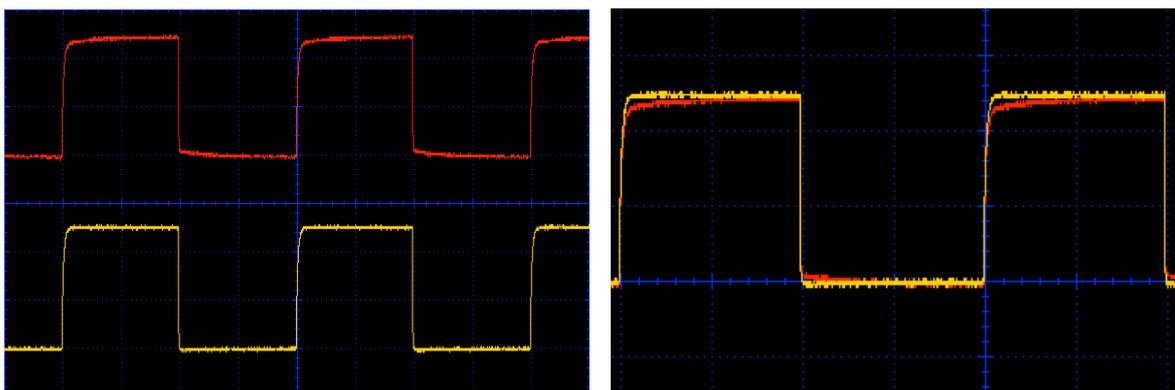
$$F_{3200} = 1.47 \times 3200 = 4704 = 4.7 \text{ KHz}$$

En base a pasos de motor, se puede observar que al usar 1600 pasos/rev se encuentra una precisión 0.1, mientras que con 3200 pasos/ rev se consigue 0.055.

Ya que 4.7KHz sigue siendo una frecuencia baja se decidió utilizar 3200 pasos/rev pero si llegara a dar problemas se debería disminuir a 1600 pasos/rev que sigue cumpliendo con las necesidades del cliente.

A continuación se muestra en amarillo la señal de salida a 5 KHz, y en rojo la señal al final del cable de 25 metros. Esto comprueba que no va a existir ninguna problema en frecuencias mayores.

**Fig 60.** Pruebas de señal en osciloscopio



**Fig 61.** Cable apantallado  
<http://hpindustrialperu.com/productos/catalogo/cable%20apantallado%20hp%20industrial%20indec0.jpg>

Conocida la frecuencia de la señal se puede afirmar que efectos de ruido generados por la señal o el canal de transferencia son muy bajos, pero el ruido generado por el plasma si puede causarnos problemas. Es por esta razón que se utiliza un cable de datos apantallado, este cable tiene un recubrimiento metálico que utiliza el efecto Jaula de Faraday para aislar la información del ambiente y reducir el ruido dentro del cable. Este cable siempre lo se encuentra con pares, y para nuestra aplicación cualquiera que tenga

cuatro pares funciona ya que se necesita utilizar 1 para la dirección, un segundo para los pasos, y un tercer par habilitar el motor.

Al igual que los cables de potencia, este cable debe viajar las mismas longitudes por lo tanto se necesita aproximadamente 25 m de cable.

## Cadena Portacables

La mesa CNC tiene 3 líneas de cables, que pasan por un eje X móvil y un eje Y móvil de la misma forma, para evitar problemas con el movimiento de los cables se decidió adquirir canaletas móviles denominadas cadenas portacables. Estos elementos tienen un precio elevado, por esta razón es muy importante determinar el mínimo largo necesario para que tengan un funcionamiento adecuado.

Como se puede observar en el gráfico a continuación, es muy importante conocer las distancias de movimiento para determinar el largo de esta cadena.

Las medidas de la mesa son 3200 x 3200 mm, pero a esto se debe añadir rangos extra de operación que fueron determinados en el diseño mecánico para obtener un total de 3300 x 3300 mm de distancia a recorrer por los motores, esto quiere decir que se necesita canaletas que puedan recorrer completamente esta distancia.

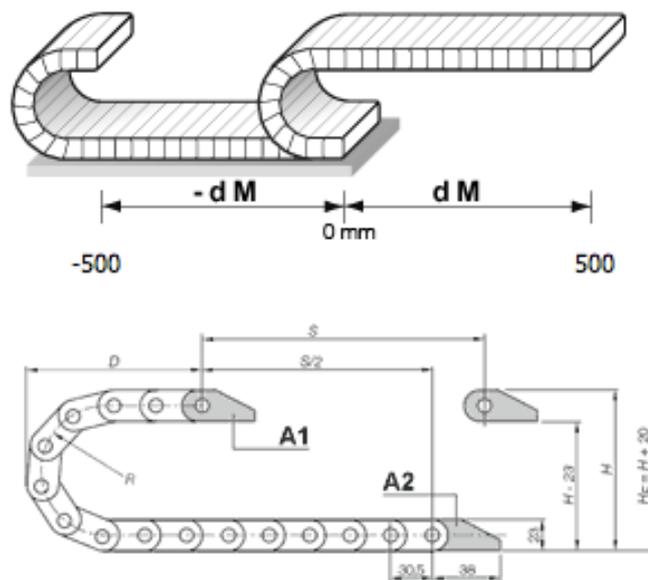
Si se centra el inicio de la cadena portacables en el centro de la mesa se puede observar que tendremos 1650 mm a cada lado, esta es la distancia que se debe mover con los cables, pero puesto que la cadena no es lineal sino forma un semicírculo se va a determinar su perímetro y así encontrar la longitud mínima de la cadena portacables.

El perímetro de un círculo es

$$P = 2\pi R$$

La cadena forma la mitad de un círculo, por lo tanto

$$P = \pi R$$



**Fig 62.** Cadena Portacables

<http://www.igus.es/iPro/>

iPro\_01\_0003\_0011\_ES  
es.htm?c=ES&l=es

En base a catálogos del proveedor (IGUS) se encuentra que una cadena con las características necesarias para la cantidad de cables que se debe movilizar. El radio de esta cadena es 100 mm.

$$P = 314mm$$

Ya que la distancia a recorrer es 1650mm mas el semicírculo de 314mm se tiene se necesita una cadena con:

$$1650 + 314 = 1964mm$$

La mesa es cuadrada por esto se utiliza 2 cadenas portacables de 2 m cada una.

## Tablero de control

Este tablero va a ser el cerebro de la mesa, ya que se encargará de dar energía a los componentes y controlar los motores para que realicen los movimientos adecuados. En la tabla a continuación se presenta una lista de las funciones del tablero.

**Tabla 8.** Características tablero

Función	Explicación
Protecciones de sobrecarga y cortocircuito.	En base a análisis se deberá determinar sistemas de protección de la máquina para proteger los elementos electrónicos.

Función	Explicación
Seguridad	Debe existir algún método rápido para detener la máquina y evitar accidentes.
Distribución de energía	El tablero es el control principal de energía, a él llega la energía eléctrica en 110 V y desde el tablero se la distribuye a las tres fuentes. También se encarga de dar energía a la tarjeta de control.
Control de motores	El sistema tarjeta de control, acople de tarjeta y dos fuentes de energía son los utilizados para controlar los motores, se utiliza una fuente externa para aislar la señal de control de los motores, de la energía de la tarjeta de control.
Comunicación con computador	Puesto que la tarjeta de control puede comunicarse mediante puerto Serial, se debe permitir la entrada de comunicación.
Activar Plasma	La mesa CNC debe poder activar el plasma, cuando empieza un corte, es por esto que se necesita algún método para activar el plasma.

## Protección de sobre carga y cortocircuito

Para cortar la energía de la máquina se va a utilizar un disyuntor o comúnmente denominado “braker”. Este elemento eléctrico puede abrir un circuito si la corriente que pasa por el excede un valor determinado. La protección será para evitar posibles cortocircuitos, que podrían darse al romper cables o aplastarlos y también para evitar que componentes eléctricos (incluidos cables) se quemen por alguna falla que genere un consumo mucho mayor de amperaje.

En la tabla a continuación se muestra el consumo de los elementos para determinar el correcto valor del elemento de protección:

**Tabla 9.** Consumo amperaje diferentes elementos

Elemento	Consumo
Fuente poder motor X derecho.	5 A
Fuente de poder motor X izquierdo.	5 A
Fuente de poder motores Y y Z.	4 A
Fuente de poder para tarjeta de control.	0.5 A

Elemento	Consumo
Fuente de poder para control de motores.	0.5 A
<b>Total</b>	<b>15 A</b>

En el mercado el valor mas cercano será el disyuntor bipolar 110 V 16 A por esta razón fue el elegido.

## Seguridad

Para evitar accidentes se a dispuesto de un selector para el encendido de la máquina, el cual controlará la energía de todos los componentes. Si el selector se apaga la energía de todos los componentes se desconectará. Se utiliza un selector de 2 posiciones ya que solo se necesita encendido o apagado.



Fig 63. Selector 2 posiciones

## Distribución de energía

Para poder distribuir la energía a todos los elementos del tablero y controlar esta distribución mediante el selector se a dispuesto el uso de un relé y un contactor. Estos dos elementos tiene la misma función pero se diferencian en la cantidad de amperios que pueden transmitir.

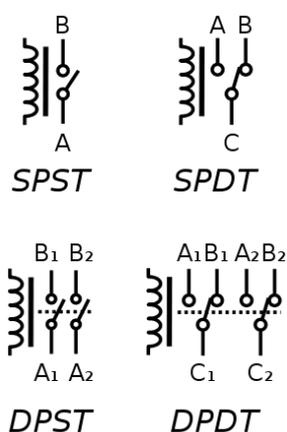


Fig 64. Funcionamiento de Relay y Contactor

Estos elementos se utilizan cuando se debe controlar un sistema mediante una señal baja sin mezclar los sistemas. Todo relé o contactor tiene un elemento que cierra un circuito cuando se le aplica energía, la mayoría de los casos usa un electro imán, que al encenderlo cierra un circuito conectando (o desconectando). Como se puede observar en la imagen, al dar energía a la bobina esta cierra el circuito.

Para dar energía las fuentes de los motores se utilizó un contactor marca Schneider con código LC1-D12, el relé que

se utiliza para las señales de control y la tarjeta de control es uno Schneider RUMC2AB1F7.

## Control de motores

Aunque la tarjeta tiene una capacidad para poder dar energía 5V para control de los motores, esto se prefirió evitar, para aislar la tarjeta de la señal de control y evitar cualquier daño posible en la tarjeta por sobre carga. También se decidió tener una fuente aparte para que pueda ser fácil de cambiar, si existiera una SNR (relación señal a ruido) muy baja se puede simplemente cambiar a una fuente con mayor voltaje y se resuelve el problema. Es por esta razón que la tarjeta de acople solo activa transistores estos son los que dan las señal a los motores.

## Comunicación con computador

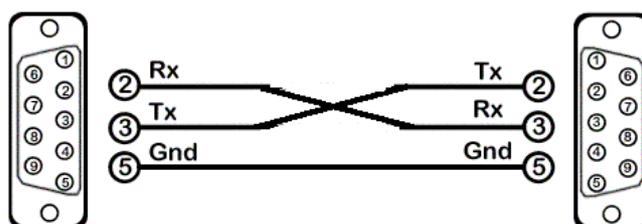
La tarjeta de control tiene incorporada una salida por puerto serial, ya que la tarjeta esta dentro de un tablero se debe

acoplar esta salida al tablero y para esto se va a conectar de forma



**Fig 66.** Cable Serial a USB

[http://astrocosmos.es/wp-content/uploads/2009/12/usb\\_serie.jpg](http://astrocosmos.es/wp-content/uploads/2009/12/usb_serie.jpg)



**Fig 65.** Conexión cable serial cruzado

<http://www.ecoustics.com/electronics/forum/home-video/323296.gif>

cruzada, esto se

hace para evitar el uso de un cable cruzado con un computador después. El un terminal va conectado a la tarjeta mientras que el otro va acoplado al exterior del tablero. Si el computador no tiene un puerto serial no hay

problema ya que se puede utilizar un convertidor de serial-usb.

## Activar Plasma

Para activar el equipo plasma no se necesita mas que un relé extra ya que solo se necesita cerrar un circuito eléctrico, se utilizo un relé de 5v DC a 110 AC, esto quiere decir que al conectar 5 voltios en sus extremos se conecta un circuito que puede ser controlado hasta con 110v AC.

## Armado del tablero.

Para el armado del tablero se utiliza unos pocos elementos extra los cuales se enumeran a continuación.



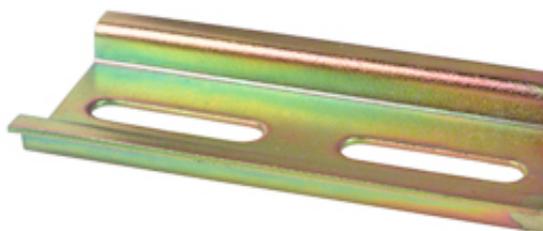
**Fig 67.** Canaleta Ranurada  
<http://www.ndu.cl/fotografias/45010604BO.jpg>

### Canaleta Plástica Ranurada

Esta canaleta permite el transporte de cables para mantener un orden dentro del tablero, al ser ranurada, permite el ingreso y salida de cables fácilmente. La canaleta consta de una base con ranura y una tapa la cual se la coloca una vez terminado todo el paso de cables, es de fácil manipulación.

## Riel DIN

Esta riel es de tamaño estándar y permite colocar elementos electrónicos como el disyuntor los relés y el contactor, es de mucho uso ya que permite cambiar fácilmente componentes y permite fácilmente organizar los elementos dentro del tablero. Al utilizar riel DIN se puede en el futuro añadir mas componentes a un tablero sin necesidad de desarmarlo ni usar ningún tipo de herramienta especial, el componente se colocará en la riel y este quedará listo para su uso.



**Fig 68.** Riel DIN  
[http://www.promelsa.com.pe/fotos/Fotos\\_Catalogo/16724024.jpg](http://www.promelsa.com.pe/fotos/Fotos_Catalogo/16724024.jpg)

## Cable y terminales U



**Fig 69.** Terminal U  
[http://img2.mlstatic.com/12-terminales-en-u-14-rojo-para-22-16-awg\\_MLM-O-47226897\\_1530.jpg](http://img2.mlstatic.com/12-terminales-en-u-14-rojo-para-22-16-awg_MLM-O-47226897_1530.jpg)

El tablero utiliza cable de control AWG 18 ya que no lleva mucha corriente para los elementos de control, como son del selector a los contactores, relés y el paso de energía a la tarjeta de control. La energía se transporta con los cables antes seleccionados. Para evitar problemas en los contactos se utilizan terminales tipo U, como se muestra en la imagen, que ayudan a que exista un buen contacto eléctrico entre el cable y sus componentes.

## Tablero eléctrico

Todo tablero se lo selecciona por su tamaño, y esto depende del número de componentes a utilizar dentro de él. Los tableros tiene un doble fondo, como se ve en la imagen para poder retirarlo y sobre el montar los componentes y después volverlo a colocar, así se puede hacer cambios en los componentes sin tener que retirar todo el tablero de su posición.

El tablero seleccionado para nuestra aplicación es de tamaño 400 X 500 x 200.



**Fig 70.** Tablero Metálico  
<http://www.inselec.com.ec/beaucoup.pdf>

## Imágenes de tablero finalizado

Como se puede ver en la fotografía se encuentra en el fondo la riel DIN sosteniendo de izquierda a derecha el Bracker, contactor y relé. Se utiliza cable negro AWG 18 con terminales negros y rojos.

Aquí se muestra la conexión de la tarjeta de control mediante puerto serial, se soldaron los cables de control apantallados a un terminal macho para conectarlo al macho que está acoplado en el tablero.

El selector colocado en el borde superior conectado por cable AWG 18 y terminales U al contactor y relés.

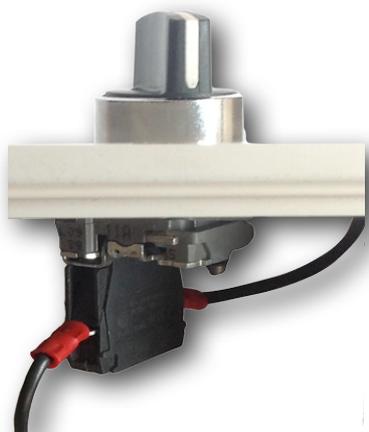
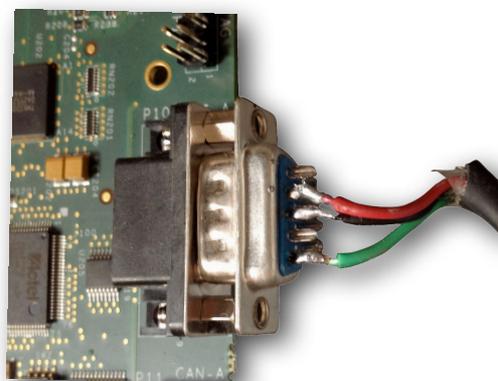
Las fuentes de poder se conectan mediante una regleta para dividir el voltaje, estas fuentes entregan 5 VDC 3 Amp.

En las imágenes a continuación se muestra el tablero terminado con prensa estopas para los cables que salen del tablero.

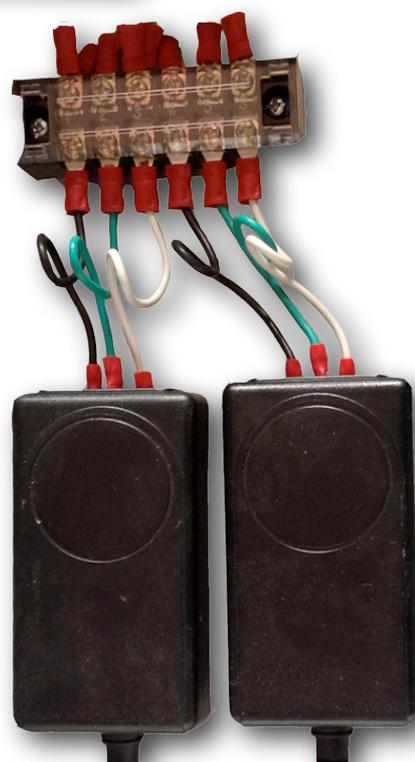


**Fig 71.** Braker, contactor y relay

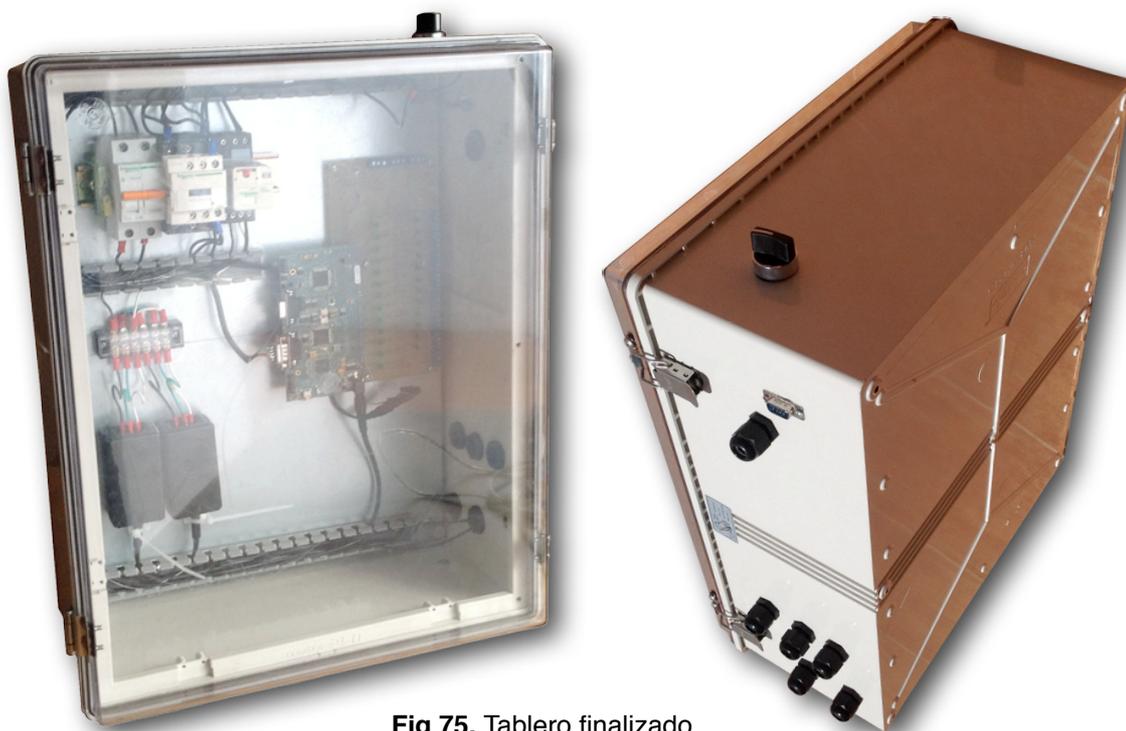
**Fig 72.** Conector puerto serial



**Fig 73.** Selector dos posiciones instalado



**Fig 74.** Fuentes para eZdsp y control



**Fig 75.** Tablero finalizado

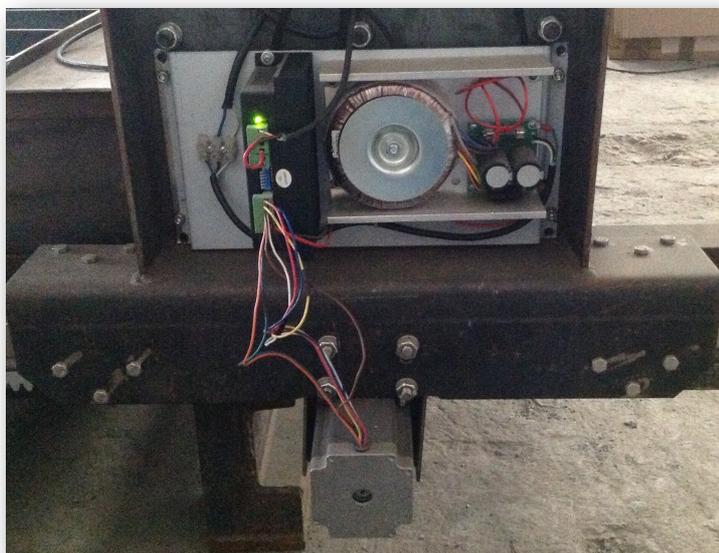
### **Armado parte eléctrica en máquina**

Como se observa en la siguiente imagen del motor X derecho, se tiene al driver, la fuente y el motor conectados. El motor esta conectado conforme lo indica el fabricante.

Para el motor X Izquierdo se hace lo mismo, con la diferencia de una regleta para distribuir la energía.

También se ve en la imagen una

placa electrónica que era para un fácil manejo de cables y desconexión de los mismos, pero esto no se utilizo que para evitar problemas en el futuro puesto que al ser una parte móvil, las partes se aflojaban y los cables se desconectaban entonces se conecto con cables directos desde el tablero de control a los drivers.



**Fig 76.** Fuente y Driver motor X2

### BIPOLAR ONLY PARALLEL

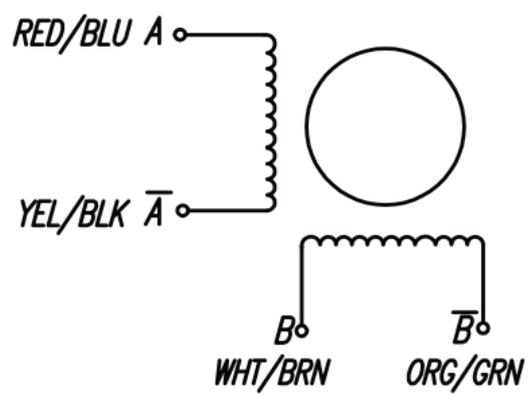


Fig 78. Conexión Driver y motor

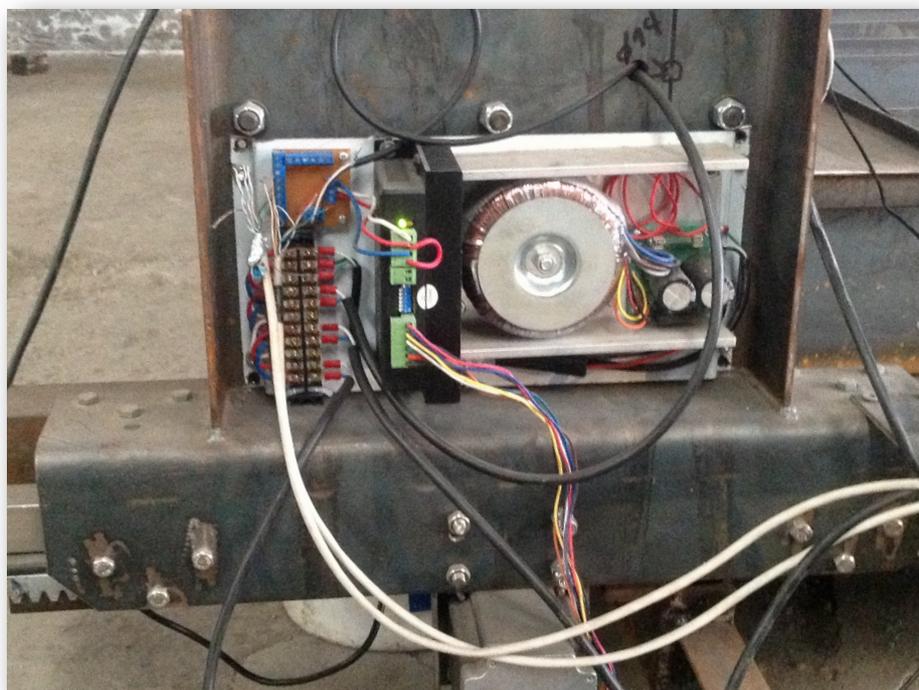


Fig 77. Fuente y Driver motor X1

## INSTALACIÓN DE SOFTWARE Y PASO DE INFORMACIÓN A LA TARJETA.

### Instalación de software

Para la instalación del software para programar y posteriormente pasar la información a la tarjeta, seguir los siguientes pasos:

- Tener Instalado Windows XP service Pack 2 (puede que funcione con otro sistema pero se lo utilizó con el OS antes mencionado)
- Insertar el CD de Spectrum Digital que viene con la tarjeta eZdspF2808.
- Instalar siguiendo el siguiente orden:
  1. C2000 Code Composer Studio V3.1
  2. eZdsp2800 Driver and Target Content
  3. CCS Falsh Plugin

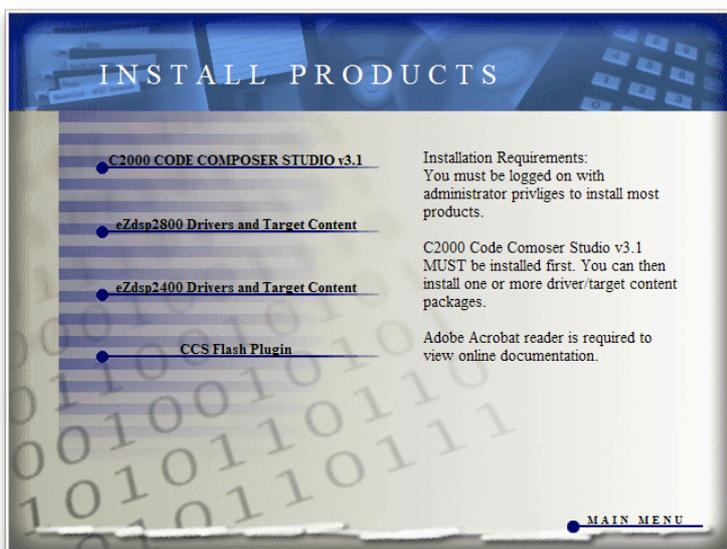
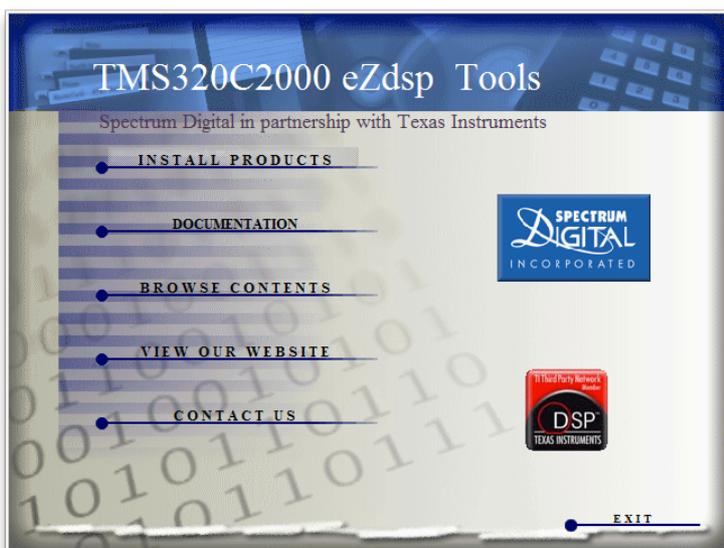


Fig 79. Ventanas instalación software

- Instalar el Code Composer Studio V4 que se lo puede obtener gratuitamente : [http://processors.wiki.ti.com/index.php/Category:Code\\_Composer\\_Studio\\_v4](http://processors.wiki.ti.com/index.php/Category:Code_Composer_Studio_v4)

- Instalar VisSim V8 que se lo puede obtener con una licencia de prueba gratis en: [http://www.vissim.com/downloads/vissim\\_software.html](http://www.vissim.com/downloads/vissim_software.html)

## Generar código para correr en tarjeta

Existen 2 formas de correr código desde VisSim, la primera consiste en ejecutar solo dentro de la memoria RAM (cuando se apaga la tarjeta se borra el mensaje), si fue generado para Flash el programa correrá al encender la tarjeta y no se borrará. Para esto se debe seguir los siguientes pasos:

- En el VisSim se abre Tools y entramos en Code Gen...

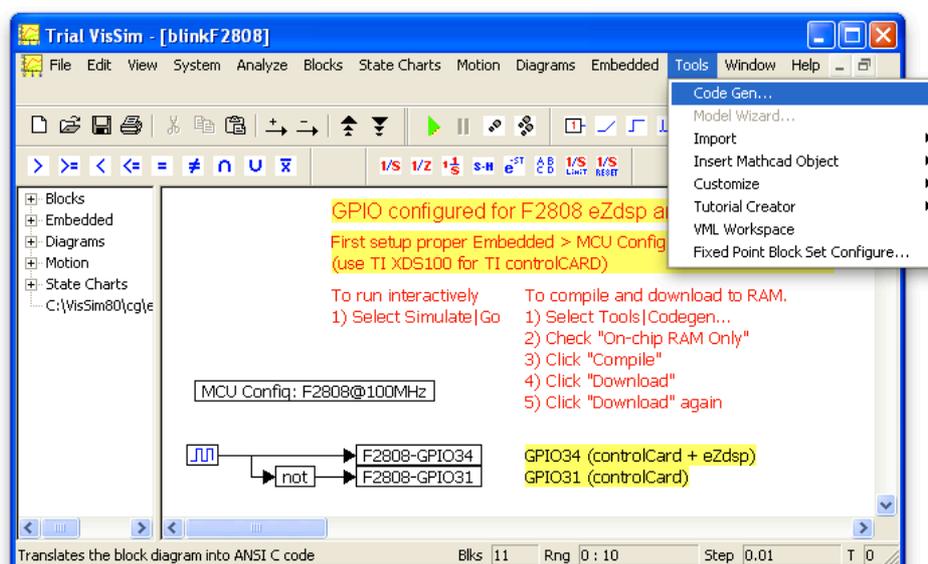


Fig 80. Generar código para tarjeta

- Si se quiere ejecutar en la memoria RAM o Flash se la selecciona. También se tiene que darle un nombre al archivo.

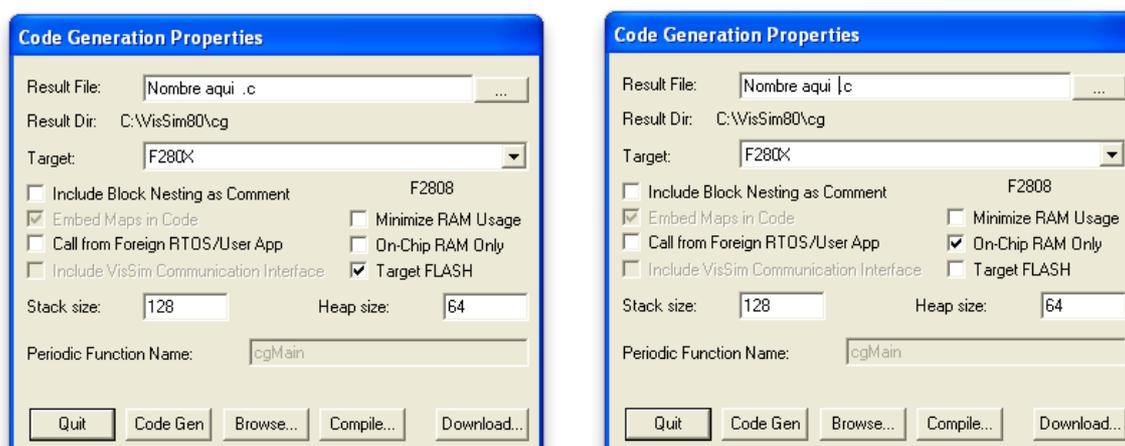


Fig 81. Generar código para tarjeta, Ram o Flash

## Probar programa en tarjeta

Siempre que se haga pruebas con nuestra tarjeta se va a utilizar el código corriendo desde la memoria RAM, esto significa que cada vez que se apaga la tarjeta se borra el programa, pero correr el programa es muy rápido. Al terminar el programa se va a utilizar la memoria Flash interna para almacenar el código y que no se borre al apagarla. Para esto debes seguir los siguientes pasos:

- Abrir Code Composer Studio V4
- En el menú "Target" se selecciona "New Target Configuration..."
- Se guarda con un nombre deseado en base a que se va a hacer con esta tarjeta.

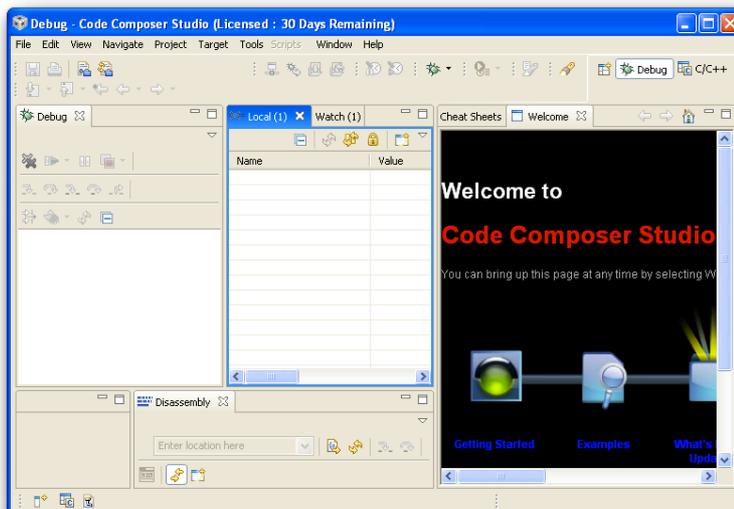


Fig 82. Crear configuración de tarjeta

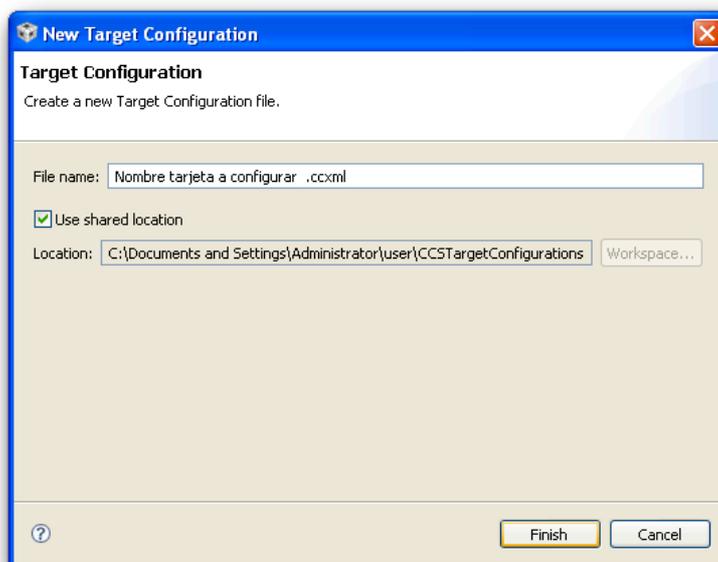
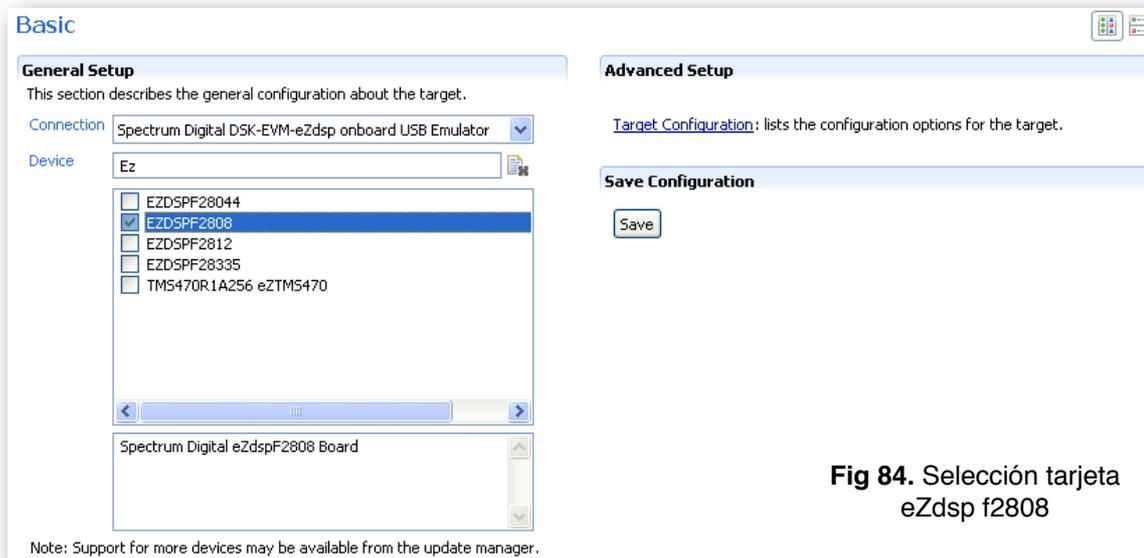


Fig 83. Final Crear configuración de tarjeta

- Se debe buscar la configuración adecuada. Si instalamos correctamente todo el software del CD que llegó con la Tarjeta se encuentra las siguientes configuraciones:

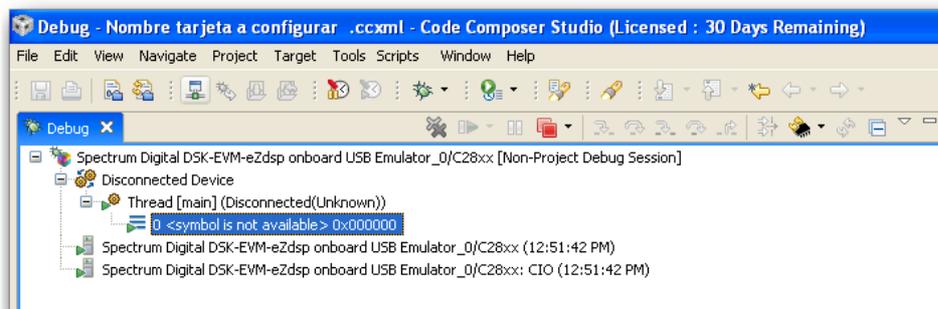


**Fig 84.** Selección tarjeta eZdsp f2808

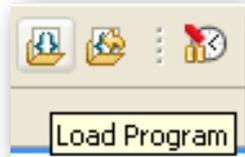
- Una vez configurada la tarjeta se conecta mediante USB, se instala el driver, si no lo encuentra ir a

C:\CCStudio\_v3.1\specdig\xds510usb

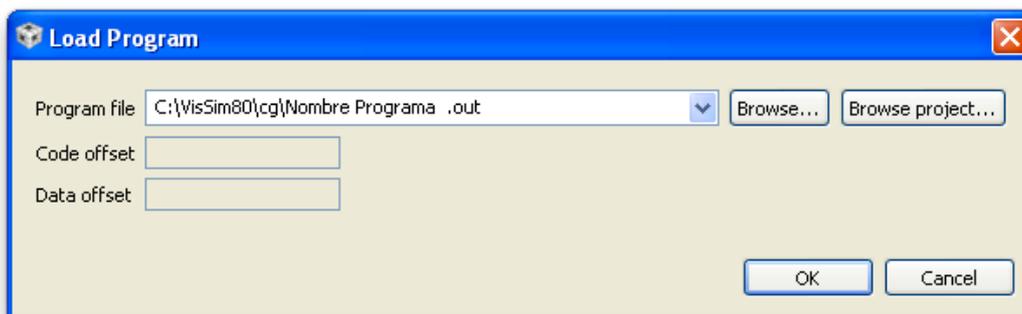
- Con el Driver instalado aplastamos el botón Debug Launch 
- Esperamos a que encuentre a la tarjeta y aplastamos el botón Connect Target 
- Aparece la ventana siguiente:



**Fig 85.** Código actual en tarjeta

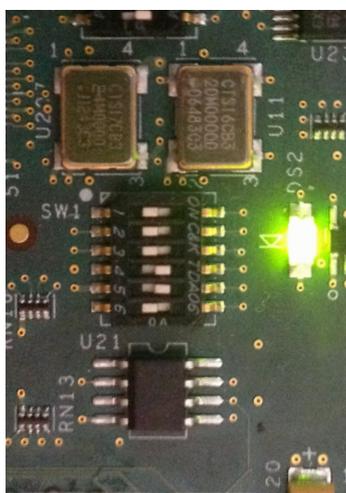
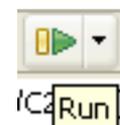
- Aquí se tiene que cargar el programa generado por VisSim aplastando el botón "Load Program" 

- En la ventana que aparece se debe seleccionar el programa .out generado por VisSim.



**Fig 86.** Cargar código a tarjeta

- Se aprieta el botón Run y con esto el programa debe correr en la tarjeta. Si el Archivo de visSim fue generado solo para RAM cuando se apaga la tarjeta se borra el mensaje. Si fue generado para Flash el programa correrá al encender la tarjeta si SW1 tiene la siguiente configuración:



Off, Off, Off, On, On, Off

**Fig 87.** Pins en SW1 para que cargue código de la memoria Flash

## RECOMENDACIONES PARTE ELECTRÓNICA

- Importar siempre repuestos de los componentes críticos para evitar tiempos muertos donde la máquina no pueda ser utilizada.
- Tener cuidado con los cables al momento de realizar pruebas, o utilizarlos en las canaletas seleccionadas para evitar cortes de los cables.
- Utilizar terminales en los cables para evitar problemas futuros de desconexiones.
- Si es posible soldar todos los cables a sus conexiones ya que es una máquina en movimiento.
- Tener mucho cuidado con la tarjeta de control en todo momento y nunca alimentar una entrada con mas de 3.2 V.
- Utilizar en todo momento elementos de seguridad eléctrica.
- Si se encuentra un error en los componentes ponerse inmediatamente en contacto con el fabricante.
- Siempre intentar utilizar componentes exclusivos para cada parte del sistema, así se detectarían mas fácilmente los errores.
- Instalar el software en Windows Xp Service Pack 2 en el siguiente orden:
  1. Cd Spectrum C2000 Code Composer Studio V3.1
  2. Cd Spectrum eZdsp2800 Driver and Target Content
  3. CCS Falsh Plugin
  4. Code Composer Studio V4
  5. VisSim V8
- Para grabar un programa en la memoria Flash utilizar Code Composer Studio V4

- Asegurarse que el SW1 de la tarjeta eZdspF2808 este en Off, Off, Off, On, On, Off, para que lea la memoria Flash al iniciar.
- Programar siempre asumiendo que existe información basura en la tarjeta y limpiar las variables antes de utilizarla.

## CAPITULO IV DISEÑO SOFTWARE DE CONTROL

### REQUERIMIENTOS 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.

## DISEÑO DE LA APLICACIÓN

### **Leer archivos DXF**

DXF (Drawing Exchange Format) es un formato de archivos en texto, esto quiere decir que se lo puede visualizar como texto en cualquier editor, esto facilita el intercambio de archivos entre programas ya que no necesita leer ningún formato en especial, simplemente con conocer las reglas y orden se puede identificar el dibujo.

Para este proyecto, se va a utilizar este tipo de archivos, que pueden ser desarrollados en AutoCad, Illustrator, etc... lo que permitirá que la mesa CNC pueda ser usada con diferentes programas.

### **Comunicación serial**

Se va a utilizar comunicación RS-232, la cual utiliza el puerto serial para intercambiar datos. Esta comunicación envía señales en voltios los cuales se interpretan como símbolos, lo que permite enviar mensajes sin protocolo lo cual permitirá desarrollar uno. Se utiliza comunicación serial como base para la comunicación y protocolo.

### **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.

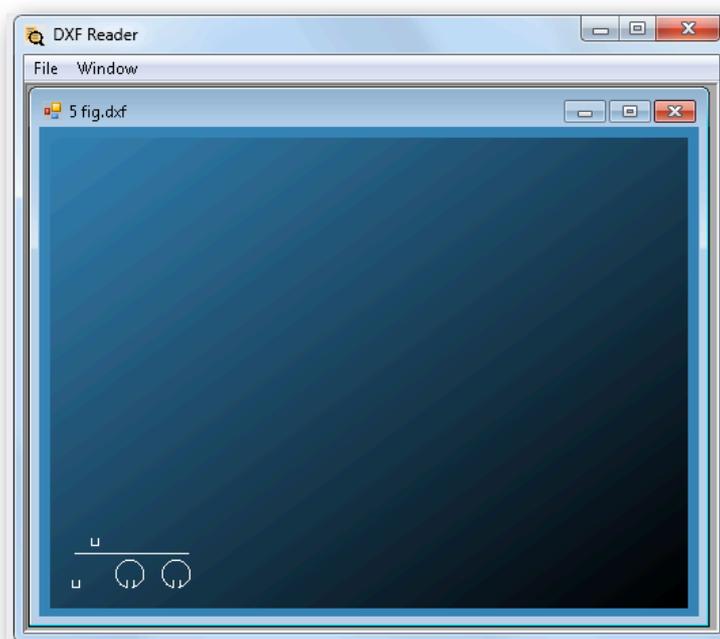
## Facilidades de uso para usuario

Ya que el usuario que generará los planos para el corte esta 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 que 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 88.** Programa DXFViewer

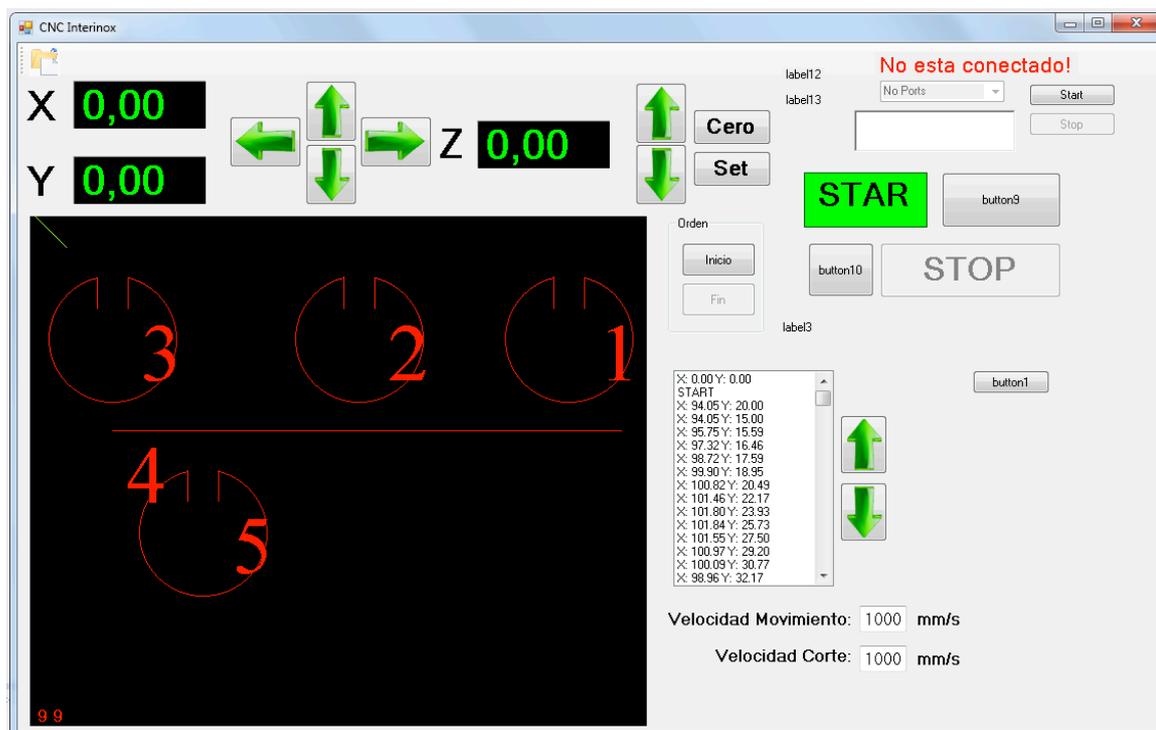


Fig 89. Programa proyecto tesis

## 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 si. A continuación se presentan tablas con los códigos y sus significados para cada una de las diferentes figuras que se presenta en el software.

Tabla 10. Formato Linea en DXF

### Linea

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

**Arco****Tabla 11.** Formato Arco en DXF

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	Angulo de fin

**Elipse****Tabla 12.** Formato Elipse en DXF

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

**Círculo****Tabla 13.** Formato Círculo en DXF

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

**Polyline****Tabla 14.** Formato Polyline en DXF

Línea 1	Línea 2	Explicación
0	POLYLINE	Inicia una figura (línea)
10	X1	Punto inicio X
20	Y1	Punto inicio Y

Línea 1	Línea 2	Explicación
42	B	Bulge
70	Open or Close	Figura abierta o cerrada
90	Num	Número de vértices

**LwPolyline** Tabla 15. Formato LwPolyline en 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

## MATEMÁTICA UTILIZADA PARA TRANSFORMAR LA INFORMACIÓN

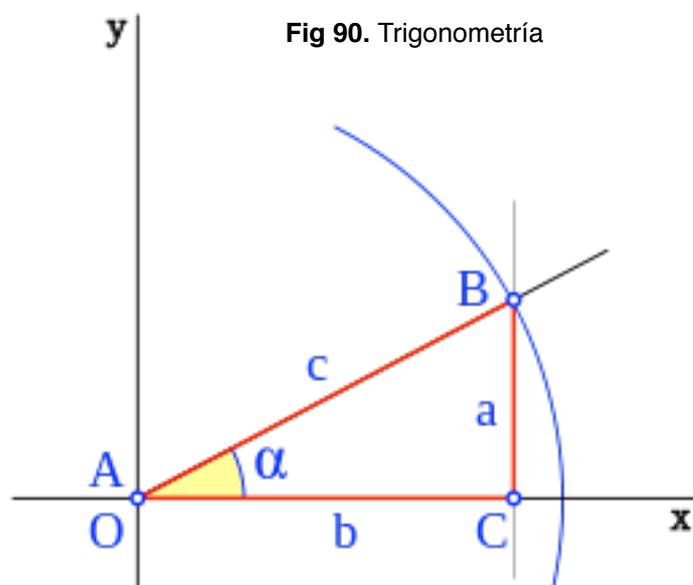
Para la generación del código que se imprime en pantalla, la información obtenida del archivo DXF, se necesita conocimiento básico de matemáticas, trigonometría y aritmética, es por esto que se presentan a continuación conceptos básicos:

### Trigonometría [9]

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

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

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



## Línea en base a dos puntos

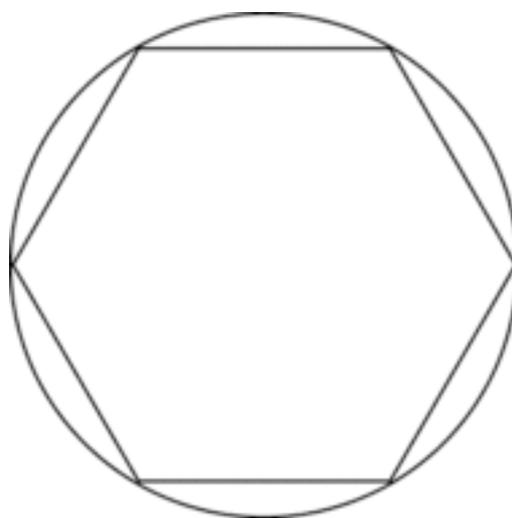
Ya que la máquina interpretara rectas es muy importante conocer la ecuación de la recta si se conoce dos puntos que la conforman.

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

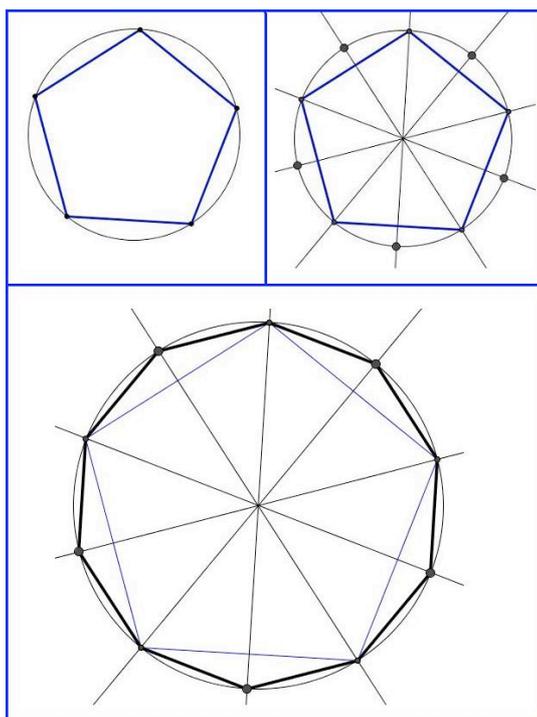
## Arco, Círculo y Elipse a líneas

Se definió que las instrucciones serán líneas es por esta razón que todas las figuras deberán transformarse a rectas. para facilitar esto se va a definir un ángulo de paso, esto es un ángulo en el cual se dividen las figuras.

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



**Fig 91.** Hexágono en círculo



divide a la mitad se tiene una figura en líneas mas parecida al círculo original.

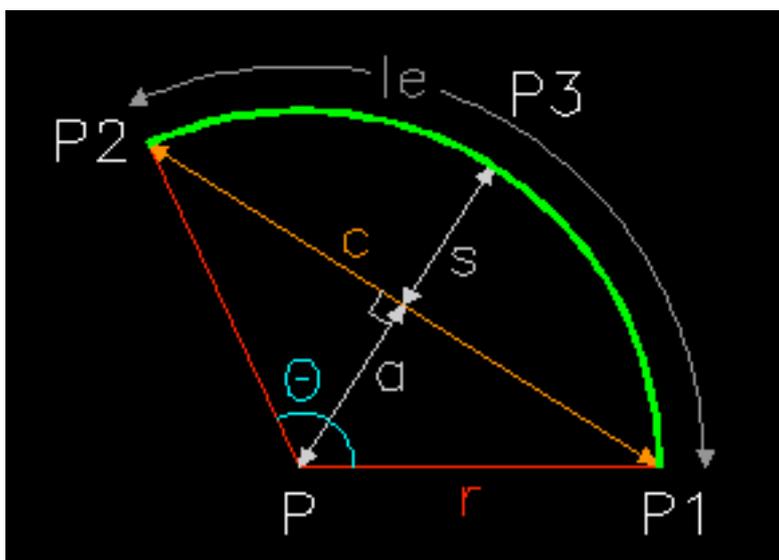
Entonces solo se necesita las ecuaciones de los arcos, círculos y elipses para poder determinar líneas que los conforman.

**Fig 92.** Polígonos en círculo

<http://gaussianos.com/images/pentagono-decagono.JPG>

## Arco definido como Bulge

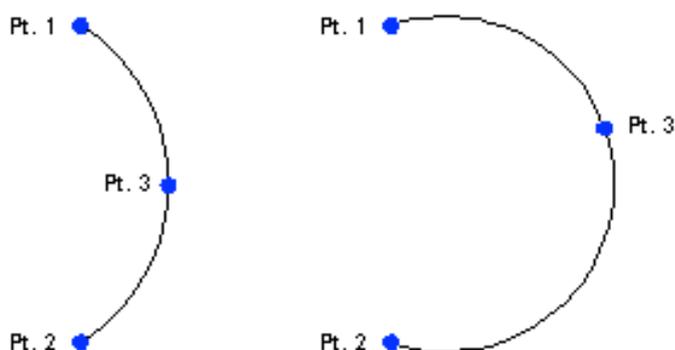
Es muy importante conocer el concepto del Bulge, ya que se vuelve algo común al momento de analizar figuras en DXF. Para generar una línea solo se necesita dos puntos, y se la puede dibujar, pero si a esos dos puntos los acompaña un bulge, esta línea se convierte en un arco. el bulge es la tangente de  $1/4$  del ángulo que se forma entre los dos puntos, es por esto que si el bulge es 1, se tiene una media circunferencia, cuando el bulge es 0 se tiene una línea recta.



**Fig 93.** 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 94.** 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.

## EXPLICACIÓN DEL CÓDIGO

El código del programa lo se lo puede encontrar en anexos digitales, a continuación se expondrán las ideas detrás de las funciones mas importantes del programa y como fueron aplicadas.

### **Código objetos que no interactua con el usuario**

En esta sección se explicara el funcionamiento de aplicaciones, lo que no observa o tiene control el usuario final.

#### Abrir archivos DXF

Como se explico 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 arraylist (matrices de datos que aumentan su tamaño al necesitar guardar mas información, no hay necesidad de declararlas con un tamaño definido.)

**Tabla 16.** ArrayList usados

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.

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 cuantos valores se debe leer del archivo datos en base a la siguiente tabla.

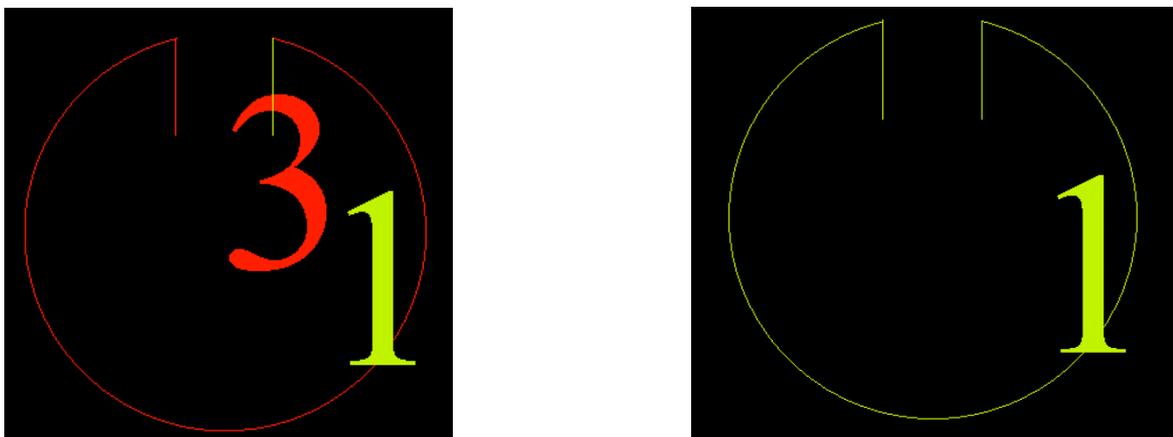
**Tabla 17.** Orden usado en ArrayList

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
Elipse	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

## Organizar Grupos

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 95.** Diferencia entre imágenes con y sin grupo

## Ordenar

Una figura al ser cortada debe poder elegirse un orden de corte, por ejemplo si se desea realizar un hueco dentro de un cuadrado, primero se debe poder cortar el círculo interno y después el cuadrado externo. Dentro de esta funcionalidad de orden también se permite eliminar figuras que no se desea cortar, dando un orden a las que se va a cortar y el fin de la operación.

## Generar instrucciones

Una vez listos con las figuras a cortar y en orden correcto, se va 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

Abierto el archivo seleccionado se debe dibujarlo en pantalla para que el usuario pueda revisar lo que desea cortar. En base a los datos se va a dibujar todos elementos en referencia a su código. En anexos se puede referir a todas las funciones con nombre dibujo. Se añade dentro de la imagen bordes de color blanco para definir los ejes y una cruz con círculo que representa la antorcha.

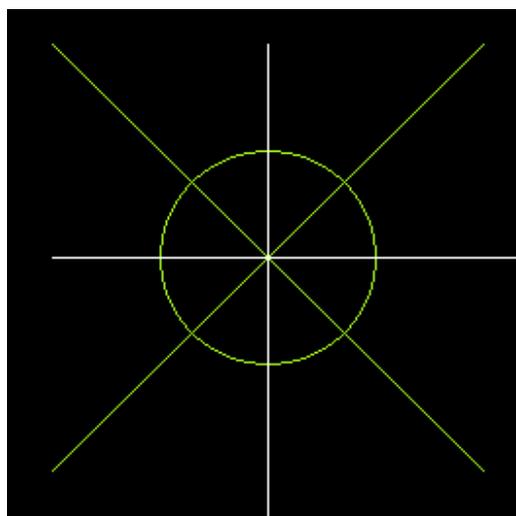


Fig 96. Representación de antorcha

### 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

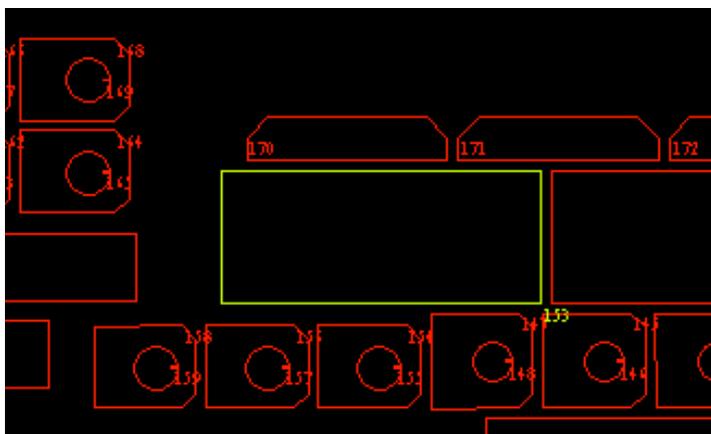
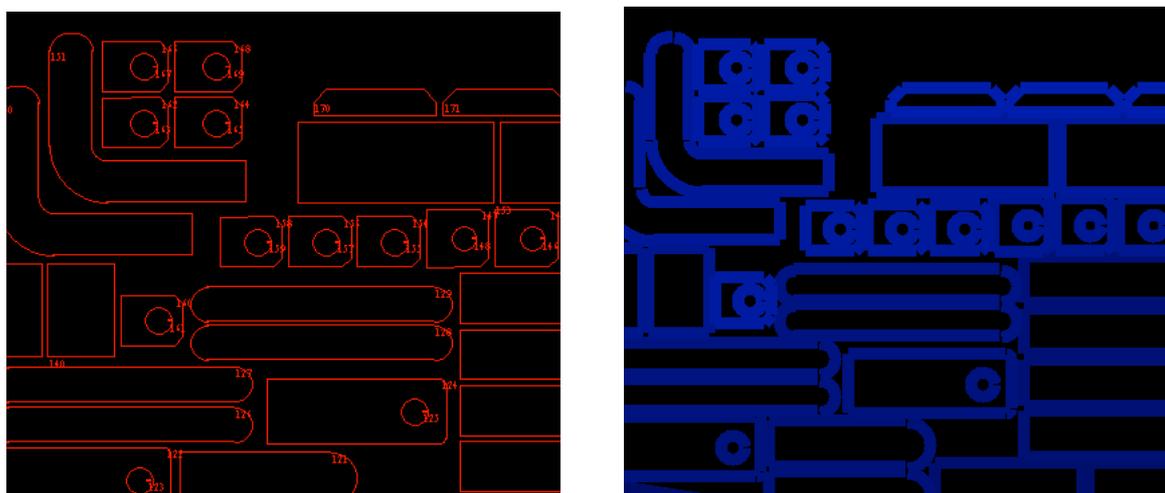


Fig 97. Figura seleccionada

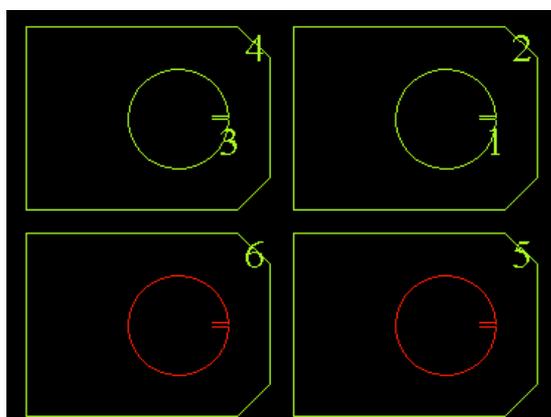
con color igual al código de la figura. La segunda imagen tiene anchos de línea mayores para que sea mas fácil seleccionarlal. Una vez desplegada la figura buscamos las coordenadas del mouse para saber que color se observa y con esto definimos el código de la figura resalada, para poder dibujarla.



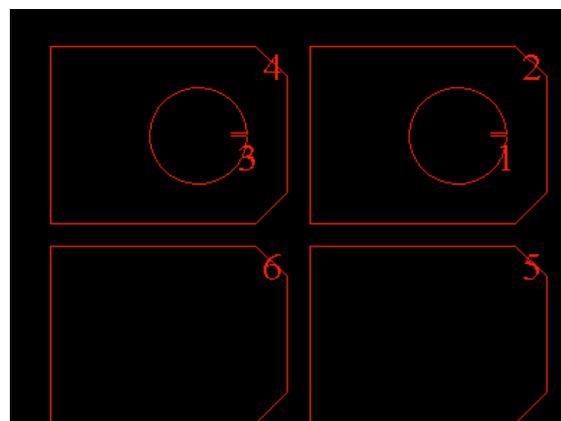
**Fig 98.** Izquierda figura vista por el usuario, derecha, figura con códigos

## Orden y eliminación de figuras

Es muy 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 99.** Orden antes de eliminar figuras



**Fig 100.** 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 mas 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 esta acostumbrado a usar en AutoCad. Se utilizo el cursor como punto centro del zoom, así a donde este apuntando se va a acercar.

## Movimientos con el Mouse



**Fig 101.** Zoom bajo

Ademas del zoom también se va a permitir otros controles con el moviendo 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 mas detalles en la imagen.

## Simulación corte

Para revisar que todas las intrusiones están correctas 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 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.

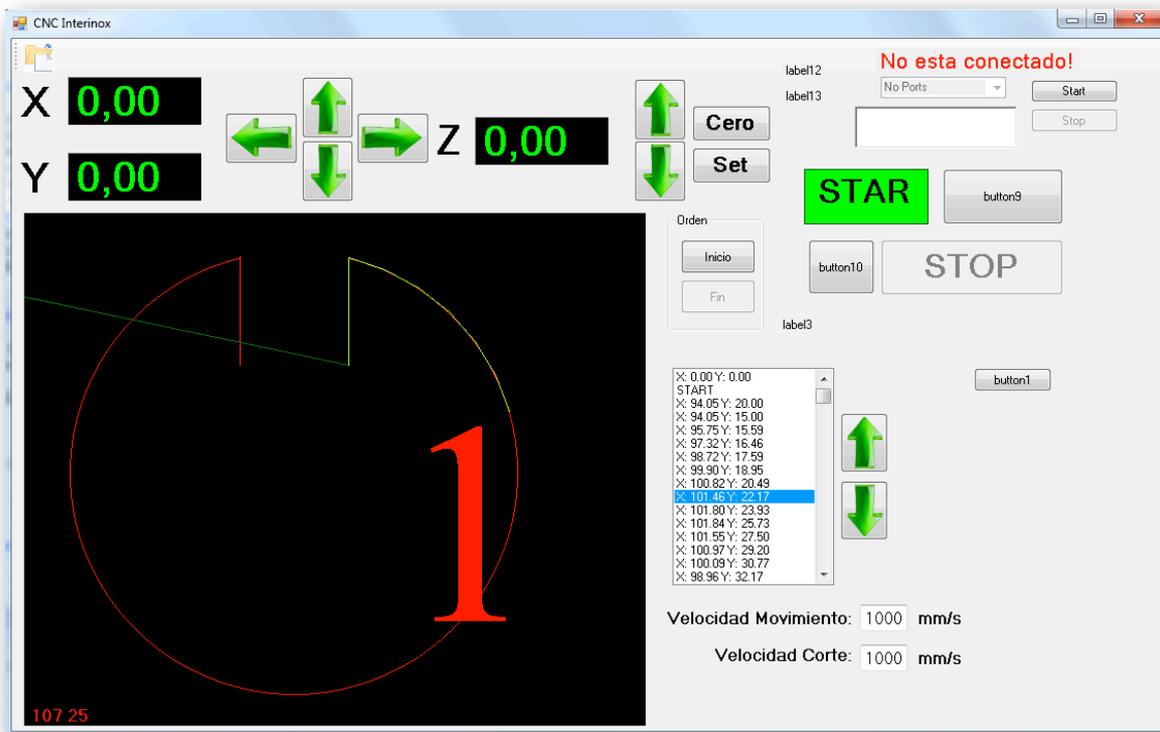


Fig 102. Simulación de corte

## CAPITULO V PROTOCOLO DE COMUNICACIÓN

### QUE ES UN PROTOCOLO?

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.

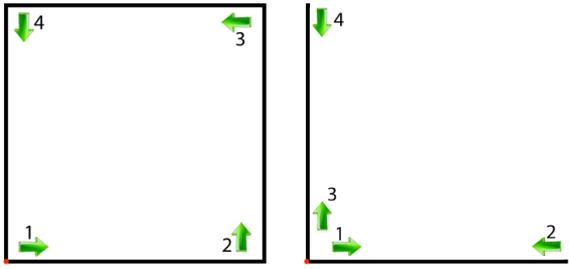
En las comunicaciones entre computadoras las reglas son mas estrictas, para garantizar que la comunicación sea correcta dependiendo el caso. No todas las comunicaciones deben ser perfectas, por ejemplo al conversar con alguien por una video llamada no importa que no se vea todo, es mas importante que la comunicación sea fluida, si por algún motivo un paquete desaparece no es importante volverlo a enviar ya que es un momento que ya paso, es mas 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 todo un cambio en la idea que se desea transmitir, en este caso se prefiere esperar antes de 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 como deben actuar en cada caso.

## DISEÑO DE PROTOCOLO

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

**Tabla 18.** 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><b>Fig 103.</b> Corte con y sin orden</p>
Pedir mensaje	El software no sabe cuanto 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.

Necesidad	Explicación
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.

## 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 primer símbolo debe ser una H
- Al finalizar el mensaje se debe enviar la suma de todos los símbolos transmitidos

La suma se se va a dar con valores Ascii de la siguiente forma:

**Tabla 19.** Ascii a número

0 = 48	= = 61	J = 75	W = 88	d = 101	q = 114
1 = 49	> = 62	K = 76	X = 89	e = 102	r = 115
2 = 50	? = 63	L = 77	Y = 90	f = 103	s = 116
3 = 51	@ = 64	M = 78	Z = 91	g = 104	t = 117
4 = 52	A = 65	N = 79	[ = 92	h = 105	u = 118
5 = 53	B = 66	O = 80	\ = 93	i = 106	v = 119
6 = 54	C = 67	P = 81	] = 94	j = 107	w = 120
7 = 55	D = 69	Q = 82	^ = 95	k = 108	x = 121
8 = 56	E = 70	R = 83	_ = 96	l = 109	y = 122
9 = 57	F = 71	S = 84	` = 97	m = 110	z = 123
: = 58	G = 72	T = 85	a = 98	n = 111	{ = 124
; = 59	H = 73	U = 86	b = 99	o = 112	= 125
< = 60	I = 74	V = 87	c = 100	p = 113	} = 126

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

Como se puede observar, se tiene  $126 - 48 = 78$  diferentes símbolos. Esto quiere decir que al enviar el símbolo ' } ' la suma total de los símbolos anteriores es 78. Como se puede observar, esto va a limitar mucho el mensaje, es por esto que se va a enviar dos símbolos al final de la suma, entonces el número máximo será

$$}} = 78\ 78$$

Este caso sería correcto, pero no se podría generar el número 7899, ya que no existe un símbolo que sea el 99, es por esta razón que el símbolo final va a ser limitado del 0 al 9, mientras que el anterior se trabajará con la tabla dando la suma máxima:

$$}9 = 789$$

Con esta limitación impuesta se puede determinar el número máximo de caracteres que se puede sumar. Si se tiene un mensaje conformado solo por el símbolo } se podría tener máximo:

$$\frac{789}{78} = 10,115$$

entonces un mensaje de 10 símbolos "}}}}}}}}}}}" será  $78 \times 10 = 780$

Este es el primer límite que se encuentra para nuestro protocolo, se puede enviar máximo 12 símbolos.

### **Mensaje Ordenado**

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 y con esto se consigue enviar un mensaje en orden correcto.

Entonces una secuencia correcta de mensajes hasta el momento será:

H 5264851 1 57 (  $H(73-48)+5+2+6+4+8+5+1+1 = 57$  )  
 H 5413210 2 43 (  $H(73-48)+5+4+1+3+2+1+0+2 = 43$  )  
 H 9653295 3 67 (  $H(73-48)+9+6+5+3+2+9+5+3 = 67$  )  
 H 82A37z5 4 >6 (  $H(73-48)+8+2+A(65-48)+3+7+z(123-48)+5+4 = 146$  )  
 (  $14+48 = 62 = '>'$  )

### 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 no cumple o no llega nada, pasado un tiempo la tarjeta pedirá el mismo mensaje nuevamente.

El formato será el siguiente "M" + # mensaje a pedir. En este caso no existe una comprobación ya que si el mensaje de pedido llega incorrecto se anulara o enviara un mensaje erróneo que la tarjeta anulará. Una comunicación correcta tendrá la siguiente forma:

T	M1
Sf	H5264851157
T	M2
Sf	H5264851157
T	M2
Sf	H5413210243
T	M3
Sf	H9653295367
T	M3
Sf	H9653295367
T	M4
Sf	H82A37z54>6
T	M5
:	:

Como se puede observar en el ejemplo, existió un error cuando el software entendió el pedido M2, por eso reenvía la información, pero la tarjeta debe descartarlo y pedir nuevamente. Otro error pero esta vez de parte de la tarjeta que no entendió el mensaje M3 pero lo pide nuevamente y el software lo envía.

## Tiempo de llegada

Para garantizar que la información de la instrucción a continuación está lista, se debe tener un buffer de instrucciones, 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 orden, este código será la letra 'F'. Un mensaje de ejecución rápida será por ejemplo un cambio en la velocidad de corte. Un ejemplo es:

H V452442 F :8 ( (73-48)+(87-48)+4+5+2+4+4+2+(71-48) = 108 = :8)

## Aviso de recepción

Todos los mensajes de pedido no necesitan confirmación pero los mensajes rápidos si lo necesitan, para enviarlo nuevamente si no llegó, por esta razón la tarjeta enviara una 'F' cuando llego 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.

Sf	HV452442F:8
:	:
Sf	HV452442F:8
T	F

## 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. La tarjeta continuara enviando z cada cierto tiempo avisando que esta 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 encuentra  
 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
- ...
- T pide el primer mensaje 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 primer mensaje enviando "M1", y repite cada 0.1s
- Sf envía mensaje solicitado asumiendo que lo recibió correctamente sin pedido de confirmación
- ...

- Para finalizar Sf envía RRRRRRRRRRRR
- T se mantiene enviando “z” para mostrar un estado de espera.

### Información en mensaje normal

Los mensajes normales que la tarjeta y el software intercambian tiene un formato específico, este mensaje debe cumplir con cierta información:

- Dirección de movimiento en X
- Dirección de movimiento en Y
- Pasos en X
- Pasos en Y
- Información si esta cortando o en movimiento.

Para cumplir todas estas necesidades se va a mantener el siguiente estructura para el mensaje:

#XXXYYY?

# = Código de dirección X, Y y antorcha prendida  
 X = Cantidad de pasos en X  
 Y = Cantidad de pasos en Y  
 ? = No tienen importancia, puede ser cualquier valor.

Para el código de dirección se va a utilizar la tabla 20:

**Tabla 20.** Dirección y corte antorcha

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

Código #	Instrucción
F	Traslado X positivo, Y negativo, antorcha prendida
G	Traslado X negativo, Y positivo, antorcha prendida
H	Traslado X negativo, Y negativo, antorcha prendida

## Tiempo de llegada de la información

Ya que en el proyecto es muy importante que las instrucciones sean continuas, ya que se necesita mantener una velocidad constante lo que se debe hacer es mantener instrucciones en un buffer, el cual se encargará de mantener instrucciones almacenadas y así nunca esperar a una instrucción. En la fase de experimentación se analizará como mantener el buffer y la cantidad de instrucciones que deben ser almacenadas.

## LIMITES Y RESTRICCIONES

Se puede observar que la cantidad de pasos para X y Y, así como para las velocidades de corte están limitadas a máximo 3 caracteres, esto genera un límite máximo de pasos por instrucción.

Se tiene 78 diferentes símbolos como se puede observar en la tabla Ascii, por lo tanto se podría tener teóricamente

$$78 \cdot 78 \cdot 78$$

Para simplificar y tener los números en una base decimal se va a restringir los dos últimos símbolos solo a números del 0 - 9 y el tercer símbolo con todos los ascii, esto queda como máximo:

$$78 \cdot 9 \cdot 9 = 7899$$

Esto quiere decir, que máximo se puede enviar instrucciones que contengan 7899 pasos, así como la velocidad máxima no podrá superar los 7899 mm/s. Si se tiene instrucciones mas grandes, el software deberá dividir las en varias instrucciones pequeñas.

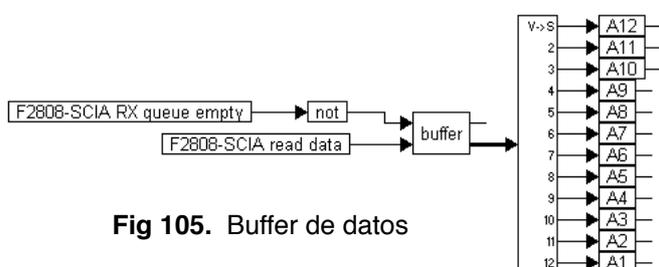
No existe ningún limite para la cantidad de instrucciones.

## PROGRAMACIÓN PROTOCOLO EN TARJETA F2808

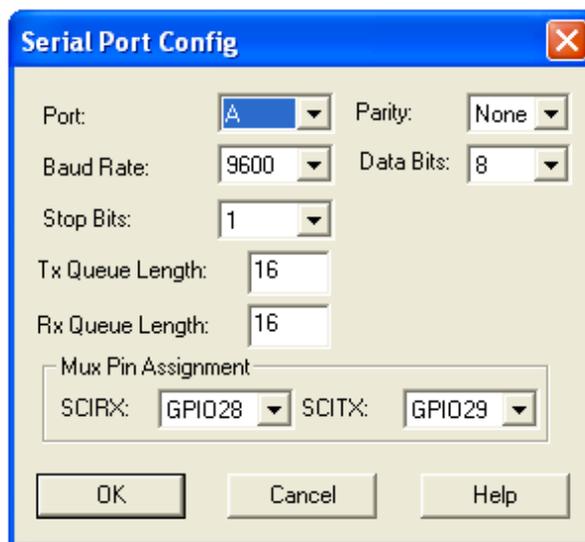
### Funciones utilizadas

#### Recibir mensaje

Estos bloques permiten la comunicación a través del puerto serial, cuando la tarjeta recibe una señal, la lee y guarda en un buffer temporal donde se almacenan 12 símbolos, después del buffer se tiene una conversión de vector a escalar, es muy importante conocer que cada símbolo que llega a la tarjeta desplaza el anterior, es por esto que la salida 12 del bloque V -> S será el primer símbolo



**Fig 105.** Buffer de datos



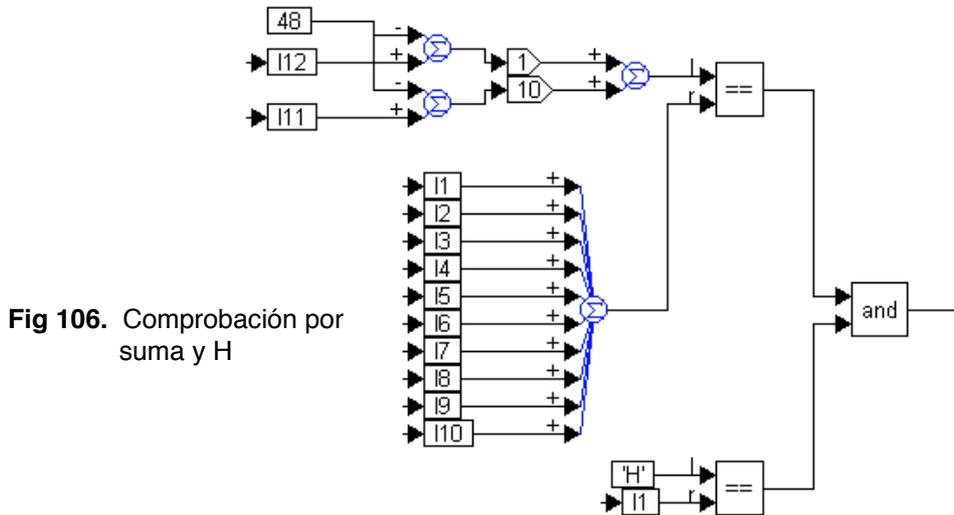
**Fig 104.** Configuración puerto serial

enviado, en nuestro protocolo la 'H', solo sin todos los otros símbolos llegaron, si un símbolo no llego la 'H' estará en la posición 11 y el 12 será un símbolo de una transición anterior. La configuración del puerto y los Baud se los realiza en la venta de configuración del Puerto Serial. Los cuadros con códigos A1 - A12 son variables, se guarda la información en estas posiciones.

#### Comprobación por suma y posición 'H'

El bloque siguiente comprueba que los dos últimos caracteres sean la suma de todos los anteriores. Se debe restar el valor 48, para obtener el valor en int de su representación en Ascii. Luego se multiplica dependiendo de su posición y se

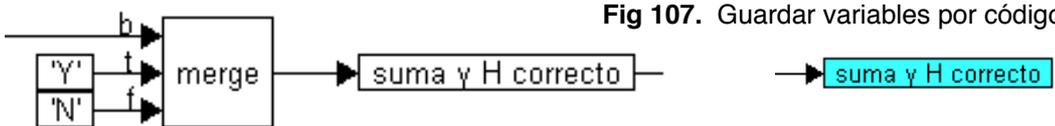
suma, para al final comparar los valores, solo si las sumas son correctas y la 'H' esta en su posición correcta se obtiene una aprobación de este bloque.



**Fig 106.** Comprobación por suma y H

### Guardar información en variables

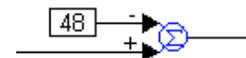
Para avisar que se tiene un mensaje correcto se va a utilizar el siguiente bloque, el cual almacena en una variable el valor 'Y' si se tiene un mensaje correcto y una 'N' si es falso, de este punto en adelante cuando se vea una variable en color celeste se tiene el bloque siguiente:



**Fig 107.** Guardar variables por código

### Transformar de Ascii a Int

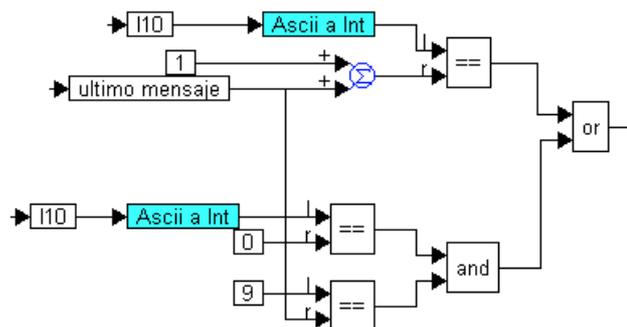
Como se explico anteriormente, se necesita disminuir el valor en Ascii del símbolo que se recibe para poder tenerlo en Int.



**Fig 108.** Ascii a int

### Orden del mensaje

Para comprobar el orden correcto se va a utilizar el siguiente bloque, el cual en base a conocer el último mensaje avisa si el siguiente es



**Fig 109.** Orden de mensaje

correcto, se comprueba que la suma del último mensaje mas 1 se igual al nuevo mensaje. También se comprueba que si el último mensaje es 9 el siguiente debe ser 0.

## Mensaje correcto

La unión de los 2 bloques anteriores deja con un bloque que avisa cuando el mensaje y el orden es correcto. El bloque S&H guarda y mantiene el último mensaje correcto para que las variables M1 - M10 sean las instrucciones.

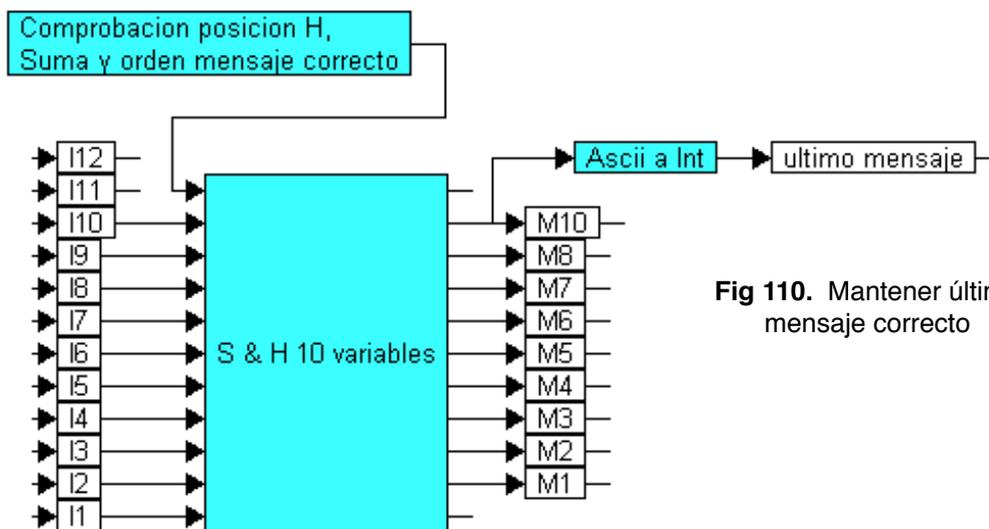


Fig 110. Mantener último mensaje correcto

## Comprobar orden correcto

Este bloque se encarga de comprobar si las entradas son consecutivas es muy parecido a la comprobación del orden del mensaje, pero en este caso se utiliza para comprobar que el mensaje anterior y el nuevo sean continuos y así pedir el mensaje siguiente.

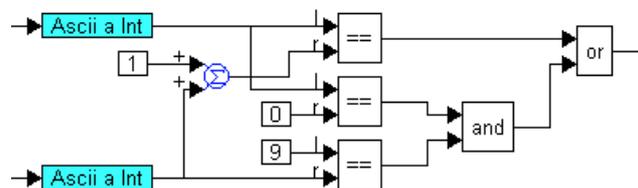


Fig 111. Comprobar mensaje correcto

### Pedir mensaje siguiente

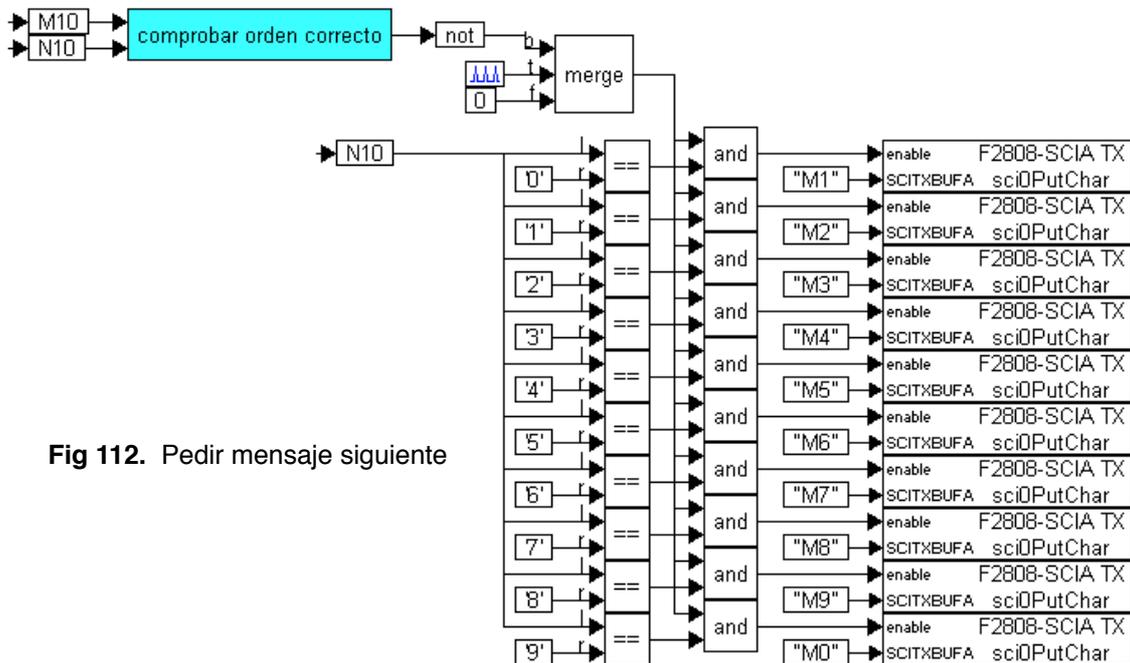


Fig 112. Pedir mensaje siguiente

Cuando al mensaje ya aprobado (N10) no le sigue el (M10), quiere decir que hubo un error o que se mantiene el mismo mensaje, es por esto que solo cuando el mensaje a continuación no sea el correcto se pide el mensaje. El bloque merge permite repetir el pedido del mensaje en base a un reloj. Se pide el mensaje a continuación del que ya se recibió.

### Buffer

Se necesita guardar una instrucción mientras se ejecuta la anterior, es por esta razón que se mantiene un buffer, que no es mas que una instrucción lista para ser ejecutada. Cuando N10 cambia (por esto el bloque 1/Z), y continua con la

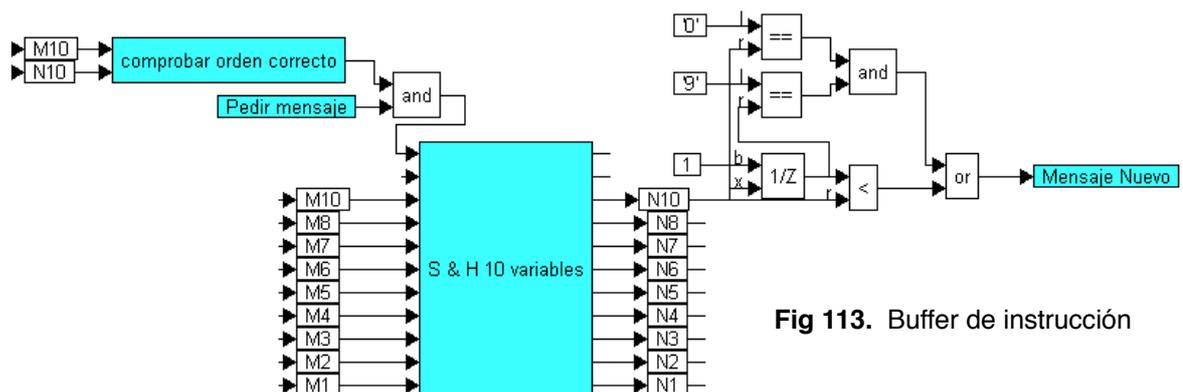


Fig 113. Buffer de instrucción

siguiente instrucción, se avisa con la variable “Mensaje nuevo” para que la tarjeta ejecute la instrucción, una vez terminada esa instrucción, avisa mediante la variable “Pedir mensaje”, y solo si el mensaje en M1 - M10 es el correcto se continua con la siguiente instrucción.

## Mensaje rápido

El mensaje rápido tiene una codificación especial, la cual se puede observar a continuación. Cuando la comprobación de la suma y la ‘H’ es correcta y si el código de mensaje es ‘F’ permite guardar en las variables F el mensaje rápido, y responde con una ‘F’ para que el Software conozca que el mensaje fue recibido. Luego se

hace una nueva comprobación para conocer si el valor a actualizar es para velocidad de corte ‘C’ o velocidad de movimiento ‘M’. Una vez comprobado esto se mantiene en la variable correspondiente el valor.

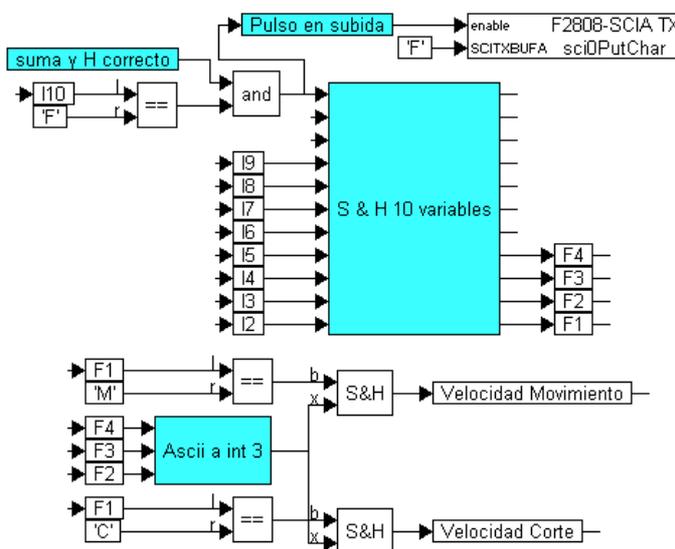


Fig 114. Mensaje rápido

## Mensaje normal (antorcha y dirección de motores)

Para el mensaje normal comprobamos que el primer símbolo indique la dirección de los motores y si se está cortando o no.

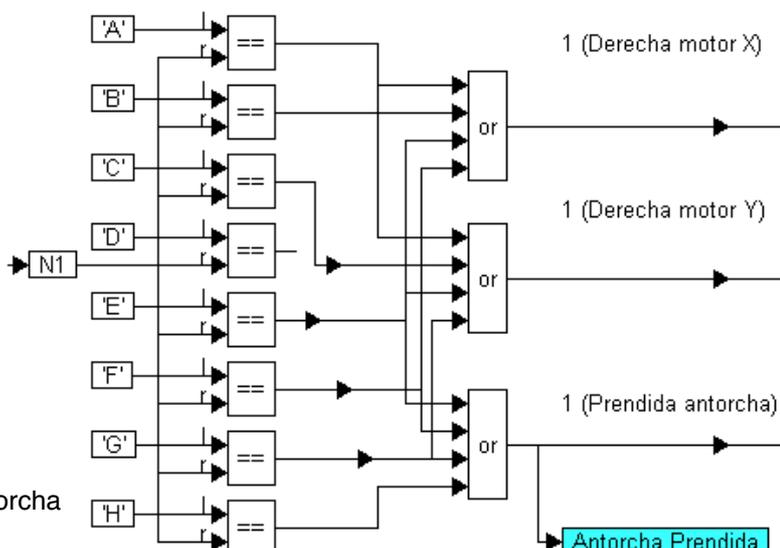


Fig 115. Dirección y corte antorcha

### Mensaje normal (guardar valores de pasos y velocidades)

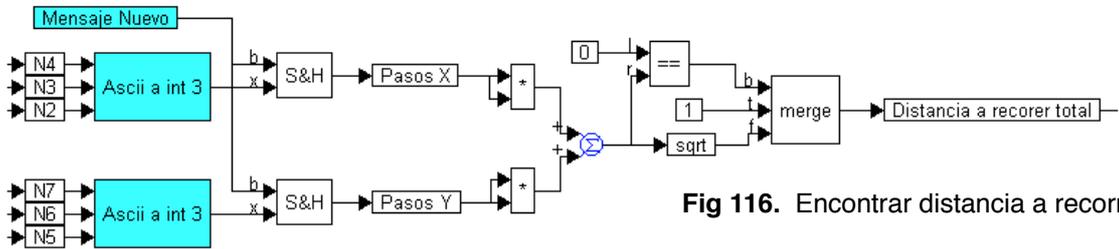


Fig 116. Encontrar distancia a recorrer

Cuando se tiene un mensaje nuevo, se actualizan los valores de pasos en X y Y, también se actualiza mediante la variable “Antorcha Prendida” cual velocidad se va a utilizar para determinar las frecuencias de X y Y.

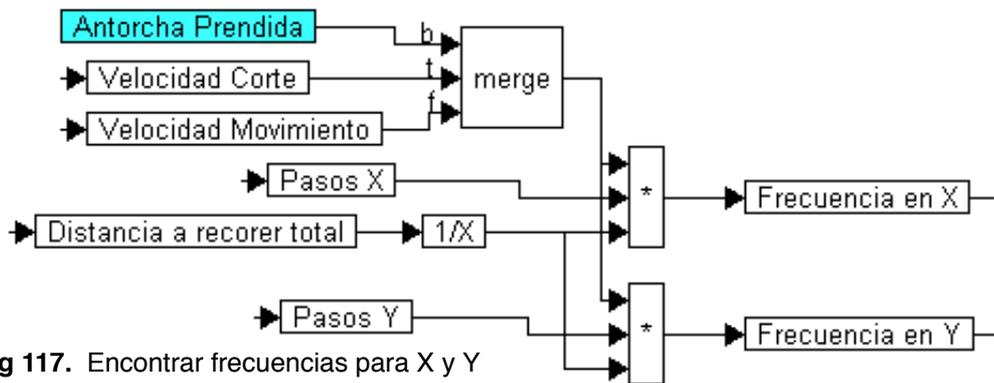


Fig 117. Encontrar frecuencias para X y Y

### Cero

Existen dos problemas importantes donde es necesario empezar desde cero la tarjeta, el primero es el momento de encenderla, para evitar basura por error o de alguna instrucción que quedo en memoria, y la segunda es al momento de empezar una transmisión, donde el protocolo indica que se debe empezar eliminado cualquier dato. En Anexos podrán ver todas las funciones anteriores modificadas para que funcionen con el proceso de inicio y cero, a continuación están los bloques que inician el

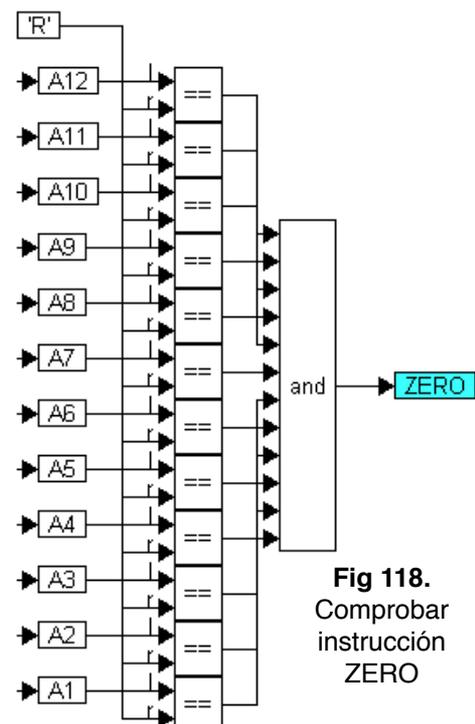


Fig 118. Comprobar instrucción ZERO

encaramiento y envían la señal 'z' anunciando que se completo.

Se comienza comprobando que la instrucción enviada sea todo 'R', cuando esto es correcto, la variable "ZERO" se activa.

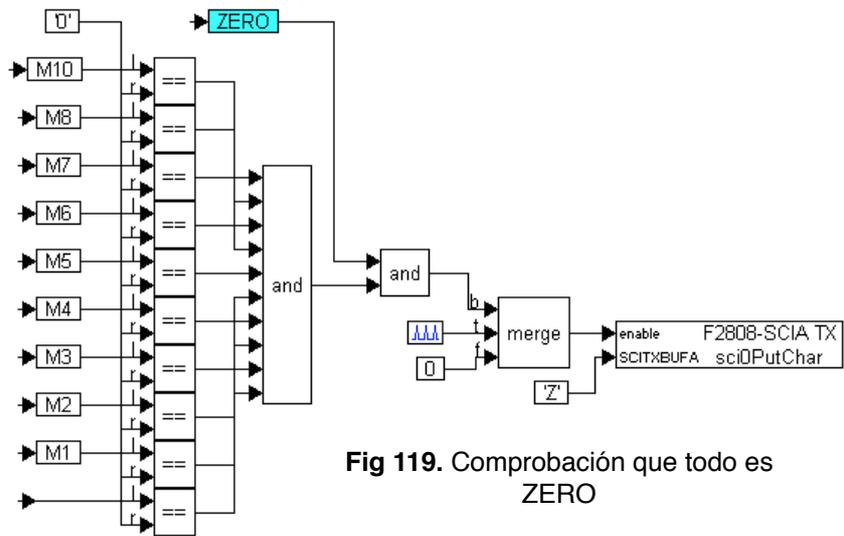


Fig 119. Comprobación que todo es ZERO

El bloque empezar en cero elimina todos los valores y pasa todo a cero, todos los bloques S&H también se activan para borrar los valores. Al final se comprueba que los valores sean cero y se envía una respuesta.

### Programa final

El programa final se resume en estos bloques. La salida GPIO14 es un led en la tarjeta que indica que esta funcionando.

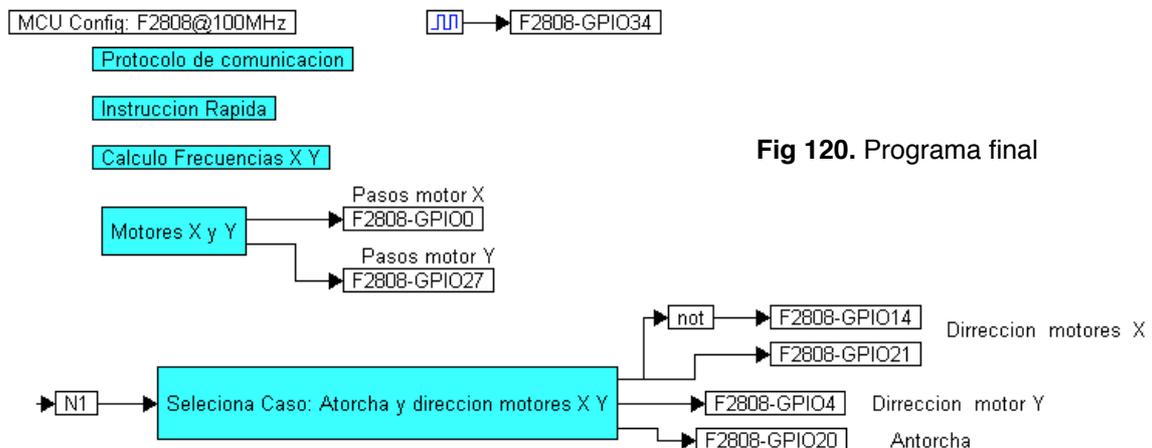


Fig 120. Programa final

## PROGRAMACIÓN PROTOCOLO EN SOFTWARE

EL protocolo requiere que se mantenga la misma comunicación en los dos extremos. Para enviar un mensaje se necesita tener el header (H) los 8 caracteres de comprobación, el orden de mensaje y la suma, para comprobar el mensaje correcto.

Se necesita una forma de empezar la transmisión, empezando por resetear la tarjeta, luego de esto se asegura que las velocidades de corte y movimiento lleguen a la tarjeta, para luego comenzar a enviar los mensajes con instrucciones. Para esto se tiene la función protocolo.

### Funciones utilizadas

#### Protocolo

```
private void Protocolo()
{
    velMove =Convert.ToInt32(Convert.ToDouble(txtvelocidadMov.Text) * 38.12);
    velCorte = Convert.ToInt32(Convert.ToDouble(txtVelosidadCort.Text) *
38.12);
    estazero = false;
    estavelcorte = false;
    estavelmov = false;
    instruccionActual = 1;
    mensajeActual = 1;
    start = true;
    tprotocolo.Interval = 600;
    tprotocolo.Enabled = true;
    serialPort.Write("RRRRRRRRRRR");
}
```

Esta función comienza obteniendo los valores de velocidad a transmitir para corte y movimiento, luego se ponen en falso unas banderas globales las cuales indican si ya se puso en cero la tarjeta y si se enviaron las velocidades. Se inicia un reloj el cual se va a encargar de enviar las velocidades cuando los pasos estén cumplidos.

#### tprotocolo

```
private void tprotocolo_Tick(object sender, EventArgs e)
{
    string inst = "";
    if (estazero == false)
    {
        serialPort.Write("RRRRRRRRRRR");
    }
    else
    {
```



```

        {
            palnum = myChar[0].ToString() + myChar[1].ToString() +
myChar[2].ToString() + myChar[3].ToString();
        }
        if (myChar.Length == 5)
        {
            char c = (char)((Convert.ToInt32(myChar[0]) - '0') * 10 +
(Convert.ToInt32(myChar[1]) - '0') + 48);
            palnum = c + myChar[2].ToString() + myChar[3].ToString() +
myChar[4].ToString();
        }

        return palnum;
    }
}

```

Se puede observar el uso de una función llamada numachar4, la cual se encarga de poner el formato al número a enviar, en este caso se va a enviar un número máximo 78999 y esta función lo transforma para que solo tenga 4 caracteres, si el número es menos de 4 caracteres añade ceros adelante para que siempre sea de 4 caracteres.

## sumaparamensaje

```

private string sumaparamensaje(string mens)
{
    char[] instchar = mens.ToCharArray();
    int t = 0;
    for (int i = 0; i < instchar.Length; i++)
    {
        t = t + Convert.ToInt32(instchar[i]);
    }
    return numachar2(t);
}

```

El mensaje que se va a enviar es separado en caracteres individuales para luego sumarlos , la suma se la divide en 2 caracteres con el número máximo de 789. Se utiliza una función igual a numachar4, pero en este caso solo devuelve 2 caracteres.

## displaytext

```

private void DisplayText(object sender, EventArgs e)
{
    int l;
    int s;
    int m;
    string t = RxString;
    string inst;
    if (t.Length > 1)
        t = t.ToCharArray()[1].ToString();
    richTextBox1.Text = richTextBox1.Text + RxString;
    richTextBox1.ScrollToCaret();

    if (start == true)
    {
        if (t == "Z")
        {

```

```

        estazero = true;
        tprotocolo.Interval = 600;
    }
    if (t == "F")
    {
        if (estavelcorte == false)
        {
            estavelcorte = true;
        }
        else
        {
            if (estavelmov == false)
            {
                estavelmov = true;
            }
        }
    }
}
try
{
    l = Convert.ToInt16(t);
    s = Convert.ToInt16("9");
    m = Convert.ToInt16("0");
}
catch
{
    return;
}
if (l <= s && l >= m && estazero == true && estavelmov == true &&
estavelcorte == true && t.Length == 1)
    if (instruccionActual < Instrucciones.Count)
    {
        if (mensajeActual == Convert.ToInt16(t))
        {
            inst = "H" + Instrucciones[instruccionActual] +
mensajeActual.ToString();
            serialPort.Write(inst + sumaparamensaje(inst));
            label13.Text = instruccionActual.ToString();
            instruccionActual++;
            mensajeActual++;
        }
        else
        {
            if ((mensajeActual - 1) < 0)
                mensajeActual = 10;
            inst = "H" + Instrucciones[instruccionActual - 1] +
(mensajeActual - 1).ToString();
            serialPort.Write(inst + sumaparamensaje(inst));
        }
        if (mensajeActual == 10)
            mensajeActual = 0;
    }
    else
    {
        btstop.PerformClick();
        tprotocolo.Enabled = false;
    }
}
ultresp = RxString.ToCharArray()[0];
}

```

Esta función recibe toda la comunicación que llega desde la tarjeta y siempre que llega un símbolo se ejecuta. Se ve al inicio que espera las respuestas de "Z" para estar seguro de que se inicio la tarjeta desde cero, luego "F" para comprobar que los dos mensajes rápidos con la velocidad de movimiento y corte fueron recibidos. Solo al cumplir las primeras tres condiciones se pasa a enviar los mensajes que se encuentran en un lista, en base al mensaje que pide la tarjeta seleccionamos el

correcto, se le da el formato necesario y lo se envía. Si se desea actualizar la velocidad se debe poner en falso la bandera, antes de enviar el siguiente mensaje se enviara la nueva velocidad. Una vez terminada la transmisión de todo el mensaje se apaga el protocolo.

## RECOMENDACIONES PARA DESARROLLO DE PROTOCOLO DE COMUNICACIÓN

- Definir claramente los pasos para el protocolo.
- Definir una longitud limitada de mensaje.
- Tener muy en cuenta las limitaciones físicas de los componentes, velocidades, etc...
- Realizar pruebas por separado de cada uno de los pasos para encontrar fácilmente los errores.

## CAPITULO VI 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.

Para esto primero se va a realizar movimientos separados en los ejes, y movimientos combinados, se utilizará un marcador, se dibujará sobre una superficie líneas, arcos y polígonos. Mediante reglas y calibradores se comprobarán los tamaños obtenidos y se compara con los planos originales, luego de esto se calibra el software para corregir cualquier error y se comprueban nuevamente los dibujos hasta que estén dentro del rango de error adecuado. 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

**Tabla 21.** 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	500	866	1,000
Real	línea	1029	30	521	887	1,029
Teórica	línea	1000	45	707	707	1,000
Real	línea	1031	45	725	733	1,031
Teórica	línea	1000	60	866	500	1,000
Real	línea	1032	60	890	522	1,032
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 22.** 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

## TIEMPO

El software antes de la prueba se calibro con una velocidad de 1000, la cual no tienen unidades ya que es un valor de referencia y en base a este valor se va a determinar la escala real con unidades.

### Datos Tiempo

Se va a realizar las siguientes pruebas:

- Línea 500 mm en diferentes ángulos.

**Tabla 23.** 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

- Línea 1000 mm en diferentes ángulos

**Tabla 24.** 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

- Círculo con diferentes Radios

**Tabla 25.** 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

## Análisis Datos Tiempo

La prueba comienza con un número 1000 calibrado en el software, en base a los datos tomados, se va a determinar la velocidad real en mm/s.

**Tabla 26.** 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

Recordando la tabla de velocidades presentada en el capítulo de diseño electrónico pero transformada a mm/s. (Datos tomados del manual Hypertherm para equipos plasma)

**Tabla 27.** 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

## Pruebas de corte

**Tabla 28.** 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

## CORRECCIONES REALIZADAS

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

$$\text{Distancia recorrida} = \text{Distancia de paso} \times \text{Número de pasos}$$

$$\text{Distancia de paso} = \text{Distancia recorrida} / \text{Número de pasos}$$

Como se puede observar se tiene constante un error 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 esta 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 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 se tenía una velocidad de 26.23 mm/s entonces:

$$\text{Constante a multiplicar} = \frac{1000}{26.23} = 38,12$$

**Tabla 29.** Código velocidades

Velocidad (mm/s)	Número a enviar por protocolo
26.23	1000
258.23	9844.83415935951
3.82	145.634769348075

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 7899, se va a modificar al programa en la tarjeta para que acepte cuatro caracteres como velocidad, gracias a la planificación del protocolo, se puede enviar mensajes de 8 caracteres, por lo tanto el cambio es muy simple.

## RECOMENDACIONES PARA CALIBRACIÓN Y PRUEBAS

- Organizar correctamente las pruebas y tener los archivos listos antes de comenzar.
- Utilizar algún método automático para la medida de tiempos en especial cuando deben ser muy cortos ya que mientras el tiempo es corto el error humano será mas grande.
- Identificar las velocidades máximas y mínimas que la máquina puede ejecutar sin problemas mecánicos de inercia, precisión, etc...

## CONCLUSIONES

- Se cumplió con el diseño y construcción de una mesa CNC de corte por plasma que incluye el software de control de la máquina el cual lee archivos de AutoCAD en formato DXF.
- Ya que este proyecto de tesis existieron objetivos personales, también se logran cumplir, permitiendo diseñar un protocolo de comunicación desde la base. Al diseñar por completo la máquina se conoce perfectamente donde pueden existir errores y corregirlos, además permite una completa manipulación para adaptarla a cualquier trabajo fácilmente. El conocimiento obtenido sobre motores a pasos, protocolo de comunicación en base a puerto serial, programación en C#, programación de eZdsp F2808 y elementos electrónicos de protección 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 facilita de gran manera el trabajo dentro de la planta al momento de cortar figuras en acero inoxidable. Se puede notar un incremento en la velocidad de producción aun cuando la máquina es nueva y los operarios están aprendiendo a utilizarla.
- En relación a las metas propuesta se logro construir una máquina a un precio mucho menor que el precio comercial. La CNC puede ser fácilmente adaptada a diferentes tipos de trabajo, pero cambiando la programación dentro de la tarjeta y el software. 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.

## BIBLIOGRAFÍA

- 1. Wikipedia. *Control numérico*. 2012 [citado 2012 13 Sep]; < [http://es.wikipedia.org/w/index.php?title=Control\\_num%C3%A9rico&oldid=56838068](http://es.wikipedia.org/w/index.php?title=Control_num%C3%A9rico&oldid=56838068) >
- 2. Hypertherm. *Aplicaciones*. 2012 [citado 2012 13 Sep];
- 3. Hypertherm. *Beneficios*. 2012 [citado 2012 13 sep];
- 4. Overby, A., *CNC machining handbook : building, programming, and implementation*. 1st ed2011, New York, NY: McGraw-Hill/TAB Electronics. xii, 260 p.
- 5. AMS. *Step101*. 2010 [citado 2012 26 Septiembre]; < <http://www.ams2000.com/pdf/step101.pdf> >
- 6. Source, C.R. *Stepper vs Servo Motors*. 2011 [citado 2012 26 Septiembre]; < <http://www.cncroutersource.com/stepper-vs-servo.html> >
- 7. Torchmate. *Stepper Motors vs. Servo Motors vs. Intelligent Motors - The Facts*. 2010 [citado 2012 26 Septiembre]; < [http://torchmate.com/resource\\_center/editorials/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/) >
- 8. Probotix. *CNC Routers, Plasma Tables, & CNC Control Systems*. 2012 [citado 2012 26 septiembre]; < <http://www.probotix.com> >
- 9. Wikipedia, C.d.e.w.o.w.i.p.t.T.C.A.o.e. *Trigonometría*. 28 de noviembre del 2012, 22:56 UTC [citado 2012 1 de diciembre]; Wikipedia, La enciclopedia libre.:< <http://es.wikipedia.org/w/index.php?title=Trigonometr%C3%ADa&oldid=61741923> >

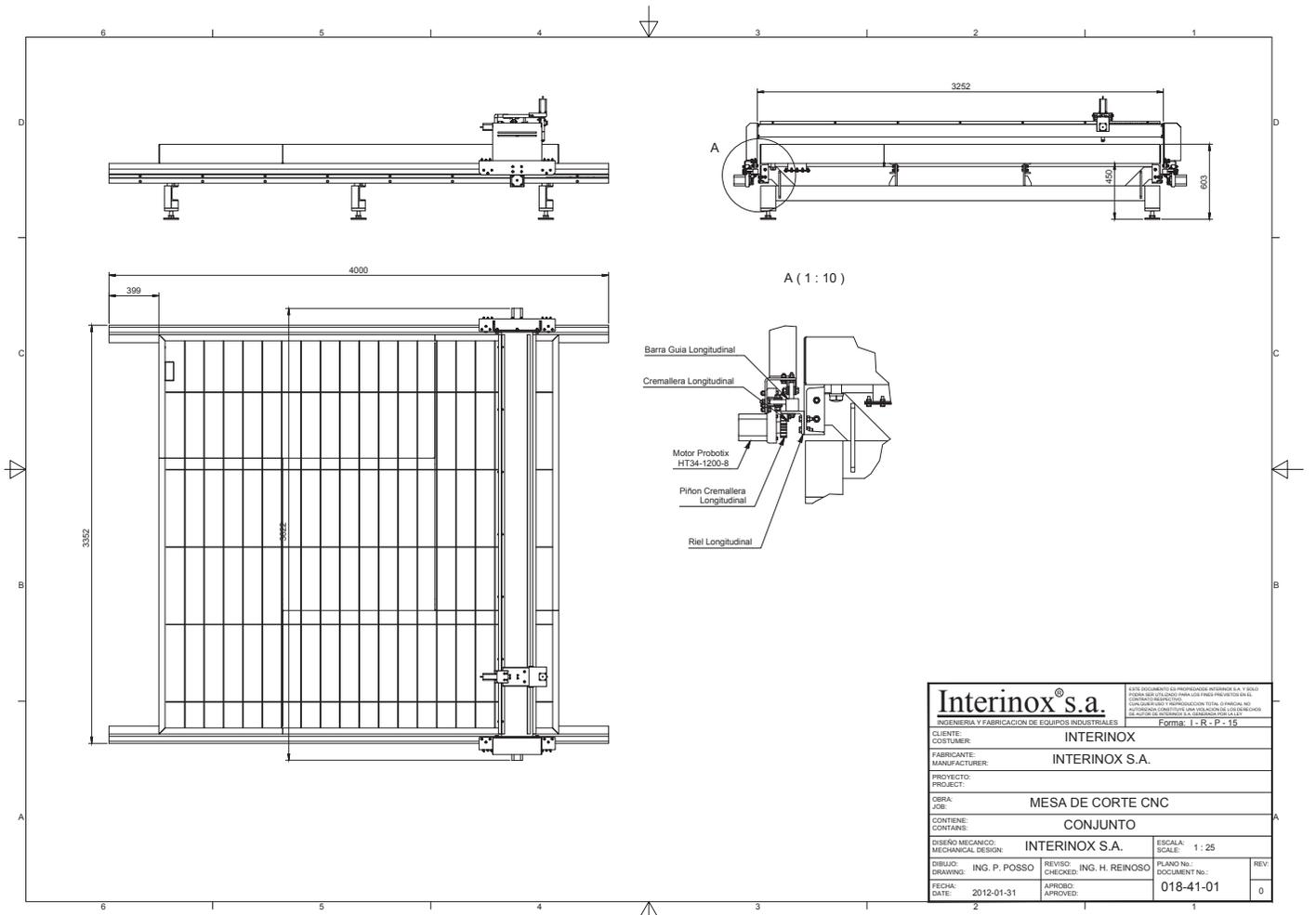
## **ANEXOS**

### **ANEXO A**

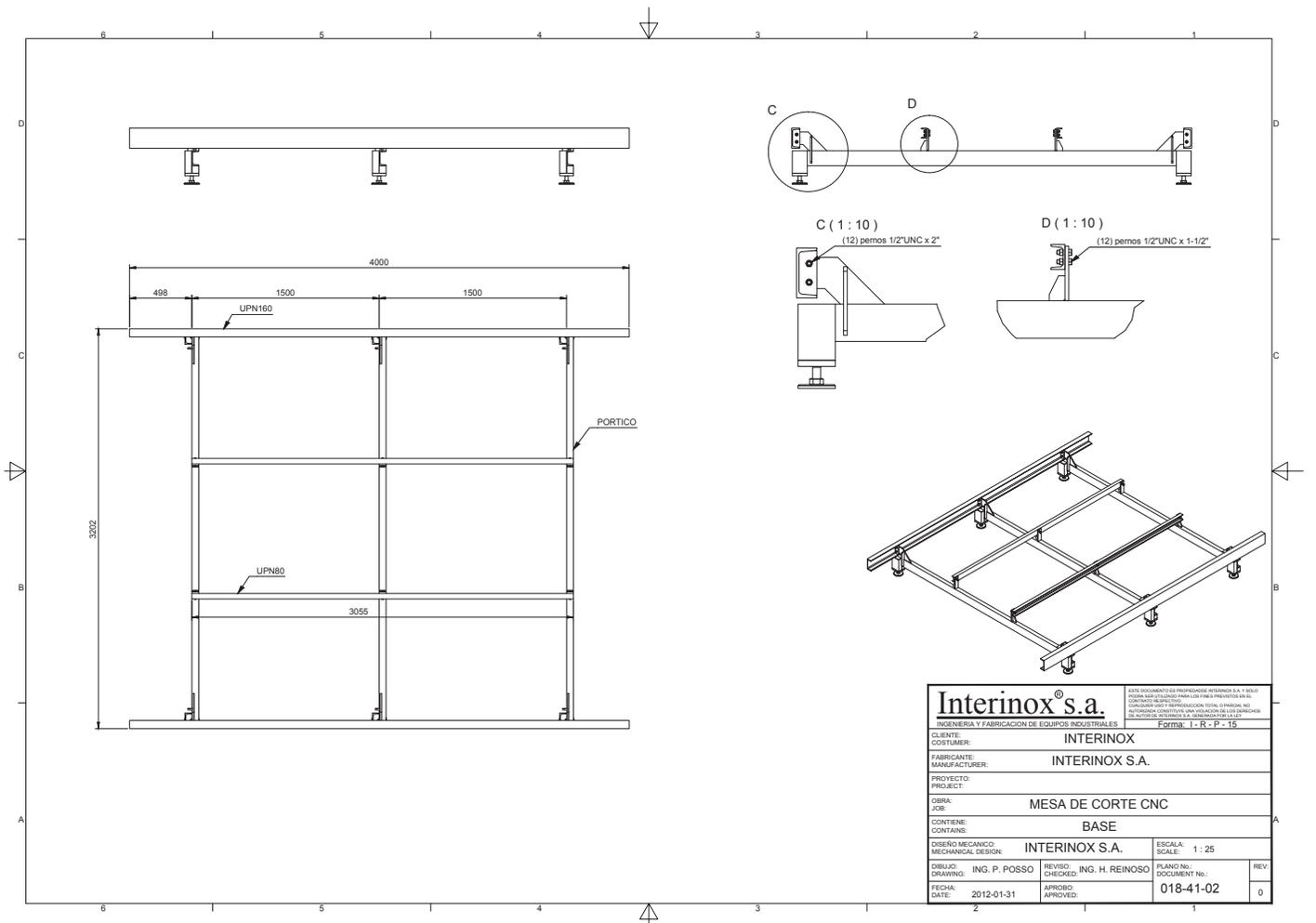
#### **Planos Construcción Mecánica**

Ver es hoja siguiente

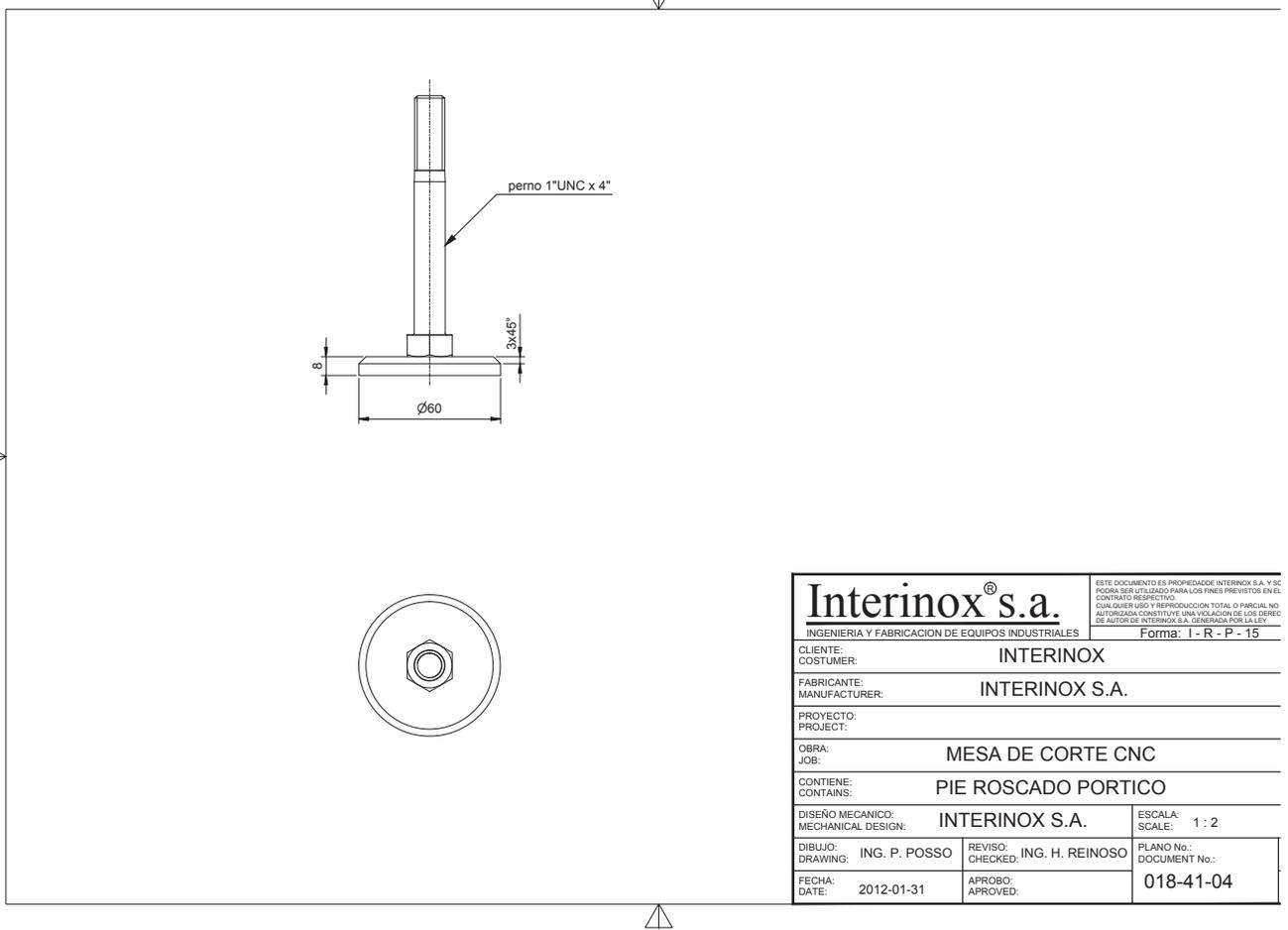
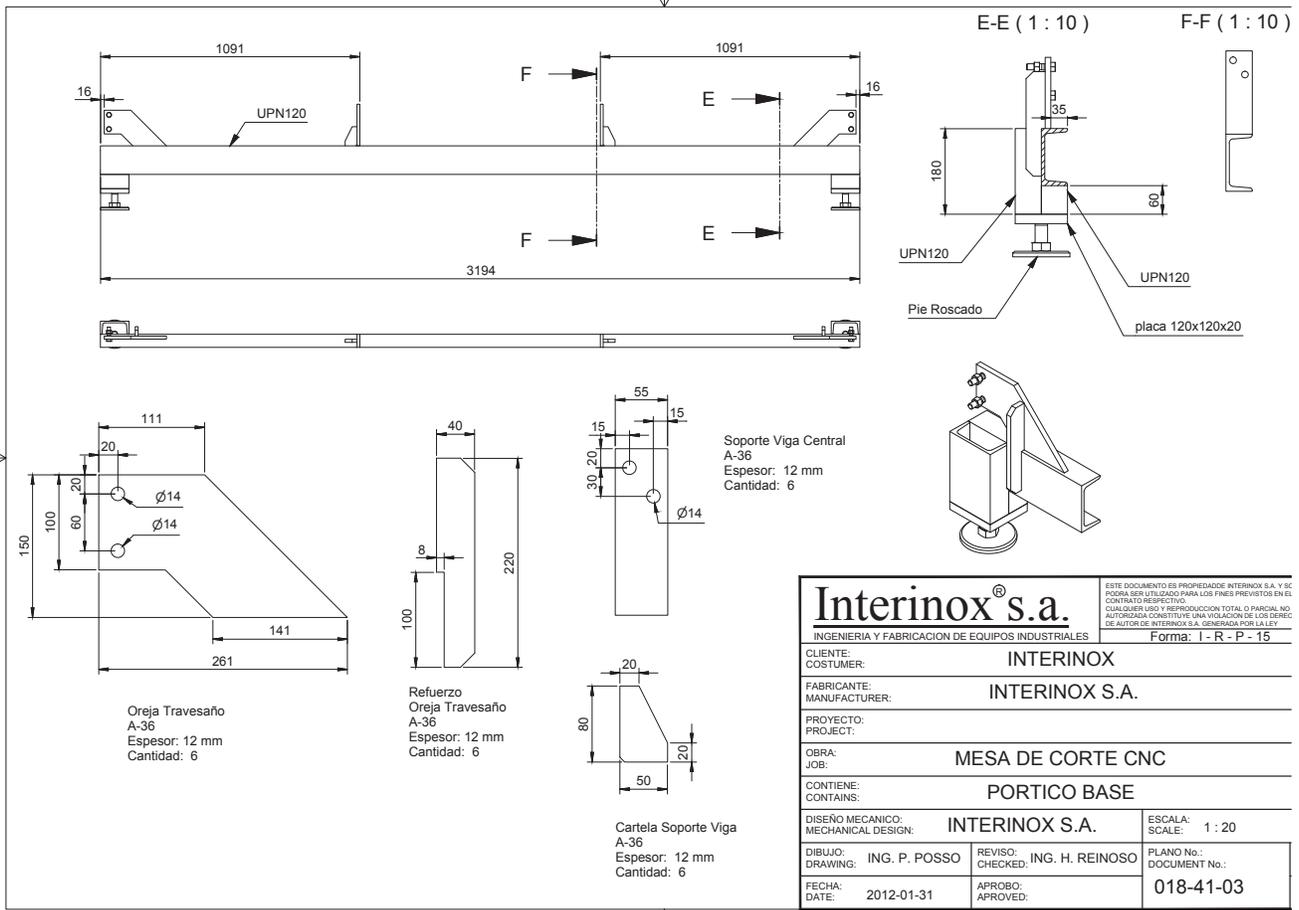
Los planos están con imágenes reducidas para que puedan entrar en el formato de este documento impreso, para ver los detalles y los planos originales, favor referirse al material digital presentado con este documento.

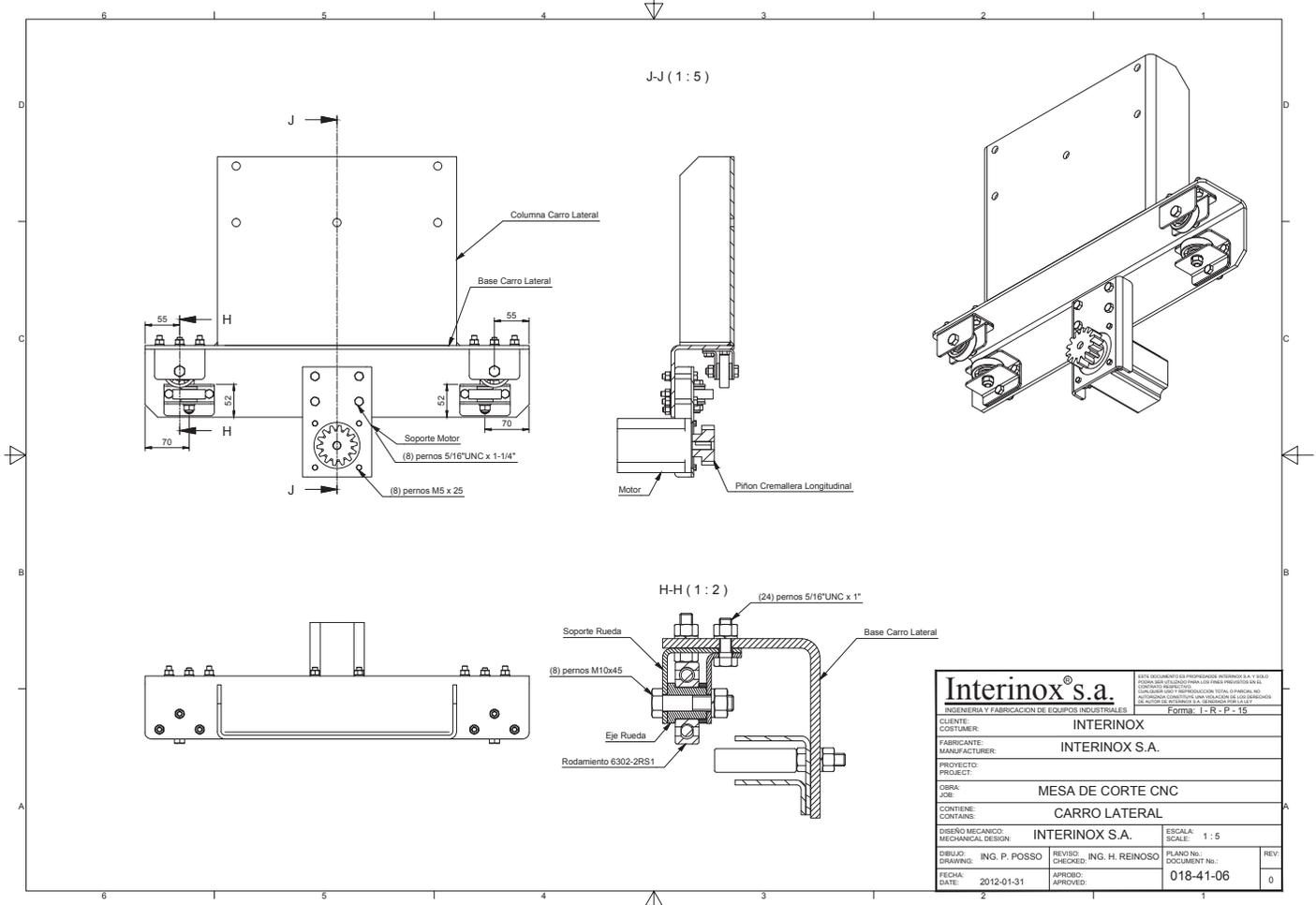
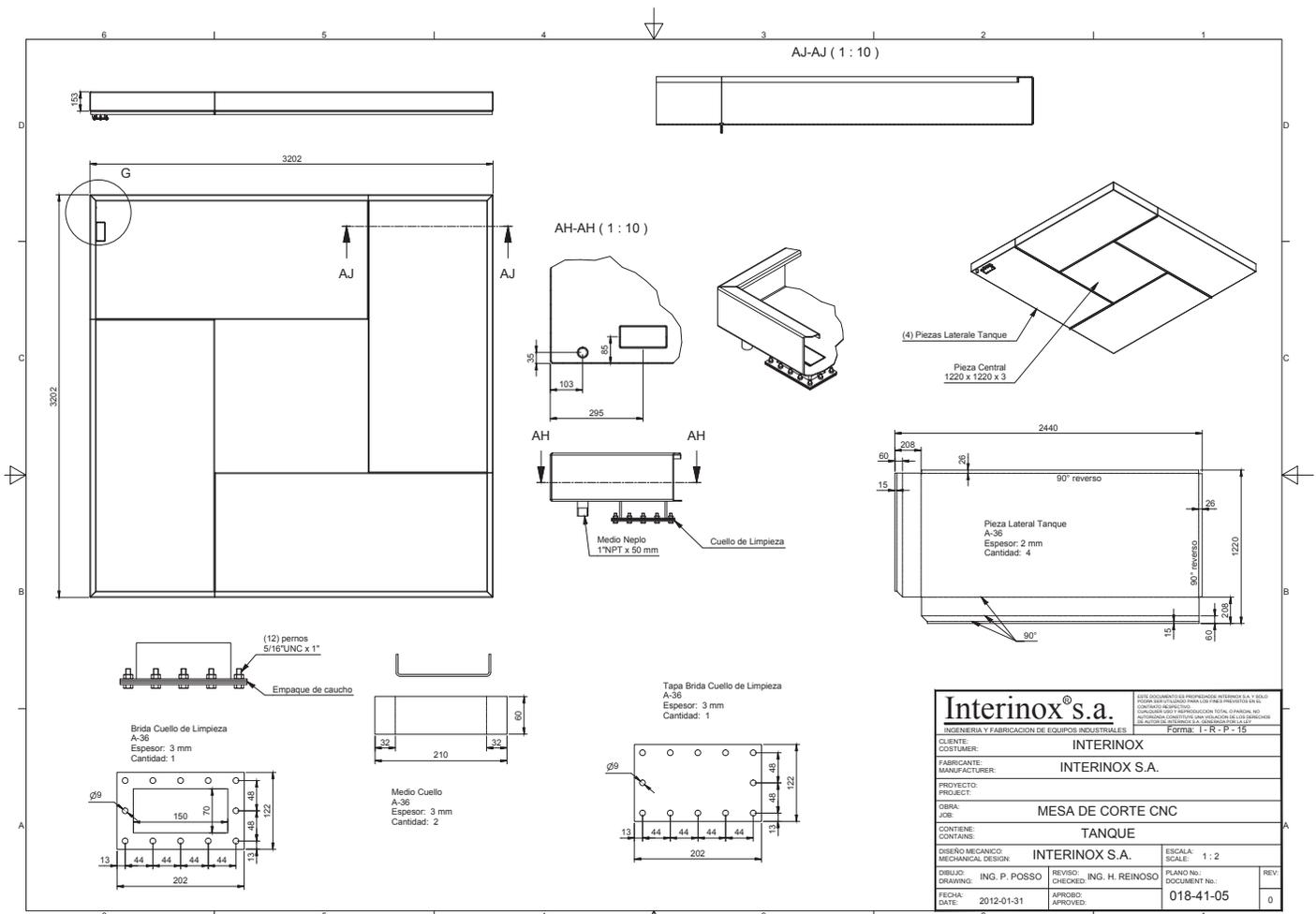


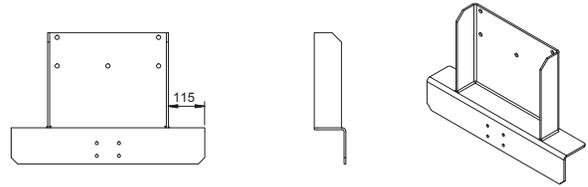
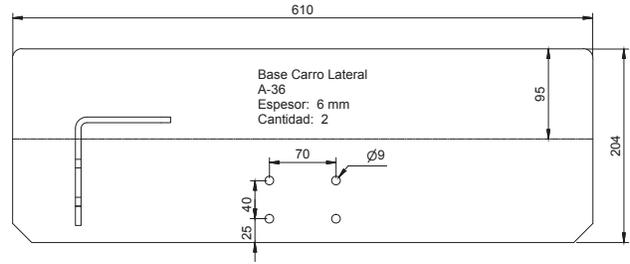
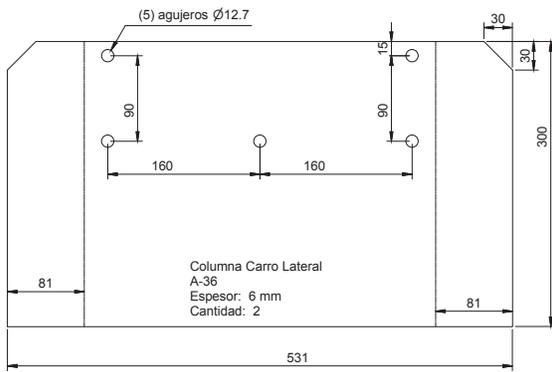
<b>Interinox s.a.</b> INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		ESTE DOCUMENTO DE PROPIEDAD INTERINOX S.A. Y DEBE QUEDAR SIN CIRCULAR PARA LOS FINES PERMITIDOS EN LA CONTRATA. INTERINOX S.A. AUTORIZA LA REPRODUCCION TOTAL O PARCIAL, SIN COSTO PARA EL USUARIO, DE LOS DISEÑOS Y DIBUJOS DE INTERINOX S.A. EN RELACION A LOS EQUIPOS DE INTERINOX S.A.	
CLIENTE:	INTERINOX		
FABRICANTE:	INTERINOX S.A.		
PROYECTO:	MESA DE CORTE CNC		
OBRA:	MESA DE CORTE CNC		
CONTIENE:	CONJUNTO		
DISEÑO MECANICO:	INTERINOX S.A.	ESCALA:	1 : 25
MECHANICAL DESIGN:	INTERINOX S.A.	SCALE:	1 : 25
DIBUJO:	ING. P. POSSO	REVISO:	ING. H. REINOSO
DRAWING:	ING. P. POSSO	CHECKED:	ING. H. REINOSO
FECHA:	2012-01-31	APROBADO:	018-41-01
DATE:	2012-01-31	APPROVED:	018-41-01
			0



<b>Interinox s.a.</b> INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		ESTE DOCUMENTO DE PROPIEDAD INTERINOX S.A. Y DEBE QUEDAR SIN CIRCULAR PARA LOS FINES PERMITIDOS EN LA CONTRATA. INTERINOX S.A. AUTORIZA LA REPRODUCCION TOTAL O PARCIAL, SIN COSTO PARA EL USUARIO, DE LOS DISEÑOS Y DIBUJOS DE INTERINOX S.A. EN RELACION A LOS EQUIPOS DE INTERINOX S.A.	
CLIENTE:	INTERINOX		
FABRICANTE:	INTERINOX S.A.		
PROYECTO:	MESA DE CORTE CNC		
OBRA:	MESA DE CORTE CNC		
CONTIENE:	BASE		
DISEÑO MECANICO:	INTERINOX S.A.	ESCALA:	1 : 25
MECHANICAL DESIGN:	INTERINOX S.A.	SCALE:	1 : 25
DIBUJO:	ING. P. POSSO	REVISO:	ING. H. REINOSO
DRAWING:	ING. P. POSSO	CHECKED:	ING. H. REINOSO
FECHA:	2012-01-31	APROBADO:	018-41-02
DATE:	2012-01-31	APPROVED:	018-41-02
			0

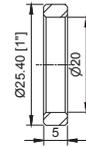
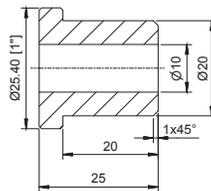
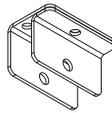
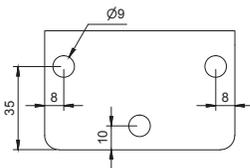
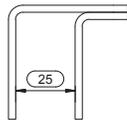
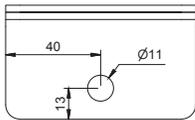






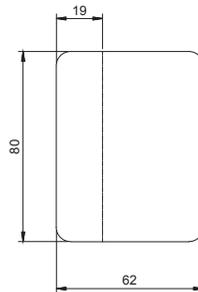
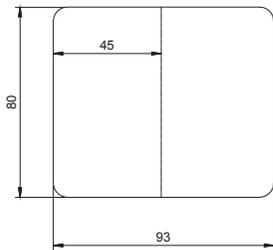
<b>Interinox® s.a.</b>		ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y SE PODRA SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCION TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACION DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY.	
INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		Forma: 1 - R - P - 15	
CLIENTE:	INTERINOX		
COSTUMER:	INTERINOX S.A.		
FABRICANTE:	INTERINOX S.A.		
MANUFACTURER:	INTERINOX S.A.		
PROYECTO:	MESA DE CORTE CNC		
PROJECT:	MESA DE CORTE CNC		
OBRA:	MESA DE CORTE CNC		
JOB:	MESA DE CORTE CNC		
CONTIENE:	BASTIDOR CARRO LATERAL		
CONTAINS:	BASTIDOR CARRO LATERAL		
DISEÑO MECANICO:	INTERINOX S.A.	ESCALA:	1 : 5
MECHANICAL DESIGN:	INTERINOX S.A.	SCALE:	1 : 5
DIBUJO:	ING. P. POSSO	REVISO:	ING. H. REINOSO
DRAWING:	ING. P. POSSO	CHECKED:	ING. H. REINOSO
FECHA:	2012-01-31	APROBO:	
DATE:	2012-01-31	APPROVED:	
		PLANO No.:	018-41-07
		DOCUMENT No.:	018-41-07

**SOPORTE RUEDA**  
Cantidad: 8



Eje Rueda  
Esc. 1 : 1  
AISI 304  
Cantidad: 8

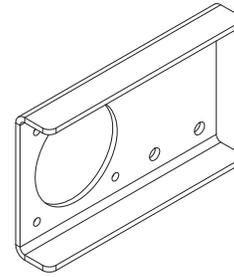
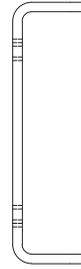
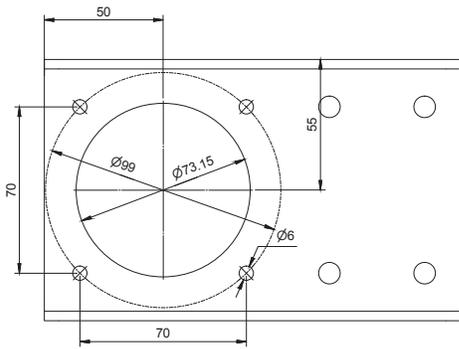
Espaciador Rueda  
Esc. 1 : 1  
AISI 304  
Cantidad: 8



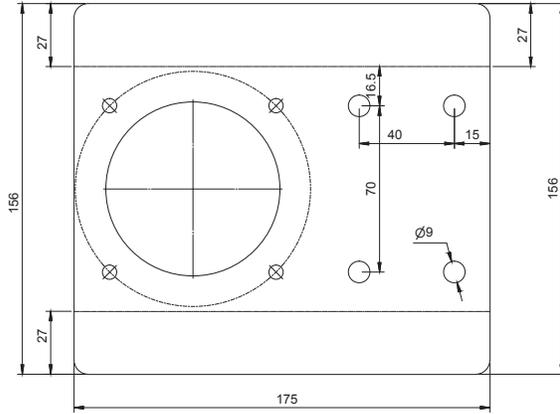
Pieza 1 Soporte Rueda  
AISI 304  
Espesor: 3 mm  
Cantidad: 8

Pieza 2 Soporte Rueda  
AISI 304  
Espesor: 3 mm  
Cantidad: 8

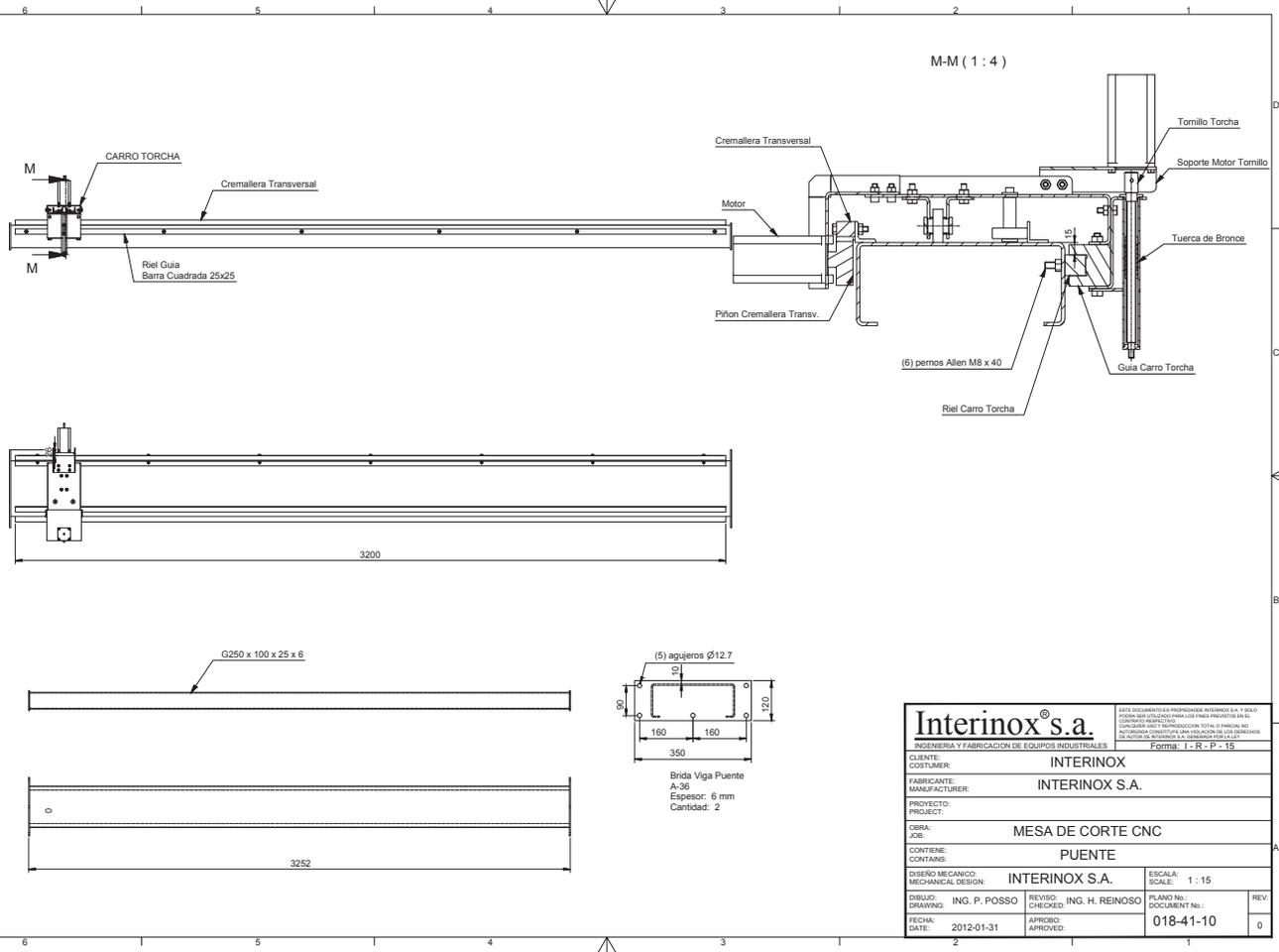
<b>Interinox® s.a.</b>		ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y SE PODRA SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCION TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACION DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY.	
INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		Forma: 1 - R - P - 15	
CLIENTE:	INTERINOX		
COSTUMER:	INTERINOX S.A.		
FABRICANTE:	INTERINOX S.A.		
MANUFACTURER:	INTERINOX S.A.		
PROYECTO:	MESA DE CORTE CNC		
PROJECT:	MESA DE CORTE CNC		
OBRA:	MESA DE CORTE CNC		
JOB:	MESA DE CORTE CNC		
CONTIENE:	PIEZAS RUEDA CARRO LATERAL		
CONTAINS:	PIEZAS RUEDA CARRO LATERAL		
DISEÑO MECANICO:	INTERINOX S.A.	ESCALA:	1 : 2 / 1 : 1
MECHANICAL DESIGN:	INTERINOX S.A.	SCALE:	1 : 2 / 1 : 1
DIBUJO:	ING. P. POSSO	REVISO:	ING. H. REINOSO
DRAWING:	ING. P. POSSO	CHECKED:	ING. H. REINOSO
FECHA:	2012-01-31	APROBO:	
DATE:	2012-01-31	APPROVED:	
		PLANO No.:	018-41-08
		DOCUMENT No.:	018-41-08



Soporte Motor Carro Lateral  
A - 36  
Espesor: 4 mm  
Cantidad: 2



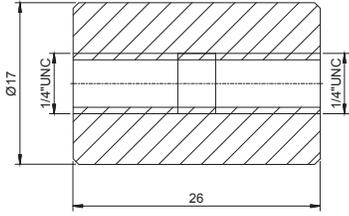
<b>Interinox® s.a.</b> INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y SE PODRA SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCION TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACION DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY
CLIENTE: COSTUMER:	INTERINOX	
FABRICANTE: MANUFACTURER:	INTERINOX S.A.	
PROYECTO: PROJECT:	MESA DE CORTE CNC	
OBRA: JOB:	SOPORTE MOTOR CARRO LATERAL	
CONTIENE: CONTAINS:	SOPORTE MOTOR CARRO LATERAL	
DISEÑO MECANICO: MECHANICAL DESIGN:	INTERINOX S.A.	ESCALA: SCALE: 1 : 2 / 1 : 1
DIBUJO: DRAWING:	ING. P. POSSO	REVISO: CHECKED: ING. H. REINOSO
FECHA: DATE:	2012-02-01	APROBO: APPROVED: 018-41-09
		PLANO No.: DOCUMENT No.:



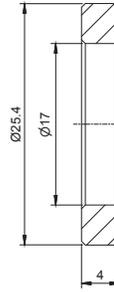
<b>Interinox® s.a.</b> INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y SE PODRA SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCION TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACION DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY
CLIENTE: COSTUMER:	INTERINOX	
FABRICANTE: MANUFACTURER:	INTERINOX S.A.	
PROYECTO: PROJECT:	MESA DE CORTE CNC	
OBRA: JOB:	PUENTE	
CONTIENE: CONTAINS:	PUENTE	
DISEÑO MECANICO: MECHANICAL DESIGN:	INTERINOX S.A.	ESCALA: SCALE: 1 : 15
DIBUJO: DRAWING:	ING. P. POSSO	REVISO: CHECKED: ING. H. REINOSO
FECHA: DATE:	2012-01-31	APROBO: APPROVED: 018-41-10
		PLANO No.: DOCUMENT No.:
		REV: 0



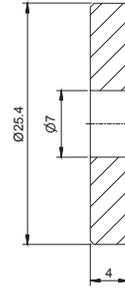




Eje Rueda Carro  
AISI 304  
Cantidad: 1

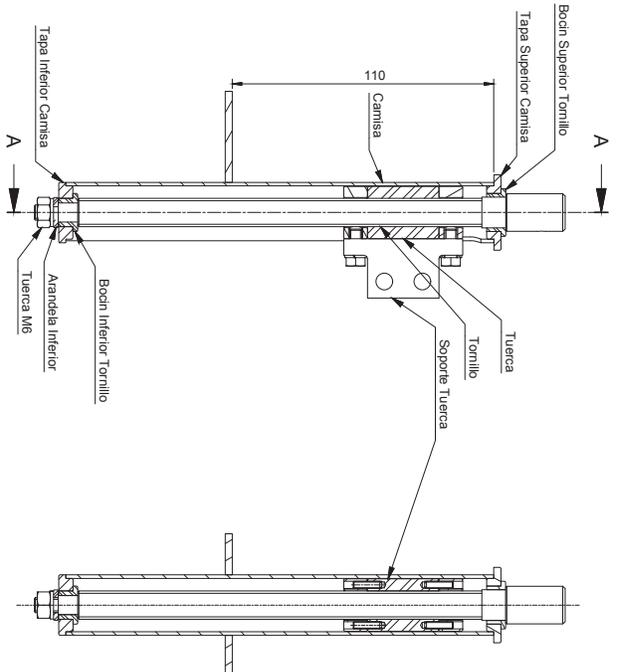
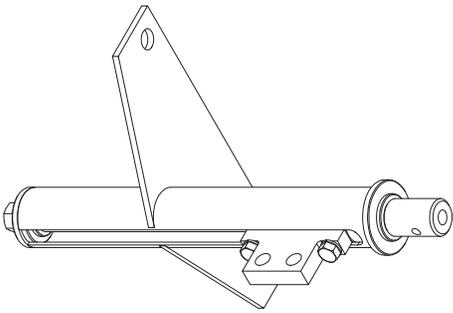


Espaciador Rueda Coche  
AISI 304  
Cantidad: 2



Tope Eje Rueda Coche  
AISI 304  
Cantidad: 2

<b>Interinox® s.a.</b>		ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y SE PODRA SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCION TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACION DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY.	
INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		Forma: 1 - R - P - 15	
CLIENTE: COSTUMER:	INTERINOX		
FABRICANTE: MANUFACTURER:	INTERINOX S.A.		
PROYECTO: PROJECT:			
OBRA: JOB:	MESA DE CORTE CNC		
CONTIENE: CONTAINS:	PIEZAS RUEDA CARRO		
DISEÑO MECANICO: MECHANICAL DESIGN:	INTERINOX S.A.		ESCALA: SCALE: 1 : 1
DIBUJO: DRAWING: ING. P. POSSO	REVISO: CHECKED: ING. H. REINOSO	PLANO No.: DOCUMENT No.:	
FECHA: DATE: 2012-01-31	APROBO: APPROVED:	018-41-15	

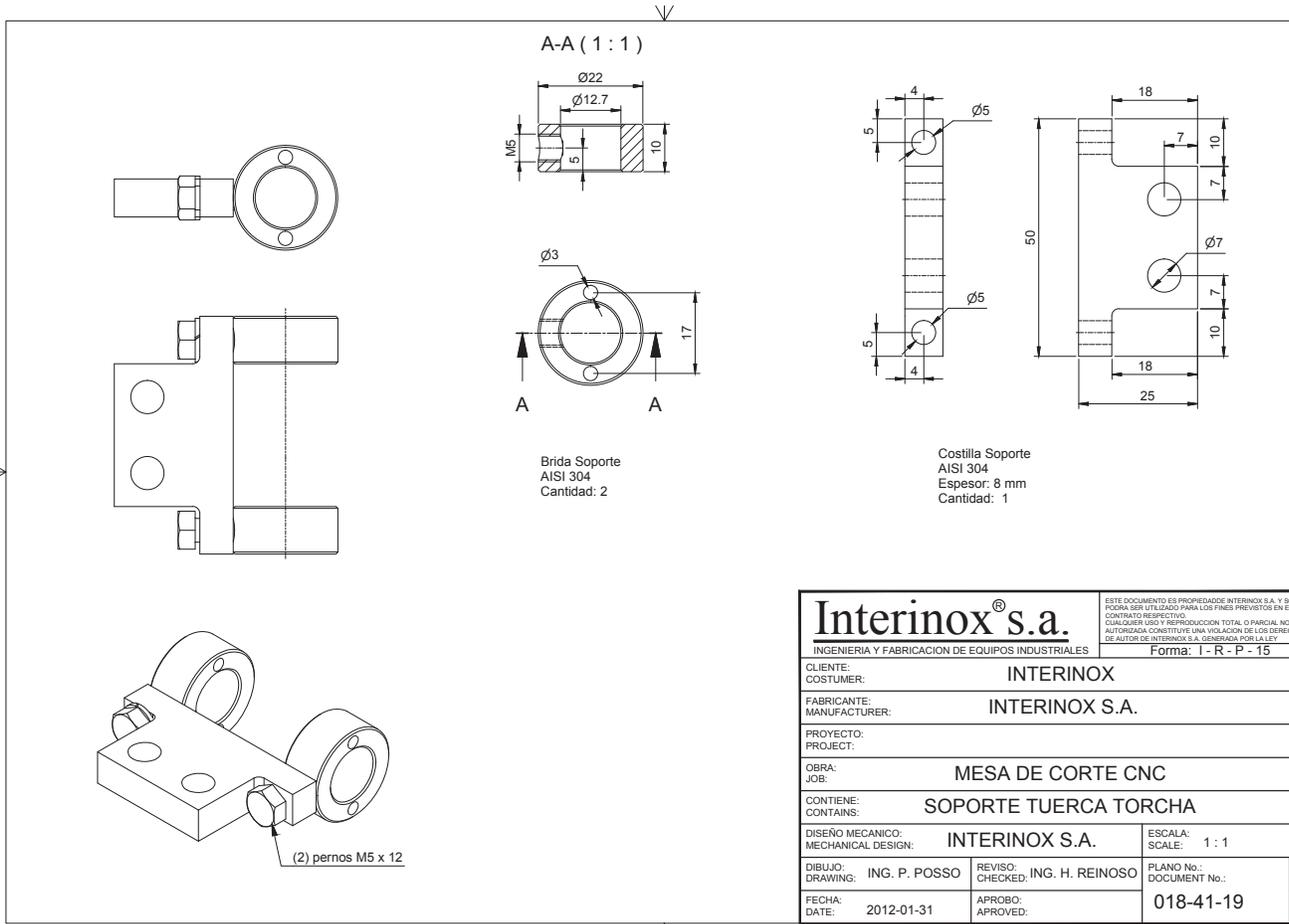


U-U (1 : 2)

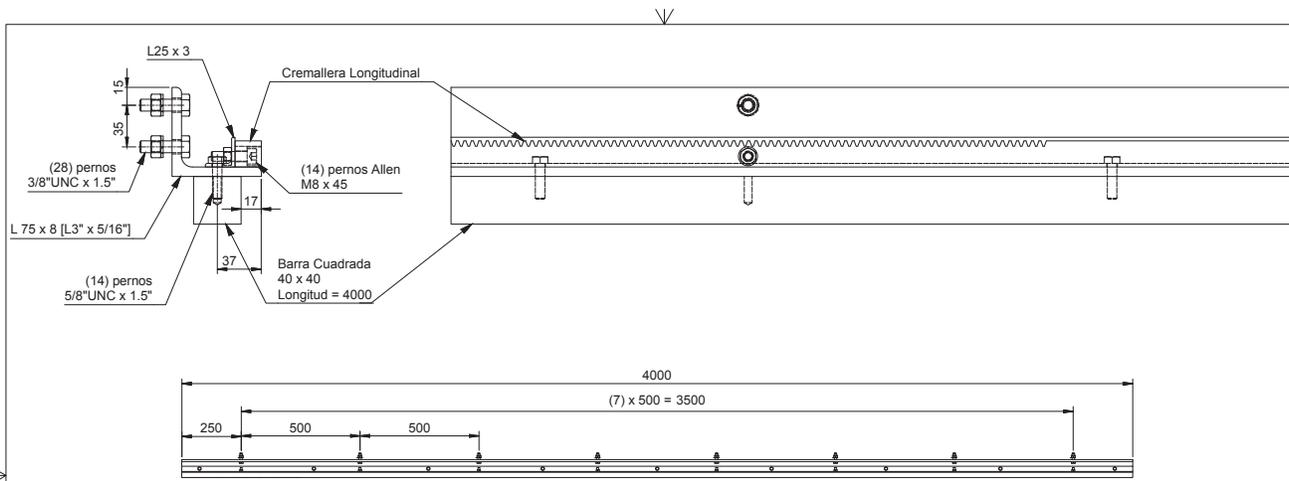
AA (1 : 2)

<b>Interinox® s.a.</b>		ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y SE PODRA SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCION TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACION DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY.	
INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		Forma: 1 - R - P - 15	
CLIENTE: COSTUMER:	INTERINOX		
FABRICANTE: MANUFACTURER:	INTERINOX S.A.		
PROYECTO: PROJECT:			
OBRA: JOB:	MESA DE CORTE CNC		
CONTIENE: CONTAINS:	TORNILLO TORCHA		
DISEÑO MECANICO: MECHANICAL DESIGN:	INTERINOX S.A.		ESCALA: SCALE: 1 : 1
DIBUJO: DRAWING: ING. P. POSSO	REVISO: CHECKED: ING. H. REINOSO	PLANO No.: DOCUMENT No.:	
FECHA: DATE: 2012-01-31	APROBO: APPROVED:	018-41-16	
			REV: 0

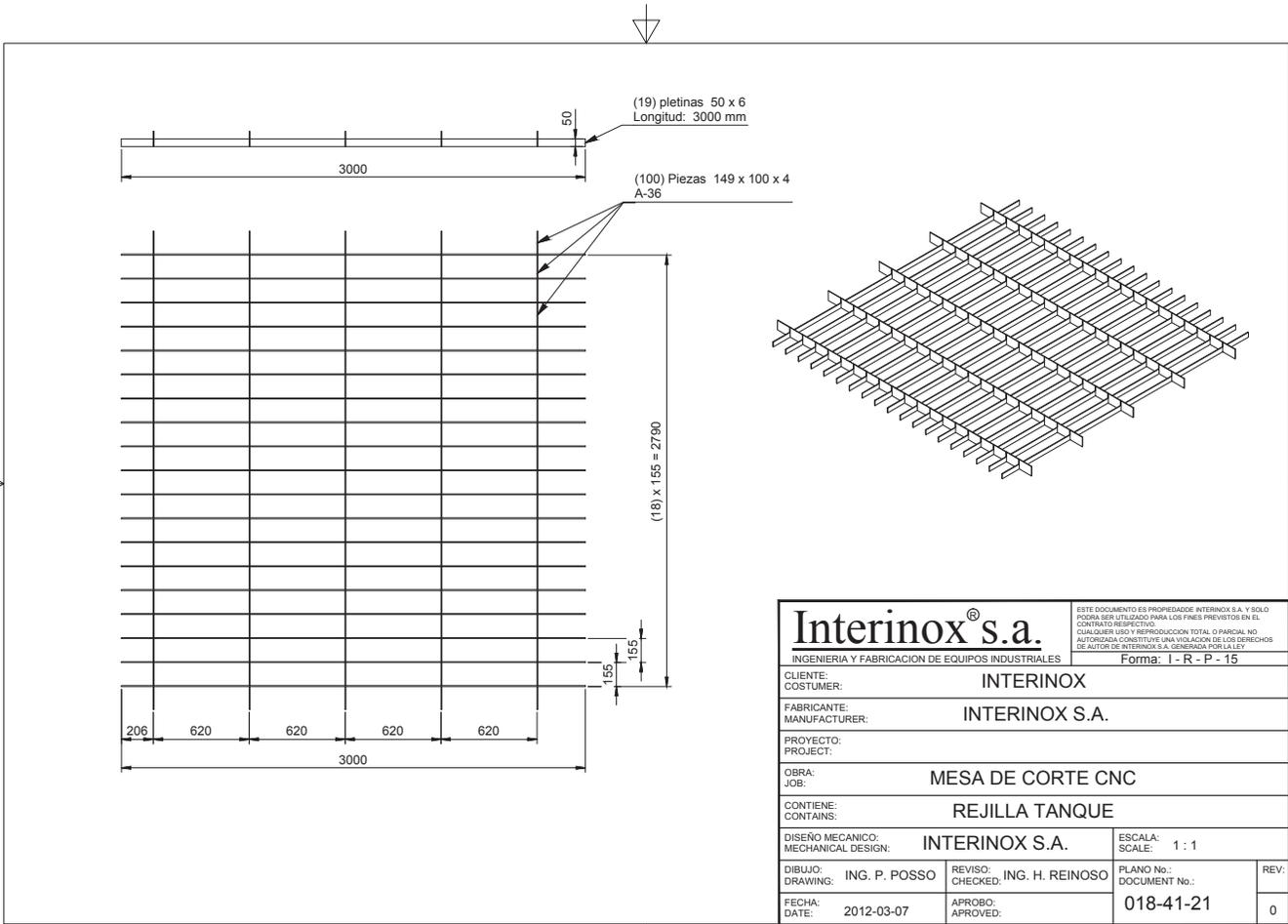




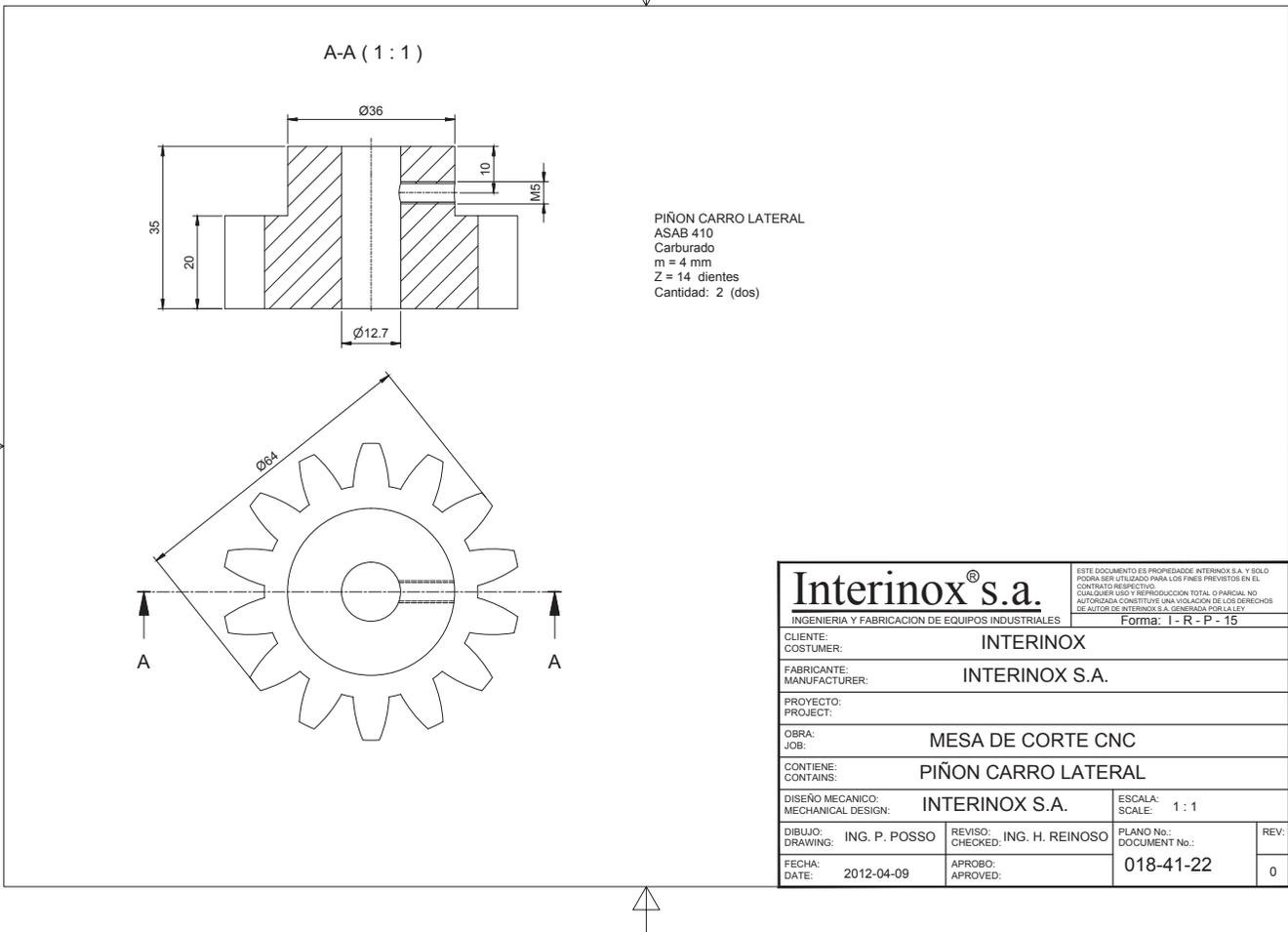
<b>Interinox® s.a.</b>		<small>ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y SE PODRA SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCION TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACION DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY.</small>	
INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		Forma: 1 - R - P - 15	
CLIENTE: COSTUMER:	INTERINOX		
FABRICANTE: MANUFACTURER:	INTERINOX S.A.		
PROYECTO: PROJECT:			
OBRA: JOB:	MESA DE CORTE CNC		
CONTIENE: CONTAINS:	SOPORTE TUERCA TORCHA		
DISEÑO MECANICO: MECHANICAL DESIGN:	INTERINOX S.A.	ESCALA: SCALE:	1 : 1
DIBUJO: DRAWING:	ING. P. POSSO	REVISO: CHECKED:	ING. H. REINOSO
FECHA: DATE:	2012-01-31	APROBO: APPROVED:	
		PLANO No.:	DOCUMENT No.:
			018-41-19



<b>Interinox® s.a.</b>		<small>ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y SE PODRA SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCION TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACION DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY.</small>	
INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		Forma: 1 - R - P - 15	
CLIENTE: COSTUMER:	INTERINOX		
FABRICANTE: MANUFACTURER:	INTERINOX S.A.		
PROYECTO: PROJECT:			
OBRA: JOB:	MESA DE CORTE CNC		
CONTIENE: CONTAINS:	RIEL LATERAL		
DISEÑO MECANICO: MECHANICAL DESIGN:	INTERINOX S.A.	ESCALA: SCALE:	1 : 1
DIBUJO: DRAWING:	ING. P. POSSO	REVISO: CHECKED:	ING. H. REINOSO
FECHA: DATE:	2012-02-01	APROBO: APPROVED:	
		PLANO No.:	DOCUMENT No.:
			018-41-20

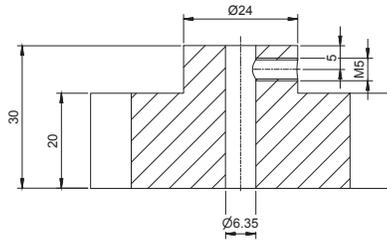


<b>Interinox® s.a.</b>		<small>ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y SOLO PODRA SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCION TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACION DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY</small>	
INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		Forma: I - R - P - 15	
CLIENTE: INTERINOX			
COSTUMER: INTERINOX			
FABRICANTE: INTERINOX S.A.			
MANUFACTURER: INTERINOX S.A.			
PROYECTO: MESA DE CORTE CNC			
PROJECT: MESA DE CORTE CNC			
OBRA: MESA DE CORTE CNC			
JOB: MESA DE CORTE CNC			
CONTIENE: REJILLA TANQUE			
CONTAINS: REJILLA TANQUE			
DISEÑO MECANICO: INTERINOX S.A.		ESCALA: 1 : 1	
MECHANICAL DESIGN: INTERINOX S.A.		SCALE: 1 : 1	
DIBUJO: ING. P. POSSO	REVISO: ING. H. REINOSO	PLANO No.: 018-41-21	REV: 0
DRAWING: ING. P. POSSO	CHECKED: ING. H. REINOSO	DOCUMENT No.:	
FECHA: 2012-03-07	APROBO: 2012-03-07	018-41-21	0
DATE: 2012-03-07	APPROVED: 2012-03-07		

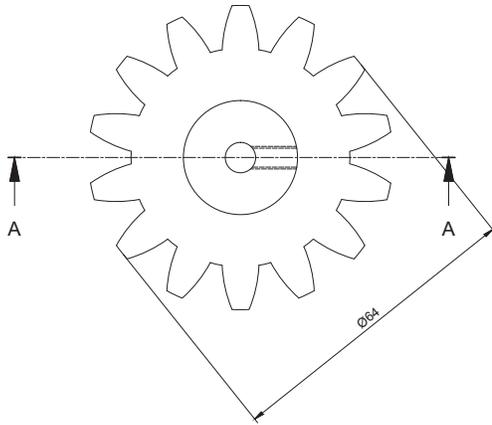


<b>Interinox® s.a.</b>		<small>ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y SOLO PODRA SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCION TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACION DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY</small>	
INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		Forma: I - R - P - 15	
CLIENTE: INTERINOX			
COSTUMER: INTERINOX			
FABRICANTE: INTERINOX S.A.			
MANUFACTURER: INTERINOX S.A.			
PROYECTO: MESA DE CORTE CNC			
PROJECT: MESA DE CORTE CNC			
OBRA: MESA DE CORTE CNC			
JOB: MESA DE CORTE CNC			
CONTIENE: PIÑON CARRO LATERAL			
CONTAINS: PIÑON CARRO LATERAL			
DISEÑO MECANICO: INTERINOX S.A.		ESCALA: 1 : 1	
MECHANICAL DESIGN: INTERINOX S.A.		SCALE: 1 : 1	
DIBUJO: ING. P. POSSO	REVISO: ING. H. REINOSO	PLANO No.: 018-41-22	REV: 0
DRAWING: ING. P. POSSO	CHECKED: ING. H. REINOSO	DOCUMENT No.:	
FECHA: 2012-04-09	APROBO: 2012-04-09	018-41-22	0
DATE: 2012-04-09	APPROVED: 2012-04-09		

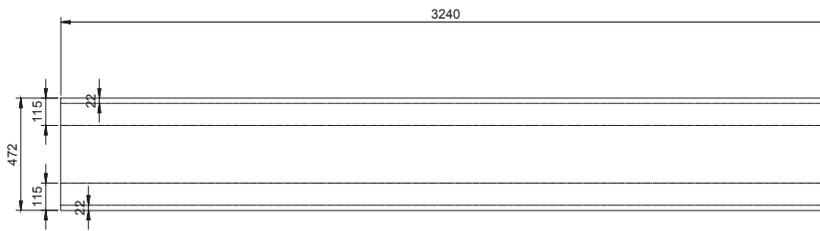
A-A (1:1)

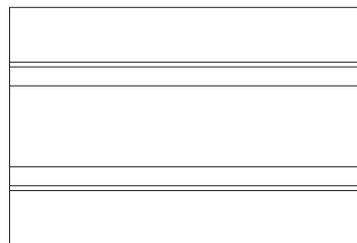
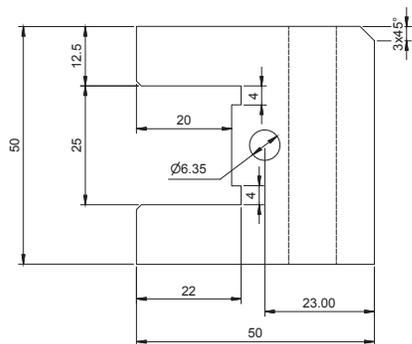


PIÑÓN CARRO PUENTE  
 ASAB 410  
 Carburado  
 m = 4 mm  
 Z = 14 dientes  
 Cantidad: 1 (uno)

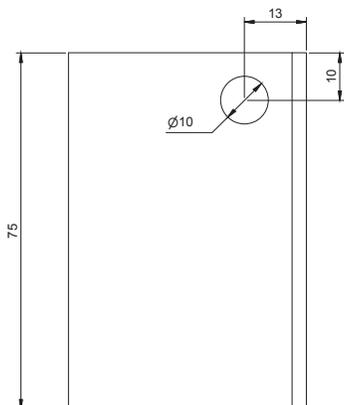


<b>Interinox® s.a.</b>		<small>ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y SOLO PODRA SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCION TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACION DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY.</small>	
INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		Forma: 1 - R - P - 15	
CLIENTE:	INTERINOX		
COSTUMER:	INTERINOX S.A.		
FABRICANTE:	INTERINOX S.A.		
MANUFACTURER:	INTERINOX S.A.		
PROYECTO:			
PROJECT:			
OBRA:	MESA DE CORTE CNC		
JOB:	MESA DE CORTE CNC		
CONTIENE:	PIÑÓN CARRO PUENTE		
CONTAINS:	PIÑÓN CARRO PUENTE		
DISEÑO MECANICO:	INTERINOX S.A.	ESCALA:	1 : 1
MECHANICAL DESIGN:	INTERINOX S.A.	SCALE:	1 : 1
DIBUJO:	ING. P. POSSO	REVISO:	ING. H. REINOSO
DRAWING:	ING. P. POSSO	CHECKED:	ING. H. REINOSO
FECHA:	2012-04-09	APROBO:	
DATE:	2012-04-09	APPROVED:	
		PLANO No.:	018-41-23
		DOCUMENT No.:	018-41-23
		REV:	0

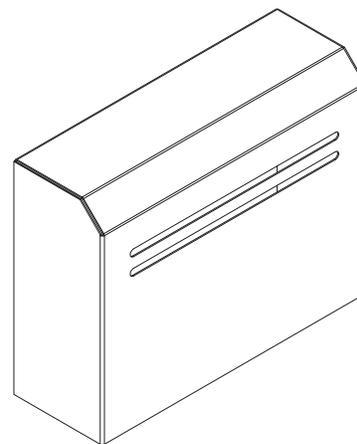
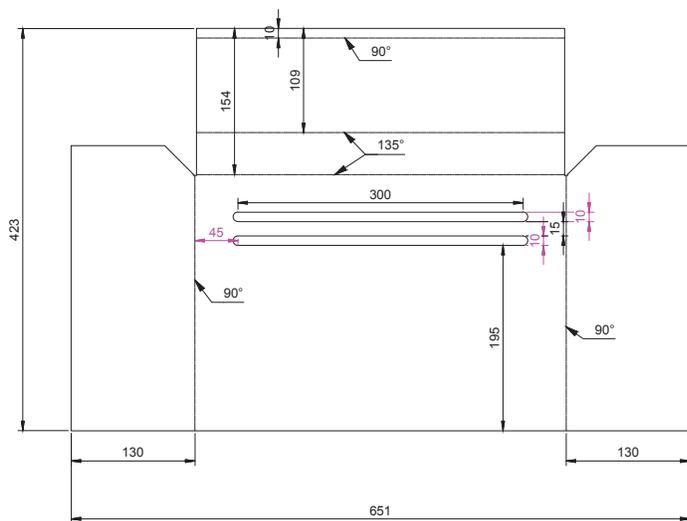




Medio Bloque Guia Carro  
Duralon  
Cantidad: 2

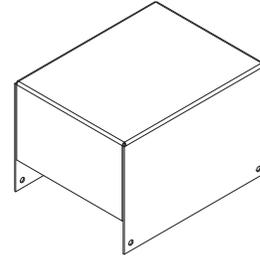
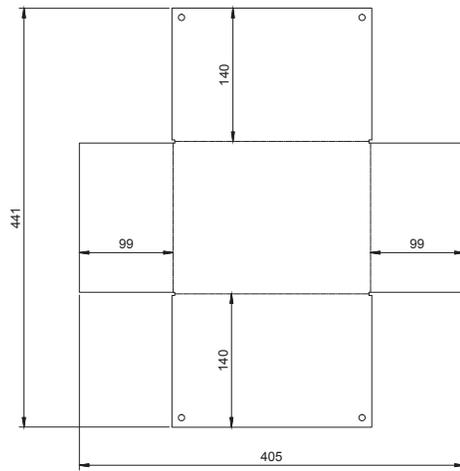


<b>Interinox® s.a.</b>		ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y SOLO PODRA SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCION TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACION DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY	
INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		Forma: I - R - P - 15	
CLIENTE: COSTUMER:	INTERINOX		
FABRICANTE: MANUFACTURER:	INTERINOX S.A.		
PROYECTO: PROJECT:			
OBRA: JOB:	MESA DE CORTE CNC		
CONTIENE: CONTAINS:	MEDIO BLOQUE GUIA		
DISEÑO MECANICO: MECHANICAL DESIGN:	INTERINOX S.A.	ESCALA: SCALE:	1 : 1
DIBUJO: DRAWING:	ING. P. POSSO	REVISO: CHECKED:	ING. H. REINOSO
FECHA: DATE:	2012-05-28	APROBO: APPROVED:	
		PLANO No.: DOCUMENT No.:	018-41-25
		REV:	0



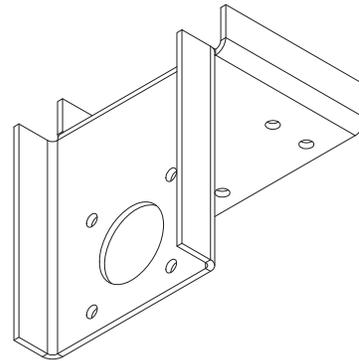
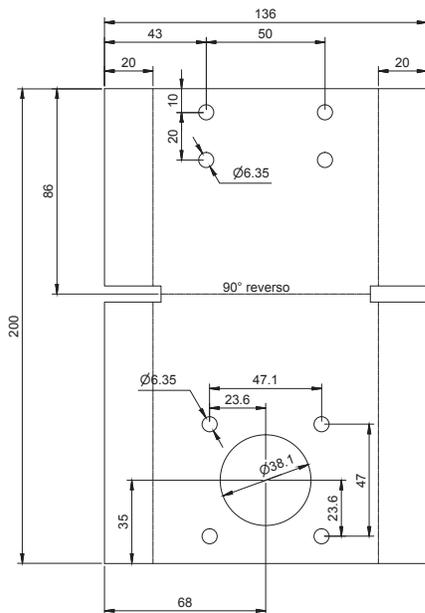
Desarrollo Guarda Carro Lateral  
AISI 304  
Espesor: 1,5 mm  
Cantidad: 2

<b>Interinox® s.a.</b>		ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y SOLO PODRA SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCION TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACION DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY	
INGENIERIA Y FABRICACION DE EQUIPOS INDUSTRIALES		Forma: I - R - P - 15	
CLIENTE: COSTUMER:	INTERINOX		
FABRICANTE: MANUFACTURER:	INTERINOX S.A.		
PROYECTO: PROJECT:			
OBRA: JOB:	MESA DE CORTE CNC		
CONTIENE: CONTAINS:	GUARDA CARRO LATERAL		
DISEÑO MECANICO: MECHANICAL DESIGN:	INTERINOX S.A.	ESCALA: SCALE:	1 : 5
DIBUJO: DRAWING:	ING. P. POSSO	REVISO: CHECKED:	ING. H. REINOSO
FECHA: DATE:	2012-08-29	APROBO: APPROVED:	
		PLANO No.: DOCUMENT No.:	018-41-25



Desarrollo  
Guarda Carro Torcha  
AISI 304  
Espesor: 1.5 mm  
Cantidad: 1

<b>Interinox® s.a.</b>		<small>ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y NO PODRÁ SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCIÓN TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACIÓN DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY</small>	
INGENIERÍA Y FABRICACIÓN DE EQUIPOS INDUSTRIALES		Forma: I - R - P - 15	
CLIENTE: COSTUMER:	INTERINOX		
FABRICANTE: MANUFACTURER:	INTERINOX S.A.		
PROYECTO: PROJECT:			
OBRA: JOB:	MESA DE CORTE CNC		
CONTIENE: CONTAINS:	GUARDA CARRO TORCHA		
DISEÑO MECANICO: MECHANICAL DESIGN:	INTERINOX S.A.	ESCALA: SCALE:	1 : 5
DIBUJO: DRAWING:	ING. P. POSSO	REVISO: CHECKED:	ING. H. REINOSO
FECHA: DATE:	2012-08-29	APROBO: APPROVED:	
		PLANO No.: DOCUMENT No.:	018-41-27

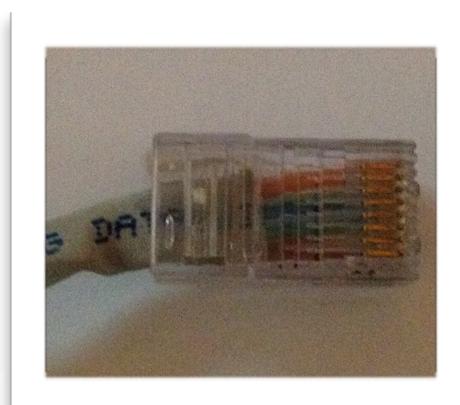
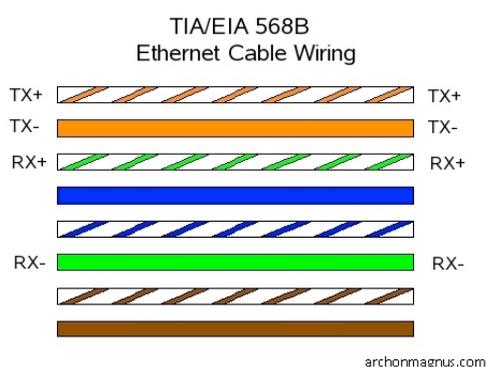


Soporte Vertical Carro Torcha  
AISI 304  
Espesor: 3 mm  
Cantidad: 1

<b>Interinox® s.a.</b>		<small>ESTE DOCUMENTO ES PROPIEDAD DE INTERINOX S.A. Y NO PODRÁ SER UTILIZADO PARA LOS FINES PREVISTOS EN EL CONTRATO RESPECTIVO. CUALQUIER USO Y REPRODUCCIÓN TOTAL O PARCIAL NO AUTORIZADA CONSTITUYE UNA VIOLACIÓN DE LOS DERECHOS DE AUTOR DE INTERINOX S.A. GENERADA POR LA LEY</small>	
INGENIERÍA Y FABRICACIÓN DE EQUIPOS INDUSTRIALES		Forma: I - R - P - 15	
CLIENTE: COSTUMER:	INTERINOX		
FABRICANTE: MANUFACTURER:	INTERINOX S.A.		
PROYECTO: PROJECT:			
OBRA: JOB:	MESA DE CORTE CNC		
CONTIENE: CONTAINS:	SOPORTE VERTICAL CARRO TORCHA		
DISEÑO MECANICO: MECHANICAL DESIGN:	INTERINOX S.A.	ESCALA: SCALE:	1 : 2
DIBUJO: DRAWING:	ING. P. POSSO	REVISO: CHECKED:	ING. H. REINOSO
FECHA: DATE:	2012-08-30	APROBO: APPROVED:	
		PLANO No.: DOCUMENT No.:	018-41-28
			REV: 0

## Conexión por Colores Cables de Control

Para la conexión de la tarjeta de control se utilizaron cables de ethernet que se los arma con el siguiente código de color, si sosse tiene el conector y se ve sus colores desde la parte contraria a la pata que lo libera se debe ver los cables en este orden.



(<http://www.archonmagnus.com/computing/elec/diagrams/ethernetCableDiagram.jpg>)

Código en EzDSP F2808	Conector en tarjeta de acople	PIN por conector	Color Cable	Operación Asignada	Conectar a Driver en Señal
GPIO14	1	4	Naranja	Pasos Motor X	Driver (X1 y X2) PUL-
GPIO21		3	Azul	Dirección X 1	Driver X1 DIR-
GPIO0		2	Verde	Dirección X 2	Driver X2 DIR-
+ 5v		5	Café Blanco	Energía	Driver (X1 y X2) PUL+ y DIR+
GPIO2	2	4	Naranja	Pasos Y	Driver Y PUL-
GPIO4		3	Azul	Dirección Y	Driver Y DIR-
GPIO27		2	Verde	Pasos Z	Driver Z PUL-
GPIO13		1	Café	Dirección Z	Driver Z DIR-

Código en EzDSP F2808	Conector en tarjeta de acople	PIN por conector	Color Cable	Operación Asignada	Conectar a Driver en Señal
+ 5v		5	Café Blanco	Energía	Driver (Y y Z) PUL+ y DIR+
GPIO7	3	4	Naranja	Libre para su utilización en actualizaciones o si algún otro puerto se daña	
GPIO16		3	Azul		
GPIO18		2	Verde		
GPIO26		1	Café		
+ 5v		5	Café Blanco	Energía	
GPIO12	4			Antorcha	
ADCINB0	5	6	Azul Blanco	Libre para futuro control de altura automático	
ADCINB1		5	Café Blanco		
ADCREFM		1	Café		
ADCREFP		2	Verde		
ADCINA0		7	Verde Blanco		
ADCINA1		8	Naranja Blanco		
VREFLO		3	Azul		
GND		4	Naranja		

## ANEXO B

### **Referencia técnica de tarjeta EzDSP F2808**

Ver es hoja siguiente

La referencia técnica de la tarjeta está con imágenes reducidas para que puedan entrar en el formato de este documento impreso, para ver los detalles, favor referirse al material digital presentado con este documento.



# **eZdsp™ F2808 USB**

## *Technical Reference*

# eZdsp™ F2808 USB Technical Reference

## IMPORTANT NOTICE

Spectrum Digital, Inc. reserves the right to make changes to its products or to discontinue any product or service without notice. Customers are advised to obtain the latest version of relevant information to verify data being relied on is current before placing orders.

Spectrum Digital, Inc. warrants performance of its products and related software to current specifications in accordance with Spectrum Digital's standard warranty. Testing and other quality control techniques are utilized to the extent deemed necessary to support this warranty.

Please be aware, products described herein are not intended for use in life-support appliances, devices, or systems. Spectrum Digital does not warrant, nor is it liable for, the product described herein to be used in other than a development environment.

Spectrum Digital, Inc. assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does Spectrum Digital warrant or represent any license, either express or implied, is granted under any patent right, copyright, or other intellectual property right of Spectrum Digital, Inc. covering or relating to any combination, machine, or process in which such Digital Signal Processing development products or services might be or are used.

## WARNING

This equipment is intended for use in a laboratory test environment only. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to subpart J of part 15 of FCC rules, which are designed to provide reasonable protection against radio frequency interference. Operation of this equipment in other environments may cause interference with radio communications, in which case the user, at his own expense, will be required to take any measures necessary to correct this interference.

507835-0001 Rev. C  
October 2005

## TRADEMARKS

**eZdsp** is a trademark of Spectrum Digital, Inc.

**SPECTRUM DIGITAL, INC.**  
12502 Exchange Dr., Suite 440 Stafford, TX. 77477  
Tel: 281.494.4505 Fax: 281.494.5310  
sales@spectrumdigital.com www.spectrumdigital.com

Copyright © 2005 Spectrum Digital, Inc.

# Contents

<b>1</b>	<b>Introduction to the eZdsp™ F2808</b>	<b>1-1</b>
	<i>Provides a description of the eZdsp™ F2808, key features, and board outline.</i>	
1.0	Overview of the eZdsp™ F2808	1-2
1.1	Key Features of the eZdsp™ F2808	1-2
1.2	Functional Overview of the eZdsp™ F2808	1-3
<b>2</b>	<b>Operation of the eZdsp™ F2808</b>	<b>2-1</b>
	<i>Describes the operation and various interfaces of the eZdsp™ F2808.</i>	
2.0	The eZdsp™ F2808 Operation	2-2
2.1	The eZdsp™ F2808 Board	2-2
2.1.1	Power Connector	2-3
2.2	eZdsp™ F2808 Memory	2-3
2.2.1	Memory Map	2-4
2.3	eZdsp™ F2808 Connectors	2-5
2.3.1	P1, JTAG Interface	2-7
2.3.2	P3, USB Port/JTAG Interface	2-8
2.3.3	P4, P8, P7, I/O Interface	2-8
2.3.4	P5, P9, Analog Interface	2-10
2.3.5	P6, Power Connector	2-11
2.3.6	P10, RS-232 Connector	2-12
2.3.7	P11, CAN Connector	2-13
2.3.8	J10, SCIB 5 x 2 Connector	2-14
2.3.9	J11, CANB 5 x 2 Connector	2-14
2.3.10	Connector Part Numbers	2-15
2.4	eZdsp™ F2808 Jumpers	2-15
2.4.1	JP4, Voltage Jumper, +3.3/5 Volts for P8, P4	2-16
2.4.2	JP5, ADCREFIN Select	2-17
2.4.3	JP6, GPIO22/GPIO24 Select	2-17
2.5	Switches	2-18
2.5.1	Switch SW1	2-18
2.5.1.1	Switch SW1, Position 1-3, Boot Mode Select	2-19
2.5.1.2	Switch SW1, Position 4, EEPROM Write Enable/Disable	2-19
2.5.1.3	Switch SW1, Position 5, Select EEPROM Pull-ups	2-20
2.5.1.4	Switch SW1, Position 6, Serial EEPROM Address A1	2-20
2.5	Switch SW2	2-21
2.6	LEDs	2-22
2.7	Test Points	2-22
<b>A</b>	<b>eZdsp™ F2808 Schematics</b>	<b>A-1</b>
	<i>Contains the schematics for the socketed and unsocketed versions of the eZdsp™ F2808</i>	
<b>B</b>	<b>eZdsp™ F2808 Mechanical Information</b>	<b>B-1</b>
	<i>Contains the mechanical information about the socketed and unsocketed versions of the eZdsp™ F2808</i>	

## List of Figures

Figure 1-1, Block Diagram eZdsp™ F2808	1-3
Figure 2-1, eZdsp™ F2808 PCB Outline	2-2
Figure 2-2, eZdsp™ F2808 Memory Space	2-4
Figure 2-3, eZdsp™ F2808 Connector and Switch Positions	2-6
Figure 2-4, Connector P1 Pin Locations	2-7
Figure 2-5, Connector P8 Connector	2-8
Figure 2-6, Connector P5/P9 Pin Locations	2-10
Figure 2-7, Connector P6 Location	2-11
Figure 2-8, eZdsp™ F2808 Power Connector	2-11
Figure 2-9, P10, DB9 Female Connector	2-12
Figure 2-10, P11, DB9 Female Connector	2-12
Figure 2-11, eZdsp™ F2808 Jumper Positions (Bottom Side)	2-16
Figure 2-12, JP4 Layout	2-16
Figure 2-13, JP6 Layout	2-17
Figure 2-14, SW1 Layout	2-18
Figure 2-15, SW2 Layout	2-21

## List of Tables

Table 2-1, eZdsp™ F2808 Connectors	2-5
Table 2-2, P1, JTAG Interface Connector	2-7
Table 2-3, P4/P8, I/O Connectors	2-9
Table 2-4, P5/P9, Analog Interface Connector	2-10
Table 2-5, P10, RS-232 Pinout	2-12
Table 2-6, P11, CANA Pinout	2-13
Table 2-7, J10, 5 x 2 Pinout	2-14
Table 2-8, J11, 5 x 2 Pinout	2-14
Table 2-9, eZdsp™ F2808 Suggested Connector Part Numbers	2-15
Table 2-10, eZdsp™ F2808 Jumpers	2-15
Table 2-11, JP4, +3.3/5 Volts to P8	2-17
Table 2-12, JP6, GPIO22/GPIO24 Select	2-18
Table 2-13, SW1, Switch Positions	2-18
Table 2-14, SW1, Positions 1-3	2-19
Table 2-15, SW1, Position 4	2-19
Table 2-16, SW1, Position 5	2-20
Table 2-17, SW1, Position 6	2-20
Table 2-18, SW2, Switch Positions	2-21
Table 2-19, LEDs	2-22
Table 2-20, Test Points	2-22

## About This Manual

This document describes board level operations of the eZdsp™ F2808 based on the Texas Instruments TMS320F2808 Digital Signal Processor.

The eZdsp™ F2808 is a stand-alone module permitting engineers and software developers evaluation of certain characteristics of the TMS320F2808 DSP to determine processor applicability to design requirements. Evaluators can create software to execute onboard or expand the system in a variety of ways.

## Notational Conventions

This document uses the following conventions.

The “eZdsp™ F2808” will sometimes be referred to as the “eZdsp”.

“eZdsp” will include the socketed or unsocketed version

Program listings, program examples, and interactive displays are shown in a special italic typeface. Here is a sample program listing.

*equations*

*!rd = !stroke&nr;*

## Information About Cautions

This book may contain cautions.

**This is an example of a caution statement.**

A caution statement describes a situation that could potentially damage your software, hardware, or other equipment. The information in a caution is provided for your protection. Please read each caution carefully.

## Related Documents

Texas Instruments TMS320F28x DSP CPU and Instruction Set Reference Guide, literature #SPRU430

Texas Instruments TMS320F28x Assembly Language Tools Users Guide, literature #SPRU513

Texas Instruments TMS320F28x Optimizing C/C++ Compiler User's Guide, literature #SPRU514

Texas Instruments Code Composer Studio Getting Started Guide, literature #SPRU509

**Table 1: Manual History**

Revision	History
A	Preliminary Release
B	Production Release
C	Corrected Table in Chapter 2

**Table 2: Board History**

Revision	History
A	Prototype Release
B	Production Release

# Chapter 1

## Introduction to the eZdsp™ F2808

This chapter provides a description of the eZdsp™ for the TMS320F2808 Digital Signal Processor, key features, and block diagram of the circuit board.

Topic	Page
1.0 Overview of the eZdsp™ F2808	1-2
1.1 Key Features of the eZdsp™ F2808	1-2
1.2 Functional Overview of the eZdsp™ F2808	1-3

### 1.0 Overview of the eZdsp™ F2808

The eZdsp™ F2808 is a stand-alone card--allowing evaluators to examine the TMS320F2808 digital signal processor (DSP) to determine if it meets their application requirements. Furthermore, the module is an excellent platform to develop and run software for the TMS320F2808 processor.

The eZdsp™ F2808 is shipped with a TMS320F2808 DSP. The eZdsp™ F2808 allows full speed verification of F2808 code. Expansion connectors are provided for any necessary evaluation circuitry not provided on the as shipped configuration.

To simplify code development and shorten debugging time, a C2000 Tools Code Composer driver is provided. In addition, an onboard JTAG connector provides interface to emulators, operating with other debuggers to provide assembly language and 'C' high level language debug.

### 1.1 Key Features of the eZdsp™ F2808

The eZdsp™ F2808 has the following features:

- TMS320F2808 Digital Signal Processor
- 100 MIPS operating speed
- 18K words on-chip zero wait state SARAM
- 64K words on-chip Flash memory
- 256K bits serial I<sup>2</sup>C EEPROM memory
- 20 MHz. clock
- Expansion Connectors (analog, I/O)
- Onboard IEEE 1149.1 JTAG Controller
- 5-volt only operation with supplied AC adapter
- TI F28xx Code Composer Studio tools driver
- On board USB JTAG emulation connector
- 2 SCI UART channels
- 2 eCAN channels

### 1.2 Functional Overview of the eZdsp™ F2808

Figure 1-1 shows a block diagram of the basic configuration for the eZdsp™ F2808. The major interfaces of the eZdsp are the JTAG interface, and expansion interface.

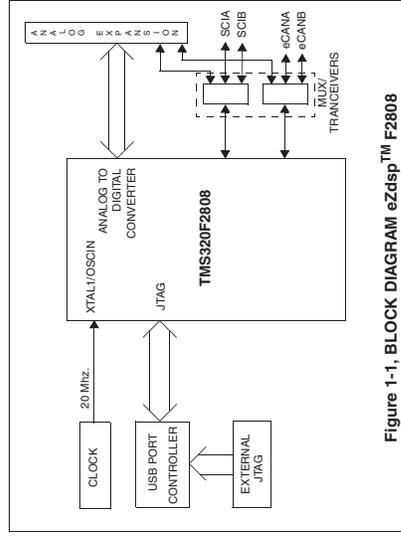


Figure 1-1. BLOCK DIAGRAM eZdsp™ F2808

# Chapter 2

## Operation of the ezdsp™ F2808

### 2.0 The ezdsp™ F2808 Operation

This chapter describes the ezdsp™ F2808, key components, and operation. Information on the ezdsp's various interfaces is also included. The ezdsp™ F2808 consists of four major blocks of logic:

- Analog Interface Connector
- I/O Interface Connector
- JTAG Interface
- USB Port/JTAG Controller Interface

### 2.1 The ezdsp™ F2808 Board

The ezdsp™ F2808 is a 5.25 x 3.0 inch, multi-layered printed circuit board, powered by an external 5-Volt only power supply. Figure 2-1 shows the layout of the top side of the ezdsp™ F2808.

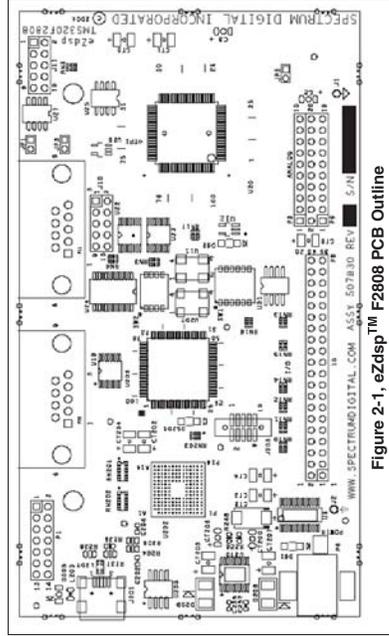


Figure 2-1, ezdsp™ F2808 PCB Outline

Topic	Page
2.0 The ezdsp™ F2808 Operation	2-2
2.1 The ezdsp™ F2808 Board	2-2
2.1.1 Power Connector	2-3
2.2 ezdsp™ F2808 Memory	2-3
2.2.1 Memory Map	2-4
2.3 ezdsp™ F2808 Connectors	2-5
2.3.1 P1, JTAG Interface	2-7
2.3.2 P3, USB Port/JTAG Interface	2-8
2.3.3 P8, I/O Interface	2-8
2.3.4 P5,P9, Analog Interface	2-10
2.3.5 P6, Power Connector	2-11
2.3.6 P10, RS-232 Connector	2-12
2.3.7 P11, CAN Connector	2-13
2.3.8 J10, SCIB 5 x 2 Connector	2-14
2.3.9 J11, CANB 5 x 2 Connector	2-14
2.3.10 Connector Part Numbers	2-15
2.4 ezdsp™ F2808 Jumpers	2-15
2.4.1 JP4, Voltage Jumper, +3.3/5 Volts for P8, P4	2-16
2.4.2 JP5, ADCREFIN Select	2-17
2.4.3 JP6, GPIO22/GPIO24 Select	2-17
2.5 Switches	2-18
2.5.1 Switch SW1	2-18
2.5.1.1 Switch SW1, Position 1-3, Boot Mode Select	2-19
2.5.1.2 Switch SW1, Position 4, EEPROM Write Enable/Disable	2-19
2.5.1.3 Switch SW1, Position 5, Select EEPROM Pull-ups	2-20
2.5.1.4 Switch SW1, Position 6, Serial EEPROM Address A1	2-20
2.5.2 Switch SW2	2-21
2.6 LEDs	2-22
2.7 Test Points	2-22

This chapter describes the operation of the ezdsp™ F2808, key interfaces and includes a circuit board outline.

### 2.1.1 Power Connector

The eZdsp™ F2808 is powered by a 5-Volt only power supply, included with the unit. The unit requires 500mA. The power is supplied via connector P6. If expansion boards are connected to the eZdsp, a higher amperage power supply may be necessary. Section 2.3.6 provides more information on connector P6.

### 2.2 eZdsp™ F2808 Memory

The eZdsp includes the following on-chip memory:

- 2 blocks of 4K x 16 single access RAM (SARAM)
- 1 block of 8K x 16 SARAM
- 2 blocks of 1K x 16 SARAM

The eZdsp can load ram for debug, or Flash can be programmed and run. For larger software projects it is suggested to do a initial debug with on eZdsp F2808 module which supports a total RAM environment. With careful attention to the I/O mapping in the software the application code can easily be ported to the F2808.

### 2.2.1 Memory Map

The figure below shows the memory map configuration on the eZdsp™ F2808.

Block Start Address	Data Space	Program Space
0x0000-0000	M0 SARAM	
0x0000-0400	M1 SARAM	
0x0000-0800	Peripheral Frame 0	
0x0000-0D00	Pie Vector Table (256 x 16)	Do Not Use !
0x0000-0E00		
0x0000-8000	Peripheral Frame 1	
0x0000-7000	Peripheral Frame 2 (16 Bit access only)	Do Not Use !
0x0000-8000	L0 SARAM	
0x0000-9000	L1 SARAM	
0x0000-A000	H0 SARAM	
0x0000-C000		
0x003D-7800		OTP (Read Only)
0x003D-7800		
0x003E-8000		Flash (Read Only) (64K x 16)
0x003F-7FFF		
0x003F-8000	L0 SARAM Mirror	
0x003F-9000	L1 SARAM Mirror	
0x003F-A000	H0 SARAM Mirror	
0x003F-C000		
0x003F-F000		Boot ROM (Read Only)
0x003F-FFFF		

Figure 2-2, eZdsp™ F2808 Memory Space

### 2.3 eZdsp™ F2808 Connectors

The eZdsp™ F2808 has five connectors. Pin 1 of each connector is identified by a square solder pad. The function of each connector is shown in the table below:

**Table 1: eZdsp™ F2808 Connectors**

Connector	Function
P1	JTAG Interface
P8	I/O Interface
P5/P9	Analog Interface
P6	Power Connector
P10	DB-9, RS-232
P11	DB9, eCAN-A
J10	2x5 Header, SCI-B
J11	2x5 header, eCAN-B
J201	USB Controller Interface

The diagram below shows the position of each connector

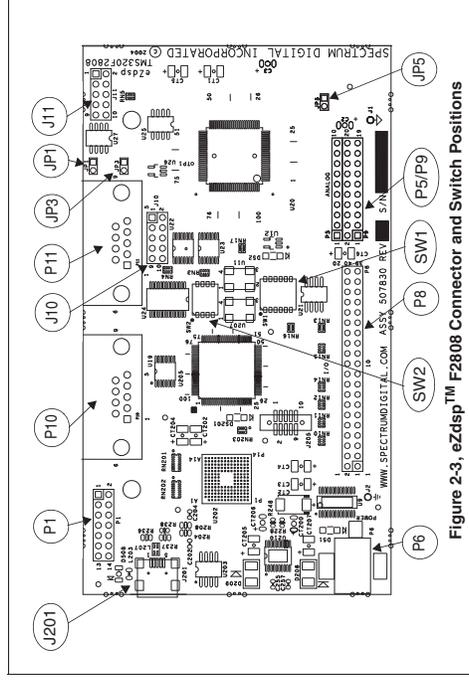
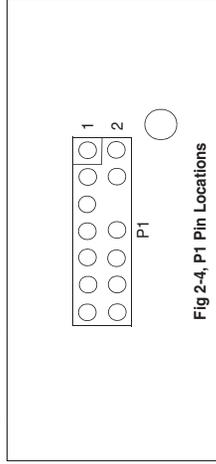


Figure 2-3, eZdsp™ F2808 Connector and Switch Positions

**2.3.1 P1, JTAG Interface**

The eZdsp™ F2808 is supplied with a 14-pin header interface, P1. This is the standard interface used by JTAG emulators to interface to Texas Instruments DSPs. The positions of the 14 pins on the P1 connector are shown in the diagram below as viewed from the top of the eZdsp.



**Fig 2-4, P1 Pin Locations**

The definition of P1, which has the JTAG signals is shown below.

**Table 2: P1, JTAG Interface Connector**

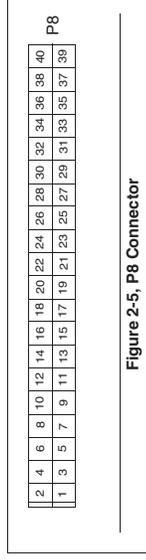
Pin #	Signal	Pin #	Signal
1	TMS	2	TRST-
3	TDI	4	GND
5	PD (+5V)	6	no pin
7	TDO	8	GND
9	TCK-RET	10	GND
11	TCK	12	GND
13	EMU0	14	EMU1

**2.3.2 P3, USB Port/JTAG Interface**

The eZdsp™ F2808 uses a custom USB port JTAG interface device. The device has direct access to the integrated JTAG interface. Drivers for C2000 Code Composer tools are shipped with the eZdsp modules

**2.3.3 P8, I/O Interface**

The connector P8 presents the I/O signals from the DSP. The layout of this connector is shown below.



**Figure 2-5, P8 Connector**

The pin definition of P4/P8 connectors are shown in the table below.

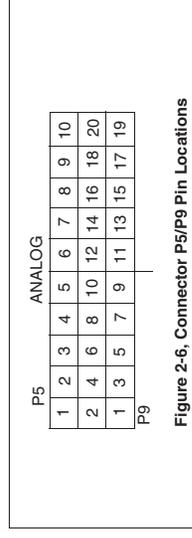
**Table 3: P4/P8, I/O Connectors**

P8 Pin #	P8 Signal	P8 Pin #	P8 Signal
1	+3.3V/+5V/NC *	2	+3.3V/+5V/NC *
3	MUX_GPIO29	4	MUX_GPIO28
5	GPI014	6	GPI020
7	GPI021	8	GPI023
9	GPI00	10	GPI01
11	GPI02	12	GPI03
13	GPI04	14	GPI05
15	GPI027	16	GPI06
17	GPI013	18	GPI034
19	GND	20	GND
21	GPI07	22	GPI015
23	GPI016	24	GPI017
25	GPI018	26	GPI019
27	MUX_GPIO31	28	MUX_GPIO30
29	MUX_GPIO11	30	MUX_GPIO8
31	MUX_GPIO9	32	MUX_GPIO10
33	GPI022/GPI024	34	GPI025
35	GPI026	36	GPI032
37	GPI012	38	GPI033
39	GND	40	GND

\* Default is No Connect (NC). User can jumper to +3.3V or +5V on backside of ezdsp with JP4.

**2.3.4 P5/P9, Analog Interface**

The position of the 30 pins on the P5/P9 connectors are shown in the diagram below as viewed from the top of the ezdsp.



**Figure 2-6, Connector P5/P9 Pin Locations**

The definition of P5/P9 signals are shown in the table below.

**Table 4: P5/P9, Analog Interface Connector**

P5 Pin #	Signal	P9 Pin #	Signal	P9 Pin #	Signal
1	ADCINB0	1	GND	2	ADCINA0
2	ADCINB1	3	GND	4	ADCINA1
3	ADCINB2	5	GND	6	ADCINA2
4	ADCINB3	7	GND	8	ADCINA3
5	ADCINB4	9	GND	10	ADCINA4
6	ADCINB5	11	GND	12	ADCINA5
7	ADCINB6	13	GND	14	ADCINA6
8	ADCINB7	15	GND	16	ADCINA7
9	ADCREFM	17	GND	18	VREFLO *
10	ADCREFP	19	GND	20	No connect

\* Connect VREFLO to AGND or VREFLO of target system for proper ADC operation.

2.3.5 P6, Power Connector

Power (5 volts) is brought onto the eZdsp™ F2808 via the P6 connector. The connector has an outside diameter of 5.5 mm, and an inside diameter of 2.5 mm. The position of the P6 connector is shown below.

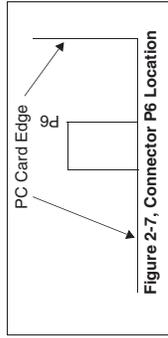


Figure 2-7, Connector P6 Location

The diagram of P6, which has the input power is shown below.

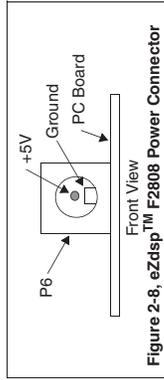


Figure 2-8, eZdsp™ F2808 Power Connector

2.3.6 P10, RS-232 Connector

The eZdsp F2808 has an RS-232 connector which brings out the SCIA transmit and receive signals to be used as UART. This UART uses the MAX3238 RS-232 line driver and is routed to a male 9 pin D-connector, P10. The pin positions for the P10 connector as viewed from the edge of the printed circuit board are shown below.

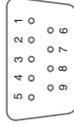


Figure 2-9, P10, DB9 Female Connector

The pin numbers and their corresponding signals are shown in the table below. This corresponds to a standard dual row to DB-9 connector interface used on personal computers.

Table 5: P10, RS-232 Pinout

Pin #	Signal Name	Direction
1	No Connect	
2	PCRXDA	Out
3	PCTXDA	In
4	No Connect	
5	GND	N/A
6	No Connect	
7	No Connect	
8	No Connect	
9	No Connect	

### 2.3.7 P11, CAN Connector

The eZdsp F2808 has a 9 Pin female D-connector which brings out the CANA transmit and receive signals. This CAN interface uses the SN65HVD235 CAN driver. The pin positions for the P11 connector as viewed from the edge of the printed circuit board are shown below.



**Figure 2-10, P11, DB9 Female Connector**

The pin numbers and their corresponding signals are shown in the table below.

**Table 6: P11, CANA Pinout**

Pin #	Signal Name
1	No Connect
2	CANLA
3	GND
4	No Connect
5	No Connect
6	No Connect
7	CANHA
8	No Connect
9	No Connect

### 2.3.8 J10, SCI B 5 x 2 Header

The SCI B signals are routed through the MAX3238 line driver to a 5 x 2 double row header, J10. The pin numbers for J10 and their corresponding signals are shown in the table below.

**Table 7: J10, 5 x 2 Pinout**

Pin #	Signal Name	Direction	Pin #	Signal Name	Direction
1	No Connect		2	No Connect	
3	PCRXDB	Out	4	No Connect	
5	PCTXDB	In	6	No Connect	
7	No Connect		8	No Connect	
9	GND	N/A	10	No Connect	

### 2.3.9 J11, CANB 5 x 2 Header

The CANB signals are routed through the SN65HVD235 CAN driver to a 5 x 2 double row header, J11. The pin numbers for J11 and their corresponding signals are shown in the table below.

**Table 8: J11, 5 x 2 Pinout**

Pin #	Signal Name	Pin #	Signal Name
1	No Connect	2	No Connect
3	CANLB	4	CANHB
5	GND	6	No Connect
7	No Connect	8	No Connect
9	No Connect	10	No Connect

2.3.10 Connector Part Numbers

The table below shows the part numbers for connectors which can be used on the eZdsp™ F2808. Part numbers from other manufacturers may also be used.

Table 9: eZdsp™ F2808 Suggested Connector Part Numbers

Connector	Male Part Numbers	Female Part Numbers
P1	SAMTEC TSW-1-10-07-G-T	SAMTEC SSW-1-10-01-G-T
P2	SAMTEC TSW-1-20-07-G-T	SAMTEC SSW-1-20-01-G-T

\*SSW or SSQ Series can be used

2.4 eZdsp™ F2808 Jumpers

The eZdsp™ F2808 has 2 jumpers, JP4 and JP5. JP4 will allow power to be supplied to the expansion headers. Jumper JP5 selects the ADCREFIN. The table below lists the jumpers and their function. The following sections describe the use of each jumper.

Table 10: eZdsp™ F2808 Jumpers

Jumper #	Size	Function	Position As Shipped From Factory
JP1	1 x 2	Terminator Resistor - CANB	Installed
JP3	1 x 2	Terminator Resistor - CANA	Installed
JP4	1 x 3	+3.3/5 Volts to P8 Pin 1,2 and P4, Pin 1	Not connected
JP5	1 x 2	Selects ADCREFIN voltage	Not connected
JP6	1 x 3	Selects GPIO22 or GPIO24 to Pin 33, P8	2 - 3, GPIO24 to Pin 33, P8

2.4.1 JP4, Voltage Jumper, +3.3/5 Volts for P8, P4

Jumpers JP4 is an unpopulated jumper on the bottom side of the board that provide either +3.3 volts or +5 volts to pins on the expansion connector. This jumper is shipped uninstalled to prevent accidental damage by connecting wires or circuitry to the expansion connector. The user may connect the jumper by installing a jumper wire or zero ohm resistor. The position of this jumper is shown in the figure below.

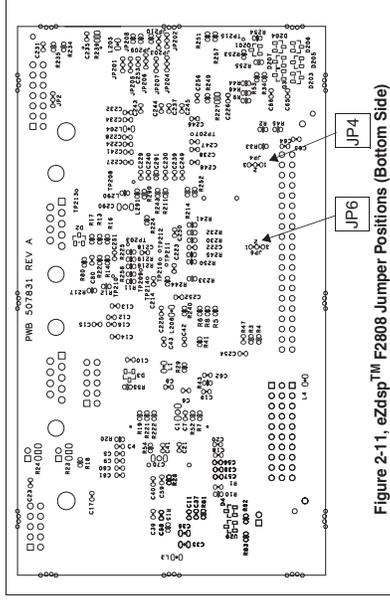


Figure 2-11, eZdsp™ F2808 Jumper Positions (Bottom Side)

Jumper JP4 allows the user to provide either +3.3 or +5 volts to pins 1 and 2 of expansion connector P8.

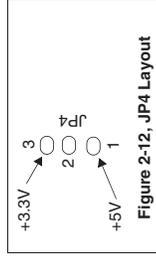


Figure 2-12, JP4 Layout

The table below shows the functions of the two positions of JP4.

Table 11: JP4, Voltage Jumper,+3.3/5 Volts for P8, P4

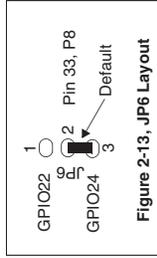
Position	Function
1 - 2	+5V connected to P8
2 - 3	+3.3 connected to P8

**2.4.2 JP5, ADCREFIN Select**

Jumper J5 is used to select ADCREFIN as the voltage reference for the analog to digital converter. When the jumper is shorted the +2.048 voltage level is routed to the ADCREFIN signal of the DSP. When the jumper is open the ADCREFIN floats.

**2.4.3 JP6, GPIO22/GPIO24 Select**

Jumper JP6 selects which signal, GPIO22 or GPIO24, to be routed to Pin 33, P8. When the 1-2 jumper position is selected GPIO22 is connected to pin 33, P8. The 2-3 selection will route GPIO24 to Pin 33, P8. The figure below shows the connector layout on the bottom side of the board.



The table below shows the functions of the two positions of JP6.

**Table 12: JP6, GPIO22/GPIO24 Select**

Position	Function
1 - 2	GPIO22 connected to Pin 33, P8
2 - 3	GPIO24 connected to Pin 33, P8 *

\* default

**2.5 Switches**

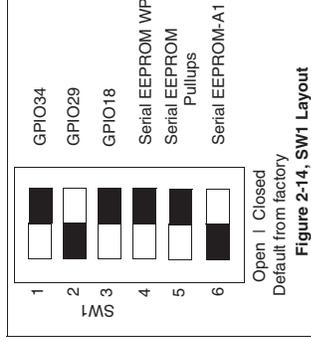
The eZdsp F2808 has 2 switches, SW1 and SW2, which are used to configure various functions on the board. The switches and their functionality are described in the following sections.

**2.5.1 Switch SW1**

The eZdsp F2808 has a 6 position switch, SW1, that allows the user to configure the board for their application. The function of each position on SW1 is shown in the table below.

**Table 13: SW1 Switch Positions**

Position	Function
1-3	Boot Mode Select
4	Serial EEPROM WP
5	Serial EEPROM Pullups
6	Serial EEPROM - A1



### 2.5.1.1 Switch SW1, Positions 1-3, Boot Mode Select

Positions 1-3 on switch SW1 are used to determine what mode the DSP will use for bootloading on power up. The options are shown in the table below.

Table 14: SW1, Positions 1-3

Position 3 GPIO18	Position 2 GPIO29	Position 1 GPIO34	Boot Mode
Open-1	Open-1	Open-1	Flash
Open-1	Open-1	Closed-0	SCH-A
Open-1	Closed-0	Open-1	SPI-A
Open-1	Closed-0	Closed-0	I <sup>2</sup> C-A
Closed-0	Open-1	Open-1	eCAN-A
Closed-0	Open-1	Closed-0	MD-SARAM *
Closed-0	Closed-0	Open-1	OTP
Closed-0	Closed-0	Closed-0	I/O

\* factory default

### 2.5.1.2 Switch SW1, Position 4, EEPROM Write Enable/Disable

Position 4 on switch SW1 is used to enable or disable the Write Enable for the EEPROM. When position 4 is in the "Closed" state the Write Protect to the EEPROM is disabled, therefore allowing the EEPROM to be written. When position 4 is in the "Open" state the EEPROM Write Protect is enabled, not allowing the EEPROM to be written. These positions are shown in the table below.

Table 15: SW1, Position 4

Position	Function
Closed-0 *	EEPROM WP Disabled (EEPROM can be written to) *
Open-1	EEPROM WP Enabled (EEPROM cannot be written to)

\* factory default

### 2.5.1.3 Switch SW1, Position 5, Serial EEPROM Pull-ups

Position 5 on switch SW1 are used to select if the EEPROM is using pull up resistors. When position 5 is in the "Closed" state pull up resistors are used. When position 5 is in the "Open" state the pull-up resistors are not used. These positions are shown in the table below.

Table 16: SW1, Position 5

Position	Function
Closed-1 *	Pull up resistors used *
Open-0	Pull up resistors not used

\* factory default

### 2.5.1.4 Switch SW1, Position 6, Serial EEPROM Address A1

Position 6 on switch SW1 are used to select if the address line A1 to the serial EEPROM is pulled high or low. When position 6 is in the "Closed" state A1 equals 1. When position 6 is in the "Open" state A1 equals 0. These positions are shown in the table below.

Table 17: SW1, Position 6

Position	Function
Closed-1	A1 = 1
Open-0	A1 = 0 *

\* factory default

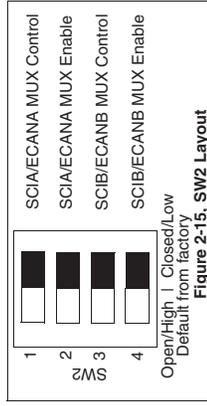
### 2.5.2 Switch SW2

Switch SW2 on the eZdsp F2808 has 4 positions that allow the user to configure the SCI/CAN MUX. The function of each position is shown in the table below.

**Table 18: SW2 Switch Positions**

Position	Function	Settings
1	SCIA/ECANA MUX Control	Open/High GPIO28-GPIO31 to expansion connector P8 Closed/Low GPIO28-GPIO31 to SCIA/CANA transceivers
2	SCIA/ECANA MUX Enable	Open/High GPIO28-GPIO31 isolated Closed/Low GPIO28-GPIO31 not isolated
3	SCIB/ECANB MUX Control	Open/High GPIO8-GPIO11 to expansion connector P8 Closed/Low GPIO8-GPIO11 to SCIB/CANB transceivers
4	SCIB/ECANB MUX Enable	Open/High GPIO8-GPIO11 isolated Closed/Low GPIO8-GPIO11 not isolated

The layout of switch SW2 is shown in the figure below.



### 2.6 LEDs

The eZdsp™ F2808 has two light-emitting diodes. DS1 indicates the presence of +5 volts and is normally 'on' when power is applied to the board. DS2 is under software control and is tied to the GPIO34 pin on the DSP through a buffer. These are shown in the table below.

**Table 19: LEDs**

LED #	Color	Controlling Signal
DS1	Green	+5 Volts
DS2	Green	GPIO34 bit (GPIO34 high = on)

### 2.7 Test Points

The eZdsp™ F2808 has two test points. The signals they are tied to are shown in the table below.

**Table 20: Test Points**

Test Point	Signal
J1	Analog Ground
J2	Ground

# Appendix A

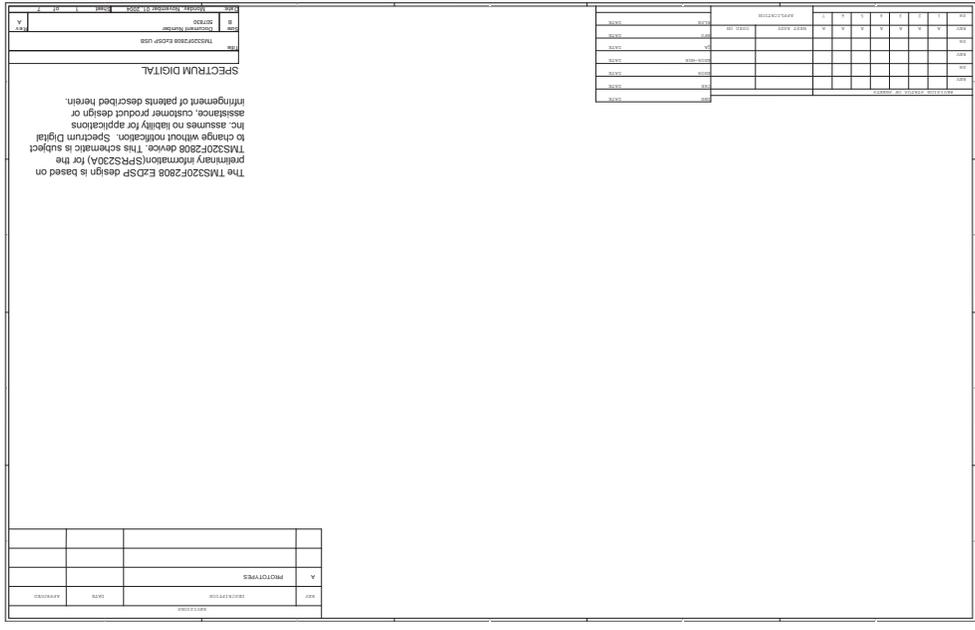
## eZdsp™ F2808 Schematics

The schematics for the eZdsp™ F2808 can be found on the CD-ROM that accompanies this board. The schematics were drawn on ORCAD.

**WARNING !**

The TMS320F2808 supports +3.3V Input/Output levels which are NOT +5V tolerant. Connecting the eZdsp to a system with +5V Input/Output levels will damage the TMS320F2808. If the eZdsp is connected to another target then the eZdsp must be powered up first and powered down last to prevent latchup conditions.

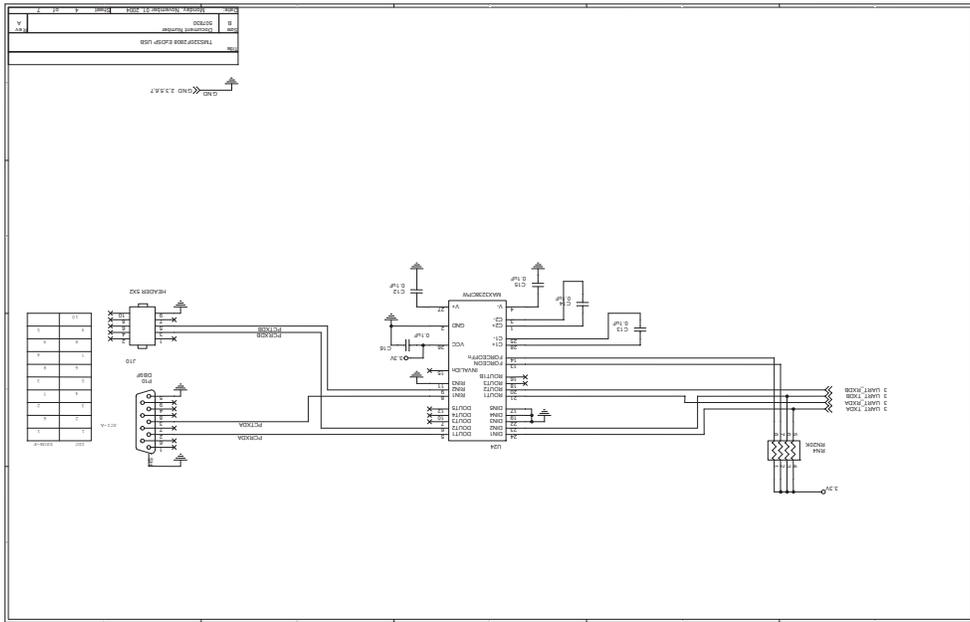
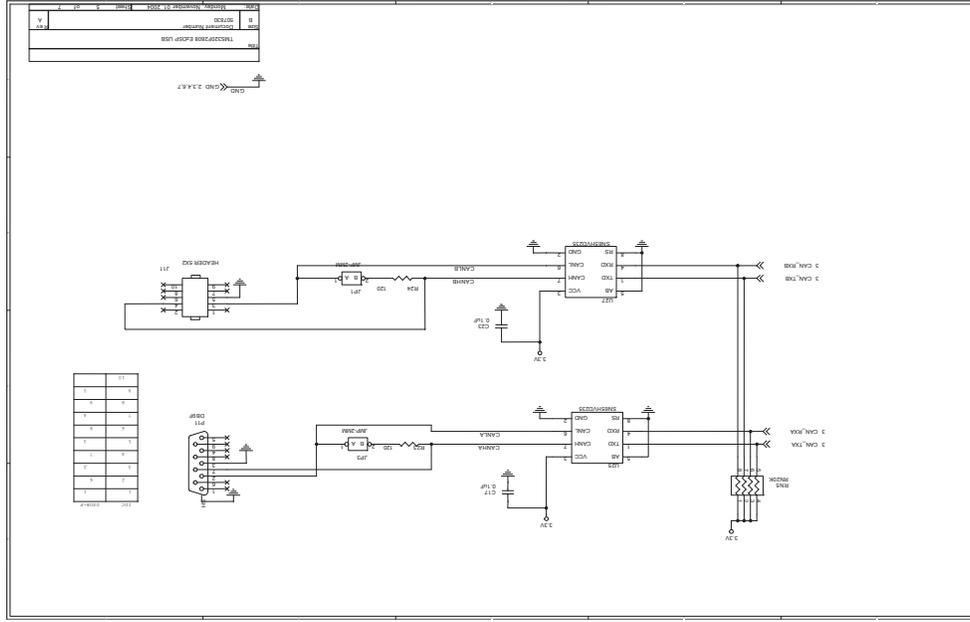
Spectrum Digital, Inc

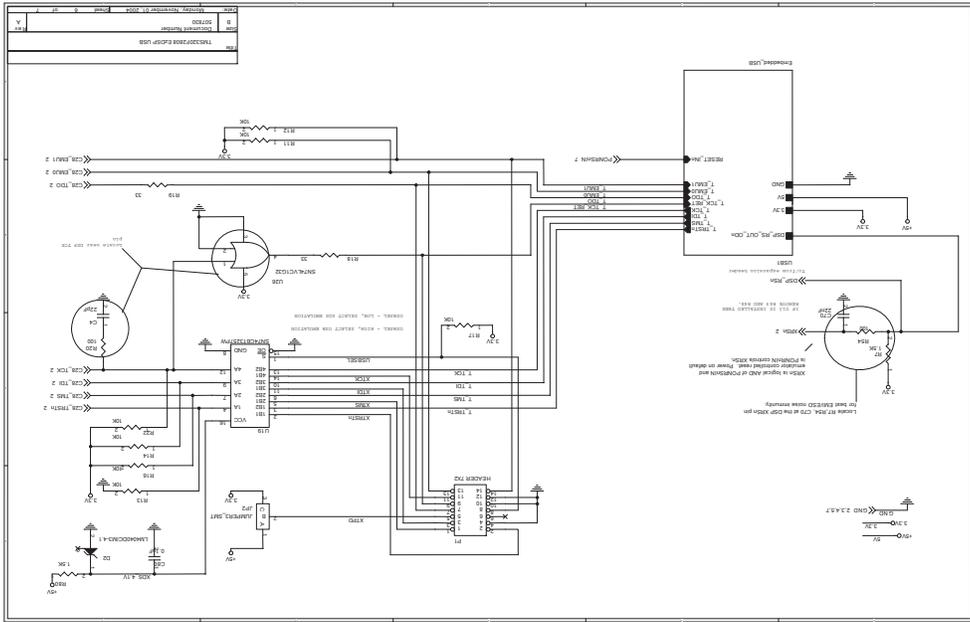
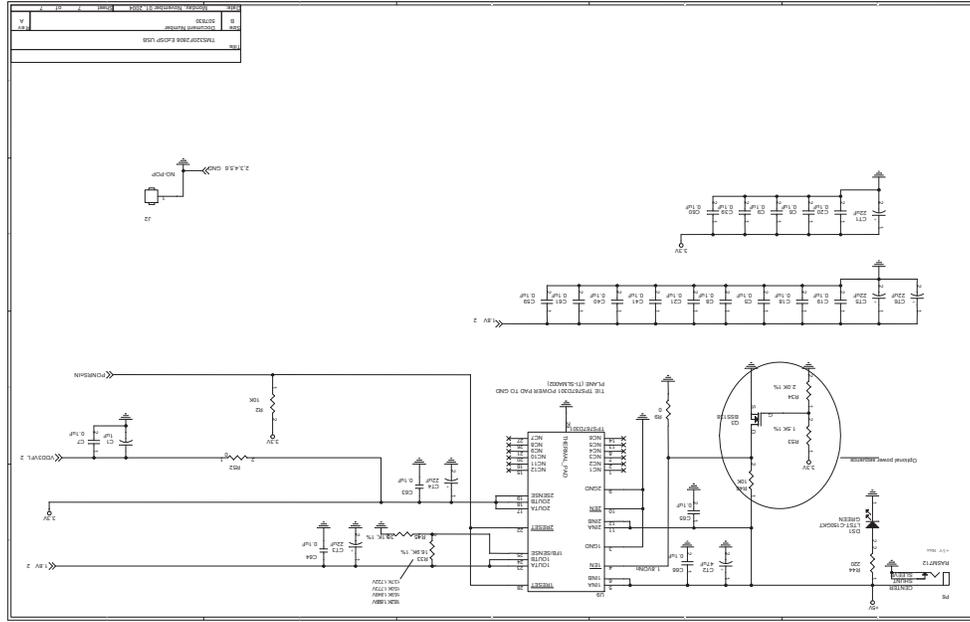


A-2 eZdsp™ F2808 Technical Reference

A-1









Spectrum Digital, Inc

Spectrum Digital, Inc

B-3

B-4

eZdsp™ F2808 Technical Reference

---



**SPECTRUM**  
*DIGITAL*  
INCORPORATED  
Printed in U.S.A. © 06/07/2005  
507635-0001 Rev. C

---



**THE GREEN MONSTER**

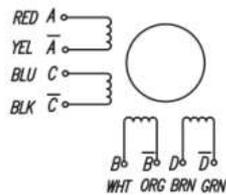


# HT23-400-8

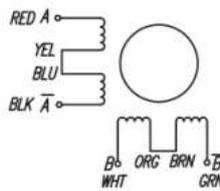
High Torque Stepper Motor  
 400 OzIn Hybrid Design  
 1.8 deg / 200 Steps Per Revolution

	Parallel	Series	UniPolar
Holding Torque (N.m +-10%)	2.8	2.8	2.0
Rated Current (Amps/phase)	4.2	2.1	3.0
Resistance (ohm/phase +-10%)	0.8	0.8	1.6
Inductance (mH/phase +-20%)	3.8	15.2	3.8
Rotor Inertia (g.cm <sup>2</sup> )	480		
Motor Weight (kg)	2.0		
Motor Length (mm)	112		
Number of Wires	8		

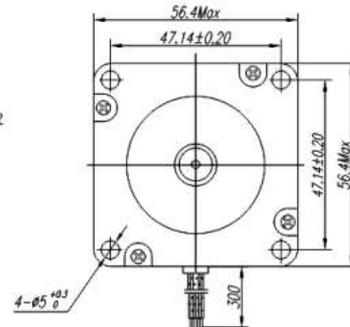
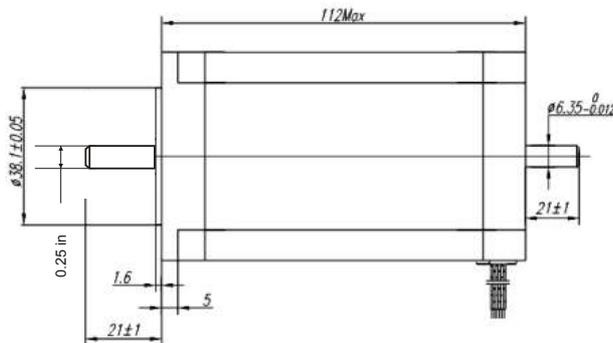
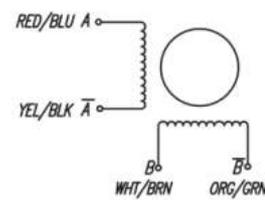
UNIPOLAR RESP. ONE WINDING



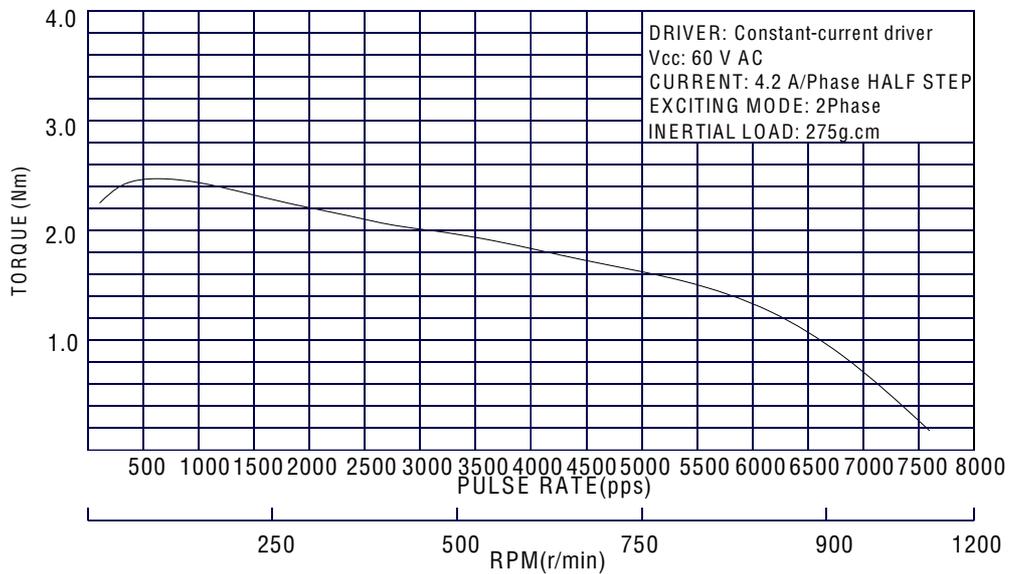
BIPOLAR RESP. BIPOLAR SERIAL



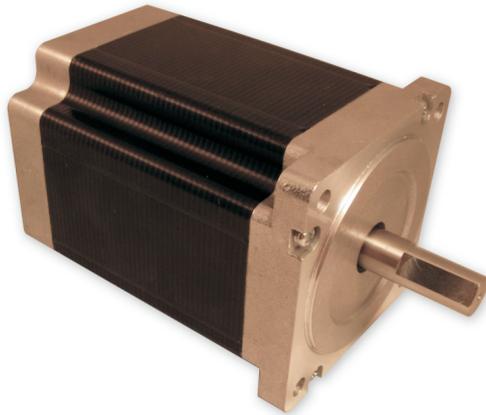
BIPOLAR ONLY PARALLEL



HT23-400-8 Torque Curve



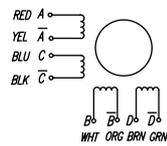
## DataSheet motores



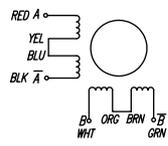
### HT34-1150-8 High Torque Stepper Motor 1150 OzIn Hybrid Design 1.8 deg / 200 Steps Per Revolution NEMA34 Frame Size 0.5" Shaft w/ Flat

	Parallel	Series	UniPolar
Holding Torque (OzIn)	1150	1150	820
Rated Current (Amps/phase)	6.0	3.0	4.2
Resistance (ohm/phase +-10%)	0.55	2.2	1.1
Inductance (mH/phase +-20%)	5.5	22	5.5
Motor Weight (Lbs)	8.25		
Motor Length (Inches)	4.5		
Number of Wires	8		

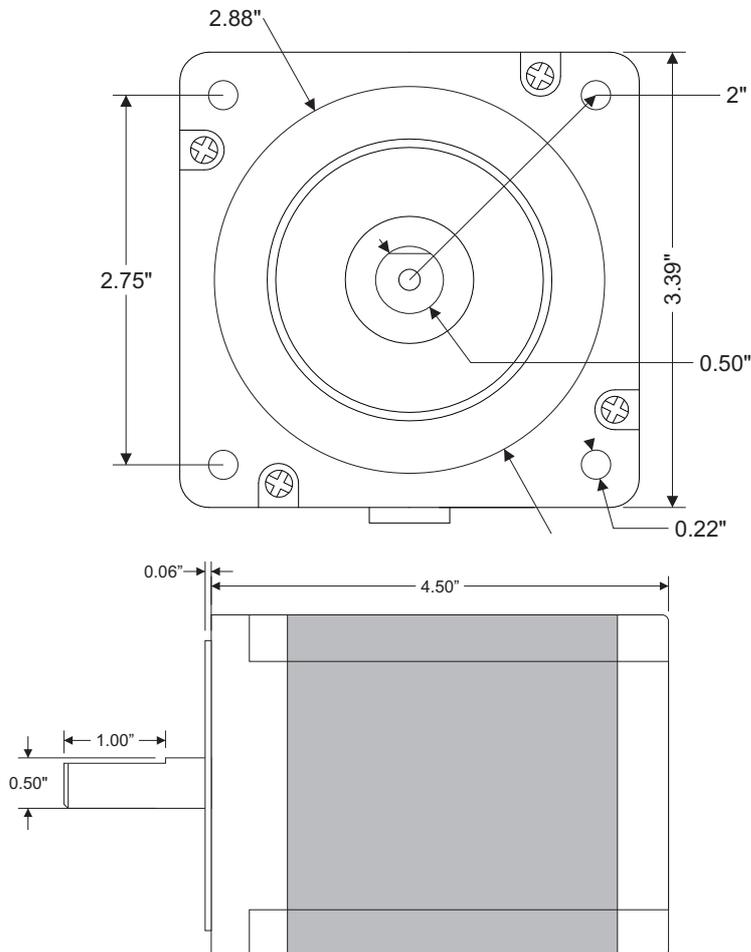
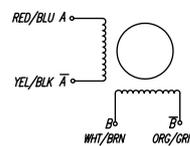
UNIPOLAR RESP. ONE WINDING



BIPOLAR RESP. BIPOLAR SERIAL



BIPOLAR ONLY PARALLEL



## **Datasheet Drivers**

Ver hojas siguientes

Los Datasheets están con imágenes reducidas para que puedan entrar en el formato de este documento impreso, para ver los detalles, favor referirse al material digital presentado con este documento.



## High Performance Microstepping Driver



## User's Manual

Version 1.0

©2010 PROBOTIX All Rights Reserved

Attention: Please read this manual carefully before using the driver!

## Table of Contents

1. Introduction, Features and Applications.....	1
Introduction.....	1
Features.....	1
Applications.....	13
Specifications.....	13
Electrical Specifications (Tj = 25°C/77°F).....	13
Operating Environment and other Specifications.....	13
Mechanical Specifications (unit: mm[inch]).....	13
Elimination of Heat .....	14
3. Pin Assignment and Description.....	14
Connector P1 Configurations.....	15
Selecting Effective Pulse Edge or Effective Level and Control Signal Mode	15
Connector P2 Configurations.....	16
4. Control Signal Connector (P1) Interface.....	16
5. Connecting the Motor.....	17
Connections to 4-lead Motors.....	17
Connections to 6-lead Motors.....	17
Half Coil Configurations.....	18
Full Coil Configurations.....	18
Connections to 8-lead Motors.....	18
Series Connections.....	18
Parallel Connections.....	19
6. Power Supply Selection.....	19
Regulated or Unregulated Power Supply.....	19
Multiple Drivers.....	20
Selecting Supply Voltage.....	20
7. Selecting Microstep Resolution and Driver Output Current.....	20
Microstep Resolution Selection.....	21

# 1. Introduction, Features and Applications

## Introduction

The MondoStep 7.8 is a high performance microstepping driver based on pure-sinusoidal current control technology. Owing to the above technology and the self-adjustment technology (self-adjusting current control parameters) according to different motors, the driven motors can run with less noise, lower heating, smoother motion and have better performance at higher speed than most other drivers in the market. It is suitable for driving 2-phase and 4-phase hybrid stepping motors.

### Features:

- High performance, cost-effective
- Supply voltage up to +80VDC
- Output current up to 7.8A
- Self-adjustment technology
- Pure-sinusoidal current control technology
- Pulse input frequency up to 300 KHz
- TTL compatible and optically isolated input
- Automatic idle-current reduction
- 16 selectable resolutions in decimal and binary, up to 51,200 steps/rev
- Suitable for 2-phase and 4-phase motors
- Support PUL/DIR and CW/CCW modes
- Short-voltage, over-voltage, over-current and short-circuit protection

Current Settings.....	21
Dynamic current setting.....	22
Standstill current setting.....	22
8. Wiring Notes.....	22
9. Typical Connection.....	23
10. Sequence Chart of Control Signals.....	23
11. Protection Functions.....	24
Short-voltage and Over-voltage protection .....	24
Over-current Protection.....	25
Short Circuit Protection.....	25
12. Frequently Asked Questions.....	25
Problem Symptoms and Possible Causes.....	26

## Applications

Suitable for a wide range of stepping motors, from NEMA size 17 to 43. It can be used in various kinds of machines, such as X-Y tables, labeling machines, laser cutters, engraving machines, pick-place devices, and so on. Particularly adapt to the applications desired with low noise, low heating, high speed and high precision.

## 2. Specifications

### Electrical Specifications ( $T_J = 25^{\circ}\text{C}/77^{\circ}\text{F}$ )

Parameters	MondoStep 7.8			Unit
	Min	Typical	Max	
Output current	1.8	-	7.8 (5.6 RMS)	A
Supply voltage	+24	+68	+80	VDC
Logic signal current	7	10	16	mA
Pulse input frequency	0	-	300	KHz
Isolation resistance	500			MΩ

### Operating Environment and other Specifications

Cooling Operating Environment	Natural Cooling or Forced cooling	
	Environment	Avoid dust, oil fog and corrosive gases
	Ambient Temperature	0°C – 50°C (32°F – 122°F)
	Humidity	40%RH – 90%RH
Storage Temperature	Operating Temperature	70°C (158°F) Max
	Vibration	5.9m/s <sup>2</sup> Max
Weight	-20°C – 65°C (-4°F – 149°F)	
	Approx. 570g (20.10 oz)	

## Mechanical Specifications (unit: mm[inch])

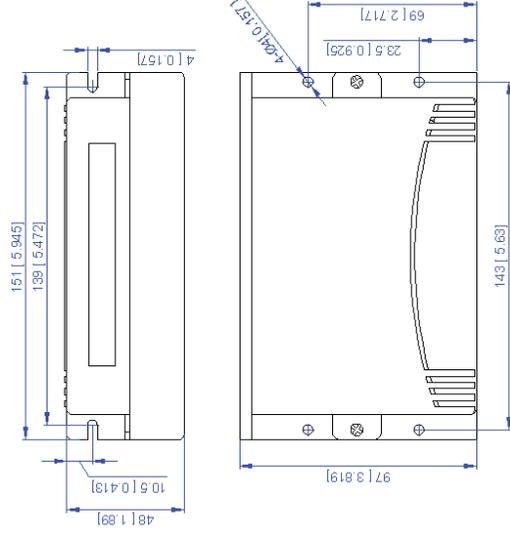


Figure 1: Mechanical specifications

**\*It is recommended to use side mounting for better heat dissipation**

## Elimination of Heat

- Driver's reliable working temperature should be  $<70^{\circ}\text{C}$  (158°F), and motor working temperature should be  $<80^{\circ}\text{C}$  (176°F);
- It is recommended to use automatic idle-current mode, this automatically reduces the motor current to 60% when the motor stops, so as to reduce driver heating and motor heating;
- It is recommended to mount the driver vertically to maximize heat sink area. Use forced cooling method to cool the system if necessary.

## 3. Pin Assignment and Description

The MondoStep 7.8 has two connectors, connector P1 for control signals connections,

Figure 2: J1 and J3 jumpers

### Connector P2 Configurations

Pin Function	Details
VDC	Power supply, 24~80 VDC, Including voltage fluctuation and EMF voltage.
GND	Power Ground.
A+, A-	Motor Phase A
B+, B-	Motor Phase B

### 4. Control Signal Connector (P1) Interface

The MondoStep 7.8 can accept differential and single-ended inputs (including open-collector and PNP output). The MondoStep 7.8 has 3 optically isolated logic inputs which are located on connector P1 to accept line driver control signals. These inputs are isolated to minimize or eliminate electrical noises coupled onto the drive control signals. Recommend use line driver control signals to increase noise immunity of the driver in interference environments. In the following figures, connections to open-collector and PNP signals are illustrated.

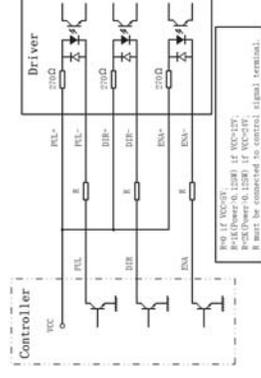


Figure 3: Connections to open-collector signal (common-anode)

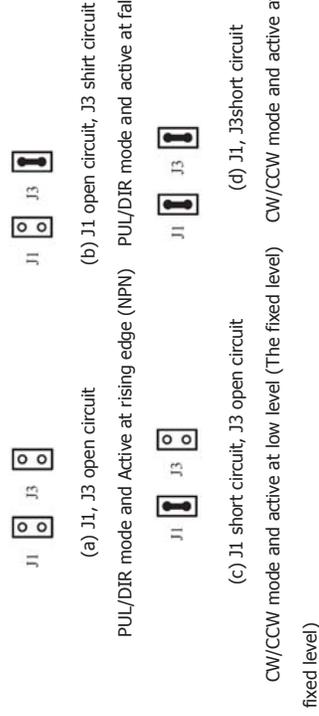
and connector P2 for power and motor connections. The following tables are brief descriptions of the two connectors. More detailed descriptions of the pins and related issues are presented in section 4, 5, 9.

### Connector P1 Configurations

Pin Function	Details
<b>PUL+</b>	<b>Pulse signal:</b> In single pulse (pulse/direction) mode, this input represents pulse signal, active at each rising or falling edge (set by inside jumper J3); 4-5V when PUL-HIGH, 0-0.5V when PUL-LOW. In double pulse mode (pulse/pulse), this input represents clockwise (CW) pulse, active at high level or low level (set by inside jumper J3). For reliable response, pulse width should be longer than 1.5μs. Series connect resistors for current-limiting when +12V or +24V used.
<b>DIR+</b>	<b>DIR signal:</b> In single-pulse mode, this signal has low/high voltage levels, representing two directions of motor rotation; in double-pulse mode (set by inside jumper J1), this signal is counter-clock (CCW) pulse, active at high level or low level (set by inside jumper J3). For reliable motion response, DIR signal should be ahead of PUL signal by 5μs at least. 4-5V when DIR-HIGH, 0-0.5V when DIR-LOW. Please note that motion direction is also related to motor-driver wiring match. Exchanging the connection of two wires for a coil to the driver will reverse motion direction.
<b>ENA+</b>	<b>Enable signal:</b> This signal is used for enabling/disabling the driver. High level (NPN control signal, PNP and Differential control signals are on the contrary, namely Low level for enabling.) for enabling the driver and low level for disabling the driver. Usually left <b>UNCONNECTED (ENABLED)</b> .
<b>ENA-</b>	

### Selecting Effective Pulse Edge or Effective Level and Control Signal Mode

There are two jumpers J1 and J3 inside the MondoStep 7.8 specifically for selecting active pulse edge or effective level and control signal mode, as shown in figure 2. Default setting is PUL/DIR mode and upward-rising edge active. **(Note: J2 inside the driver is used to reverse the default rotation direction.)**



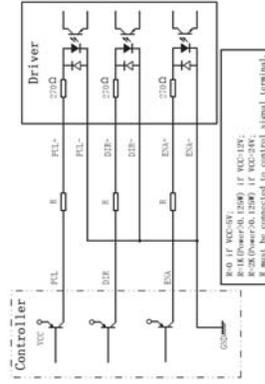


Figure 4: Connection to PNP signal (common-cathode)

## 5. Connecting the Motor

The MondoStep 7.8 can drive any 2-phase and 4-phase hybrid stepping motors.

### Connections to 4-lead Motors

4 lead motors are the least flexible but easiest to wire. Speed and torque will depend on winding inductance. In setting the driver output current, multiply the specified phase current by 1.4 to determine the peak output current.

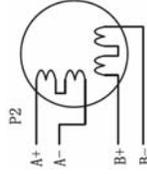


Figure 5: 4-lead Motor Connections

### Connections to 6-lead Motors

Like 8 lead stepping motors, 6 lead motors have two configurations available for high speed or high torque operation. The higher speed configuration, or half coil, is so described because it uses one half of the motor's inductor windings. The higher torque configuration, or full coil, uses the full windings of the phases.

#### Half Coil Configurations

As previously stated, the half coil configuration uses 50% of the motor phase windings. This gives lower inductance, hence, lower torque output. Like the parallel connection of 8 lead motor, the torque output will be more stable at higher speeds. This configuration is

also referred to as half chopper. In setting the driver output current multiply the specified per phase (or unipolar) current rating by 1.4 to determine the peak output current.

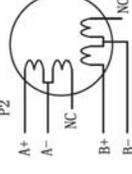


Figure 6: 6-lead motor half coil (higher speed) connections

#### Full Coil Configurations

The full coil configuration on a six lead motor should be used in applications where higher torque at lower speeds is desired. This configuration is also referred to as full copper. In full coil mode, the motors should be run at only 70% of their rated current to prevent over heating.

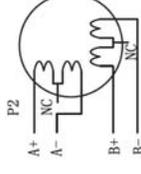


Figure 7: 6-lead motor full coil (higher torque) connections

### Connections to 8-lead Motors

8 lead motors offer a high degree of flexibility to the system designer in that they may be connected in series or parallel, thus satisfying a wide range of applications.

#### Series Connections

A series motor configuration would typically be used in applications where a higher torque at lower speeds is required. Because this configuration has the most inductance, the performance will start to degrade at higher speeds. In series mode, the motors should also be run at only 70% of their rated current to prevent over heating.

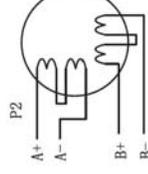


Figure 8: 8-lead motor series connections

## Parallel Connections

An 8 lead motor in a parallel configuration offers a more stable, but lower torque at lower speeds. But because of the lower inductance, there will be higher torque at higher speeds. Multiply the per phase (or unipolar) current rating by 1.96, or the bipolar current rating by 1.4, to determine the peak output current.

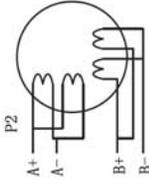


Figure 9: 8-lead motor parallel connections

## 6. Power Supply Selection

The MondoStep 7.8 can match medium and small size stepping motors (from NEMA frame size 17 to 43) made by us or other motor manufacturers around the world. To achieve good driving performances, it is important to select supply voltage and output current properly. Generally speaking, supply voltage determines the high speed performance of the motor, while output current determines the output torque of the driven motor (particularly at lower speed). Higher supply voltage will allow higher motor speed to be achieved, at the price of more noise and heating. If the motion speed requirement is low, it's better to use lower supply voltage to decrease noise, heating and improve reliability.

### Regulated or Unregulated Power Supply

Both regulated and unregulated power supplies can be used to supply the driver. However, unregulated power supplies are preferred due to their ability to withstand current surge. If regulated power supplies (such as most switching supplies.) are indeed used, it is important to have large current output rating to avoid problems like current clamp, for example using 4A supply for 3A motor-driver operation. On the other hand, if unregulated supply is used, one may use a power supply of lower current rating than that of motor (typically 50% ~ 70% of motor current). The reason is that the driver draws current from the power supply capacitor of the unregulated supply only during the ON duration of the PWM cycle, but not during the OFF duration. Therefore, the average

current withdrawn from power supply is considerably less than motor current. For example, two 3A motors can be well supplied by one power supply of 4A rating.

## Multiple Drivers

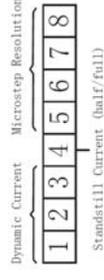
It is recommended to have multiple drivers to share one power supply to reduce cost, if the supply has enough capacity. To avoid cross interference, **DO NOT** daisy-chain the power supply input pins of the drivers. (Instead, please connect them to power supply separately.)

## Selecting Supply Voltage

The power MOSFETS inside the MondoStep 7.8 can actually operate within +24 ~ +80VDC, including power input fluctuation and back EMF voltage generated by motor coils during motor shaft deceleration. Higher supply voltage can increase motor torque at higher speeds, thus helpful for avoiding losing steps. However, higher voltage may cause bigger motor vibration at lower speed, and it may also cause over-voltage protection or even driver damage. Therefore, it is suggested to choose only sufficiently high supply voltage for intended applications, and it is suggested to use power supplies with theoretical output voltage of +24 ~ +75VDC, leaving room for power fluctuation and back-EMF.

## 7. Selecting Microstep Resolution and Driver Output Current

This driver uses an 8-bit DIP switch to set microstep resolution, and motor operating current, as shown below:



### Microstep Resolution Selection

Microstep resolution is set by SW5, 6, 7, 8 of the DIP switch as shown in the following table:

Microstep	Steps/rev.(for 1.8°motor)	SW5	SW6	SW7	SW8
2	400	ON	ON	ON	ON

4	800	OFF	ON	ON	ON
8	1600	ON	OFF	ON	ON
16	3200	OFF	OFF	ON	ON
32	6400	ON	ON	OFF	ON
64	12800	OFF	ON	OFF	ON
128	25600	ON	OFF	OFF	ON
256	51200	OFF	OFF	OFF	ON
5	1000	ON	ON	ON	OFF
10	2000	OFF	ON	ON	OFF
20	4000	ON	OFF	ON	OFF
25	5000	OFF	OFF	ON	OFF
40	8000	ON	ON	OFF	OFF
50	10000	OFF	ON	OFF	OFF
100	20000	ON	OFF	OFF	OFF
200	40000	OFF	OFF	OFF	OFF

## Current Settings

For a given motor, higher driver current will make the motor to output more torque, but at the same time causes more heating in the motor and driver. Therefore, output current is generally set to be such that the motor will not overheat for long time operation. Since parallel and serial connections of motor coils will significantly change resulting inductance and resistance, it is therefore important to set driver output current depending on motor phase current, motor leads and connection methods. Phase current rating supplied by motor manufacturer is important in selecting driver current, however the selection also depends on leads and connections.

The first three bits (SW1, 2, 3) of the DIP switch are used to set the dynamic current. Select a setting closest to your motor's required current.

### Dynamic current setting

Peak Current	RMS Current	SW1	SW2	SW3
2.8 A	2.0 A	ON	ON	ON
3.5 A	2.5 A	OFF	ON	ON
4.2 A	3.0 A	ON	OFF	ON
4.9 A	3.5 A	OFF	OFF	ON
5.7 A	4.1 A	ON	ON	OFF
6.4 A	4.6 A	OFF	ON	OFF
7.0 A	5.0 A	ON	OFF	OFF

## 7.8 A

## 5.6 A

## OFF

**Notes:** Due to motor inductance, the actual current in the coil may be smaller than the dynamic current setting, particularly under high speed condition.

### Standstill current setting

SW4 is used for this purpose. OFF meaning that the standstill current is set to be half of the selected dynamic current, and ON meaning that standstill current is set to be the same as the selected dynamic current.

The current automatically reduced to 60% of the selected dynamic current one second after the last pulse. Theoretically, this will reduce motor heating to 36% (due to  $P=I^2 \cdot R$ ) of the original value. If the application needs a different standstill current, please contact us.

## 8. Wiring Notes

- In order to improve anti-interference performance of the driver, it is recommended to use twisted pair shield cable.
- To prevent noise incurred in PUL/DIR signal, pulse/direction signal wires and motor wires should not be tied up together. It is better to separate them by at least 10 cm, otherwise the disturbing signals generated by motor will easily disturb pulse direction signals, causing motor position error, system instability and other failures.
- If a power supply serves several drivers, separately connecting the drivers is recommended instead of daisy-chaining.
- It is prohibited to pull and plug connector P2 while the driver is powered ON, because there is high current flowing through motor coils (even when motor is at standstill). Pulling or plugging connector P2 with power on will cause extremely high back-EMF voltage surge, which may damage the driver.

## 9. Typical Connection

A complete stepping system should include stepping motor, stepping driver, power supply and controller (pulse generator). A typical connection is shown as figure 10.

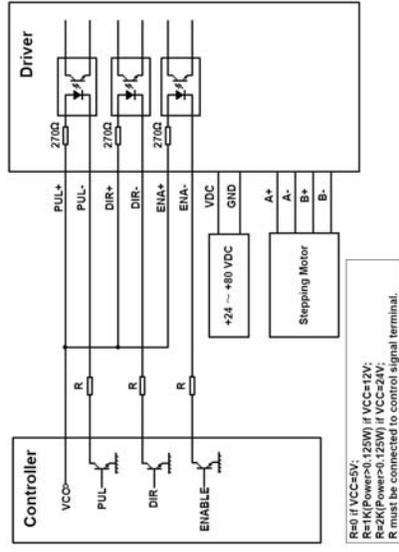


Figure 10: Typical connection

## 10. Control Signal Timing Diagram

In order to avoid some fault operations and deviations, PUL, DIR and ENA should abide by some rules, shown as following diagram:

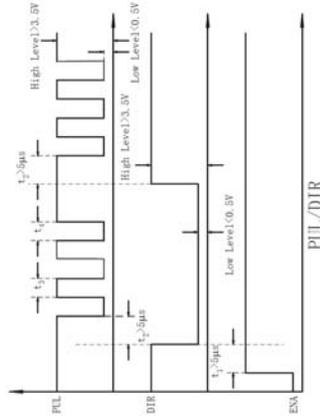


Figure 11: Sequence chart of control signals

### Remark:

- ENA must be ahead of DIR by at least  $5\mu s$ . Usually, ENA+ and ENA- are NC (not connected). See "Connector P1 Configurations" for more information.
- DIR must be ahead of PUL effective edge by  $5\mu s$  to ensure correct direction;
- Pulse width not less than  $1.5\mu s$ ;
- Low level width not less than  $1.5\mu s$ .

## 11. Protection Functions

To improve reliability, the driver incorporates some built-in protections features.

### Short-voltage and Over-voltage protection

When power supply voltage is lower than +18VDC, over-voltage protection will be activated and power indicator LED will turn off. When power supply voltage exceeds +94VDC, over-voltage protection will be activated and the Alarm indicator LED will turn on.

### Over-current Protection

Protection will be activated when continuous current reaches to 16A.

### Short Circuit Protection

Protection will be activated in case of short circuit between motor coils or between motor coil and ground.

## 12. Frequently Asked Questions

In the event that your driver doesn't operate properly, the first step is to identify whether the problem is electrical or mechanical in nature. The next step is to isolate the system component that is causing the problem. As part of this process you may have to disconnect the individual components that make up your system and verify that they operate independently. It is important to document each step in the troubleshooting process. You may need this documentation to refer back to at a later date, and these details will greatly assist our Technical Support staff in determining the problem should you need assistance.

Many of the problems that affect motion control systems can be traced to electrical noise, controller software errors, or mistake in wiring.

### Problem Symptoms and Possible Causes

Symptoms	Possible Problems
<b>Motor is not rotating</b>	No power
	Microstep resolution setting is wrong
	DIP switch current setting is wrong
<b>Motor rotates in the wrong direction</b>	Fault condition exists
	The driver is disabled
<b>The driver in fault</b>	Motor phases may be connected in reverse
	DIP switch current setting is wrong
<b>Erratic motor motion</b>	Something wrong with motor coil
	Control signal is too weak
	Control signal is interfered
<b>Motor stalls during acceleration</b>	Wrong motor connection
	Something wrong with motor coil
	Current setting is too small, losing steps
	Motor is undersized for the application
<b>Excessive motor and driver heating</b>	Acceleration is set too high
	Power supply voltage too low
	Inadequate heat sinking / cooling
	Automatic current reduction function not being utilized
	Current is set too high

# MondoStep 4.2

## Bi-polar Stepper Motor Driver



- **Features:**
- High performance, cost-effective
- Supply voltage up to +50VDC
- Output current up to 4.2A
- Pure-sinusoidal current control
- Pulse input frequency up to 300 kHz
- TTL compatible optically isolated inputs
- Automatic idle-current reduction
- 15 selectable microstep resolutions, up to 25600 steps/rev
- Supports Step/Direction and CW/CCW modes
- Short-circuit, over-voltage, over-current and over-temperature protection

### Introduction:

The MondoStep 4.2 drivers are high performance microstepping drivers based on the most advanced stepper motor driver technologies in the world today. They are suitable for driving any 2-phase hybrid stepper motors. By using advanced bipolar constant-current chopping technique, the MondoStep can output more speed and power from the same motor. With 3-state current control technology, coil current is maintained with relatively small current ripple which results in greater positioning accuracy and less motor heating.

### Applications:

Suitable for a wide range of stepping motors of size Nema23 and NEMA 34, and usable for various kinds of machines, such as X-Y tables, labeling machines, laser cutters, engraving machines, and pick-place devices. Extremely suitable for applications expected to be low vibration, high speed and high precision.

### Electric Specifications (Tj=25°C)

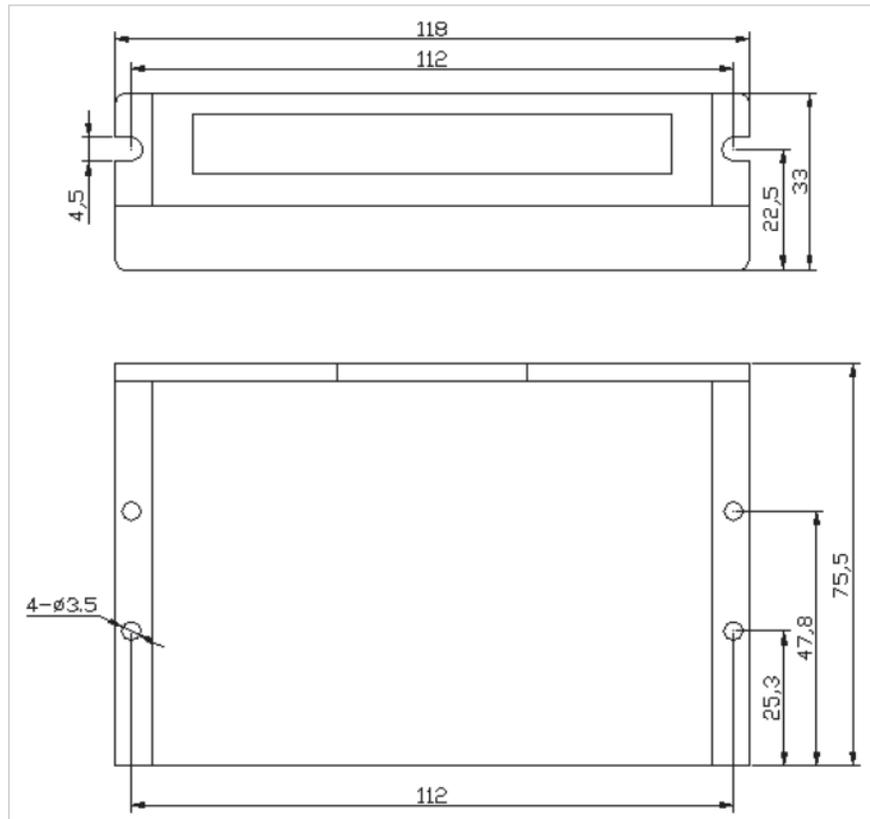
Parameters	Min	Typical	Max	Units
Output Current	1.0	-	4.2	Amps
Supply Voltage	+20	+36	+50	VDC
Logic Signal Current	7	10	16	mA
Step Frequency	0	-	300	Khz
Isolation Resistance	500	-	-	Mohm
Step Pulse Width	1.2			uS
Direction Setup	5			uS
Input High	4	-	5	VDC
Input Low	0	-	0.5	VDC

### Operating Environment and Other Specifications

Cooling	Natural Cooling or Forced Air Cooling
Operating Environment	Avoid dust, oil, mist, and corrosive gases
Ambient Temperature	0°C - 50°C (32°F - 122°F)
Humidity	40%RH - 90%RH
Operating Temperature	70°C (158°F) Max
Vibration	5.9m/s <sup>2</sup>
Storage Temperature	-20°C - 65°C (-4°F - 149°F)
Weight	280g (10oz)

### Mechanical Specifications:

Unit: mm



Pin Assignment and Description

Control Signal Connector P1 pins

Pin Function	Details
PUL+ (+5VDC)	Step Signal: In Step/Direction mode, this input is the step signal. In Up/Down mode, this signal is the clockwise (CW) step signal. Internal jumper JP1 determines whether the indexer steps on the rising or falling edge of the step pulse.
PUL- (GND)	
DIR + (+5VDC)	Direction Signal: In Step/Direction mode, this input is the direction signal. The motor will spin in one direction when this input is high, and the other direction when low. In Up/Down mode, this input is the counter-clockwise (CCW) step signal.
DIR - (GND)	
ENA+ (+5VDC)	Enable Signal: This signal will disable the driver when pulled low. Left unconnected, the driver will be enabled.
ENA- (GND)	

Power connector P2 pins

Pin Function	Details
GND	DC Power Ground
+V	DC Power Supply, 24-80VDC. Allow for voltage fluctuations and back EMF.
A+, A-	Motor Phase A
B+, B-	Motor Phase B

Microstep Resolution Selection

Microsteps	Steps/Rev	SW5	SW6	SW7	SW8
2	400	OFF	ON	ON	ON
4	800	ON	OFF	ON	ON
8	1600	OFF	OFF	ON	ON
16	3200	ON	ON	OFF	ON
32	6400	OFF	ON	OFF	ON
64	12800	ON	OFF	OFF	ON
128	25600	OFF	OFF	OFF	ON
5	1000	ON	ON	ON	OFFN
10	2000	OFF	ON	ON	OFF
20	4000	ON	OFF	ON	OFF
25	5000	OFF	OFF	ON	OFF
40	8000	ON	ON	OFF	OFF
50	10000	OFF	ON	OFF	OFF
100	20000	OFF	ON	OFF	OFF
125	25000	OFF	OFF	OFF	OFF

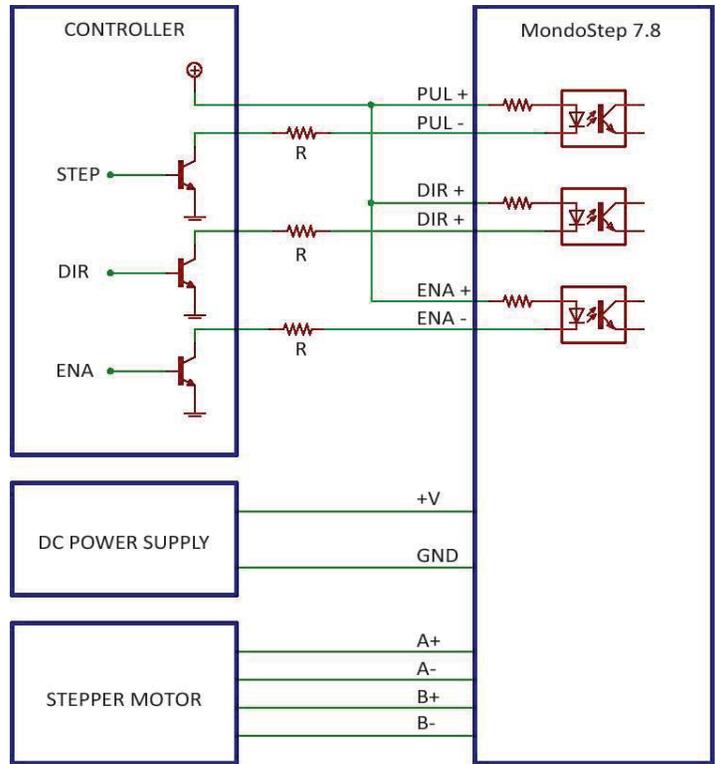
Current Setting

Peak Current	SW1	SW2	SW3
1.0 A	ON	ON	ON
1.46 A	OFF	ON	ON
1.91 A	ON	OFF	ON
2.84 A	OFF	OFF	ON
3.31 A	ON	ON	OFF
3.76 A	OFF	ON	OFF
4.2 A	ON	OFF	OFF

Idle Current Reduction

SW4 is used to enable or disable idle current reduction. When switched OFF, the motor current is reduced to half after one second of receiving the last step pulse. If switched ON the current will remain at the full current.

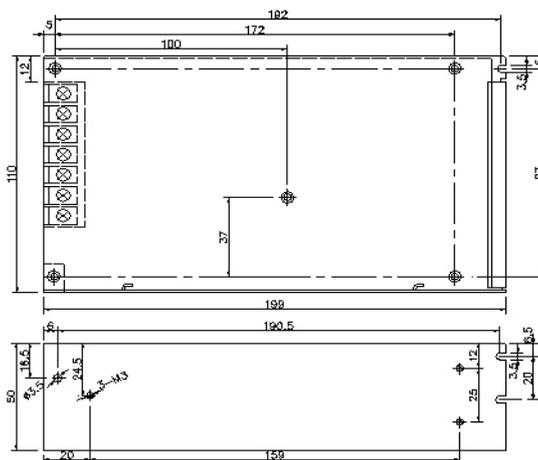
Typical Connections



## Datasheet fuentes



### Industrial Power Supply IM150W Series



## SPECIAL FEATURES

- AC input range selected by switch
- High efficiency
- DC adjust range
- LED Power indicated
- Output reverses protection
- Free air cooling convection
- 199\* 97\* 50 mm
- Weight 0.73 kgs/pcs
- 1 year warranty

## SPECIFICATION

### Output

- Maximum power.....150 Watts
- DC adjust range..... ± 10% of rating output voltage
- Hold-up time.....> 20 ms at 230Vac/ 50Hz
- Rise time.....< 20ms
- Over Voltage Protection... 110% ~ 135% of rated output
- Over Power Protection..... 105% ~ 150%, auto-recovery
- Short Circuit Protection.....shut off, AC recycle to restart
- Connector.....7P / 9.5mm pitch terminal block

### Input

- Input range.....90~132Vac/180~264Vac selected by switch
- Frequency.....47-63Hz
- AC inrush current...Cold start, 30A @ 115Vac; 60A @ 220Vac
- EMI/RFI.....FCC Part 15J, Class A & CISPR 22 Class A

### Operating temperature

- 0 °C to +40 °C @ 100%; -10 °C @ 80%; 50 °C @ 60% load
- Storage temperature: -20 °C to +80 °C
- Free air cooling convection

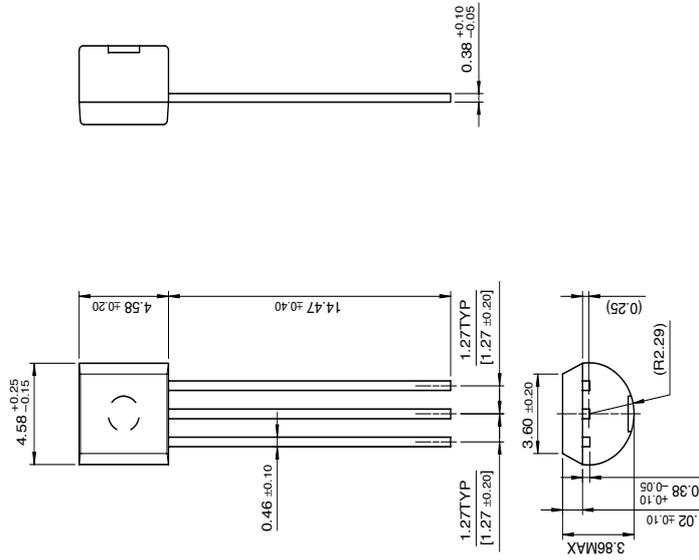
Model	IM151C12-1M	IM151C24-1M
Output	V1	V1
Voltage	+12V	+24V
Max. Load	12.5A	6.5A
Efficiency	>80% @115Vac/60 Hz	>80% @ 115 Vac/60 Hz

PN2222

Package Dimensions

TO-92

Datasheet transistor 2N2222



Dimensions in Millimeters

Rev. A, November 2004

©2004 Fairchild Semiconductor Corporation

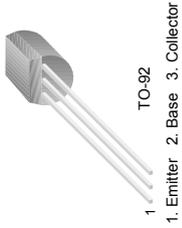
PN2222

PN2222

General Purpose Transistor

NPN Epitaxial Silicon Transistor

Absolute Maximum Ratings  $T_a=25^{\circ}\text{C}$  unless otherwise noted



TO-92  
1. Emitter 2. Base 3. Collector

Symbol	Parameter	Value	Units
$V_{CB0}$	Collector-Base Voltage	60	V
$V_{CE0}$	Collector-Emitter Voltage	30	V
$V_{EB0}$	Emitter-Base Voltage	5	V
$I_C$	Collector Current	600	mA
$P_C$	Collector Power Dissipation	625	mW
$T_J$	Junction Temperature	150	$^{\circ}\text{C}$
$T_{STG}$	Storage Temperature	-55 ~ 150	$^{\circ}\text{C}$

Electrical Characteristics  $T_a=25^{\circ}\text{C}$  unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Max.	Units
$BV_{CBO}$	Collector-Base Breakdown Voltage	$I_C=10\mu\text{A}, I_E=0$	60		V
$BV_{CEO}$	Collector-Emitter Breakdown Voltage	$I_C=10\text{mA}, I_B=0$	30		V
$BV_{EBO}$	Emitter-Base Breakdown Voltage	$I_E=10\mu\text{A}, I_C=0$	5		V
$I_{CO}$	Collector Cut-off Current	$V_{CB}=50\text{V}, I_E=0$		0.01	$\mu\text{A}$
$I_{EO}$	Emitter Cut-off Current	$V_{EB}=3\text{V}, I_C=0$		10	nA
$h_{FE}$	DC Current Gain	$V_{CE}=10\text{V}, I_C=0, I_B=1\text{mA}$ $V_{CE}=10\text{V}, I_C=150\text{mA}$	35	300	
$V_{CE(sat)}$	* Collector-Emitter Saturation Voltage	$I_C=500\text{mA}, I_B=50\text{mA}$		1	V
$V_{BE(sat)}$	* Base-Emitter Saturation Voltage	$I_C=500\text{mA}, I_B=50\text{mA}$		2	V
$f_T$	Current Gain Bandwidth Product	$V_{CE}=20\text{V}, I_C=20\text{mA}, f=100\text{MHz}$	300		MHz
$C_{ob}$	Output Capacitance	$V_{CB}=10\text{V}, I_E=0, f=1\text{MHz}$		8	pF

\* Pulse Test: Pulse Width=300 $\mu\text{s}$ , Duty Cycle=2%

Rev. A, November 2004

©2004 Fairchild Semiconductor Corporation

### Tarjeta acople EzDSP F2808

Este no es el tamaño real. Para imprimir las imágenes correctas ver documento digital adjunto a este proyecto.

