

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE CUENCA

FACULTAD DE INGENIERÍA

CARRERA: ELECTRÓNICA

Trabajo de Grado previo a la obtención del

Título de Tecnólogo en Electrónica

**“DISEÑO Y CONSTRUCCIÓN DE UN TALADRO XYZ CONTROLADO
POR MICROCONTROLADOR CON INTERFAZ GRÁFICA MEDIANTE
LABVIEW PARA LA FABRICACIÓN DE CIRCUITOS IMPRESOS”**

Autores:

Juan Carlos Galarza Galarza

Miguel Alberto Dávila Sacoto

Director:

Ing. Marco Carpio

CUENCA, ENERO DE 2010

DEDICATORIA

**Al Gran Arquitecto del Universo, quien me
guía por el camino del conocimiento, al
Poeta maldito, cuyas letras acompañaron
mis eternas noches de delirio, y a ti, pobre
ser que no entiende que “*en el placer de
hacer el mal se encuentra toda voluptuosidad*”**

Miguel Alberto Dávila Sacoto

DEDICATORIA

**A mis padres y al loco amor por la ciencia,
que cada día te convierte en un loco para
unos y en un sabio para otros, pero a la final
lo que queda es el placer de haber cumplido
un sueño, y con ganas de seguir adelante sin
que nada se te oponga.**

Juan Carlos Galarza Galarza

AGRADECIMIENTO

A todas las personas que hicieron posible el desarrollo de este trabajo de grado, a nuestros amigos que nos apoyaron y motivaron y en especial al Ing. Wilson Calle, director del laboratorio de Mecánica de la Universidad Politécnica Salesiana, quien desinteresadamente nos proporcionó los conocimientos y recursos mecánicos necesarios para la realización de este proyecto.

Ing. Marco Carpio

CERTIFICA

Haber dirigido y revisado prolijamente cada uno de los capítulos de este trabajo de tesis, realizada por los señores Juan Carlos Galarza Galarza y Miguel Alberto Dávila Sacoto.

Por cumplir los requisitos, autorizo su presentación.

Ing. Marco Carpio

DECLARATORIA DE RESPONSABILIDAD

Nosotros, Juan Carlos Galarza Galarza y Miguel Alberto Dávila Sacoto, autores del Presente Trabajo de Tesis “DISEÑO Y CONSTRUCCIÓN DE UN TALADRO XYZ CONTROLADO POR MICROCONTROLADOR CON INTERFAZ GRAFICA MEDIANTE LABVIEW PARA LA FABRICACION DE CIRCUITOS IMPRESOS” declaramos que:

Los conceptos desarrollados, análisis realizados y las conclusiones del presente trabajo, son de exclusiva responsabilidad de los autores.

Cuenca, diciembre de 2009

Juan Carlos Galarza Galarza

Miguel Alberto Dávila Sacoto

**LOS MIEMBROS DEL TRIBUNAL DE CALIFICACIÓN Y
SUSTENTACIÓN DE LA TESIS**

**“DISEÑO Y CONSTRUCCIÓN DE UN TALADRO XYZ CONTROLADO
POR MICROCONTROLADOR CON INTERFAZ GRAFICA MEDIANTE
LABVIEW PARA LA FABRICACION DE CIRCUITOS IMPRESOS”**

Que la presente tesis ha sido debidamente revisada y por lo tanto

APROBADA

Presidente Del Tribunal

Miembro Del Tribunal

Miembro Del Tribunal

Cuenca, enero de 2009

Índice:

Introducción.....	4
Capítulo 1. Diseño y construcción de la parte mecánica	
1.1 Generalidades: Robots industriales.....	5
1.2 Morfología del robot industria.....	5
1.2.1 Estructura Mecánica.....	6
1.2.2 Transmisiones y reductores.....	7
1.2.3 Sistemas de accionamiento.....	10
1.2.4 Actuadores.....	11
1.2.4.1 Actuadores Eléctricos.....	12
1.2.4.2 Motores de pasos.....	12
1.2.4.3 Tipos de motores de pasos.....	13
1.2.5 Selección del motor apropiado.....	17
1.2.6 Sistema sensorial.....	22
1.2.7 Elementos terminales.....	23
1.3 Máquinas CNC.....	25
1.3.1 Principio de funcionamiento del CNC.....	25
1.3.2 Características del CNC.....	26
1.3.3 Ejes principales de una máquina CNC.....	27
1.3.4 Sistemas de sujeción.....	31
1.4 El CNC en la fabricación de circuitos impresos.....	32
1.4.1 Proceso de fabricación de circuito impresos...32	
1.4.2 Máquinas de prototipos.....	36
1.5 Selección del diseño de mesa.....	37
1.5.1 Fresadora basada en el movimiento del área de trabajo.....	37
1.5.2 Fresadora basada en el movimiento ejes.....	38
1.6 Diseño y Construcción.....	39
1.6.1 Eje Z.....	42
1.6.2 Carro para el eje X e Y.....	44
1.6.3 Mecanizado.....	45
1.6.4 Montaje.....	52

1.7 Análisis de precisión mecánica.....	53
Capítulo 2. Diseño e implementación de la parte eléctrica y electrónica	
2.1 Circuito Eléctrico.....	54
2.1.1 Diseño del transformador.....	54
2.1.2 Fuente de poder.....	74
2.2 Circuito electrónico.....	76
2.2.1 Diseño modular.....	77
2.2.2 El microcontrolador PIC18f4550.....	77
2.2.3 Comunicación USB.....	80
2.2.4 Diseño de la comunicación con el computador.....	98
2.2.4.1 Configuración del HID en Protón.....	99
2.2.5 Circuito de control.....	106
2.2.6 Potencia.....	108
Capítulo 3. Programación de la pantalla gráfica de control mediante LabView	
3.1 Control numérico por computador.....	110
3.1.1 Código NC-Drill.....	110
3.1.2 Funciones Preparatorias (G) Código Gerber.....	116
3.2 Identificación de los archivos generados por software de circuitos impresos.....	123
3.3 Comunicación con LabView.....	124
3.3.1 Programa en LabView.....	126
3.3.2 Comunicación USB mediante LabView.....	132
3.4 Decodificador del archivo NC-Drill en LabView.....	137

3.5 Seguridad industrial.....	140
3.5.1 Tipos de accidentes.....	141
3.5.2 Fuentes de peligros en robots industriales.....	141
3.5.3 Tipos de control de seguridad.....	142
Conclusiones.....	143
Bibliografía.....	145
Anexos	

Introducción

El constante adelanto de la tecnología, requiere procesos de manufactura cada vez más precisos y rápidos por lo que las máquinas herramienta han jugado un papel fundamental en el desarrollo tecnológico del mundo.

Para la mecanización total de un número de piezas es necesario realizar las operaciones de fresado, mandrinado y perforado, la cual alcanzaría mayor eficacia si este grupo de máquinas herramientas estuvieran automatizadas. Esta necesidad, sumada a numerosos y nuevos requerimientos que día a día forzaron la utilización de nuevas técnicas que reemplazaran al operador humano. De esta forma se introdujo el control numérico en los procesos de fabricación, impuesto por varias razones, como la necesidad de fabricar productos que no se podían conseguir en cantidad y calidad suficientes sin recurrir a la automatización del proceso de fabricación, la necesidad de obtener productos hasta entonces imposibles o muy difíciles de fabricar, por ser excesivamente complejos para ser controlados por un operador humano y la necesidad de fabricar productos a precios suficientemente bajos.

En cuanto al desarrollo de la fabricación de circuitos impresos el control numérico no se queda de lado, siendo así que las máquinas de prototipos funcionan mediante éste código, lo que resulta en mayor precisión al realizar agujeros y pistas en los mismos. Por lo cual se vio la necesidad de comprender el funcionamiento de éste tipo de máquinas para así poder diseñarlas y posteriormente comercializarlas.

Por lo tanto en esta tesis se analizará una parte importante del CNC, que se encarga de los agujeros en piezas, en nuestro caso en PCB's (placas de circuito impreso), y se construirá una mesa para dicho objetivo. Se describirá el proceso de construcción de la máquina y el control electrónico necesario para su correcto funcionamiento.

Capítulo 1. Diseño y construcción de la parte mecánica

1.1 Generalidades: Robots industriales

La robótica industrial puede definirse como el estudio, diseño y aplicación de robots para la ejecución de procesos industriales. Más formalmente, el estándar ISO 8373:1994 de Robots industriales manipuladores dado por la Federación Internacional de Robótica (IFR), define a un robot industrial como un manipulador multifuncional reprogramable con varios grados de libertad, el cual es capaz de posicionar y orientar objetos según trayectorias programadas para los mismos, los cuales son utilizados en diferentes etapas de producción industrial. La IFR, clasifica a los robots industriales según generaciones en tres tipos básicos:

1. **Robots de Primera Generación:** Repiten una tarea programada secuencialmente, no posee una retroalimentación de las posibles alteraciones de su entorno.
2. **Robots de Segunda Generación:** Este tipo de robots adquieren información limitada de su entorno y actúa en consecuencia a estos datos. Pueden localizar, clasificar y detectar esfuerzos, y así adaptar sus movimientos.
3. **Robots de Tercera Generación:** Los cuales poseen capacidad para la planificación automática de tareas (inteligencia artificial).

1.2 Morfología del robot industrial

Un robot está formado por seis elementos principales:

1. Estructura mecánica
2. Transmisiones y reductores
3. Sistema de accionamiento
4. Actuadores
5. Sistema sensorial

- 6. Sistema de control
- 7. Elementos terminales

1.2.1 Estructura mecánica

Mecánicamente, un robot industrial está constituido por elementos o eslabones unidos mediante articulaciones que permiten su movimiento relativo. Su constitución física guarda similitud con la anatomía del brazo humano, por lo cual cuando se hace referencia a elementos del robot, se utilizan términos como cuerpo, brazo, codo y muñeca. Cada articulación puede realizar desplazamientos lineales, angulares o una combinación de ambos. Existen seis tipos diferentes de articulaciones posibles en un robot, las cuales se muestran en la figura 1.2.1.1

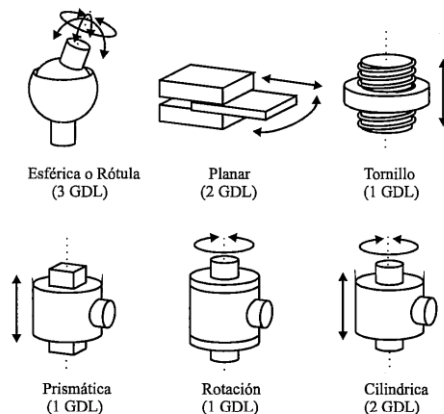


Figura1.2.1.1 Tipos de articulaciones de un robot industrial [1]

El movimiento independiente de una articulación respecto a la anterior se denomina grado de libertad, por lo cual el grado de libertad de un robot viene dado por la suma de los grados de libertad de las articulaciones que lo componen. El empleo de diferentes combinaciones de articulaciones, da lugar a diferentes configuraciones de robots, las cuales se muestran en la figura1.2.1.2. El diseño de este tipo de configuraciones depende íntegramente de la aplicabilidad del robot.

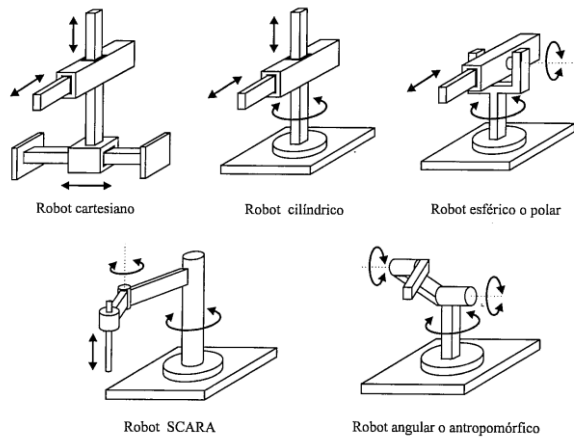


Figura1.2.1.2 Configuraciones de robots industriales [1]

En este proyecto se utilizará la configuración de robot cartesiano, el cual presenta movimientos lineales en los ejes “x”, “y”, y “z” permitiendo el movimiento de un punto a otro en una base, acoplándose así perfectamente para el trazado de pistas y agujeros en tarjetas electrónicas. Este tipo de aplicación de robots se centra en las máquinas de control numérico, las cuales se especificarán más adelante.

1.2.2 Transmisiones y reductores

Las transmisiones son elementos encargados de transmitir el movimiento desde los actuadores hasta las articulaciones, junto a la transmisión se incluyen los reductores, los cuales adaptan el par y la velocidad de salida del actuador a valores adecuados al diseño y funcionalidad del robot.

Dado que un robot mueve su extremo con aceleraciones elevadas debido al uso de motores de corriente continua, se debe reducir al máximo su momento de inercia. Del mismo modo, los pares estáticos que deben vencer los actuadores dependen directamente de la distancia de las masas del actuador. Por estos motivos se procura que los actuadores pesados estén lo más cerca posible de la base del robot.

El sistema de transmisión debe cumplir las siguientes características básicas:

- Debe tener un tamaño y peso reducido
- No debe presentar juegos u holguras considerables
- Alto rendimiento

Los sistemas de transmisión usados con mayor frecuencia se especifican en la tabla 1.2.2.1. Cabe destacar que el sistema de transmisión a utilizar no debe afectar el movimiento que transmite, ya sea por rozamiento o por holguras causadas por desgaste. Por lo tanto el sistema de transmisión debe soportar un par elevado y un funcionamiento continuo.

Tabla 1.2.2.1 Sistemas de transmisión comunes [1]

Entrada- Salida	Denominación	Ventajas	Desventajas
	Engranaje	Pares altos Distancia grande	Holguras
Circular- Circular	Correa dentada	Distancia grande	-
	Cadena	grande	Ruido
	Paralelogramo	-	Giro limitado
	Cable	-	Deformabilidad
Circular- Lineal	Tornillo sin fin Cremallera	Poca Holgura Poca Holgura	Rozamiento Rozamiento
Lineal- Circular	Paral, articulado Cremallera	- Holgura media	Control difícil Rozamiento

En cuanto a reductores, estos son utilizados en robótica para condiciones de funcionamiento muy restrictivas. Estos elementos son los que aseguran precisión y

velocidad de posicionamiento. Las características que debe tener un reductor son las siguientes:

- Bajo peso
- Tamaño reducido
- Bajo rozamiento
- Deben realizar una reducción elevada de velocidad en un único paso
- Deben minimizar el momento de inercia de los motores
- Deben soportar pares elevados puntuales
- El juego angular o backlash (el cual es el ángulo que gira el eje de salida cuando se cambia su sentido de giro sin que llegue a girar el eje de entrada) debe ser mínimo
- Deben poseer alta rigidez torsional (par que se debe aplicar sobre el eje de salida para que, manteniendo bloqueado el de entrada, aquél gire un ángulo unitario)

En la tabla 1.2.2.2 se muestran los valores típicos de reductores utilizados comúnmente en robótica.

Tabla 1.2.2.2 Características y valores típicos de reductores

Características	Valores Típicos
Relación de reducción	50-300
Peso y tamaño	0,1-30 kg
Momento de inercia	0,0010 kg m ²
Velocidades de entrada máxima	6000-7000rpm
Par de salida nominal	5700 Nm
Par de salida máximo	7900 Nm
Juego angular	0-2 "
Rigidez torsional	100-200 Nm/rad
Rendimiento	85%-98%

Por lo tanto, tomando en cuenta estos criterios se procedió a diseñar el sistema de transmisión del proyecto. Así se tomaron las siguientes decisiones:

1. Para mover los ejes se utilizará una transmisión circular-lineal por medio de un tornillo sin fin, así la estructura será más robusta, y el juego angular será mínimo
2. Se utilizará un sistema de transmisión por engranaje entre los motores de pasos a utilizarse y los tornillos sin fin, así aunque se perderá velocidad, la ganancia en torque será suficiente para mover la estructura
3. El deslizamiento de los ejes se hará a través de un sistema diseñado de manera que el rozamiento sea eliminado, esto es mediante un juego de rodamientos y una estructura para los mismos

1.2.3 Sistemas de accionamiento

Accionamiento directo.- Es el de accionamiento en el que el eje del actuador se conecta directamente a la carga o articulación, sin la utilización de un reductor intermedio, exclusivamente se utiliza para robots con accionamiento eléctrico.

Este tipo de accionamiento permite combinar gran precisión con alta velocidad, presentando las siguientes ventajas:

- Posicionamiento rápido y preciso
- Evita rozamientos
- Evita juegos en transmisiones y reductores
- Aumenta las posibilidades de controlabilidad del sistema pero demandan una mayor complejidad
- Simplifica el sistema mecánico al eliminarse el reductor

Desventajas:

- Debe utilizarse motores con un par elevado a bajas revoluciones manteniendo la máxima rigidez posible, los cuales son pesados y voluminosos

- Se debe acoplar un codificador de posición en el eje, el cual debe tener una resolución muy alta

Analizando las ventajas y desventajas de este tipo de accionamiento se decidió colocarlo como una opción, ya que facilita el control de la mesa pero demanda una complejidad de construcción mecánica de estos elementos, conocidos también como matrimonios entre el eje del motor y el tornillo sin fin es mucho mayor al del acoplamiento de engranes.

1.2.4 Actuadores

Son los elementos que generan movimiento del robot, según las órdenes dadas por la unidad de control. Estos pueden ser neumáticos, hidráulicos o eléctricos. Cada uno presenta características diferentes las cuales deben ser tomadas en cuenta al diseñar un robot, entre las cuales tenemos las siguientes:

- Potencia
- Controlabilidad
- Peso y Volumen
- Precisión
- Velocidad
- Mantenimiento
- Coste

Debido a que el presente proyecto utilizará una potencia relativamente baja, y por cuestiones de coste, se decidió utilizar actuadores eléctricos, los cuales presentan características favorables para el control numérico, siendo bastante precisos y de fácil control.

1.2.4.1 Actuadores Eléctricos

Como se mencionó antes, las características de control, sencillez y precisión de los actuadores eléctricos ha hecho que sean los más utilizados en robots industriales. Estos actuadores se clasifican en tres tipos diferentes:

1. Motores de corriente continua
 - a. Controlados por inducido
 - b. Controlados por excitación
2. Motores de corriente alterna
 - a. Síncronos
 - b. Asíncronos
3. Motores de pasos

En cuanto se refiere al diseño de máquinas CNC de gran potencia, los actuadores más utilizados son los motores de corriente continua con reducción por engranaje, conocidos también como servomotores, los cuales presentan torques elevados y alta precisión, pero en sistemas de baja potencia, como en el caso de la máquina que se diseña para este proyecto se utilizarán motores de pasos.

1.2.4.2 Motor de pasos

Los motores de pasos, también llamados motores paso a paso, son dispositivos electromagnéticos que convierten pulsos digitales en rotación mecánica, son un tipo especial de motores DC utilizados por sus movimientos precisos y de velocidad controlable.

Funcionan basados en el principio de repulsión y atracción de polos magnéticos, su rotor consiste de imanes permanentes y su estator de 2 bobinas que actúan como electroimanes, que al interactuar con los imanes del rotor provocan un movimiento circular exacto del rotor de unos cuantos grados.

La cantidad de rotación es directamente proporcional al número de pulsos y la velocidad de rotación es relativa a la frecuencia de dichos pulsos.

Ventajas de un motor a pasos:

- Los motores de pasos no tienen escobillas, por lo cual no demandan mayor mantenimiento
- Facilitan un diseño efectivo y un bajo costo.
- Presentan alta confiabilidad, pero dependiente de la carga
- Posicionamiento en lazo abierto, pues en condiciones de carga ideales, su giro es confiable
- Pueden mantenerse estacionarios, es decir pueden detener un torque externo, dentro de los límites de carga
- Límite conocido al “error de posición dinámica”
- Arranque con carga
- Respuesta favorable a un escalón de arranque, respuesta rápida

Desventajas de un motor de pasos

- Desplazamiento dependiente de la construcción del motor
- Necesitan de un control de PWM cuando se mantiene energizada una bobina para evitar sobrecalentamiento del bobinado
- Fuerza limitada, en comparación a servomotores o motores de corriente continua de igual tamaño
- Pérdida de pasos cuando si se excede la carga nominal

1.2.4.3 Tipos de motores de pasos

Para proceder con el diseño de la mesa, se necesita establecer el tipo de motores de pasos que se utilizarán para mover los carros. Para esto debemos conocer los tipos

de motores existentes en el mercado, por lo que se analizará a continuación su clasificación.

Motor de pasos de reluctancia variable.- Tiene un rotor multipolar de hierro y un estator devanado laminado, y rota cuando los dientes del rotor son atraídos a los dientes del estator electromagnéticamente energizados. La inercia del rotor de un motor de paso de reluctancia variable es pequeña y la respuesta es muy rápida, pero la inercia permitida de la carga es pequeña. Cuando los devanados no están energizados, el par estático de este tipo de motor es cero. Poseen de tres a cinco bobinados conectados por un terminal común

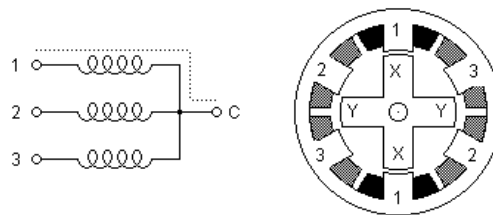


Figura 1.2.4.1. Motor de reluctancia Variable [2]

La figura 1.2.4.1 muestra un motor de reluctancia variable de treinta grados por paso, El rotor de este motor tiene cuatro dientes y el estator tiene seis polos. El diente del rotor “X” se encuentra magnetizado por la bobina “1”. El motor gira en sentido de las manecillas del reloj cuando “1” se desactiva y “2” se activa, entonces el diente del rotor “Y” es atraído por la bobina “2”. La secuencia de activación de un motor de reluctancia variable de 30° por paso, con giro en sentido horario, para una vuelta completa, es decir 12 pasos, se muestra en la tabla 1.2.4.2

Tabla 1.2.4.2. Secuencia de un motor híbrido

Bobina	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
1	1	0	0	1	0	0	1	0	0	1	0	0	1
2	0	1	0	0	1	0	0	1	0	0	1	0	0
3	0	0	1	0	0	1	0	0	1	0	0	1	0

Motor de pasos híbrido.- Se caracteriza por tener varios dientes en el estator y en el rotor, el rotor con un imán concéntrico magnetizado axialmente alrededor de su eje. Este tipo de motor tiene una alta precisión y alto par y se puede configurar para suministrar un paso angular tan pequeño como 1.8°. Su estructura básica se muestra en la figura 1.2.4.3

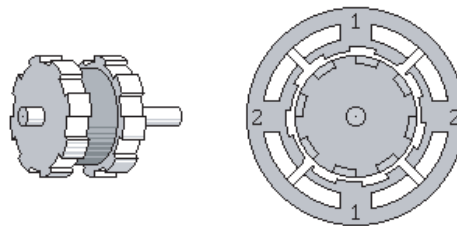


Figura1.2.4.3. Motor de pasos híbrido [2]

Motor de pasos bipolar.- Estos tienen generalmente 4 cables de salida. Para ser controlados requieren del cambio de dirección de flujo de corriente a través de las bobinas en la secuencia apropiada para realizar un movimiento. Presentan un buen torque, pero el bobinado tiende a calentarse excesivamente pues manejan corrientes elevadas (alrededor de 3 o 4 amperios por bobina).

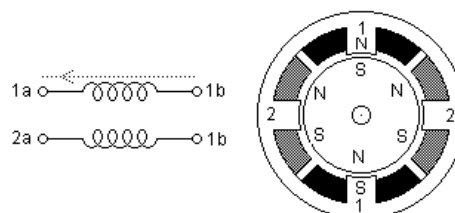


Figura1.2.4.4. Motor de pasos bipolar [2]

La figura 1.2.4.4 muestra un motor de pasos bipolar treinta grados por paso, en el cual, como se puede observar, el rotor tiene polaridades fijas, mientras que los cuatro dientes del estator deben cambiar su polaridad para atraer al rotor. La secuencia de activación de un motor bipolar de 30° por paso, con giro en sentido horario, para una vuelta completa, es decir 12 pasos, se muestra en la tabla 1.2.4.5.

Tabla 1.2.4.5. Secuencia para un motor bipolar

Bobina	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
1	V+	V-	V+	V-	V+	V-	V+	V-	V+	V-	V+	V-	V+
2	V+	V-	V-	V+	V+	V-	V-	V+	V+	V-	V-	V+	V+
3	V-	V+	V-	V+	V-	V+	V-	V+	V-	V+	V-	V+	V-
4	V-	V+	V+	V-	V-	V+	V+	V-	V-	V+	V+	V-	V-

Motor de pasos unipolar.- estos motores suelen tener 5 o 6 cables de salida dependiendo de su conexión interna. Este tipo se caracteriza por ser más simple de controlar que un motor bipolar. El torque obtenido por este tipo de motores no supera al bipolar, pero la facilidad de su manejo compensa esta deficiencia, además, puede manejar corrientes altas pero el bobinado, al ser dividido, no se calienta tanto como en el motor bipolar

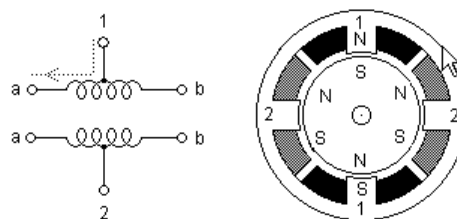


Figura 1.2.4.6. Motor de pasos unipolar [2]

La figura 1.2.4.6 muestra un motor de pasos bipolar treinta grados por paso, en el cual, como se puede observar, el rotor tiene polaridades fijas, mientras que los cuatro dientes del estator deben cambiar su polaridad para atraer al rotor, igual que en un motor bipolar, pero en este caso las bobinas tienen un común, el cual facilita el

control del cambio de dirección de la corriente en el devanado, así se pueden unir los comunes y conectarlos a Vcc, con lo cual la secuencia de activación de un motor bipolar de 30° por paso, con giro en sentido horario, para una vuelta completa, es decir 12 pasos, se muestra en la tabla 1.2.4.7.

Tabla 1.2.4.7. Secuencia para un motor unipolar (paso simple)

Bobina	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
1	1	0	0	0	1	0	0	0	1	0	0	0	1
2	0	1	0	0	0	1	0	0	0	1	0	0	0
3	0	0	1	0	0	0	1	0	0	0	1	0	0
4	0	0	0	1	0	0	0	1	0	0	0	1	0

Además éste tipo de conexión, nos permite elevar el torque de retención de un motor unipolar, lo cual se logra energizando dos bobinas a la vez. Esto se conoce como secuencia de paso completo, la cual se muestra en la tabla 1.2.4.8

Tabla 1.2.4.8. Secuencia para un motor unipolar (paso completo)

Bobina	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
1	1	0	0	1	1	0	0	1	1	0	0	1	1
2	1	1	0	0	1	1	0	0	1	1	0	0	1
3	0	1	1	0	0	1	1	0	0	1	1	0	0
4	0	0	1	1	0	0	1	1	0	0	1	1	0

1.2.5 Selección del motor apropiado

Existen muchos factores que se deben tomar en consideración al momento de elegir un motor de pasos para una aplicación, entre los cuales influyen los siguientes aspectos:

- Tipo de motor
- Torque requerido por el sistema
- Complejidad del controlador
- Características físicas del motor
- Stock nacional, debido a las limitantes en tecnología de nuestro país

Analizando estos factores, los motores que mejor se adaptan a nuestros requerimientos, y que sobretodo se pueden encontrar en el mercado nacional son los motores unipolares y bipolares. En la tabla se observa una comparación entre estos dos tipos de motores 1.2.5.1

Tabla 1.2.5.1. Comparación entre motores unipolares y bipolares

	Tipo de motor	
	Unipolar	Bipolar
Torque	Presentan un buen torque pero menor al de un bipolar de igual tamaño	Presentan un torque 30% mayor al de un unipolar
Control	Fácil	Más complejo que un unipolar
Costo	Costo bajo, debido a la simplicidad de control	Costo alto, debido a la necesidad de drivers

Por estas características se decidió utilizar motores unipolares, los cuales se muestran en la Figura 1.2.5.1

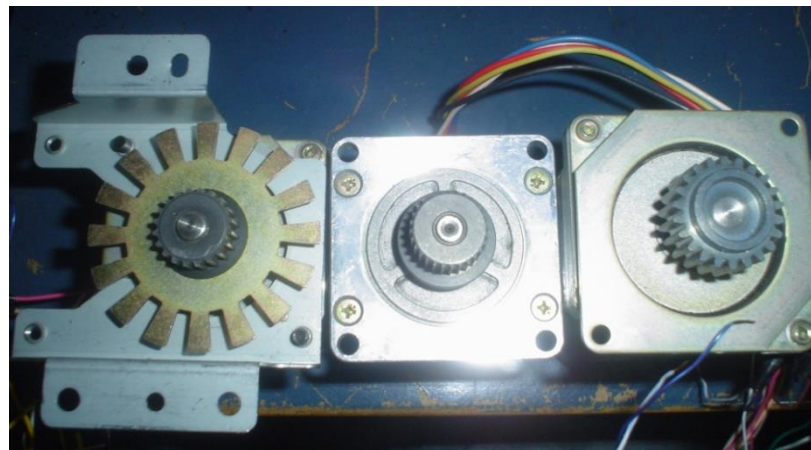
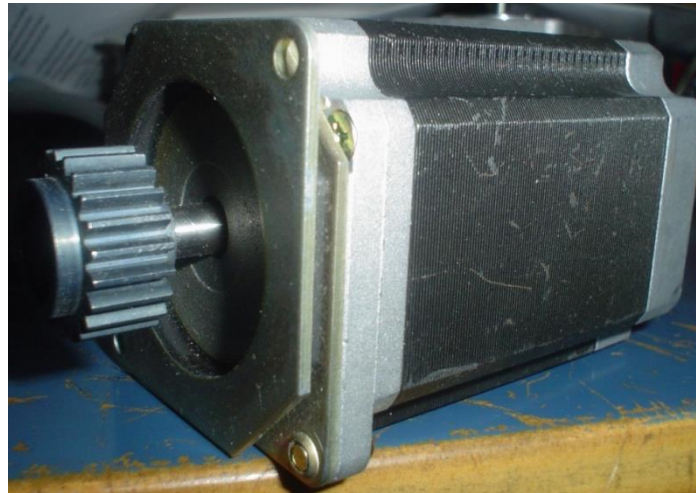


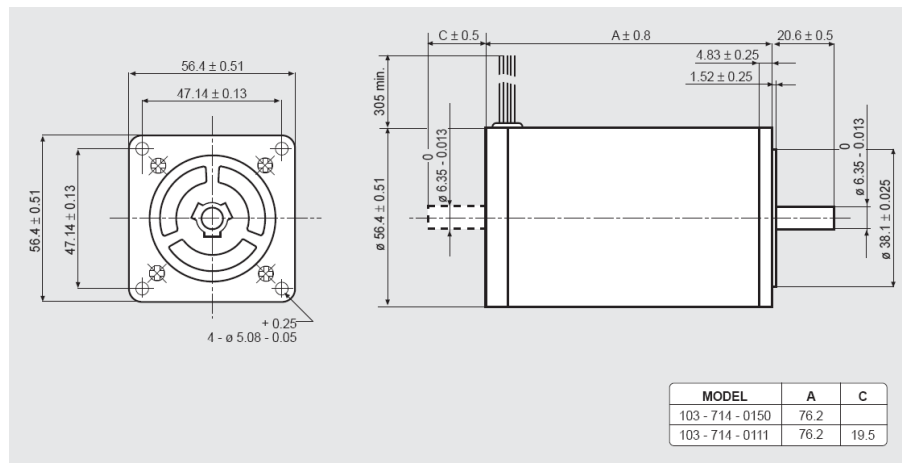
Figura 1.2.5.1 Motores a utilizar en los ejes

Motor 1

El primer motor se encargará de realizar el ascenso o el descenso del taladro, es decir el movimiento del eje z, en la figura 1.2.5.2 se muestra el motor.



(a)



(b)

Figura 1.2.5.2 características físicas del motor para el eje z (a) motor de pasos (b) dimensiones

Características

Corriente: 2 A por bobina

Resistencia: 2.6 Ω

Inductancia: 4.9 mH

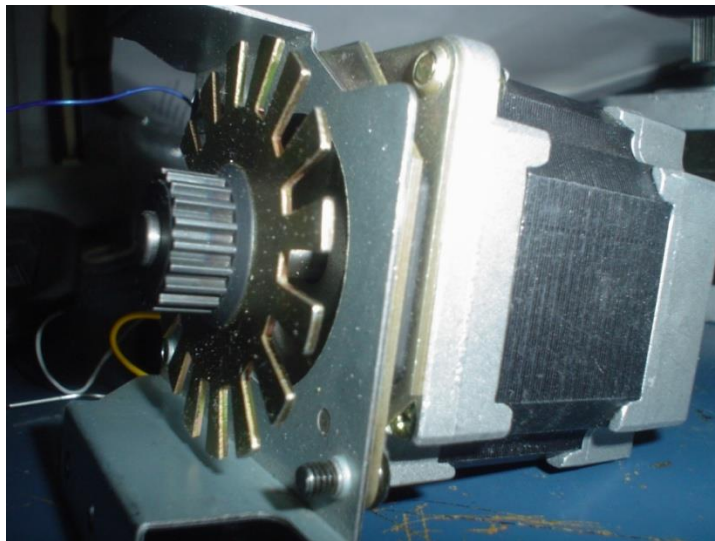
Torque: 88 Ncm

Peso: 0.95 kg

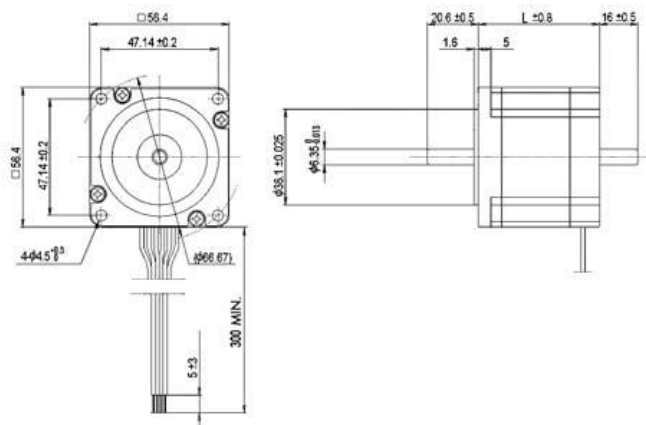
Función: Se encarga del acenso y descenso del eje z

Motor 2

El segundo motor se encargará de mover al carro del eje z y el eje y, es decir el eje x, en la figura 1.2.5.3 se muestra el motor.



(a)



(b)

Figura 1.2.5.3 características físicas del motor para el eje x (a) motor de pasos (b) dimensiones

Características:

Corriente: 3 A por bobina

Resistencia: 0.7Ω

Inductancia: 1.3 mH

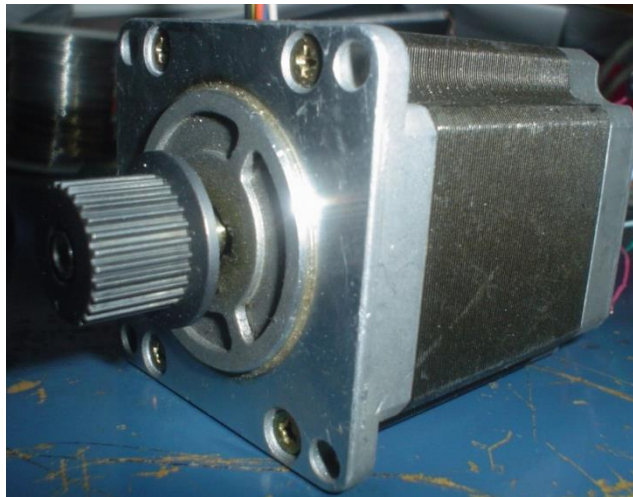
Torque: 1.3 kg.cm

Aceleración 51000 rad/seg^2

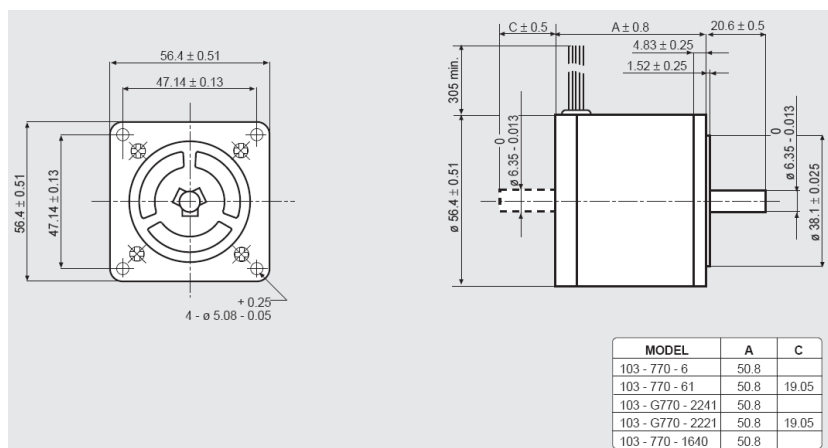
Peso: 0.71 kg

Motor 3

El segundo motor se encargará de mover al carro del eje z para adelante y para atrás, es decir el eje y, en la figura 1.2.5.4 se muestra el motor.



(a)



(b)

Figura 1.2.5.4 características físicas del motor para el eje y (a) motor de pasos (b) dimensiones

Características:

Corriente: 2 A por bobina

Resistencia: 2.4 Ω

Inductancia: 5.2 mH

Torque: 78 Ncm

Peso: 0.84 kg

1.2.6 Sistema sensorial

Para conseguir que el robot realice una tarea con adecuada precisión y velocidad es necesario que el sistema de control tenga conocimiento tanto de su estado interno como el de su entorno. El reconocimiento interno se logra mediante el uso de sensores internos utilizados en las articulaciones y el conocimiento del entorno se adquiere con sensores externos, en nuestro caso se utilizarán fines de carrera para este propósito

1.2.6.1 Sensor final de carrera

Dentro de los componentes electrónicos, se encuentra el final de carrera o sensor de contacto (también conocido como "interruptor de límite") o limit switch, son dispositivos eléctricos, neumáticos o mecánicos situados al final del recorrido de un elemento móvil, como por ejemplo una cinta transportadora, con el objetivo de enviar señales que puedan modificar el estado de un circuito. Internamente pueden contener interruptores normalmente abiertos (NA o NO en inglés), cerrados (NC) o conmutadores dependiendo de la operación que cumplan al ser accionados, de ahí la gran variedad de finales de carrera que existen en mercado.

Generalmente estos sensores están compuestos por dos partes: un cuerpo donde se encuentran los contactos y una cabeza que detecta el movimiento. Su uso es muy

diverso, empleándose, en general, en todas las máquinas que tengan un movimiento rectilíneo de ida y vuelta o sigan una trayectoria fija, es decir, aquellas que realicen una carrera o recorrido fijo, como por ejemplo ascensores, montacargas, robots, etc. Los finales de carrera están fabricados en diferentes materiales tales como metal, plástico o fibra de vidrio. En la figura 1.2.6.1 se puede observar un fin de carrera común en robots industriales.

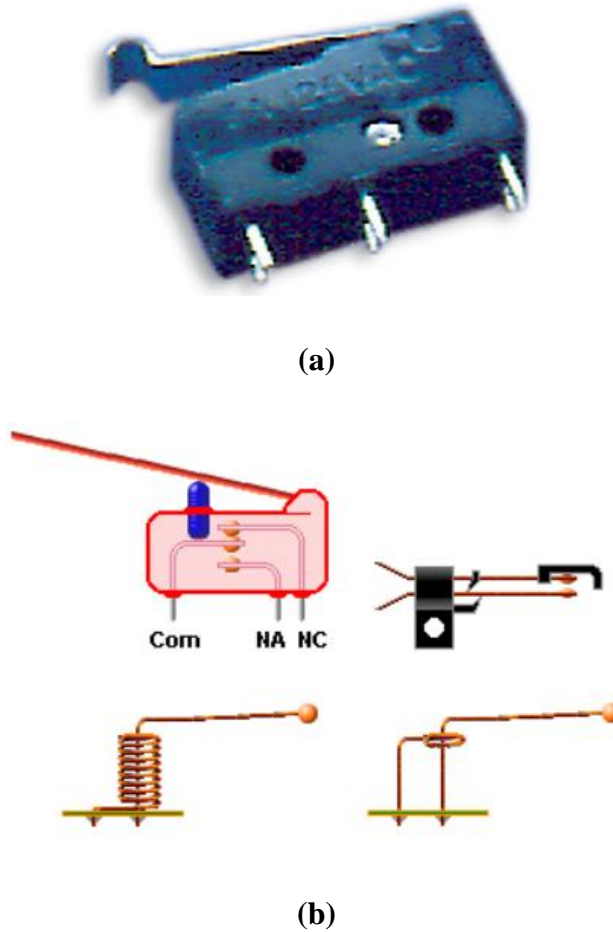


Figura 1.2.6.1. Fin de carrera de contacto (a) aspecto físico (b) funcionamiento interno

1.2.7 Elementos terminales

Los elementos terminales, o efectores finales, son los encargados de interactuar directamente con el entorno del robot. En el caso de robots industriales pueden ser

elementos de aprehensión; como pinzas o manipuladores, o pueden ser herramientas, como una fresadora, un taladro, etc.

El elemento terminal es específicamente diseñado para un tipo de trabajo, pero en el caso de herramientas esta puede ser un poco más flexible, ya que un mismo robot, dependiendo de su control, puede fresar o taladrar. En la tabla se especifican algunos tipos de herramientas utilizadas en robots industriales

Tabla 1.2.7.1. Herramientas terminales comunes en robots industriales

Tipo de herramienta	Generalidades
Pinza soldadura por puntos	Dos electrodos que se cierran sobre la pieza a soldar
Soplete soldadura al arco	Aporta el flujo de electrodo que se funde
Cucharon para colada	Utilizada en trabajos de fundición
Atornillador	Suelen incluir la alimentación de tornillos
Fresa-lijabroca	Para perfilar, eliminar rebabas, pulir, debastar, etc
Pistola de pintura	Por pulverización de pintura
Cañón láser	Para corte de materiales, soldadura o inspección
Cañón de agua a presión	Para corte de materiales.

En los extremos de una herramienta o en su acople con el robot, se suelen situar sensores para detectar estados en la misma, en el caso específico de una fresadora industrial se pueden acoplar sensores de velocidad para la fresa, sistemas de presión en el mandril, sensores de proximidad, etc

En el caso de nuestra máquina, la herramienta de trabajo es una fresadora, específicamente:

1.3 Máquinas CNC

Un robot se puede considerar de Control Numérico por Computador, o también llamado CNC (Computer Numerical Control) si es capaz de dirigir el posicionamiento de un elemento mecánico móvil mediante órdenes elaboradas de forma automática a partir de informaciones numéricas en tiempo real.

Entre las operaciones de maquinado que se pueden realizarse en una máquina CNC se encuentra el torneado y fresado de piezas. Sobre la base de esta combinación es posible generar la mayoría de las piezas necesarias en la industria.

1.3.1 Principio de funcionamiento del CNC

Para mecanizar una pieza se usa un sistema de coordenadas que especificarán el movimiento de la herramienta de corte. El sistema se basa en el control de los movimientos de la herramienta de trabajo con relación a los ejes de coordenadas de la máquina, usando un programa ejecutado por un ordenador, o en nuestro caso un microcontrolador.

En el caso de un torno, hace falta controlar los movimientos de la herramienta en dos ejes de coordenadas: el eje de las X para los desplazamientos laterales del carro y el eje de las Z para los desplazamientos transversales de la torre.

En el caso de las fresadoras se controlan los desplazamientos verticales, que corresponden al eje Z. Para ello se incorporan motores eléctricos en los mecanismos de desplazamiento del carro y la torreta, en el caso de los tornos, y en la mesa en el caso de la fresadora; dependiendo de la capacidad de la maquina, esto puede no ser limitado únicamente a tres ejes.

En la figura se muestra una fresadora CNC de seis ejes. Este tipo de máquinas reducen el tiempo de mecanizado de piezas en la industria y disminuyen al máximo

los errores de precisión, lo cual es difícil de lograr en una línea de producción manual.



Figura 1.3.1. Fresadora de seis ejes modelo DMU 80P, cortesía de DMG Electronics

Aparte de aplicarse en las máquinas-herramienta para modelar metales, el CNC se usa en la fabricación de muchos otros productos de ebanistería, carpintería, etc. La aplicación de sistemas de CNC en las máquinas-herramienta han hecho aumentar enormemente la producción, al tiempo que ha hecho posible efectuar operaciones de conformado que era difícil de hacer con máquinas convencionales, por ejemplo la realización de superficies esféricas manteniendo un elevado grado de precisión dimensional. Finalmente, el uso de CNC incide favorablemente en los costos de producción al propiciar la baja de costes de fabricación de muchas máquinas, manteniendo o mejorando su calidad.

1.3.2 Características del C.N.C

Ventajas de la automatización mediante CNC:

- Mayor precisión y mejor calidad de productos.
- Mayor uniformidad en los productos producidos.
- Un operario puede operar varias máquinas a la vez.

- Fácil procesamiento de productos de apariencia complicada.
- Flexibilidad para el cambio en el diseño y en modelos en un tiempo corto.
- Fácil control de calidad.
- Reducción en costos de inventario, traslado y de fabricación en los modelos y abrazaderas.
- Es posible satisfacer pedidos urgentes.
- No se requieren operadores con experiencia.
- Se reduce la fatiga del operador.
- Mayor seguridad en las labores.
- Aumento del tiempo de trabajo en corte por maquinaria.
- Fácil control de acuerdo con el programa de producción lo cual facilita la competencia en el mercado.
- Fácil administración de la producción e inventario lo cual permite la determinación de objetivos o políticas de la empresa.
- Permite simular el proceso de corte a fin de verificar que este sea correcto.

Desventajas:

- Alto costo de la maquinaria.
- Falta de opciones o alternativas en caso de fallas.
- Es necesario programar en forma correcta la selección de las herramientas de corte y la secuencia de operación para un eficiente funcionamiento.
- Los costos de mantenimiento aumenta, ya que el sistema de control es más complicado y surge la necesidad de entrenar al personal de servicio y operación.
- Es necesario mantener un gran volumen de producción a fin de lograr una mayor eficiencia de la capacidad instalada.

1.3.3 Ejes principales de una máquina CNC

En la descripción de las MHCNC (máquinas herramientas de control numérico computarizado) se utiliza siempre el concepto de "eje", es decir, direcciones de los

desplazamientos principales de las partes móviles de la máquina como la mesa portapiezas, cabezal, torreta

- Eje Y: desplazamiento transversal del carro portador del cabezal de dentro a fuera
- Eje Z: desplazamiento transversal de la mesa portapiezas de arriba abajo.
- Eje X: desplazamiento transversal de la mesa portapiezas de izquierda a derecha

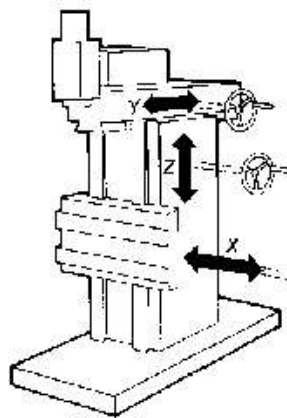


Figura1.3.3.1. Fresadora CNC, ejes de movimiento

Las MHCN están provistas de un número de ejes principales característico que hace factibles los trabajos de mecanizado sobre la pieza. Estos ejes se designan convencionalmente como X, Y y Z

Generalmente las maquinas convencionales tienen de dos a tres ejes de desplazamiento, como los tornos y las fresadoras respectivamente, pero, en trabajos de mecanizado de formas complejas se requieren MHCN dotadas de más ejes de desplazamiento.

La designación y descripción de los ejes de cada tipo de MHCNC se encuentra normalizada. La disposición de los carros móviles en las MHCNC puede ser muy sofisticada, dando origen a una gran variedad de diseños / modelos tanto en fresadoras como tornos.

Los fabricantes de MHCNC determinan dichas disposiciones en función de los requerimientos en cuanto a capacidad de carga y precisión de posicionado. Esta disposición viene condicionada por:

- La forma de la trayectoria a recorrer.
- Las propiedades de las superficies de contacto.
- Las exigencias de apriete o sellado

Los recorridos de la herramienta en el seno de la pieza se originan por la acción combinada de los desplazamientos en cada uno de sus ejes principales como se puede ver en la figura

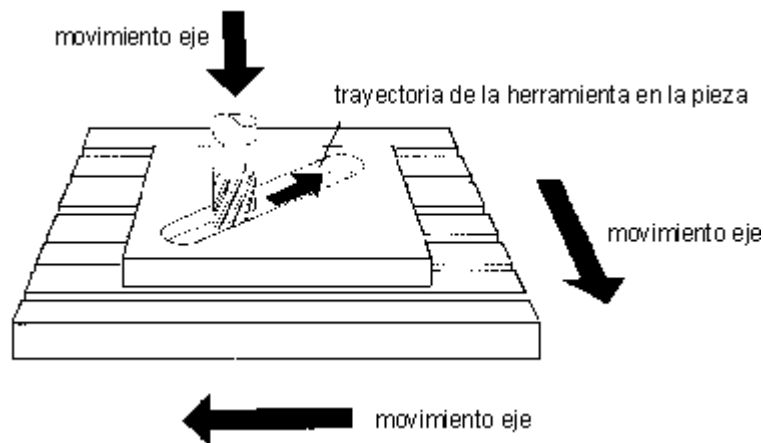


Figura 1.3.3.2. Mesa de trabajo de una máquina CNC

Los sistemas de transmisión producen traslaciones rectilíneas en los ejes principales a partir del giro básico generado por el grupo del motor-reductor. Consiste en un sinfín acanalado y un acoplamiento al que se fija el conjunto mecánico a desplazar. Cuando el grupo del motor gira, su rotación se transmite al sinfín y el cuerpo del acoplamiento se traslada longitudinalmente a través de este arrastrando consigo a la mesa de trabajo en el sentido oportuno.

Para conocer la posición exacta de cualquier elemento móvil de una MHCNC a lo largo de un eje de desplazamiento se emplean un conjunto de dispositivos electrónicos y unos métodos de cálculo. Estos elementos constan, básicamente, de una escala graduada (similar a un escalímetro) y el resolver capaz de "leer" dicha escala. Atendiendo a al método de lectura y forma de la escala se distingue entre:

Posiciones absolutas.- para la medición de los desplazamientos supone que las posiciones estimadas son independientes del estado puntual de la máquina o de su

control al estar referido a un punto invariante conocido como “origen absoluto o cero máquina”.

Posiciones por incrementos.-(incremento = desplazamiento pequeño de longitud fija) se emplea para designar los movimientos relativos a algún punto significativo distinto del origen absoluto y que, además, puede variar. Durante el movimiento la UC lleva a cabo un conteo del número de incrementos (divisiones) en las que la nueva posición difiere de la anterior.

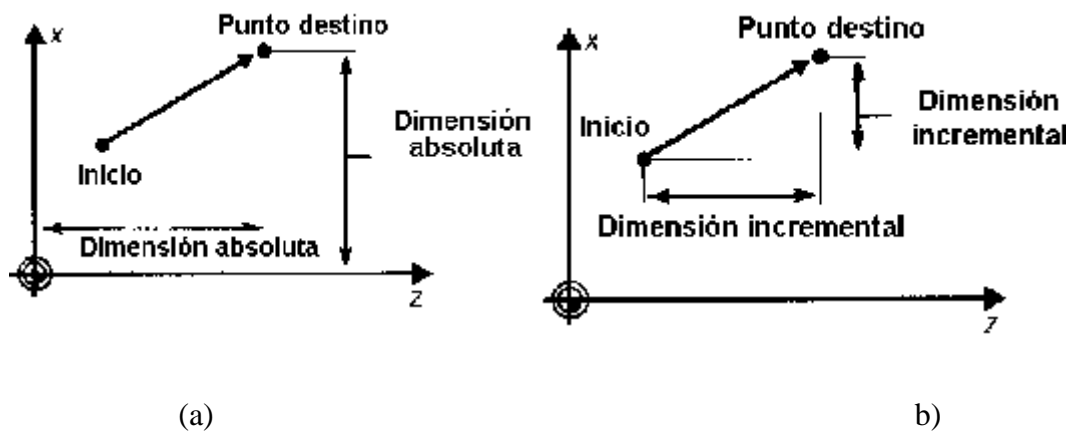


Figura 1.3.3.3 Tipos de incrementos (a) Dimensión absoluta (b) dimensión por incrementos

La medición de posiciones absolutas emplea un sistema de escalas codificadas y ordenadas por múltiplos similares a un escalímetro. Para conocer la posición actual del desplazamiento se hace siempre referencia al cero máquina (origen absoluto) que es un punto físico, conocido e invariante de la MHCN.

Es imprescindible que la lectura pueda llevarse a cabo en todo el rango de desplazamiento del eje en cuestión. A cada posición definida dentro de ese rango la UC le asigna un valor numérico.

La escala se codifica generalmente en sistema binario. La medición de posiciones por incrementos emplea una escala con un sistema de división simple. La rejilla esta

dividida en sectores blanco / negro sobre los que pasa el resolver durante el movimiento. Este cuenta el número de sectores blanco / negro obteniendo el valor del desplazamiento por diferencia respecto a su posición previa. Para garantizar que la medida se realiza correctamente, inmediatamente después de inicializarse la UC se debe de medir la posición inicial respecto al cero máquina. A esta posición de inicio se le conoce como "punto de referencia". Tan pronto como la máquina a asignado el punto de referencia el resolver comienza a suministrar posiciones relativas al último punto mediante lectura / conteo de la escala.

1.3.4 Sistemas de sujeción

En fresado se emplean las siguientes formas de sujeción:

- Sargentos y apoyos con formas escalonadas, ajustables en altura o bloques con varias facetas de contacto, con pernos y resortes de apriete de montaje-desmontaje rápido.
- Placas angulares de apoyo.
- Palancas de apriete. Mordazas mecánicas autocentrables
- Platos o mesas magnéticas.
- Mesas y dispositivos modulares de uso universal.
- Apoyos de diseño específico o especial.

Los dispositivos de sujeción permiten asegurar la pieza a la mesa de trabajo. El número de funciones controlables que están relacionadas con estos sistemas depende de la forma de alimentación de piezas (manual o automática) y de la complejidad del sistema de amarre.

También se puede establecer por programa la presión de cierre de las garras. La elección de la fuerza de apriete depende generalmente de la velocidad de giro del cabezal; velocidades elevadas demandan las presiones mayores al aumentar la acción de la fuerza centrífuga. Como es habitual que las MHCN trabajen a velocidades de giro (corte) elevadas y esto podría suponer presiones que dañasen la pieza, estas

incorporan mecanismos de compensación de las fuerzas centrífugas. El diseño de las mismas se basa de mantener una presión estable del accionamiento de cierre hidráulico a velocidades de giro elevadas.

En fresado las presiones de apriete no resultan tan críticas. El aspecto más crítico en la sujeción en estas máquinas es la rapidez de montaje / desmontaje y la precisión en el posicionado de la pieza en la mesa de trabajo.

El sistema de amarre debe permitir una fácil carga / descarga de la pieza de trabajo y garantizar la repetitibilidad en la colocación estable y precisa de la misma en el seno de la MHCN. Compatibilizar todo ello puede resultar costoso en tiempo y dinero.

Los sistemas de sujeción específicos mediante componentes normalizados y modulares se utilizan frecuentemente. Estos dispositivos deben permitir el mecanizado completo sin operaciones de montaje / desmontaje

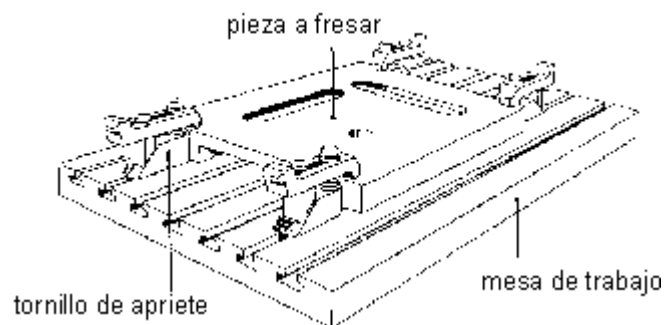


Figura 1.3.4.1. Sistema de sujeción de piezas en la mesa de una fresadora

1.4 El CNC en la fabricación de circuitos impresos

1.4.1 Proceso de Fabricación de un circuito impreso

El sistema de producción involucrado en la fabricación de circuitos impresos dividido en seis áreas



Figura 1.4.1.1. Diagrama de flujo del proceso de fabricación de circuitos impresos

Programación y Perforación

Se da inicio al proceso productivo, su función principal es realizar el programa del circuito impreso indicando las perforaciones según el diseño, los diámetros de las brocas, medidas y estructuras de la placa, así como la velocidad de perforación.

Todas estas especificaciones se generan cuando el usuario realiza su esquema y lo convierte a un PCB (printed circuit board) en cualquier software, entre los más utilizados: Altium Designer Protel, Proteus, y EAGLE, los cuales generan un archivo de texto, en el cual están las posiciones de los agujeros a realizar, las pistas, y el tipo de broca a utilizar, todo codificado en código G (Gerber), el cual se tratará más adelante.

Metalización

Se encarga de lijar, metalizar electrolíticamente, atacar y fundir los circuitos que recibió de perforación. Los siguientes procesos que se lleva a cabo dentro de esta área son:

Deposito de cobre electrolítico: se deposita cobre en las perforaciones por medio de electrolisis en las pistas e isletas, esta con la finalidad de que estas adquieran una mejor resistencia a la conductividad.

Deposito de plomo / estaño: las pistas e isletas adquieren una segunda capa de estaño para hacerlas mas resistentes y proteger el cobre de alguna oxidación y brindar una mejor soldabilidad de los componentes.

Desentintado: se realiza con una solución de oxido de sodio y su función es descubrir el cobre que no será utilizado para que sea atacado.

Atacado: Consiste en desprender el cobre de las placas, para que solo quede la fibra de vidrio o la baquelita y se aprecien únicamente las pistas e isletas del circuito impreso. Dependiendo del tipo de máquina el atacado puede ser químico o mecánico.

- 1. Atacado Químico:** Es el método utilizado en máquinas industriales para producción en serie. Consiste en atacar al cobre con ácido clorhídrico o cloruro férrico dejando sólo las pistas y retirando el cobre sobrante
- 2. Atacado mecánico o por Fresado:** Este método es utilizado para producción de prototipos (pequeña escala), consiste en retirar el cobre sobrante y trazar las pistas y agujeros con una fresadora CNC de pequeñas dimensiones

Control de Calidad

Es una de las áreas más importantes en la fabricación el circuito impreso ya que tiene a su cargo la responsabilidad de revisar el trabajo efectuado. Se analiza las placas mediante señales eléctricas para encontrar defectos en el ataque del cobre.

Impresión

Aplica sobre las placas unas capas de pintura, llamadas de primera impresión, mascara y leyendas, las cuales son necesarias para que el circuito tome la vista de cómo quedara al termino del proceso productivo.

Acabado

Recibe las placas del área de impresión y realiza los cortes exactos especificados a través de la maquina routeadora. Como podemos observar el proceso para llevar un circuito electrónico desde el esquemático hasta la tarjeta de circuito impreso es complicado para quien sólo realiza prototipos. Entre las opciones que se tiene para baja escala de producción, está el procesado químico netamente artesanal (en el caso de estudiantes y aficionados), el cual presenta resultados favorables para circuitos simples, pero cuando se dispone de un diseño complejo, el método artesanal es ineficiente, en este caso el ruteado en máquinas CNC es la mejor opción.

En cuanto a las máquinas de ruteado, estas consisten en mesas (robots cartesianos) de tres ejes, los cuales toman el archivo gerber (el cuál será explicado en el capítulo pertinente) generado por los programas más comunes de diseño de circuitos impresos, y mediante la fresa en el cabezal desbasta el cobre sobrante de una placa virgen.

1.4.2 Máquinas de prototipos

En el mercado existen muchas opciones de centros de mecanizado para circuitos impresos. En la figura 1.4.2.1 se muestra el modelo ProtoMat s42 de LPKF. Este modelo de fresadora CNC está diseñada para trabajar en materiales suaves, como el cobre y la madera, por lo tanto se puede decir que es una versión más simple de una fresadora CNC industrial.



Figura 1.4.2.1. Fresadora de prototipos ProtoMat s42. Cortesía de LPKF

Este tipo de dispositivos disponibles en la industria ofrecen las siguientes ventajas:

- Área de trabajo, 229mm x 305mm,
- Motor AC monofásico de velocidad variable, de hasta 42.000 rpm
- Mínimo tamaño de Pista/Aislación: 0,1mm/0,1mm (4 mils)
- Mínimo tamaño de orificio: 0,15mm
- Instalación de Herramienta manual
- Incluye paquete de software CircuitCAM S42
- Materiales de trabajo: FR3, FR4, G10 y plástico
- Conexión USB al PC a través RS 232
- Recorrido a través del eje X, Y y Z

Características de Funcionamiento

- 50mm de desbaste por segundo
- Motor con variación de velocidad
- de hasta 42.000 rpm.
- 8 Orificios por segundo

1.5 Selección del diseño de mesa

Para el desarrollo de este proyecto es necesario escoger un modelo de mesa. Existen dos modelos básicos para una fresadora de prototipos, los cuales radican su diferencia en cómo realizan el movimiento cartesiano, lo cual depende del tamaño del área de trabajo y del tamaño y peso de las torres de sujeción de la fresadora.

1.5.1 Fresadora basada en el movimiento del área de trabajo

El primer tipo de mesa, la cual basa su movimiento en el área de trabajo, se muestra en la figura 1.5.1.1.

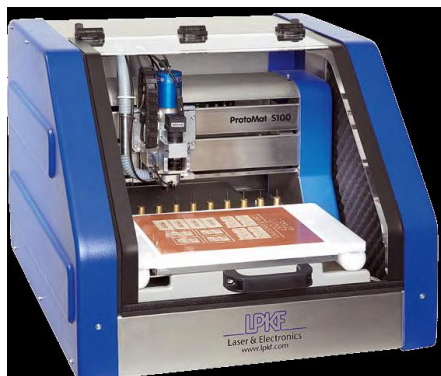


Figura 1.5.1.1. Fresadora de mesa con movimiento del área de trabajo cortesía de LPKF electronics

Como podemos observar, el eje Z es fijo, mientras los ejes X e Y son los que se mueven para realizar el fresado, es decir la fresa se mantiene fija, mientras que la placa se desliza por debajo de la fresa, lo que ocasiona el desgaste del cobre. Este diseño es propio para áreas de trabajo pequeñas.

1.5.2 Fresadora basada en el movimiento de los ejes

El segundo tipo de mesa, la cual basa su movimiento en los ejes, se muestra en la figura 1.5.2.1.

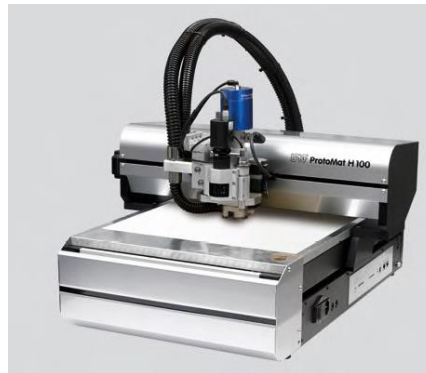


Figura 1.5.2.1. Fresadora de mesa con movimiento de ejes cortesía de LPKF electronics

Como podemos observar, el eje X es fijo, mientras los ejes Y y Z son los que se mueven para realizar el fresado, es decir la fresa se desliza por la placa para retirar el cobre, mientras que la placa se queda fija. Este diseño se utiliza para áreas de trabajos relativamente grandes, y para piezas de considerable altura.

Por lo tanto, observando ambos modelos y teniendo en cuenta que la mesa de trabajo que se va a diseñar debe tener un área útil de trabajo amplia, se escogió el segundo tipo de mesa.

1.6 Diseño y construcción

El diseño empezó planteando los materiales a trabajar y el tamaño del área de trabajo, es una maquina pensada para materiales blandos, como ser madera, aglomerados acrílico, plásticos, polifan, pero principalmente entraría la parte del perforado de PCB, con un área de trabajo del orden de los **350 x 435** mm con un recorrido en el eje Z de unos 150mm, las guías tienen un algo mayor según el ancho de cada carro en dicho eje. También la idea sería el grabado en metales blandos como ser Aluminio cobre o bronce, no siendo la finalidad primordial el mecanizado de metales, para esto habría que replantear de entrada el tipo de guías a utilizar, sobre todo en estas dimensiones, para no tener pandeos en las guías o vibraciones al trabajar materiales duros.

Para las Guías se utilizó un sistema propio en base de rodamientos de bolas que recorren el eje. Las medidas de las barras son las siguientes:

Eje X: Barra \varnothing 1/2 pulg de 600mm

Eje Y: Barra \varnothing 1/2 pulg de 400mm

Eje Z: Barra \varnothing 1/4 pulg de 200mm

Materiales a utilizar para la construcción

- 1 Plancha de aluminio de 500mm x 1000mm x 10mm para la construcción de soportes
- 2 m de hierro para eje de 1/2 pulg
- 500mm de hierro para eje de 1/4 pulg
- 8 m de perfil de aluminio (ver anexo)
- 2.5m de tornillo rosca withword de 14 hilos/cm
- 100 tornillos tipo allen de 2 pulgadas

- 12 tuercas M10
- 6 tuercas M12
- 15 rodamientos M12
- 10 rodamientos M9

Herramientas:

- 1 juego de machuelos M4
- 1 juego de machuelos M5
- 1 porta machuelos
- 1 terraja M10
- 1 terraja M12
- 1 porta terrajas
- 1 Fresa M12
- 1 Fresa M10
- 1 Fresa 1 pulg
- 1 broca M4
- 1 broca M5
- 1 broca M10
- 1 broca M12
- 1 broca M22
- 1 broca M46
- 1 broca de centro
- 1 cuchilla de acero rápido
- 1 cuchilla para interiores

Maquinaria:

- Torno
- Fresadora
- Taladradora

- Cierra de cinta
- Cierra circular
- Lijadora
- Cierra de mano
- Lima

Aparatos de medición

- Calibrador
- Micrómetro
- Reloj comparador
- Cabezal divisor y punto

Otros:

- Aceite
- Taladrina

Con esto llegamos al diseño de la mesa que se muestra en la figura 1.6.1. El diseño fue realizado en Inventor 2009.

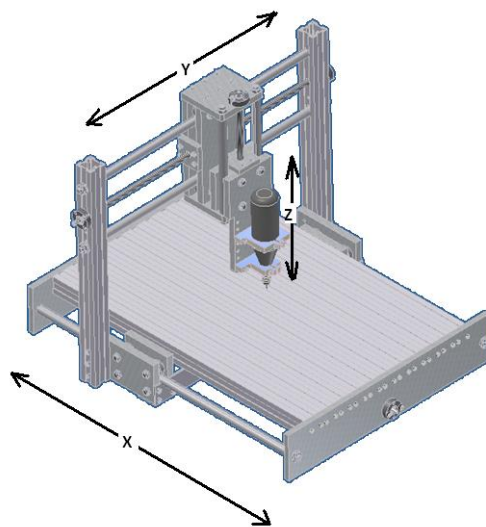
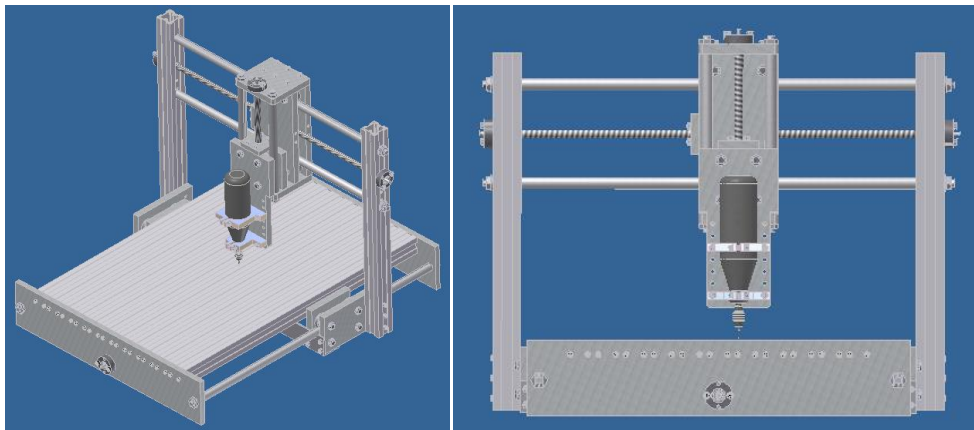
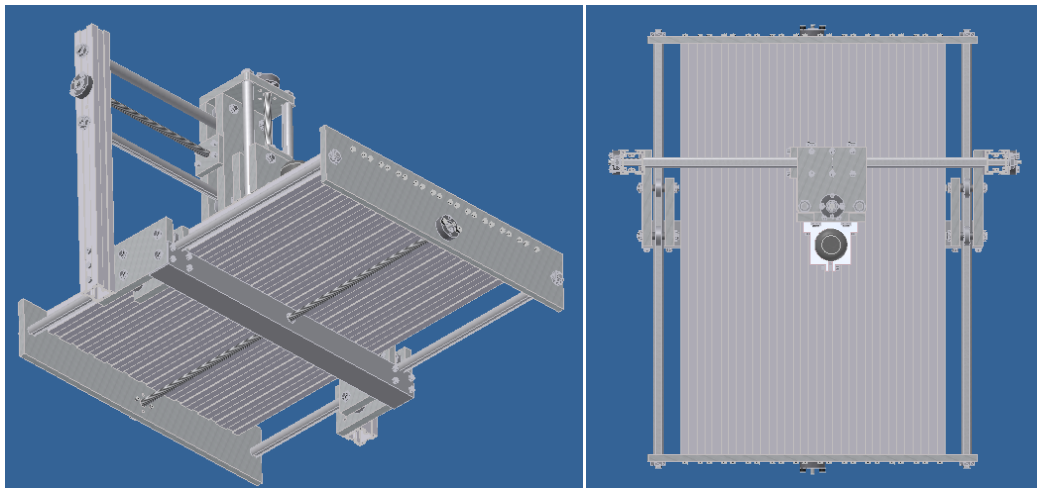


Figura 1.6.1 Diseño final de la mesa



a

b



c

d

Figura 6.1.2 Vistas de ensamblaje

1.6.1 Eje z

Para el diseño del eje z y el carro de sujeción del taladro se tomó como referencia el recorrido planteado al inicio del diseño. De esta manera el eje consta de los siguientes elementos.

1. Tapas de sujeción superiores e inferiores
2. Tapas laterales para sujeción de rodamientos
3. Tapas para sujeción de la herramienta
4. Ejes de deslizamiento para ascenso y descenso del taladro
5. Tornillo sin fin para movimiento del taladro
6. Rodamientos para el desplazamiento en el eje Y
7. Sujeción ajustable para la herramienta

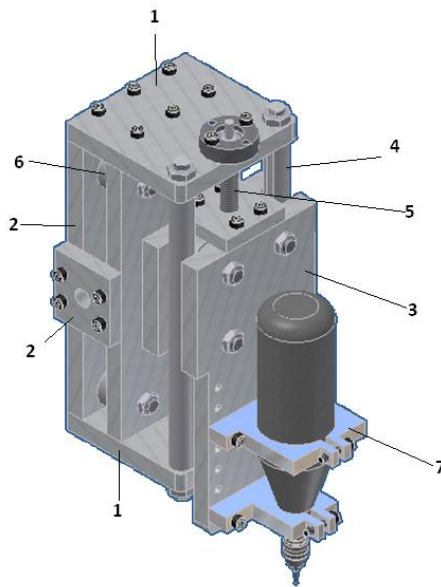


Figura 1.6.1.1 eje Z

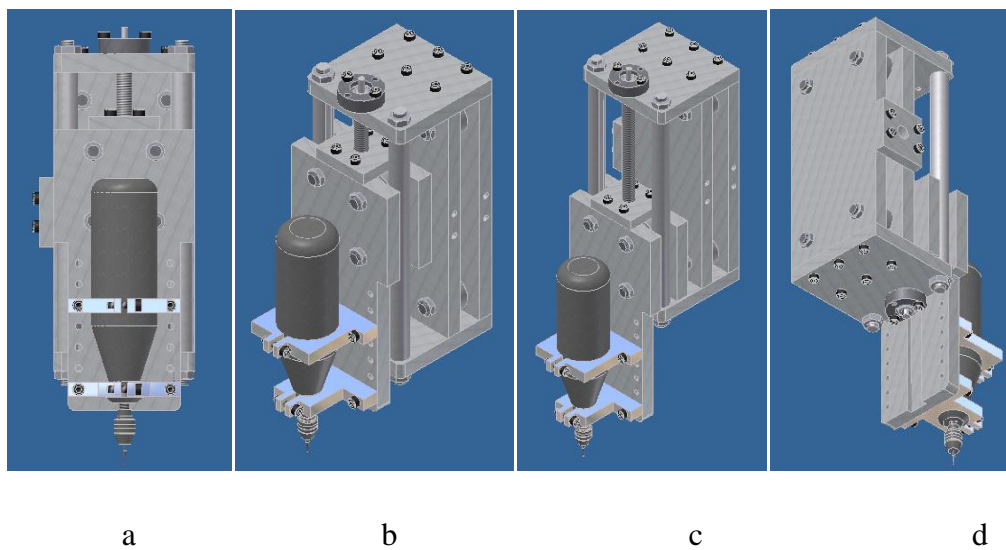


Figura 1.6.1.2 vistas y recorrido del eje z

1.6.2 Carro para el eje X e Y

Para el diseño del carro que mueve la herramienta por el eje “x” e “y” del taladro se tomó como referencia el área útil planteada al inicio del diseño. De esta manera el carro consta de los siguientes elementos:

1. Torres de sujeción del carro
2. Tornillo sin fin para movimiento
3. Estructura con rodamientos para deslizamiento sobre el eje x
4. Soporte para movimiento sobre el eje X

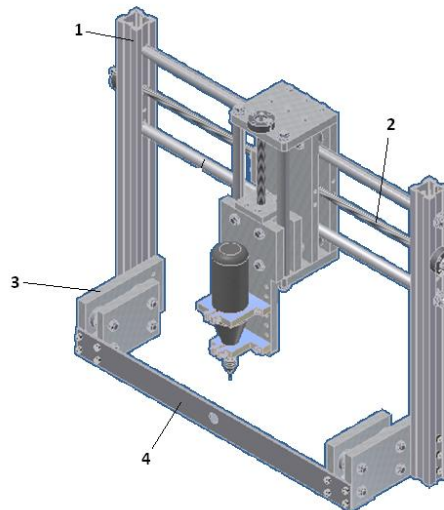
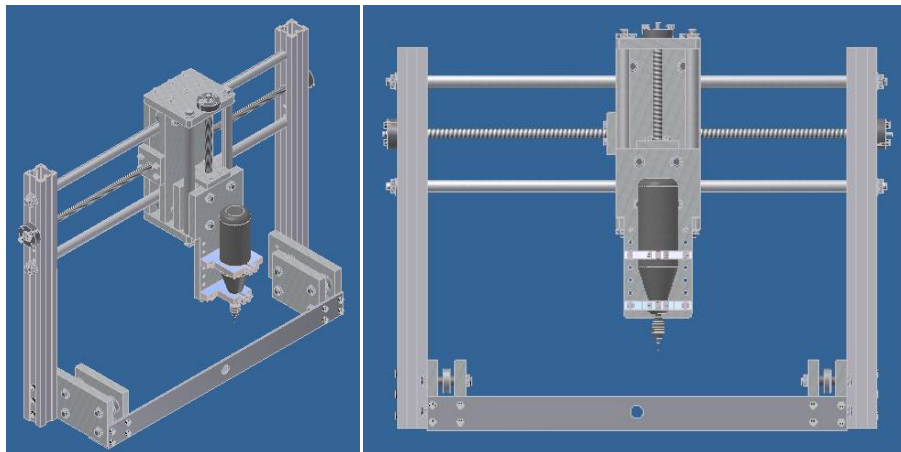
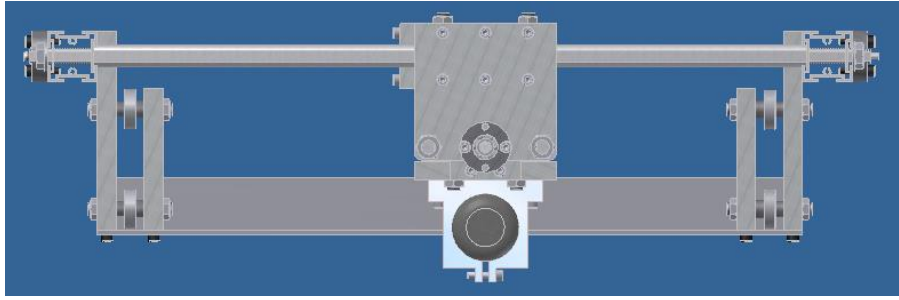


Figura 1.6.2.1 Carro de sujeción para movimiento sobre ejes X-Y



a

b



c

Figura 1.6.2.2 Vistas y recorrido eje X-Y

1.6.3 Mecanizado



Figura 1.6.3.1 Taladrado y fresado de soportes

Para la transmisión se optó por tornillos de rosca whitworth de 3/4 de pulgada de diámetro y 1/4 de paso, que serían unos 19mm de diámetro y paso 6.35mm. La idea

era utilizar un tornillo de buen paso para obtener altas velocidades, lo ideal sería usar un tornillo de bolas recirculantes, pero estaba fuera del presupuesto, esto de dejo para un futuro, ya que la actualización es muy simple.

Figura 1.6.3.2 Torneado de tornillos sin fin

El tornillo que mueve el carro de cada eje al girar convierte el movimiento radial del motor en uno lineal, por ende en el tornillo aparecen cargas axiales, o sea fuerzas hacia los costados en sentido contrario al que se mueve el Carro del eje en cuestión. Es importante contar con un sistema capaz de absorber esos esfuerzos sin juego y permitiendo girar libremente al Tornillo.



Figura 1.6.3.3 Fresado del eje z



Figura 1.6.3.4 Fresado de ejes X e Y

En el Extremo opuesto tenemos rodamientos, es un rodamiento radial de los mas comunes, un 6200, solo se va a limitar a hacer de guía para el tornillo y aguantar las cargas justamente radiales que le va a oprimir la transmisión por correa al motor Paso a Paso. En este lado el tornillo también esta torneado a 10mm para encastrar en el rodamiento y luego a 8mm que es donde se montara la polea dentada, fíjense que el tornillo no hace tope sobre este rodamiento sino que la parte del torneado de 10mm comienza unos 2mm antes, de esta manera no hace falta una media tan estricta en el largo de los torneados ya demás esa luz absorbe cualquier dilatación que pueda tener el tornillo, recordemos que la sujeción axial esta en el otro extremo.



Figura 1.6.3.5 Tornillo sin fin torneado y tuercas de sujeción

Los laterales del puente están hechos en acero de 8mm de espesor, en la parte inferior tiene un ancho de 200 mm que es la separación que se le dio a los rodamientos lineales para el eje X y de alto tienen unos 400 mm. Las guías del Eje X por donde corre el puente están sujetas por dos plegados en chapa de 4.75mm de espesor en forma de L que a su vez hace de apoyo para la maquina y también sujetan los perfiles de aluminio que conforman la mesa. Por debajo del puente va un perfil en forma de u también hecho con un plegado de chapa, en este caso en chapa de 3mm, sobre este perfil ira montada la tuerca que mueve todo el puente. La mesa esta formada por 6 perfiles, de 40 x 80mm y uno central de 40 x 40, se optó por esto ya que son muy rígidos y sobre todo son los únicos disponibles en el país. Para sujetar los Perfiles a los plegados en L de los extremos se rosco sobre las perforaciones internas que tienen con un macho M12. De amañera similar están sujetas las barras de las guías a este mismo perfil y sobre los laterales



Figura 1.6.3.6 Soporte de las torres de los ejes Y-Z

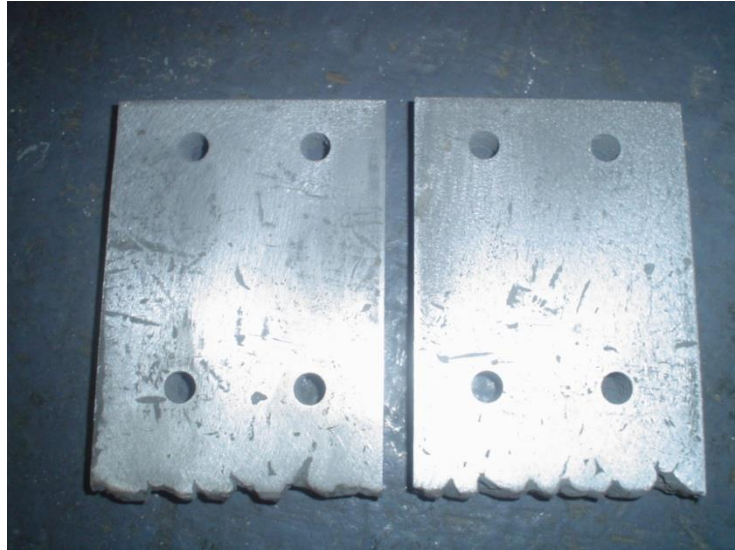


Figura 1.6.3.7 Estructura para los rodamientos

El Eje Z esta construido íntegramente en aluminio, de 12mm para las partes donde se sujetan los soportes de los rodamientos lineales y 10mm las tapas que sujetan las barras, los soportes de los motores de los 3 ejes también están hechos en aluminio de 10mm y todas estas piezas han sido mecanizadas por CNC. Cabe mencionar el trabajo de roscado, toda la maquina esta armada sin ninguna tuerca, todo va roscado, y se utilizado en su mayoría tornillos Allen de 6mm salvo para sujetar los perfiles de aluminio de la mesa que por el diámetro del agujero la rosca a realizar fue de 12mm y en los soporte de los rodamientos de los tornillos que su uso tornillos Allen de 4mm.



Figura 1.6.3.8 Tornillos Allen de sujeción

Como ya adelante, la transmisión entre los motores y los tornillos esta hecha por correas y poleas dentadas. Esto surgió por varios motivos, el primero es que es un excelente acople, nunca es recomendable un acople mecánico totalmente rígido, si se puede poner directo al tronillo pero siempre es aconsejable usar un acople flexible para evitar el deterioro de los rodamientos y perdidas de fuerza si no queda exactamente alineado, permite jugar con las reducciones cambiando las poleas, y por último ofrece la posibilidad de colocar el motor hacia el costado con lo cual queda mas compacta la maquina, sino con el motor a continuación del tornillo mas el largo del acople nos roba unos cuantos centímetros de espacio. Se utilizaron correas de paso T5 de 10mm de ancho montando una reducción de 1:1.2 para tener un paso que ronde los 5mm por vuelta de motor, exactamente se tienen 5.291mm, se podría haber usado una reducción 1:1 sin problemas.

Figura 1.6.3.9 Soporte del carro para el eje z



Figura 1.6.3.10 Sujeción inferior de herramienta

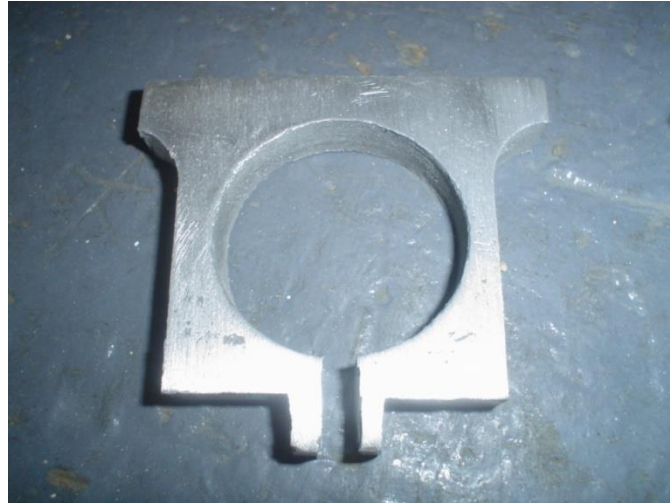


Figura 1.6.3.11 Sujeción superior de herramienta

Los Motores utilizados son de 60mm de brida, de 3.1Nm de torque en configuración bipolar paralela que consume unos 2.8a. Por ultimo en lo que hace a la estructura de la Máquina se coloco una chapa de aluminio de 3mm en la parte trasera del puente uniendo los dos laterales, esto es para rigidizar mas el puente ya que actúa como escuadra entre ambos laterales manteniéndolos siempre paralelos. Y como se ve en las fotos se ha montado las cadenas porta cables y todos los cables de los motores, finales de carrera y alimentación del husillo. Dejando por delante solo la instalación del Husillo, prueba de la maquina y desarme posterior para pintar las partes de acero y terminar con el ensamble final.

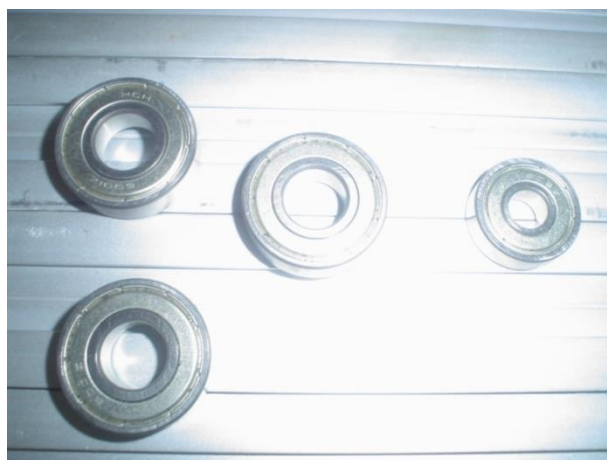


Figura 1.6.3.11 Rodamientos para los ejes



Figura 1.6.3.12 Rodamientos para la estructura

1.6.4 Montaje



Figura 1.6.3.1 Eje Z



Figura 1.6.3.2 Mesa

1.7 Análisis de precisión mecánica

Calculo de desplazamiento

Tornillo tiene 13 hilos por pulgada, los motores a pasos son de 1.8° o 200 pasos por vuelta y no existe ningún tipo de reducción o aumento de velocidad.

$$\frac{14 \text{ hilos}}{1 \text{ pul}} \cdot \frac{1 \text{ pul}}{25.4 \text{ mm}} \cdot \frac{200 \text{ pasos}}{1 \text{ hilo}} = 110.2362 \text{ pasos/mm}$$

Con los cálculos realizados podemos programar el número de pasos que deben dar los motores a pasos para que el taladro consiga colocarse en la posición necesaria para realizar la perforación de los agujeros.

Por lo tanto la precisión de la máquina sería de 0.009 mm lo que nos significaría una respuesta relativamente lenta de la máquina pero se conseguiría una precisión cercana a las máquinas profesionales.

Capítulo 2. Diseño e implementación de la parte eléctrica y electrónica

2.1 Circuito Eléctrico

Es necesario un sistema eléctrico que soporte el funcionamiento del taladro y la mesa, el cual, básicamente consta de la fuente de poder y las debidas conexiones a la línea de alimentación.

2.1.1 Diseño del transformador

El Transformador es una máquina eléctrica estática, la cual nos sirve para convertir voltajes y corrientes en un circuito de corriente alterna. Esta conversión está dada por las características constructivas del transformador. En la figura se muestra el esquema eléctrico básico de un transformador.

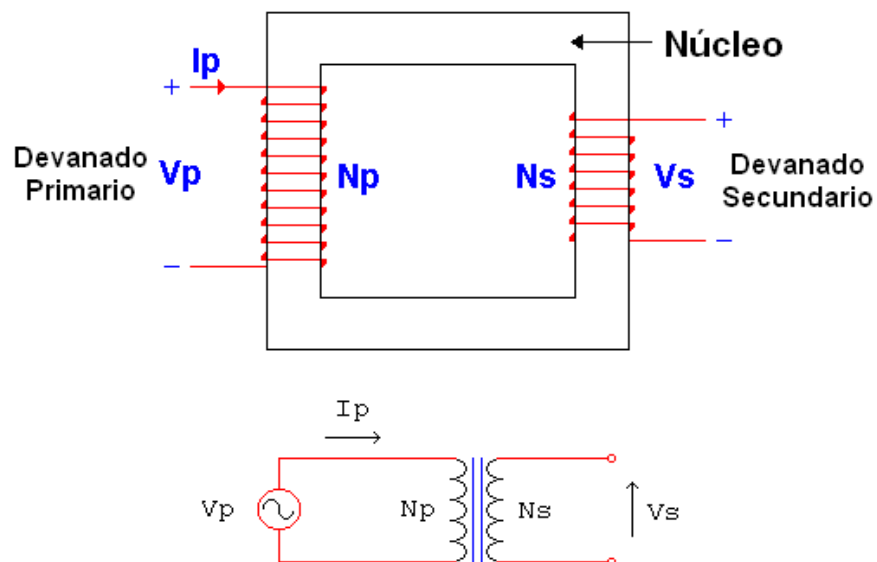


Figura 2.1.1.1 Esquema básico del transformador

En un transformador ideal se establece la siguiente relación de transformación.

$$\frac{V_1}{V_2} = \frac{N_1}{N_2} = \frac{I_2}{I_1} = a$$

Con estas relaciones podemos establecer las siguientes relaciones de transformación las cuales nos pueden simplificar los cálculos y son:

Secundario a Primario

$$Z'2 = a^2 \cdot Z2$$

$$V'2 = a \cdot V2$$

$$I'2 = a \cdot I2$$

Primario a Secundario

$$Z'1 = \frac{Z1}{a^2}$$

$$V'1 = \frac{V1}{a}$$

$$I'1 = \frac{I1}{a}$$

Una vez establecidas las relaciones, ahora se establecerán las ecuaciones para analizar las pérdidas de un transformador no ideal:

$$P_1 = P_2 + P_{p\acute{e}rdidas}$$

Estas pérdidas son de 2 tipos:

Circuito Magnético: conocido también como pérdidas en el hierro.

- Ciclo de Histéresis
- Corrientes Parásitas.
- Flujo de Dispersión.

Circuito Eléctrico: conocido también como pérdidas en el cobre.

- Efecto Joule.

MODELO STEIMETZ

Es un modelo matemático el cual nos sirve para establecer el funcionamiento de un transformador, en este modelo se presentan las perdidas del mismo por medio de resistencias y reactancias inductivas.

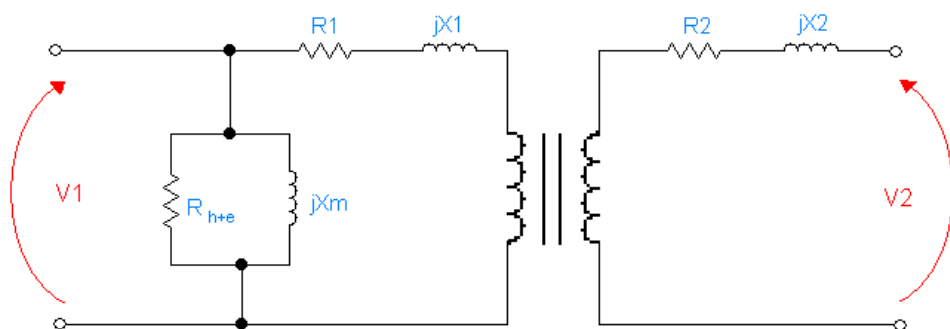


Figura 2.1.1.2 Modelo Steimetz magnético

Para facilitar el cálculo del transformador todas las impedancias se transfieren al primario o al secundario por medio de las relaciones y se puede obtener, las impedancias del secundario al primario obtenemos:

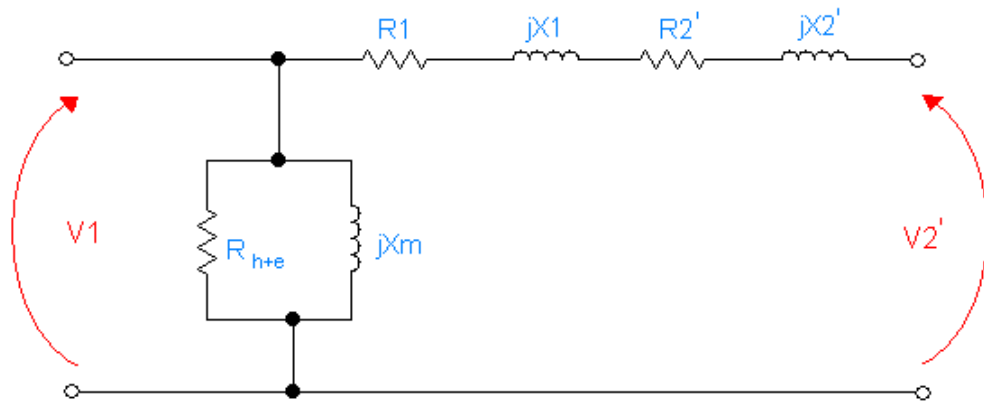


Figura 2.1.1.3 Modelo Steimetz simplificado

El cable para el bobinado

El cable de conexión representa el componente indispensable para el transporte de la energía eléctrica entre los diferentes bloques que integran el sistema. Resulta inevitable que parte de esta energía se pierda en forma de calor, ya que la resistencia eléctrica de un conductor nunca es nula. El material más indicado para la fabricación del cable conductor para bobinado representa un compromiso entre un bajo valor de resistividad y el costo del mismo. El cobre ofrece hoy día la mejor solución. La información contenida en este capítulo está dada para este tipo de material.

Norma AWG

La dependencia entre el diámetro y el área del conductor permite establecer un método de clasificación para los cables. A determinados diámetros se les asigna un número en una escala arbitraria, al que se conoce como el calibre del conductor. Esta escala se la conoce como el AWG (American Wire Gauge, calibre americano para conductores), y es utilizada dentro y fuera de los EEUU. El rango de calibres comienza con el calibre 4/0 (4 ceros), al que corresponde el mayor diámetro. El número de ceros disminuye hasta alcanzar el valor 1/0. A partir de este valor el

calibre del cable está asociado a un valor numérico creciente (2, 4, 6, etc). Es importante recordar que para estos calibres el diámetro del conductor se reduce cuando el valor numérico asignado aumenta. El máximo valor numérico que se utiliza en maquinaria es el 10, ya que la resistencia eléctrica por unidad de longitud resulta excesiva para calibres superiores a este valor. Los calibres 4/0 y 3/0 son raramente usados, pues son difíciles de instalar, tienen un elevado peso por unidad de longitud y un costo muy elevado.

Diámetros relativos

La Figura muestra, en forma comparativa, los diámetros de varios de los calibres AWG. Las características eléctricas y mecánicas de los mismos están resumidas en el anexo. El diámetro en mm especificado para cada calibre corresponde al del conductor sin aislación alguna. Los valores resistivos, ohms por cada 100m, corresponden al valor de ese calibre a una temperatura de 25°C.

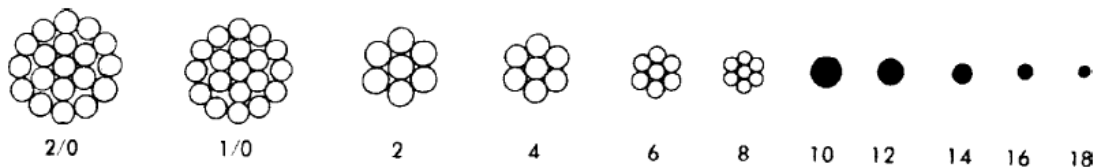


Figura 2.1.1.4 Diámetros de cable AWG

Cobertura aislante

Los materiales aislantes que cubren a los conductores no sólo proveen aislación eléctrica, además proporcionan protección ambiental y resistencia mecánica a la fricción (tirado de cables dentro de un conducto o expansión y contracción con variaciones de temperatura). Al respecto, cuando la protección mecánica debe

mejorarse, se usa una capa adicional, la que se conoce, en inglés, como “jacket”. Nylon suele ser el material más usado para este propósito. En aplicaciones industriales la aislación debe, a veces, evitar que salte un arco a través de ella, o proveer una protección ambiental contra la acción corrosiva de sustancias químicas (gasolina, aceites, ácidos). En sistemas de bajo voltaje en entornos industriales la protección ambiental es la más importante ya que los cables exteriores estarán expuestos a disolventes, aceite, polvo, etc, así como a altas o bajas temperaturas y/o humedad. De ser posible, entierre el cable, a fin de protegerlos de temperaturas extremas. Para estos casos se requiere una aislación especial, la que se reconoce por las designaciones UF (Underground Feeder), alimentador bajo tierra o SEU (Service Entrance Underground), cable de entrada de servicio bajo tierra.

Cálculo del transformador

Para poder diseñar el transformador para nuestro sistema se parte de los siguientes datos:

Voltaje primario	(Vp)	120 V
Voltaje Secundario	(Vs)	3 V
Factor potencia	(Fr)	0.85
Rendimiento	(n)	0.9
Densidad primaria	(dp)	5A/mm ²
Densidad secundaria	(ds)	5A/mm ²
Sección bruta	cm ²	10.89 cm ²

Potencia primaria (Pp)

Esta potencia primaria lo obtenemos a partir de la relación con la sección bruta

$$S = 1.3 \sqrt{P_p}$$

$$P_p = \left(\frac{S}{1.3}\right)^2$$

$$P_p = \left(\frac{10.89}{1.3}\right)^2$$

$$P_p = 70.17 \text{ W}$$

Potencia secundaria (Ps)

Para obtener el valor de la potencia secundaria lo que debemos hacer es multiplicar el valor de la potencia primaria por el rendimiento y esto nos dará como resultado la potencia que saldrá a la carga.

$$P_s = P_p \cdot \eta$$

$$P_s = 70.17 \cdot 0.9$$

$$P_s = 63.15 \text{ W}$$

Corriente de primario (Ip)

Por ley de Ohm sabemos que $P = V I \cos\Phi$, entonces con esta fórmula y los datos como la potencia primaria, el voltaje primario y el $\cos\Phi$ podemos calcular la corriente en el primario.

$$I_P = \frac{P_P}{V_P \cdot \cos \phi}$$

$$I_P = \frac{70.17}{120 \cdot 0.85}$$

$$I_P = 0.6879 \text{ A}$$

Corriente de secundario (Is)

De igual manera que con el primario calculamos en el secundario.

$$I_S = \frac{P_S}{V_S \cdot \cos \phi}$$

$$I_S = \frac{63.15}{3 \cdot 0.85}$$

$$I_S = 24.76 \text{ A}$$

Voltios por espira (Volt/esp)

Ahora calculamos el voltaje que habrá en cada espira, esto se calcula evaluando la siguiente formula $E = 4,44 \cdot f \cdot \phi \cdot N$, y llegamos a la siguiente fórmula.

$$\text{Volt/esp} = \frac{38}{S}$$

$$\text{Volt/esp} = \frac{38}{10.89}$$

$$\text{Volt/esp} = 3.66$$

Número de espiras del devanado primario (N_p)

Ahora calculamos el número de espiras en el primario, sabiendo las espiras por voltio y multiplicando por el voltaje primario.

$$N_p = \text{espira/V} \cdot V_p$$

$$N_p = 3.66 \cdot 120$$

$$N_p = \mathbf{439.2 \text{ espiras}}$$

Número de espiras del devanado secundario (N_s)

Al igual que en el primario, ahora calculamos el número de espiras del secundario.

$$N_s = \text{espira/V} \cdot V_s$$

$$N_s = 3.66 \cdot 3$$

$$N_s = \mathbf{9.99 \text{ espiras}}$$

Sección del conductor para el devanado primario ($S_{\Phi p}$)

Ahora calculamos la sección del conductor, al principio nos impusimos la densidad primaria 5 porque este transformador no va a ser utilizado en grandes lapsos de tiempo.

$$S\phi_p = \frac{I_p}{d_p}$$

$$S\phi_p = \frac{0,6879}{5}$$

$$S\phi_p = 0,13758$$

$$\phi_p = \sqrt{\frac{4 \cdot S\phi_p}{\pi}}$$
$$\phi_p = \sqrt{\frac{4 \cdot 0,13758}{\pi}}$$

$$\phi_p = 0,418mm \qquad \Phi = 0,511mm$$

Por lo tanto podemos observar que necesitamos un conductor con sección de 0.418mm, así que observando la tabla de normalización AWG en el anexo, este valor no es comercial, por lo tanto se debe escoger un cable con una superficie mayor. Así se elije cable calibre 24 AWG

Sección del conductor para el devanado secundario (SΦs)

Ahora calculamos la sección del secundario siguiendo el mismo principio.

$$S\phi_s = \frac{I_s}{d}$$

$$S\phi_s = \frac{24.76}{5}$$

$$S\phi_s = 4.952$$

$$\phi_p = \sqrt{\frac{4 \cdot S\phi_p}{\pi}}$$

$$\phi p = \sqrt{\frac{4 \cdot 4.952}{\pi}}$$

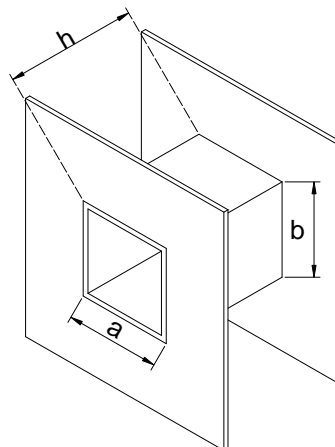


$$\phi p = 2.51mm$$

$$\Phi = 2.59mm$$

Por lo tanto podemos observar que necesitamos un conductor con sección de 2.51mm, así que observando la tabla de normalización AWG en el anexo, este valor no es comercial, por lo tanto se debe escoger un cable con una superficie mayor. Así se elije cable calibre 10 AWG

Cálculo del carrete para los devanados



$$a = 36mm$$

$$b = 36mm$$

$$h_{libre} = 44mm$$

Figura. 2.1.1.5. Medidas libres del carrete

Espiras por capa en el devanado primario (Espiras/cap P)

Es el coeficiente entre la altura del carrete y el diámetro del conductor.

$$espiras / capaP = \frac{h_{libre}}{\Phi p}$$

$$espiras / capaP = \frac{44}{0,511}$$

$$espiras / capaP = 86$$

Capas de conductor en el devanado primario (capasP)

Es el cálculo de cuantas capas se necesitan para contener el número de espiras, y es el cociente entre el número de espiras y el número de espiras por capa.

$$capasP = \frac{N_1}{esp / capaP}$$

$$capasP = \frac{440}{86}$$

$$capasP = 5,11 \text{ Capas} \approx 6 \text{ capas}$$

Espiras por capa en el devanado secundario (Esp/cap S)

Se sigue el mismo procedimiento que en el primario.

$$esp / capaS = \frac{h_{libre}}{\Phi S}$$

$$esp / capaS = \frac{44}{2,59}$$

$$esp / capaS = 17$$

Capas de conductor en el devanado secundario (capasS)

Seguimos el mismo procedimiento que en el primero.

$$capasS = \frac{N_2}{esp / capasS}$$

$$capasS = \frac{10}{17}$$

$$capasS = 1Capas$$

Peso del conductor para el devanado primario (Pes.co.p.)

Primero debemos calcular el perímetro medio que se obtiene multiplicando el diámetro del conductor por la mitad del número de capas totales y a ese resultado se le suma las dimensiones A y B del carrete.

$$LmP = 2a + 2b + 4(5 \cdot \Phi)$$

$$LmP = 2 \cdot (36) + 2 \cdot (36) + 4(5 \cdot 0,511)$$

$$LmP = 154.22mm$$

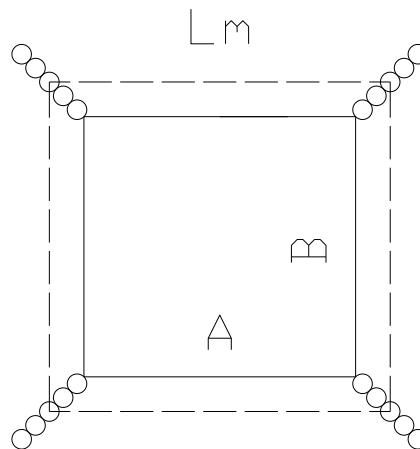


Figura.2.1.1.6. Perímetro medio del conductor primario

Una vez calculado el perímetro medio podemos calcular la longitud total del conductor primario mediante la multiplicación del valor del perímetro medio por el número de espiras totales primarias.

$$L_T P = LmP \cdot espP$$

$$L_T P = 154.22 \cdot 440$$

$$L_T P = 67.85m$$

Ahora ya podemos calcular el peso esto se hace multiplicando la longitud del conductor por una constante del peso del conductor por metro, que se encuentra en tablas.

1.82g/m

$$Peso = L_T P \cdot \frac{gr}{m}$$

$$Peso = 67.85 \cdot 1.82$$

$$Peso = 123.5gr = 0,1235Kg$$

Ahora pasamos a libras

$$Peso = P_{Kgr} \cdot \frac{2,2lib}{1Kgr}$$

$$Peso = 0,1235 \cdot \frac{2,2lib}{1Kgr}$$

$$Peso = 0,2717lbs$$

$$Peso = P_{lbr} \cdot \frac{16onzas}{1lb}$$

$$Peso = 0.2717 \cdot \frac{16onzas}{1lb}$$

$$Peso = 4.34onzas \Rightarrow 5onzas$$

Peso del conductor para el devanado secundario (Pes.co.S.)

Seguimos el mismo procedimiento que en el primario.

$$LmP = 2a + 2b + 4(2 \cdot \Phi)$$

$$LmP = 2 \cdot (36) + 2 \cdot (36) + 4(2 \cdot 2.59)$$

$$LmP = 164.72mm$$

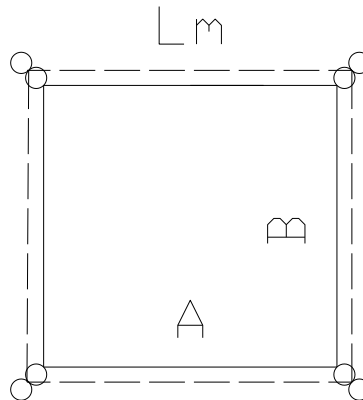


Figura.2.1.1.7. Perímetro medio del conductor secundario.

Con el perímetro medio del secundario y multiplicando por el número de espiras del secundario tenemos ya la longitud del conductor secundario.

$$L_T S = LmP \cdot espS$$

$$L_T S = 164.72 \cdot 10$$

$$L_T S = 1.6472m$$

Así como el conductor primario tiene su constante en g/m el conductor secundario de 16AWG tiene su constante y multiplicada por la longitud se tiene el peso de cable secundario en gramos. De igual manera debemos expresar el valor del peso en libras.

$$Peso = L_T S \cdot \frac{gr}{m}$$

$$Peso = 1.6472 \cdot 46.8$$

$$Peso = 77.088gr = 0,077088Kg$$

Ahora pasamos a libras 46.8kg/m

$$Peso = P_{Kgr} \cdot \frac{2,2lib}{1Kgr}$$

$$Peso = 0,077088 \cdot \frac{2,2lib}{1Kgr}$$

$$Peso = 0,17lbs$$

$$Peso = P_{lbr} \cdot \frac{16onzas}{1lb}$$

$$Peso = 0.17 \cdot \frac{16onzas}{1lb}$$

$$Peso = 2.72onzas \Rightarrow 3onzas$$

CONSTRUCCIÓN DEL TRANSFORMADOR

a) Construcción de los carretes con sus tacos.

Materiales:

- Cartón Piedra (e=3mm).
- Caladora.
- Guillotina.
- Escuadra, Lápiz, Cuchilla.
- Solución.
- Un taco de Madera.
- Taladro, brocas.

Construcción:

1.- Señalamos las dimensiones en el cartón piedra.

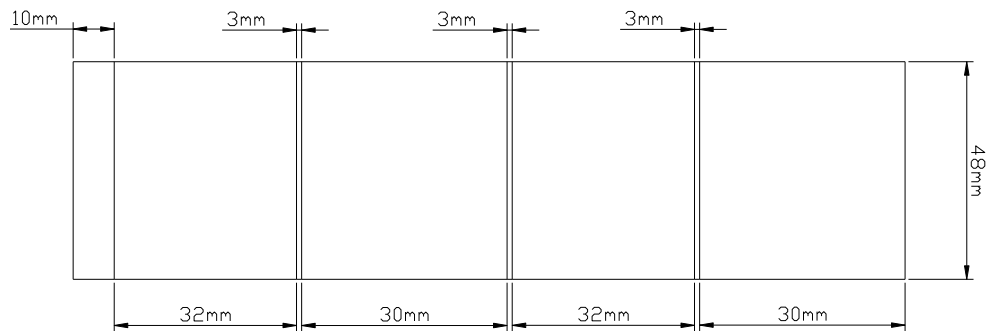


Figura 2.1.1.8. Dimensiones del Cuerpo del Carrete

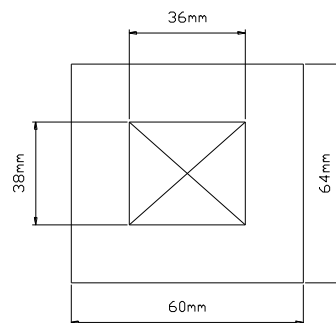


Figura. 2.1.1.9. Dimensiones de la tapa del Carrete

2.- Ya cortado el cuerpo del carrete y las dos tapas se procede armar el carrete colocando y pegando con la solución las tapas al cuerpo con la medida extra del cuerpo del carrete y la cruz de las tapas del carrete tratando que quede lo más uniforme.

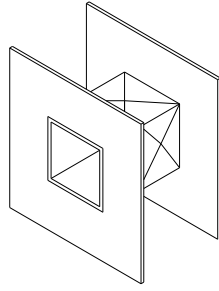


Fig. 2.1.1.10. Estructura del Carrete

4.-Damos las dimensiones al taco de madera según las medidas del cuerpo del carrete y del núcleo, este nos servirá para que al bobinar el carrete no se deforme y nos traiga problemas al colocar en la columna central del núcleo.

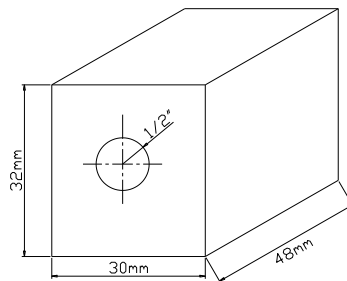


Figura. 2.1.1.11. Dimensiones del taco de Madera

b) Construcción de las bobinas.

Materiales:

- Carrete
- Taco.
- 4 onzas de alambre #26 AWG (bobinado primario).
- 4 onzas de alambre #16 AWG (bobinado secundario).
- Bobinadora.
- Espagueti.
- Cinta masquen (aislante).
- Taladro, brocas.
- Papel minarbón.

Construcción:

1.- Realizamos los orificios en las tapas para el ingreso y salida de cada bobinado; 2 para el bobinado primario y 2 para secundario.

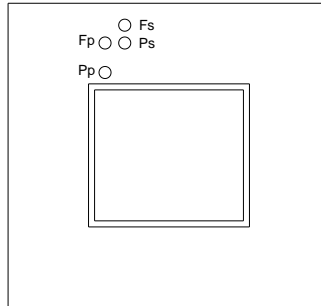


Figura. 2.1.1.12. Salidas del carrete

2.- Colocamos el taco dentro del carrete y este a la bobinadora.

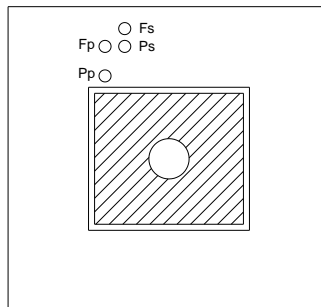


Figura. 2.1.1.13. Carrete con taco de madera

3.- Comenzamos a bobinar con alambre **24 AWG**, colocando el espaguete en el principio del bobinado se introduce por el primer orificio de la tapa y se comienza a dar las 476 espiras y por ultimo sacamos la toma con su respectivo espaguete por el segundo orificio; luego comenzamos con el bobinado secundario con el alambre **10 AWG** colocando el espaguete en el principio del bobinado por el tercer orificio; y se comienza a bobinar sobre el bobinado primario hasta tener 48 espiras, luego se le saca el final del bobinado colocándole espaguete por el cuarto orificio.

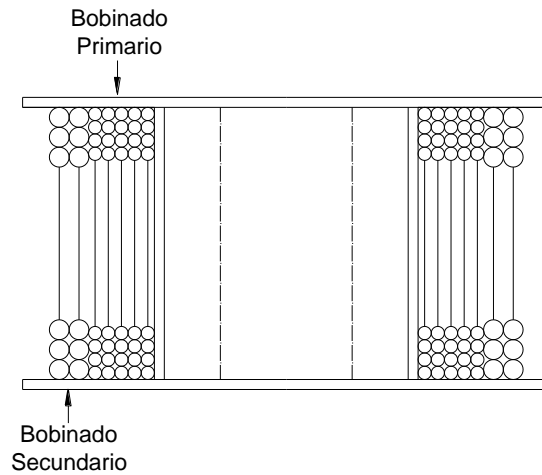


Figura. 2.1.1.14. Capas de cable en el bobinado

4.- Por ultimo se recubre los bobinados con papel minarbón.

c) **Armado total del transformador.**

Materiales:

- Carretes (con las 2 bobinas).
- Núcleo.
- Pernos.
- Bornera.
- Papel Minarbón.

Construcción:

1.- Armamos el núcleo colocando el carrete con los bobinados en la columna central ya que el armado del núcleo es en zig-zag; es decir que se intercambia cada chapa arriba, abajo y así sucesivamente.

2.- Procedemos a colocar los pernos para sujetar y apretar las chapas y así logren compactarse más y no obtener pérdidas.

3.- Procedemos a pelar los terminales de las bobinas y a su vez colocarlos en una bornera.

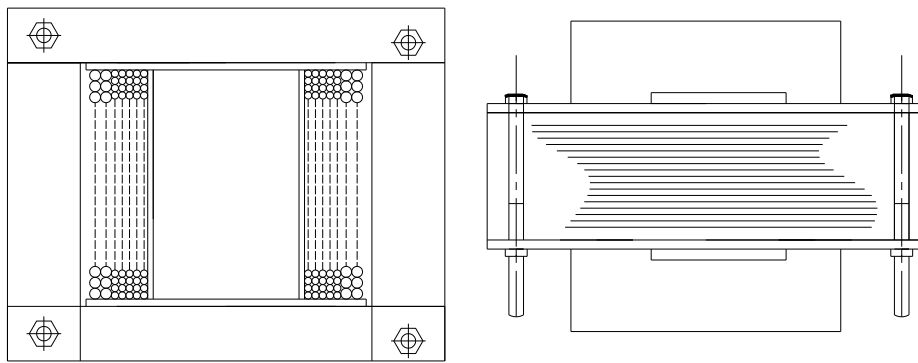


Figura. 2.1.1.15. Transformador armado

2.1.2 Fuente de Poder

Todo dispositivo electrónico necesita poseer una fuente de poder para su funcionamiento, que recibe una tensión senoidal, la cual debe ser convertida en continua. Para realizar esto tenemos muchas opciones en el mercado pero si queremos un fuente que no varié su valor de voltaje a la salida tenemos estas opciones que pueden ser la utilización de diodos tener o integrados reguladores entre otras. Específicamente en este proyecto se utilizará una fuente rectificadora puramente con diodos condensadores ya que es la única manera de rectificar un voltaje tan bajo y una corriente tan elevada, pero presenta el inconveniente de que su funcionamiento está ligado a la corriente que necesita el circuito en un instante determinado. Para estimar el consumo de corriente del circuito se realizarán las siguientes aproximaciones:

Elemento	Corriente
Motor X	6,0 A
Motor Y	6,0 A
Motor Z	6,0 A
Lógica	0,5 A
Potencia	1,5 A
Total	20,0 A

Rectificación monofásica de onda completa con transformador de toma central.

En este tipo de rectificación se utiliza solo dos diodos y se utiliza los pulsos positivos o los negativos el esquema del circuito, y los voltajes de los componentes se presentan a continuación.

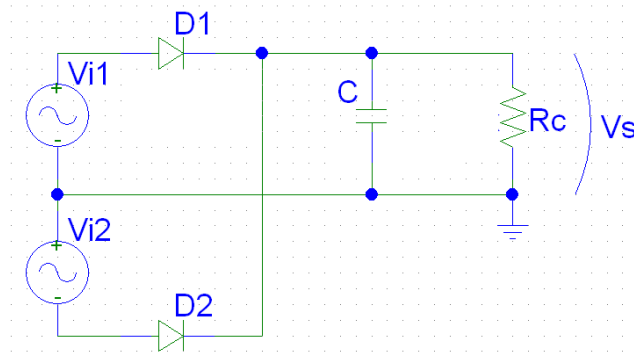


Figura. 2.1.2.1. Esquema de la fuente de poder a diseñar

Las formulas para su cálculo se presentan a continuación:

$$V_p = 4.243 V$$

$$C = \frac{I * t}{\Delta v}$$

Como necesitamos que el rizo sea mínimo pues el voltaje que se va a manejar es relativamente bajo, entonces:

$$C = \frac{24.76 * 8.33E - 3}{0.3}$$

$$C = 687500.026 \mu F$$

$$I_D = I + \frac{1}{3} I$$

$$I_D = 20 + \frac{1}{3} 20$$

$$I_D = 26.66 A$$

$$V_{C_{max}} = V_p - V_D$$

$$V_{C_{max}} = 4.243 - 1.4$$

$$V_{C_{max}} = 2.84 V$$

$$V_{C_{max}} = 2.84 V$$

$$\Delta v = 0.3 V$$

$$V_s = V_{C_{max}} - \Delta v$$

$$V_s = 2.84 - 0.3$$

$$V_s = 2.54 V$$

Para el circuito de control

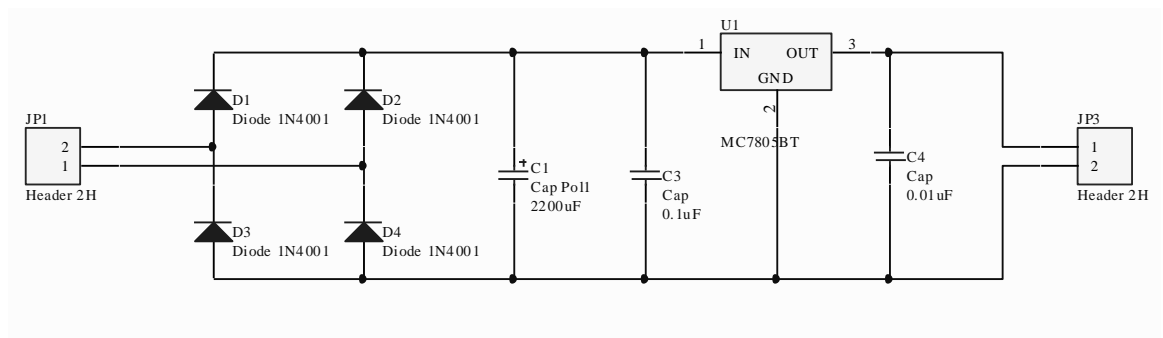


Figura 2.1.1.2 Esquema del Sistema de rectificación

2.2 Circuito electrónico

El objetivo de este capítulo es adentrarnos en el control automático de la máquina a realizarse, para esto el control consta de dos partes, la primera de software, de la cual se encarga el computador, y la segunda de hardware, de la cual se encarga el microcontrolador.

2.2.1 Diseño modular

En todos los equipos electrónicos de consumo masivo se destaca una característica en su diseño, la cual es el diseño modular, es decir, el diseño separado de las partes de control, alimentación y potencia. Esta topología de diseño permite un desarrollo más rápido de aplicaciones, además facilita el mantenimiento del equipo, lo cual desemboca en costos bajos para el usuario final.

En nuestro proyecto se separó el sistema electrónico en cinco circuitos por separado que mediante conexiones de buses de datos funcionan como un todo. Así se separaron en los siguientes

1. Circuito de alimentación de alimentación, el cual consta del transformador y la rectificación. Específicamente son dos circuitos, el primero para la alimentación de los motores (3V, 20A) y otro para el microcontrolador y los sensores (5V, 2A).
2. Tres placas de potencia, cada una consta de una interfaz con el microcontrolador mediante optoacopladores, la lógica necesaria para el control de PWM y la parte de potencia con TIPs 127 para el manejo de los motores.
3. Circuito de control, el cual consta de una sola placa en la cual se encuentra el microcontrolador 18F4550, el conector usb y la circuitería necesaria para la comunicación con la placa de potencia y el computador.

2.2.2 El microcontrolador PIC18f4550

Los microcontroladores son dispositivos electrónicos digitales con la propiedad de ser programables, esto hace que se bastante usado, ya que puede cumplir una gran variedad de procesos que con otros dispositivos nos tomaría mas espacio y tiempo de diseño.

En el mercado existen varias empresas que fabrican este tipo de dispositivos en nuestro prototipo utilizamos uno de MICROCHIP, mas conocido como PIC por sus siglas Peripheral Integrated Controller que significa Controlador de Periféricos Integrados.

Características Generales Microcontrolador PIC18F4550:

- Compatible con USB V2.0
- Velocidad baja (1.5 Mb/s) y alta (12 Mb/s)
- Soporta transferencia por control, interrupción, asíncrona y Bulk
- RAM de doble acceso por USB de 1 Kbyte
- Transceptor USB con regulador de voltaje integrado
- Tiene dos salidas de PWM
- Módulos de captura de 16 bits de resolución
- Puerto SPI
- Puerto I2C
- Comunicación USART asíncrono
- Divisor PLL para frecuencia de clock
- 32K de memoria flash (16384 palabras)

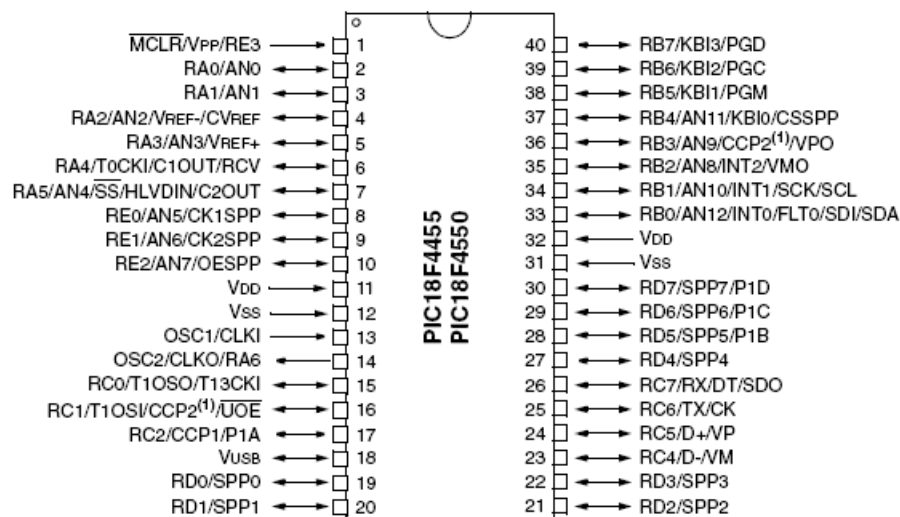


Figura 2.2.2.1 Distribución de pines del PIC16F871[3]

Generalidades de puertos y periféricos:

(PORT A, B, C, D, E): Son los pines externos del uC estos pueden ser programados para actuar como entradas o salidas de datos.

Periféricos Específicos: Son un conjunto de circuitos especializados que realizan funciones especiales (Temporizadores, Conversor Analógico Digital, etc.), son pines específicos los cuales son utilizados por los periféricos especiales y no se pueden utilizar los puertos como simples entrada y salidas.

Pin de Reset: Cuando se da un cero (tierra) en esta pin el uC vuelve a iniciar el programa grabado es decir el contador de programa regresa a la casilla cero.

Oscilador Externo y Ciclo de Instrucción: El uC necesita externamente un circuito de reloj, que genere pulsos digitales de tal manera que se puedan ejecutar as órdenes de la memoria de programa. Existen vanas formas de producir estos pulsos, pero la más usada es la que utiliza un oscilador de cuarzo en conjunto con 2 condensadores, que son conectados a los pines del uC denominados OSC1 y OSC2

Tipo de Oscilador	Frecuencia del Cristal	Capacitor Requerido C1	Capacitor Requerido C2
LP	32kHz	33 pF	33 pF
	200 kHz	15 pF	15pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15pF	15pF
	4MHz	15pF	15pF
HS	4 MHz	15 pF	15pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF

Tabla 2.2.2.2: Cristales y capacitores a utilizar para la oscilacion

2.2.3 Comunicación USB

El Universal Serial Bus (bus universal en serie) o Conductor Universal en Serie (CUS), abreviado comúnmente USB, es un puerto que sirve para conectar periféricos a una computadora. Fue creado en 1996 por siete empresas: IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipmen Corporation y NEC.

El diseño del USB tenía en mente eliminar la necesidad de adquirir tarjetas separadas para poner en los puertos bus ISA o PCI, y mejorar las capacidades plug-and-play permitiendo a esos dispositivos ser conectados o desconectados al sistema sin necesidad de reiniciar.

Características de transmisión

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

Baja velocidad (1.0): Tasa de transferencia de hasta 1,5 Mbps (192 KB/s). Utilizado en su mayor parte por dispositivos de interfaz humana (Human interface device, en inglés) como los teclados, los ratones, hornos microondas y artículos del hogar.

Velocidad completa (1.1): Tasa de transferencia de hasta 12 Mbps (1,5 MB/s), según este estándar pero se dice en fuentes independientes que habría que realizar nuevamente las mediciones. Ésta fue la más rápida antes de la especificación USB 2.0, y muchos dispositivos fabricados en la actualidad trabajan a esta velocidad. Estos dispositivos dividen el ancho de banda de la conexión USB entre ellos, basados en un algoritmo de impedancias LIFO.

Alta velocidad (2.0): Tasa de transferencia de hasta 480 Mbps (60 MB/s) pero por lo general de hasta 125Mbps (16MB/s). Está presente en el 99% de los CPUs actuales.

Super alta velocidad (3.0): Actualmente se encuentra en fase experimental y tiene una tasa de transferencia de hasta 4.8 Gbps (600 MB/s). Esta especificación será diez

veces mas veloz que la anterior 2.0 y será lanzada a mediados de 2009 por Intel, según se estima, o quizá por otra empresa de Hardware, de acuerdo con información recabada de Internet; Aunque actualmente cualquier distribución GNU/Linux es capaz, de soportar el nuevo estándar sin embargo, aun no hay hardware disponible. La velocidad del bus será diez veces más rápida que la del USB 2.0, debido a que han incluido 5 conectores extra, desechando el conector de fibra óptica propuesto inicialmente, y será compatible con los estándares anteriores. Se espera que los productos fabricados con esta tecnología lleguen al consumidor entre 2009 y 2015.

Las señales del USB se transmiten en un cable de par trenzado con impedancia característica de $90 \Omega \pm 15\%$, cuyos hilos se denominan D+ y D-. Estos, colectivamente, utilizan señalización diferencial en full dúplex para combatir los efectos del ruido electromagnético en enlaces largos. D+ y D- suelen operar en conjunto y no son conexiones simples. Los niveles de transmisión de la señal varían de 0 a 0'3 V para bajos (ceros) y de 2'8 a 3'6 V para altos (unos) en las versiones 1.0 y 1.1, y en ± 400 mV en alta velocidad (2.0). En las primeras versiones, los alambres de los cables no están conectados a masa, pero en el modo de alta velocidad se tiene una terminación de 45Ω a tierra o un diferencial de 90Ω para acoplar la impedancia del cable. Este puerto sólo admite la conexión de dispositivos de bajo consumo, es decir, que tengan un consumo máximo de 100 mA por cada puerto; sin embargo, en caso de que estuviese conectado un dispositivo que permite 4 puertos por cada salida USB (extensiones de máximo 4 puertos), entonces la energía del USB se asignará en unidades de 100 mA hasta un máximo de 500 mA por puerto.

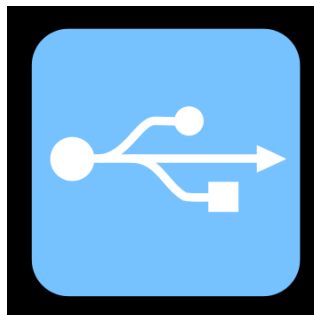


Figura 2.2.2.3.1 Logo USB [4]

El sistema de bus serie universal USB consta de tres componentes:

- Controlador
- Hubs o Concentradores
- Periféricos

Controlador

Reside dentro del PC y es responsable de las comunicaciones entre los periféricos USB y la CPU del PC . Es también responsable de la admisión de los periféricos dentro del bus, tanto si se detecta una conexión como una desconexión. Para cada periférico añadido, el controlador determina su tipo y le asigna una dirección lógica para utilizarla siempre en las comunicaciones con el mismo. Si se producen errores durante la conexión, el controlador lo comunica a la CPU, que, a su vez, lo transmite al usuario. Una vez se ha producido la conexión correctamente, el controlador asigna al periférico los recursos del sistema que éste precise para su funcionamiento.

El controlador también es responsable del control de flujo de datos entre el periférico y la CPU . Concentradores o hubs. Son distribuidores inteligentes de datos y alimentación, y hacen posible la conexión a un único puerto USB de 127 dispositivos. De una forma selectiva reparten datos y alimentación hacia sus puertas descendentes y permiten la comunicación hacia su puerta de retorno o ascendente. Un hub de 4 puertos, por ejemplo, acepta datos del PC para un periférico por su puerta de retorno o ascendente y los distribuye a las 4 puertas descendentes si fuera necesario.

Los concentradores también permiten las comunicaciones desde el periférico hacia el PC, aceptando datos en las 4 puertas descendentes y enviándolos hacia el PC por la puerta de retorno.

Además del controlador, el PC también contiene el concentrador raíz. Este es el primer concentrador de toda la cadena que permite a los datos y a la energía pasar a uno o dos conectores USB del PC, y de allí a los 127 periféricos que, como máximo, puede soportar el sistema. Esto es posible añadiendo concentradores adicionales. Por

ejemplo, si el PC tiene una única puerta USB y a ella le conectamos un hub o concentrador de 4 puertas, el PC se queda sin más puertas disponibles . Sin embargo, el hub de 4 puertas permite realizar 4 conexiones descendentes . Conectando otro hub de 4 puertas a una de las 4 puertas del primero, habremos creado un total de 7 puertas a partir de una puerta del PC . De esta forma, es decir, añadiendo concentradores, el PC puede soportar hasta 127 periféricos USB.

La mayoría de los concentradores se encontrarán incorporados en los periféricos. Por ejemplo, un monitor USB puede contener un concentrador de 7 puertas incluido dentro de su chasis. El monitor utilizará una de ellas para sus datos y control y le quedarán 6 para conectar allí otros periféricos.

Periféricos

USB soporta periféricos de baja y media velocidad. Empleando dos velocidades para la transmisión de datos de 1 . 5 y 12 Mbps se consigue una utilización más eficiente de sus recursos. Los periféricos de baja velocidad tales como teclados, ratones, joysticks, y otros periféricos para juegos, no requieren 12 Mbps . Empleando para ellos 1,5 Mbps, se puede dedicar más recursos del sistema a periféricos tales como monitores, impresoras, módems, scanner, equipos de audio, que precisan de velocidades más altas para transmitir mayor volumen de datos o datos cuya dependencia temporal es más estricta .

Diagrama de capas

En el diagrama de capas de la figura podemos ver cómo fluye la información entre las diferentes capas a nivel real y a nivel lógico.

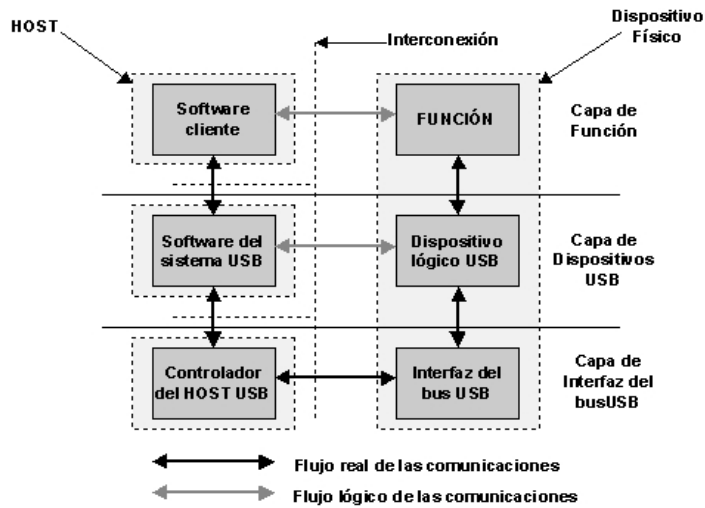


Figura 2.2.2.3.2 Diagrama de capas de una comunicación USB [4]

En dicha figura está materializada la conexión entre el controlador anfitrión o host y un dispositivo o periférico . Este está constituido por hardware al final de un cable USB y realiza alguna función útil para el usuario.

El software cliente se ejecuta en el host y corresponde a un dispositivo USB; se suministra con el sistema operativo o con el dispositivo USB . El software del sistema USB, es el que soporta USB en un determinado sistema operativo y se suministra con el sistema operativo independientemente de los dispositivos USB o del software cliente.

El controlador anfitrión USB está constituido por el hardware y el software que permite a los dispositivos USB ser conectados al anfitrión . La conexión entre un host y un dispositivo requiere la interacción entre las capas . La capa de interfaz de bus USB proporciona la conexión física entre el host y el dispositivo. La capa de dispositivo USB es la que permite que el software del sistema USB realice operaciones genéricas USB con el dispositivo .

La capa de función proporciona capacidades adicionales al host vía una adecuada capa de software cliente. Las capas de función y dispositivos USB tienen cada una de ellas una visión de la comunicación lógica dentro de su nivel, aunque la comunicación entre ellas se hace realmente por la capa de interfaz de bus USB .

Nivel Físico

A nivel físico, USB utiliza un cable de 4 conductores para transmitir una señal diferencial (D+ y D-) y alimentación (VBus = 5V y GND) por medio de conexiones punto a punto. Los dispositivos LS van obligatoriamente equipados con un cable de longitud adecuada (hasta unos 3 m, dependiendo de sus características eléctricas), mientras que los FS pueden ir equipados con un cable o utilizar cables independientes de hasta 5 m (también dependiendo de sus características eléctricas).

La comunicación es bidireccional y semi-dúplex, y utiliza codificación autoreloj NRZI (la línea cambia de nivel si se transmite un 0 y no cambia si transmite un 1) con "bit stuffing" (inserción de un cero tras la transmisión de 6 unos, para asegurar transiciones en la línea y permitir que la PLL del receptor se mantenga sincronizada). Los dispositivos disponen de un transmisor diferencial, receptores diferencial y S/E y resistencias de terminación con los que pueden transmitir y detectar varios estados eléctricos distintos en la línea:

- Transmisión/Recepción diferencial de bits: Estados DIFF0 y DIFF1, denominados también estados J y K.
- SE0 (Single-Ended 0): Ambas señales D+ y D- a 0V. Se utiliza para detectar la conexión/desconexión de dispositivos, para indicar el EOP (fin de paquete) y para generar reset.
- IDLE: reposo o línea en alta impedancia, necesario para permitir transferencias semi-dúplex, detectar la conexión y desconexión de dispositivos y discriminar entre dispositivos FS y LS.
- El SOP (principio de paquete) se indica mediante una transición IDLE a K.
- El EOP (fin de paquete) se indica mediante una secuencia SE0 (2 bits) + J (1 bit) + IDLE.
- Detección de dispositivo y discriminación FS/LS: cuando el transmisor deja la línea en IDLE, si hay un dispositivo conectado su polarización fuerza un estado J (DIFF0 si LS ó DIFF1 si FS), y si no lo hay, la polarización del transmisor fuerza un estado SE0.
- Reset: transmisión de SE0 durante ≥ 10 ms.

Conectores y cables

El USB dispone en su diversidad diferentes tipos de conectores o clavijas dependiendo del uso que le vayamos a dar. Para ello se dispone en el mercado diferentes tipos de conectores USB, de los cuales indicaremos a continuación aquellos de uso común.



Figura2.2.2.4.1 Conector usb Tipo A

Los puertos y conectores de Tipo A son pequeños y rectangulares, y suelen utilizarse para enchufar un dispositivo en un puerto USB de Tipo A en un ordenador o en un hub. En ocasiones se denominan de "flujo descendente", porque los datos salen y se alejan del ordenador o del dispositivo.

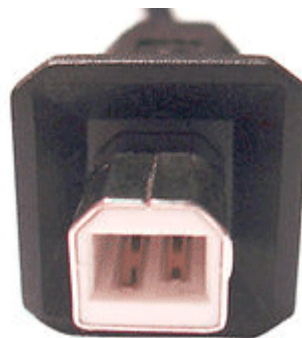


Figura2.2.2.4.2 Conector usb Tipo B

Los puertos y conectores de Tipo B son pequeños y rectangulares, y sirven para conectar un cable USB a un dispositivo USB. En ocasiones se les denomina de "flujo ascendente", porque los datos van del dispositivo al ordenador o a un hub USB.

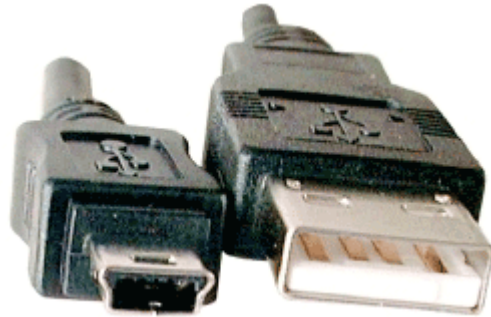


Figura2.2.2.4.3 Comparativa mini usb macho tipo B y usb macho tipo A

Dependiendo del conector, el número de pines difiere entre ellos y también la disposición de las señales, podemos observarlo en la siguiente tabla:

USB TYPE A and B			
 Mating face of USB type A female		 Mating face of USB type B female	
PIN#	SIGNAL	PIN#	SIGNAL
1	+5	3	+Data
2	-Data	4	GND
Mini USB TYPE B			
 Mating face of 5-pin Mini USB Type B female			
PIN#	SIGNAL	PIN#	SIGNAL
1	+5	4	NC
2	-DATA	5	GND
3	+DATA		
		 Mating face of 4-pin Mini USB Type B female	
PIN#	SIGNAL	PIN#	SIGNAL
1	+5	3	+DATA
2	-DATA	4	GND

Figura2.2.2.4.4 Disposición de señales de los distintos conectores usb [4]

Conexión y Desconexión en Caliente

USB permite conectar un nuevo dispositivo sin tener que reiniciar el sistema. Cuando un Hub detecta una nueva conexión se lo comunica al Controlador USB, el sistema interroga al nuevo dispositivo, determina sus propiedades y posibilidades y lo configura. Adicionalmente el sistema carga el manejador adecuado al tipo de dispositivo con lo que el usuario puede empezar a trabajar con el mismo inmediatamente. Algunos sistemas operativos llaman a este proceso "Bus Enumeration".

Protocolo

El protocolo de nivel físico se basa en tokens (testigos). El controlador USB transmite tokens que incluyen la dirección del dispositivo destino, y el dispositivo que detecta su dirección en el token responde y lleva a cabo la transferencia de datos con el controlador. De esta manera, el Controlador USB maneja la parte más compleja del protocolo, generando los tokens de transferencias de datos a 12 Mbps o a 1,5 Mbps, y controlando la conexión lógica entre el sistema y las funciones internas de cada dispositivo. El controlador USB también maneja el consumo en el bus a través de las funciones Suspender/Continuar, por medio de las cuales controla los modos Reposo/Activo de los dispositivos. Esta arquitectura permite el diseño de dispositivos extremadamente simples y de bajo coste USB divide el tiempo en espacios de 1 ms denominados Tramas, durante las cuales se llevan a cabo las comunicaciones a través de Transacciones, las cuales se componen a su vez de Paquetes. Las Transacciones se compone de 3 fases: Token, Dato y Validación (Handshake):

- La fase de Token se compone de un paquete de token enviado por el Controlador USB, y siempre está
- presente en toda transacción. El paquete contiene los campos:

- PID (identifica el tipo de paquete). Todos los PIDs van protegidos por bits redundantes.
- Dirección del elemento destino (7 bits de dispositivo + 4 bits de elemento interno al dispositivo), y \square CRC5.
- La fase de Datos (opcional) se compone de los paquetes de datos que se transfieren entre el Controlador
- USB y el dispositivo. Cada paquete se compone de los campos PID, Datos, y CRC16.
- La fase de Validación (opcional) se usa para indicar el resultado de la Transacción. Se compone sólo de un campo PID.

Adicionalmente, el Controlador USB indica el principio de cada Trama y la transmisión hacia dispositivos LS mediante tokens especiales.

Tipos de transferencia de datos

USB soporta 4 tipos de transferencias de datos:

- **Control**, para configuración y control de dispositivos y para manejo del bus.
- **Isócrono**, para transmisión de información con ancho de banda y latencia garantizados, necesario para aplicaciones como audio, telefonía y vídeo. Permite una comunicación periódica y continua entre el sistema y el dispositivo.
- **Interrupción**, para transferencias de pocos datos, no periódicas, de baja frecuencia pero con unos ciertos límites de latencia.
- **Bulk**, para transferencias de grandes cantidades de datos con dispositivos asíncronos, como impresoras, escáneres, cámaras de fotos (foto fija), etc.

Transferencias de Control

Se desarrollan en 3 Transacciones:

- Transacción de Configuración (Setup), en la que se envía al dispositivo un paquete que especifica la operación a ejecutar. Ocupa 8 bytes.
- Cero o más Transacciones de Datos, en las que se transfieren los paquetes de datos en el sentido indicado por la Transacción de Configuración. La información útil por paquete puede ser de 8, 16, 32 ó 64 bytes para Endpoints FS, y de 8 bytes para Endpoints LS.
- Transacción de Estado, en la que el receptor informa del estado final de la operación. Se procesan por medio de un mecanismo "best effort", según el cual el Controlador USB las va procesando en función del tiempo disponible en cada Trama. Como mínimo se reserva el 10% del tiempo de Trama, y se puede utilizar tiempo adicional siempre que las necesidades de los tráficos isócrono y de interrupción lo permitan.

Transferencias Isócronas

- Sólo son utilizables por dispositivos FS.
- La información útil por paquete puede oscilar entre 1 y 1,023 bytes.
- En cada Trama se transfiere un paquete por cada conexión isócrona establecida.
- El sistema puede asignar como máximo el 90% del tiempo de Trama para transferencias isócronas y de interrupción. Si el sistema ya tiene asignado un tiempo de Trama de forma que no garantiza tiempo suficiente como para manejar una nueva conexión isócrona (transmitir un nuevo paquete por Trama), simplemente no se establece la conexión.
- Los posibles errores no se recuperan (la información que no llega a su tiempo, se descarta).

Transferencias de Interrupción

- Aseguran una transacción (paquete) dentro de un periodo máximo (los dispositivos FS pueden solicitar entre 1 y 255 ms, y los LS entre 10 y 255 ms de periodo máximo de servicio).

- Incorpora detección de errores y retransmisión de datos.
- La información útil por paquete puede oscilar entre 1 y 64 bytes para dispositivos FS y entre 1 y 8 bytes para dispositivos LS.
- El sistema puede asignar como máximo el 90% del tiempo de Trama para transferencias isócronas y de interrupción. Si el sistema no puede garantizar tiempo suficiente como para manejar una nueva conexión de interrupción (transmitir un nuevo paquete dentro del periodo máximo requerido), simplemente no se establece la conexión.

Transferencias Bulk

- Sólo son utilizables por dispositivos FS.
- Se procesan por medio de un mecanismo "good effort", en el que el sistema aprovecha cualquier ancho de banda disponible y en el momento en que esté disponible (en otras palabras, no se garantiza una latencia ni un ancho de banda mínimos). Se puede utilizar el tiempo de Trama reservado y no consumido por transferencias de Control (10%).
- Incorporan mecanismos de control de errores para garantizar la entrega de datos.
- La información útil por paquete puede ser de 8, 16, 32 ó 64 bytes.

Estos 4 tipos de transferencias están disponibles como interfaces software que el sistema pone a disposición de los manejadores de dispositivo, estando los manejadores obligados a comunicarse con los dispositivos única y exclusivamente a través de estos 4 interfaces de programación. Esto viene a significar que un manejador de dispositivo USB jamás accede directamente al hardware del dispositivo, y por otro lado significa que todos los dispositivos USB deben cumplir necesariamente unas especificaciones básicas comunes, ya que deben gestionar adecuadamente los tipos de transferencias que soportan. Adicionalmente, los dispositivos USB se agrupan en Clases, de forma que todos los dispositivos de una misma Clase cumplen además con las especificaciones de dicha Clase, ya que la clase incide directamente en la manera en que el software interactúa con el dispositivo.

Modelo Lógico

Los dispositivos USB pueden tener una o más Configuraciones posibles, que definen distintas formas de funcionamiento. A nivel lógico, una determinada Configuración es un conjunto de Interfaces, donde cada Interfaz especifica qué partes del hardware del dispositivo se comunican con el sistema, donde cada una de estas partes de hardware se denomina Endpoint. En resumen, cada posible Configuración de un dispositivo USB es un conjunto de Interfaces y cada Interfaz es un conjunto de Endpoints. Los Endpoints son unidireccionales, y se direccionan por un número y por el sentido en que transfieren la información (IN (entrada) si transfieren información hacia el sistema, y OUT (salida) si transfieren información hacia el dispositivo).

La comunicación entre una aplicación y los distintos Endpoints de un dispositivo se realiza a través de USB por medio de unos caminos lógicos de transferencias de datos denominados Pipes, de forma que cada Pipe comunica la aplicación con un determinado Endpoint en el dispositivo. Las Pipes pueden ser de tipo Control (también denominadas de Mensaje), que son bidireccionales y con formato especificado por la norma, y de tipo Stream, que son unidireccionales (tipo FIFO) y con formato libre no especificado por la norma. Las Pipes de Control conectan la aplicación con un Endpoint de Control (formado por una pareja de Endpoints uno IN y otro OUT) para realizar transferencias bidireccionales de Control. Las Pipes Stream conectan la aplicación con un Endpoint para realizar transferencias unidireccionales Isócronas, Interrupción y Bulk.

Todos los dispositivos USB deben implementar los dos Endpoints 0 (IN y OUT) para permitir que el sistema pueda establecer la Pipe de Control por Defecto y pueda acceder a información de identificación y requisitos de configuración y pueda configurar el dispositivo. Adicionalmente, USB permite direccionar otros 15 Endpoints IN y 15 Endpoints OUT por dispositivo FS y otras 2 Pipes de Control y/o Interrupción por dispositivo LS. Estos Endpoints adicionales son opcionales y dependientes de los requisitos de implementación del dispositivo.

Clases USB

Una Clase USB es un grupo de dispositivos (o interfaces) con atributos o características similares. Las especificaciones para cada Clase permiten el desarrollo de dispositivos que pueden controlarse por medio de un manejador adaptativo, es decir, que se configura según la Clase reportada por el dispositivo. Dos dispositivos (o interfaces) pertenecen a la misma Clase si por ejemplo utilizan una misma forma de comunicarse con el sistema, o si por ejemplo utilizan el mismo formato de datos.

Las Clases USB se usan principalmente para describir la manera en que los dispositivos (o interfaces) se comunican con el sistema, incluyendo los mecanismos de control y datos, y adicionalmente algunas Clases se usan para identificar en todo o en parte la funcionalidad del dispositivo (o interfaz). En este caso, la Clase se puede utilizar para identificar qué manejador debe controlar dicho dispositivo (o interfaz).

Adicionalmente, los dispositivos de una Clase pueden agruparse en Subclases, lo que facilita aún más el que los manejadores puedan explorar el bus y seleccionar todos aquellos dispositivos que pueda controlar. El uso principal de una Clase USB es la de describir la forma en que un interfaz se comunica con el sistema, tanto a nivel de datos como a nivel de control. También existe un uso secundario, que es el de proporcionar información sobre la funcionalidad que proporciona dicho interfaz. De esta manera, la información de Clase proporcionada por el dispositivo puede utilizarse para que el sistema localice un driver que pueda controlar tanto la conectividad entre el interfaz y el sistema, como la propia funcionalidad del dispositivo.

HID

Un dispositivo de interfaz humana o HID por sus siglas en inglés (Human Interface Device), es un tipo de dispositivo para computadores que interactúa directamente con, y que toman entradas de humanos, y que también pueden entregar una salida a los humanos. El termino "HID" comúnmente se refiere a la especificación USB-HID.

El termino fue acuñado por Mike Van Flandern de Microsoft cuando el le propuso a la comitiva USB, crear un grupo de trabajo para la clase Human Input Device. El nombre del grupo de trabajo fue cambiado a la clase Human Interface Device en sugerencia de Tom Schmidt de DEC, porque la norma propuesta soportaba la comunicación bidireccional.

Las principales motivaciones para HID eran las de permitir inovaciones en los dispositivos de entrada a la computadora y simplificar el proceso de instalar esos dispositivos. Antes de HID, los dispositivos normalmente se ajustaban a muy estrictos protocolos para el ratón, teclados y joysticks. Cualquier innovación en el hardware, requería sobrecargar el uso de datos en un protocolo existente o la creación de un nuevo driver y la evangelización de un nuevo protocolo para los desarrolladores de aplicación. En contraste con todos los dispositivos HID, entregan paquetes autodescriptivos que pueden contener una infinidad variada de tipos de datos y formatos. Un solo driver HID en la computadora parsea los datos y permite una asociación dinámica de datos de entrada y salida (I/O) con la funcionalidad de la aplicación. Este tiene habilitado la innovación rápida y proliferación de nuevos dispositivos de interfaz humana.

Los objetivos principales y subyacentes de la definición de clase HID son:

- Ser lo más compacto posible para ahorrar espacio de datos del dispositivo.
- Permitir la aplicación de software para omitir información desconocida.
- Ser extensible y robusto.
- Que soporte la anidación y las colecciones.
- Ser autodescriptivo para permitir aplicaciones de software genéricas.

El estándar del HID fue desarrollado por una comisión de trabajo con representantes de varias empresas, esta lista de participantes puede ser encontrada en el documento "Device Class Definition for Human Interface Devices (HID)". El concepto de un autodescribible protocolo libre, fue inicialmente concebido por Mike Van Flandern y Manolito Adan, cuando se encontraban trabajando en el proyecto llamado Raptor en Microsoft e independientemente por Steve McGowan mientras trabajaba en el protocolo de un dispositivo para Access Bus en Forte. Después de comparar notas en la conferencia de Consumer Game Developer, Steve y Mike aceptaron colaborar en un nuevo estándar para el emergente Universal Serial Bus ([USB]).

La mayoría de los sistemas operativos reconocen dispositivos estándares USB HID, como lo son los teclados y los ratones, sin necesidad de un driver especial. Cuando se conectan, generalmente se despliega un mensaje en la pantalla diciendo que "un dispositivo HID-compliant ha sido reconocido"

En comparación, este mensaje usualmente no aparece para los dispositivos que son conectados a conectores de 6 pines [PS/2] lo cuales preceden al USB. El conector PS/2 no soporta [plug-and-play], lo que significa que conectar un teclado o ratón PS/2 con el computador encendido no siempre funcionarán. Además, PS/2 no soporta el protocolo HID.

Relación Driver-Dispositivo

USB define una relación entre drivers y dispositivo totalmente diferente a la filosofía tradicional. En vez de permitir que el driver tenga acceso directo al hardware del dispositivo, USB sólo permite al driver comunicarse con el dispositivo a través de las "pipes" establecidas entre el sistema USB y los distintos endpoints del dispositivo. Una vez establecidas las pipes, el Sistema Operativo las pone a disposición del driver en forma de interfaces software. Los tipos de transferencias a través de dichas pipes dependen del tipo de endpoint, y pueden ser de 4 tipos: Bulk, Control, Interrupción e Isócrono.

Por esta razón, las Clases USB se basan en la forma en que el dispositivo o interfaz se comunica con el sistema, y no simplemente en el tipo de servicio proporcionado por el dispositivo. Por ejemplo, en la Clase de Dispositivos de Impresión no interesa cuántos cartuchos de tinta o qué colores soporta la impresora, sino si se envían los datos a través de una pipe tipo Bulk-OUT y si tiene o no una pipe tipo Bulk-IN para reportar información de estado. Asimismo, en la Clase de Dispositivos de Almacenamiento Masivo no interesa si se trata de un disco duro o de un disquete, ni el número de cabezas o cilindros, ni siquiera la capacidad del dispositivo. Lo que interesa es si las lecturas y escrituras se van a realizar a través de pipes tipo Bulk-IN y Bulk-OUT o a través de una pipe de Control, y si se va a utilizar una pipe de

Interrupción para reportar información de estado o si se realiza mediante otros mecanismos.

Las Clases USB también pueden definir el formato de los datos que se transmiten. Por ejemplo, la Clase de Dispositivos de Almacenamiento Masivo define varios métodos opcionales para encapsular (transportar) distintos juegos de comandos estándares, en los paquetes de datos que se transfieren a través de las pipes. Un dispositivo concreto puede soportar uno o varios de dichos métodos de transporte, y uno o varios juegos de comandos estándar (SCSI, UFI, ATA, ATAPI, etc.), de forma que cuando el sistema lee la información proporcionada por el dispositivo, puede buscar y asociar un Driver de Clase compatible con alguno de los métodos de transporte y juegos de comandos.que el dispositivo soporta.

Descriptores

Desde el punto de vista del sistema USB, un dispositivo puede tener varias posibles Configuraciones, en cada una de las cuales el dispositivo puede funcionar de una manera distinta. En cada una de las posibles Configuraciones, el dispositivo queda organizado como un conjunto de Interfaces, donde cada Interfaz especifica qué partes del hardware del dispositivo interactúa con el sistema USB. Cada una de esas partes de hardware se denomina Endpoint. Entonces, de una manera jerárquica, un dispositivo es una colección de posibles Configuraciones, cada Configuración es una colección de Interfaces, y cada Interfaz es una colección de Endpoints. A su vez los Interfaces pueden admitir configuraciones alternativas, con distintas colecciones de Endpoints en cada una de ellas.

Los dispositivos proporcionan toda la información descriptiva al sistema a través de unas estructuras de datos denominados Descriptores. Existen distintos descriptores que proporcionan información a nivel de dispositivo, de configuración, de interfaz y de endpoint. Las especificaciones de Clase USB definen las configuraciones, interfaces (y sus configuraciones alternativas) y endpoints que los dispositivos pertenecientes a dicha Clase o Subclase deben soportar.

Clases, Subclases y Protocolos

Los descriptores de dispositivo y de interfaz contienen una serie de campos que permiten al sistema clasificar a los dispositivos. Estos campos son la Clase, la Subclase y el Protocolo. El Sistema Operativo puede utilizar estos campos para localizar y asociar al dispositivo o interfaz un determinado Driver de Clase, de entre todos los Drivers de esa Clase disponibles en el sistema. También puede seleccionar una determinada configuración del dispositivo, o una determinada configuración alternativa de un interfaz, en función de los protocolos soportados por los distintos Drivers de Clase disponibles en el sistema para esa Clase y Subclase de dispositivo.

Peticiones específicas de Clase USB y peticiones específicas del fabricante

La norma USB denomina “peticiones” (requests) a las distintas funciones que el sistema USB puede solicitar a los dispositivos, lo cual es distinto de los comandos que las aplicaciones pueden enviar, y que dependerán del juego de comandos que se esté utilizando en concreto con cada dispositivo. La norma USB define una serie de peticiones estándar que deben implementar todos los dispositivos, mientras que las especificaciones de Clase USB y los fabricantes de dispositivos pueden definir peticiones adicionales, denominadas respectivamente peticiones específicas de Clase y peticiones específicas del Fabricante.

La forma de enviar al dispositivo una petición USB es siempre a través de una Transferencia de Control dirigida a la pipe de Control por Defecto, en cuya fase de SETUP se indica el tipo de petición (Estándar, de Clase o de Fabricante) y el destinatario de la misma (el dispositivo, un interfaz o un endpoint). Si la petición es Estándar, está definida en la propia norma USB, pero si es de Clase, la Clase a la que pertenece el destinatario de la petición indica en qué Especificación de Clase está definida dicha petición.

Por ejemplo, si el destinatario es el dispositivo, entonces la Clase indicada en el descriptor del dispositivo indica la Especificación de Clase donde está definida la

petición. Si el destinatario es un interfaz o endpoint, entonces la Clase indicada en el descriptor del interfaz indica la Especificación de Clase donde está definida la petición. Si la petición es de Fabricante, entonces es el propio fabricante quien ha definido dicha petición.

2.2.4 Diseño de la comunicación con el computador

La programación del microcontrolador fue hecha en Protón 4.0. En la siguiente figura 3.3 se muestra en diagrama de bloques del sistema a realizar.

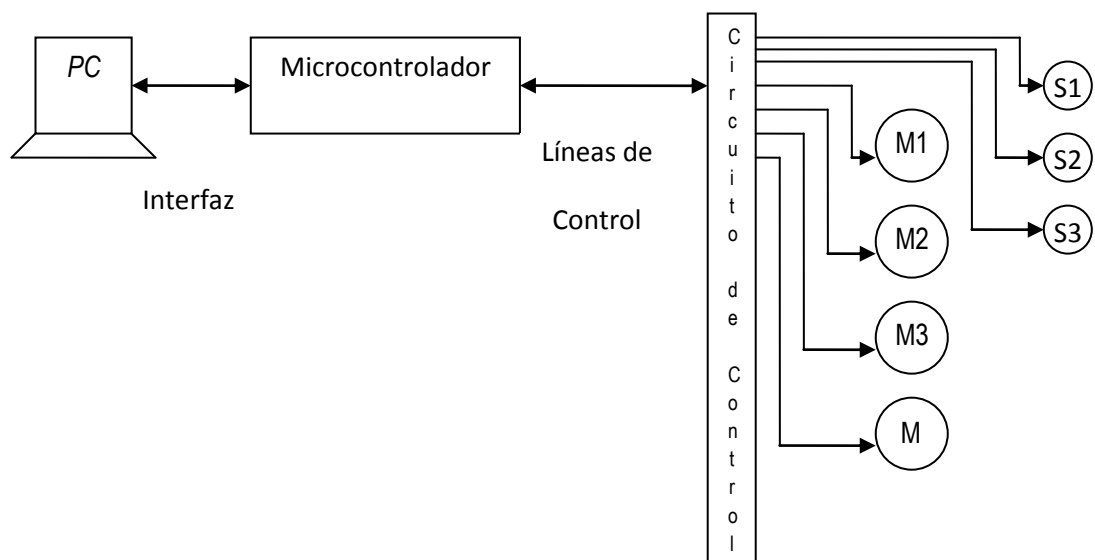


Figura 2.2.4.1 Diagrama de Bloques del Taladro XYZ

Donde:

PC: es el instrumento el cual controla el taladro a partir de un software, en nuestro caso LabView

Interfaz: es por donde se comunica la PC con PIC en esta caso es una comunicación USB.

Microcontrolador PIC: recibe las órdenes de la PC y controla los actuadores.

Líneas de Control: es el camino para transmitir las órdenes del PIC al taladro.

M: son los componentes que nos sirven para realizar la perforación, el taladro y su circuito de activación

Circuito de Control: es el conjunto de elementos que nos sirven para alimentar, y activar los distintos motores con que cuenta la mesa.

M1, M2, M3: motores de pasos que controlan las distintas posiciones del taladro.

S1, S2, S3: sensores que controlan que el taladro este en posición inicial.

Al principio el microcontrolador sondea si el brazo y el taladro están en posición inicial esto lo hace por medio de los fines de carrera, luego de estar en posición inicial el microcontrolador manda un dato de aviso el cual avisa a la PC que ya empiece a enviar las coordenadas para la perforación después de enviado el dato si la PC ya esta lista empieza a enviar los datos los cuales son interpretados por el PIC, luego de enviadas las coordenadas el microcontrolador manda un dato pidiendo que la PC le envíe que tarea realizar, recibido el dato de control se inicia la perforación y cada paso dados por los motores a pasos el microcontrolador pide un nuevo dato de control a la PC, luego de realizado el primer agujero el microcontrolador regresa a pedir las coordenadas y así sucesivamente hasta que ya no existan mas agujeros que realizar, en ese momento la PC envía un dato el cual cancela la perforación.

2.2.4.1 Configuración del HID en Protón

Las antiguas conexiones de puerto serie o paralelo permitían velocidades de transferencia de datos de hasta 115 kbps., mientras que una conexión USB 2.0 alcanza 480 Mbps. Sin contar con la sencillez que supone utilizar un único tipo de conector para todos los periféricos necesarios, la posibilidad de conectar y desconectar sin tener que reiniciar el ordenador a cada cambio de equipo es una ventaja que ha llevado al trabajo móvil hasta su máximo potencial. Disponer de instrumentos de campo o de equipos portátiles que permitan monitorear el clima en

el lugar de los hechos es, gracias al ordenador móvil, un avance que complementado con la practicidad de los puertos USB se ha ganado la preferencia de la mayoría de los usuarios.

El programa Proton Development Suite posee entre sus herramientas más importantes la que conocemos como EasyHID USB Wizard. Esta prestación se encarga de manera automática de generar los programas necesarios para poder utilizar los microcontroladores de la línea PIC 18F2455, 18F2550, 18F4455 y 18F4550, que será el que utilizaremos para nuestro proyecto. El código para el PIC, lo genera tanto para Pic Basic Pro como para Proton, mientras que para el ordenador, se encarga de generar el código en Borland Delphi, Visual C++ y Visual Basic. Tanto desde el Microcode Studio Plus como desde Proton Development Suite, al iniciar la aplicación, nos encontraremos con pantallas a las que les iremos cargando los datos que el sistema nos solicite.

La primera pantalla en aparecer es la de la figura donde se solicita el nombre del producto, el fabricante y el serial del dispositivo. En nuestro caso el producto se llamará CNC-GADA, pero el resto de nombres no se los debe cambiar pues estamos utilizando un nombre no-comercial

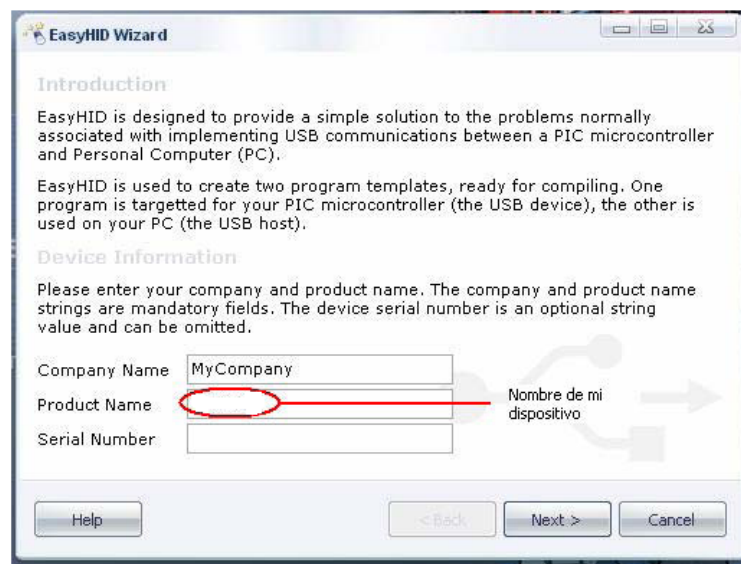


Figura 2.2.4.1.1 Pantalla easy hid, nombre de dispositivo

Al hacer click en siguiente aparecerá en la segunda pantalla (figura) donde se debe especificar el Vendor ID y el Product ID del dispositivo, los cuales son únicos en el mundo, por lo cual no se deben cambiar. Estos números son asignados por el organismo que regula la autenticidad de los productos USB donde se los puede adquirir por U\$S 4mil una membresía anual con números otorgados exclusivamente para desarrollo.

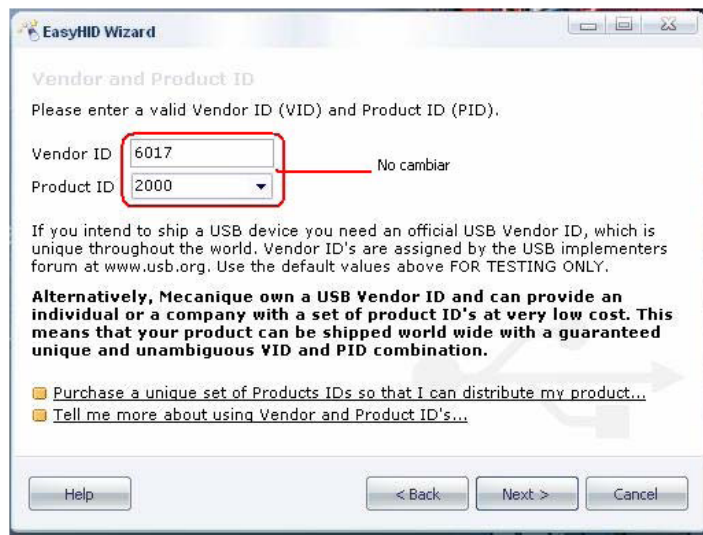


Figura 2.2.4.1.2 Pantalla easy hid, ID de dispositivo

Luego se avanza a la pantalla que permite ajustar los valores de tiempo en que el ordenador controlará la existencia de la conexión USB Polling (figura). Este punto es muy importante porque el ordenador interrogará por el enlace y, si no lo encuentra (Attached), expulsará la conexión y quedará inactivo (Dettached). Esta acción de comprobación puede ser realizada en forma constante mediante una interrupción con un Timer (TMR1) del microcontrolador o a través de un programa organizado y estructurado de manera apropiada dentro del PIC. El tamaño de los Buffers de entrada y salida quedará en 8 bytes.

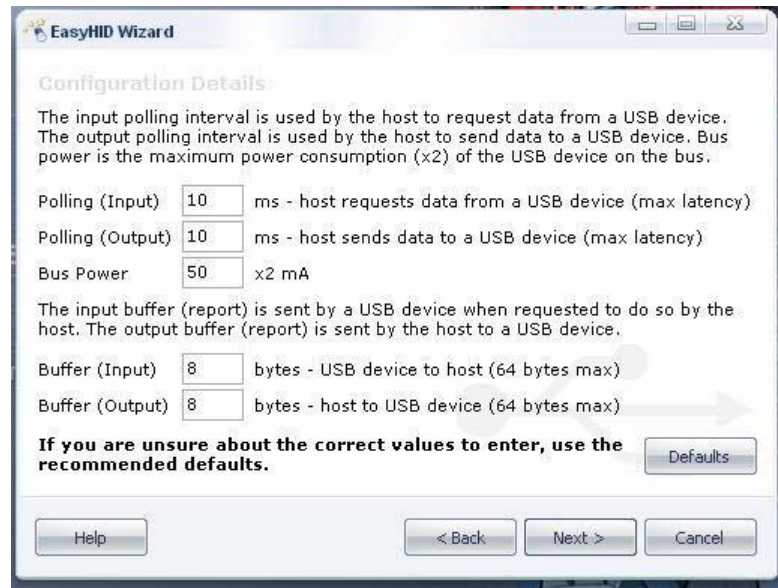


Figura 2.2.4.1.3 Pantalla easy hid, tiempos de interrupción

En la pantalla siguiente se puede seleccionar el lugar donde guardar los archivos que se cargarán en dos carpetas individuales: una para el programa base del microcontrolador y otra para el software de aplicación que se utilizará en el ordenador. Debajo de la selección del PIC, aparece un casillero para habilitar la generación automática de un programa manejado con interrupciones que mantengan “viva” la conexión USB durante todo el tiempo que esté conectado

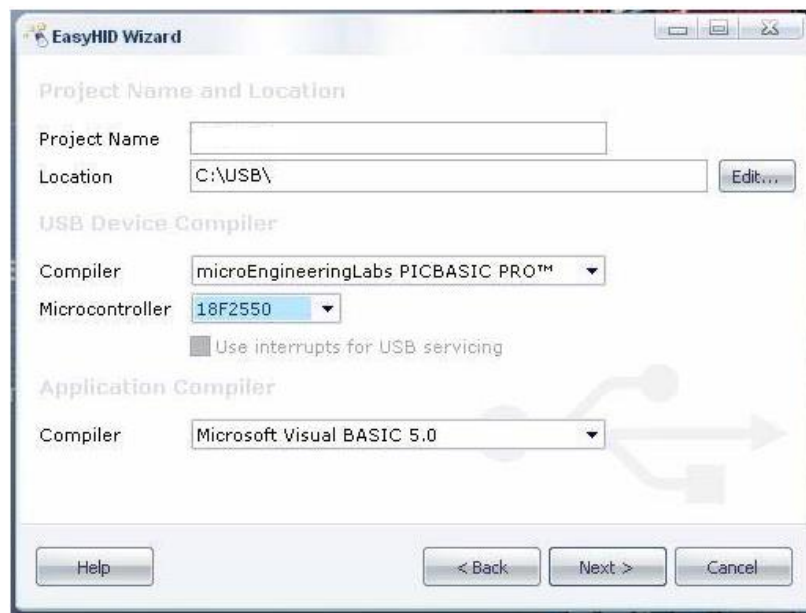


Figura 2.2.4.1.4 Pantalla easy hid, creación de proyecto

Luego aparecerá la pantalla que se observa en la figura, en la cual el Protón genera el código necesario.

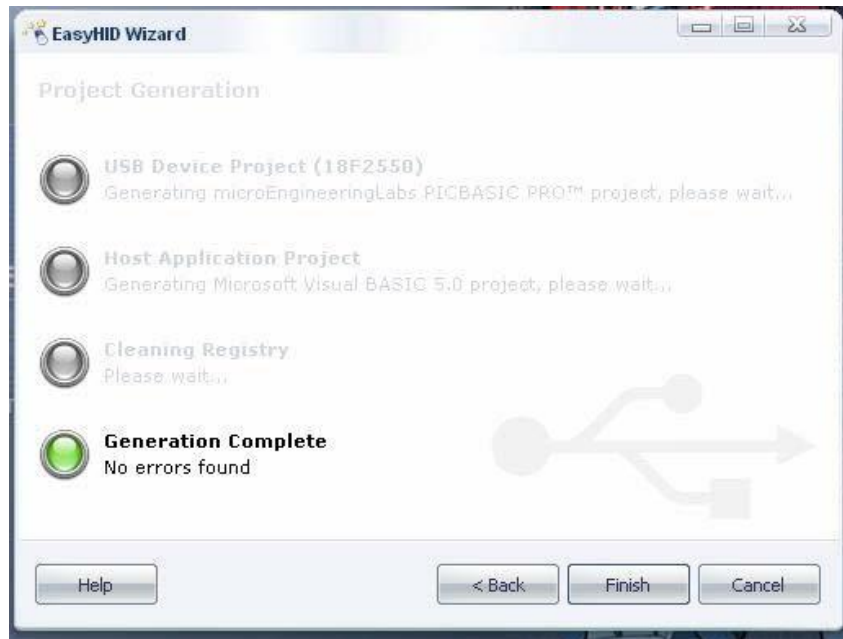


Figura 2.2.4.1.5 Pantalla easy hid, compilación de proyecto

Una vez que se han hecho los pasos anteriores, el código generado por el HID, sin ninguna modificación adicional es el siguiente:

```

DEFINE OSC 48
DEFINE LOADER_USED 1
USBBufferSizeMax con 8 ' maximum buffer size
USBBufferSizeTX con 8 ' input
USBBufferSizeRX con 8 ' output
' the USB buffer...
USBBuffer Var Byte[USBBufferSizeMax]
USBBufferCount Var Byte
' *****
' * main program loop - remember, you must keep the USB *
' * connection alive with a call to USBService every couple *
' * of milliseconds or so... *

```

```

' *****
usbinit ' initialise USB...
ProgramStart:
gosub DoUSBIn
gosub DoUSBOut
goto ProgramStart
' *****

' * receive data from the USB bus *
' *****

DoUSBIn:
USBBufferCount = USBBufferSizeRX ' RX buffer size
USBService ' keep connection alive
USBIn 1, USBBuffer, USBBufferCount, DoUSBIn ' read data, if available
return
' *****

' * wait for USB interface to attach *
' *****

DoUSBOut:
USBBufferCount = USBBufferSizeTX ' TX buffer size
USBService ' keep connection alive
USBOut 1, USBBuffer, USBBufferCount, DoUSBOut ' if bus available, transmit
data
Return

```

El nombre del producto seleccionado en la figura deberá aparecer al conectar el dispositivo avisando que ha encontrado un nuevo hardware. Luego de esto veremos el mensaje que indica que se trata de un dispositivo de interfaz humana USB. Si todo marcha sin inconvenientes, finalizará la instalación indicando que el dispositivo se ha instalado con éxito y que funciona correctamente.



Figura 2.2.4.1.6 reconocimiento de dispositivo

Si todo es así, dentro de Configuración > Panel de Control > Sistema > Hardware > Administrador de Dispositivos, se puede observar el ID del dispositivo

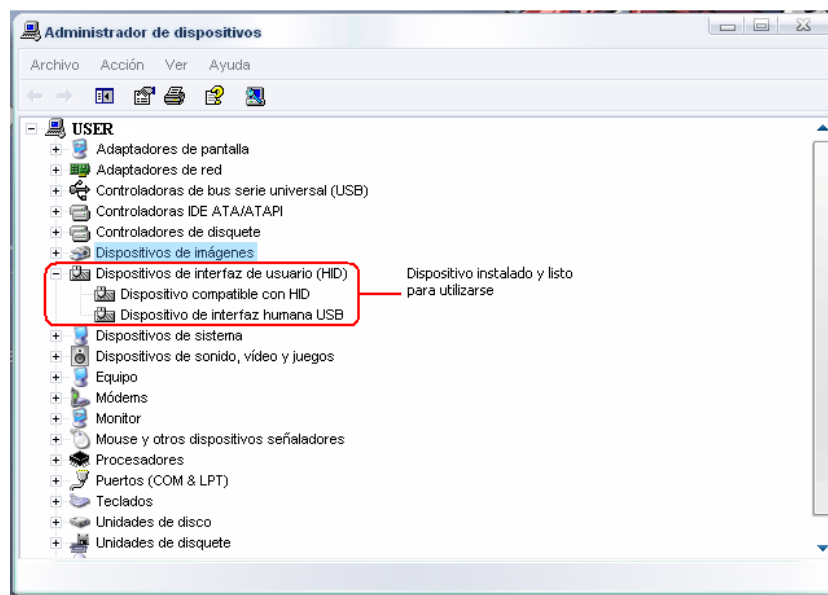


Figura 2.2.4.1.7 reconocimiento de dispositivo, administrador

El programa dentro del PIC, en el código que genera EasyHID, tiene una lógica muy sencilla pero a la vez muy rígida. La rutina se basa en interrogar, en primera instancia, si el dispositivo está conectado al puerto USB. Si no lo está, queda en un bucle infinito del que no sale hasta detectar la conexión o hasta que se reinicia el PIC. Esto sucede en la última parte donde vemos la etiqueta AttachToUSB. Como se puede observar, se repite en un lazo infinito hasta que (UNTIL) el registro interno del PIC detecte la conexión al puerto USB.

Luego se procedió a modificar el código para que cumpla con las especificaciones de nuestro proyecto. Lo que se agregó fue la declaración de los pines asignados para los

fines de carrera y el handshake necesario entre la máquina y LabView. El código completo se lo puede observar en el Anexo

2.2.5 Circuito de control

Partiendo de la hoja de datos del microcontrolador, se realizó el esquema necesario para la conexión usb, el cual se puede observar en la figura. Como se describió anteriormente se trató de llevar el diseño modular, así la placa consta de borneras para la conexión con las otras tarjetas. El circuito consta de las siguientes partes:

1. El microcontrolador PIC18F4550, cristal y masterclear (que será utilizado como pulsante de emergencia)
2. El circuito necesario para la conexión USB: el conector y el condensador de Vusb
3. Tres borneras para el control de los motores
4. Una bornera para la conexión del circuito de control del taladro
5. Una bornera principal de alimentación

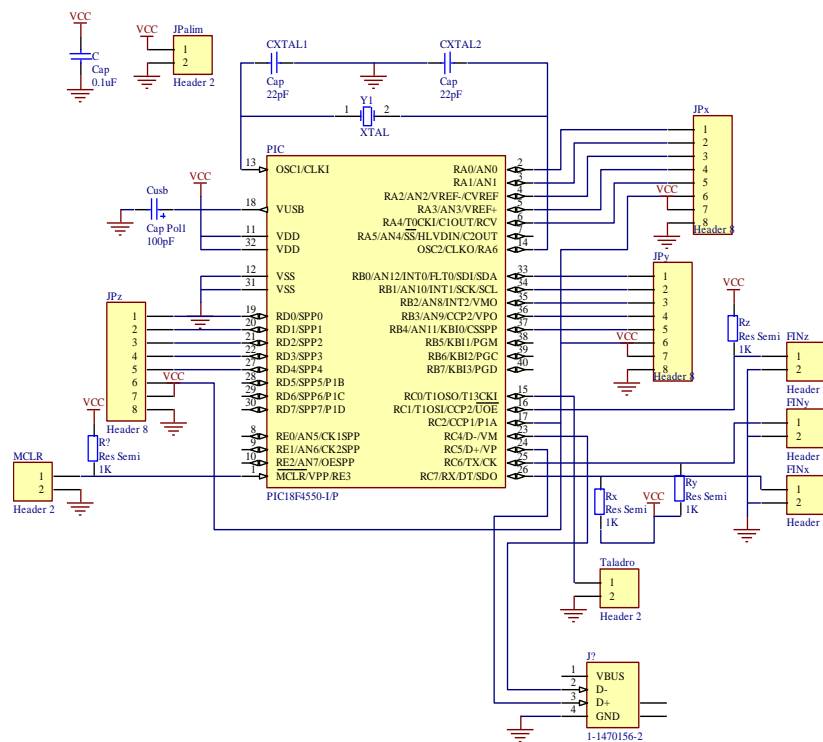


Figura 2.2.5.1 Circuito de control

El diseño de la placa de circuito impreso se realizó en Protel DXP 2004, la cual se puede observar en la figura

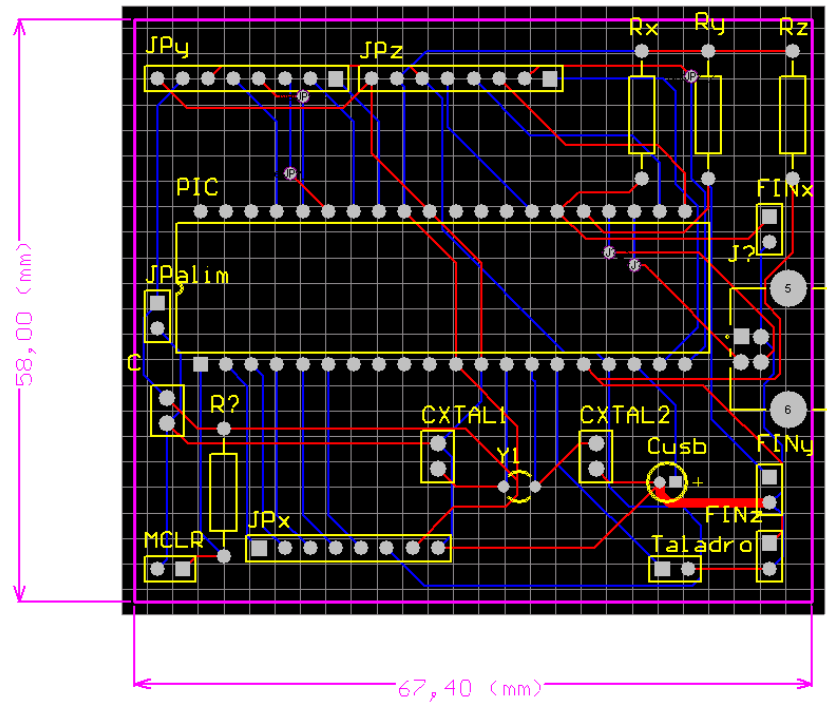


Figura 2.2.5.2 Circuito de control, placa de circuito impreso

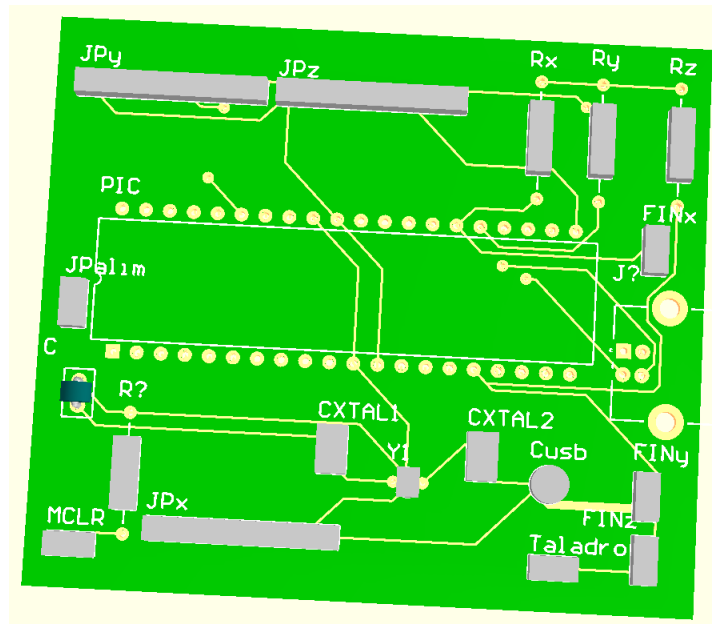


Figura 2.2.5.3 Circuito de control, placa en 3D

2.2.6 Potencia

La etapa de potencia consta de tres etapas principales

1. Lógica y bloqueo de PWM
2. Acoplamiento óptico
3. Potencia y protección

En cuanto a la lógica se utilizaron compuertas tipo AND para comandar el PWM que va a las bobinas de los motores. El uso de PWM reduce la corriente mientras los motores se encuentran “parados” en el proceso de perforado, evitando así que se sobrecalienten.

El acoplamiento óptico se realizó mediante optotransistores 4N35, evitando así ruido causado por switcheo de los transistores y la parte de potencia y protección se diseñó con transistores PNP TIPs 127 para la conmutación y con diodos en antiparalelo, así se evitan ruidos inductivos. El circuito se observa en la figura:

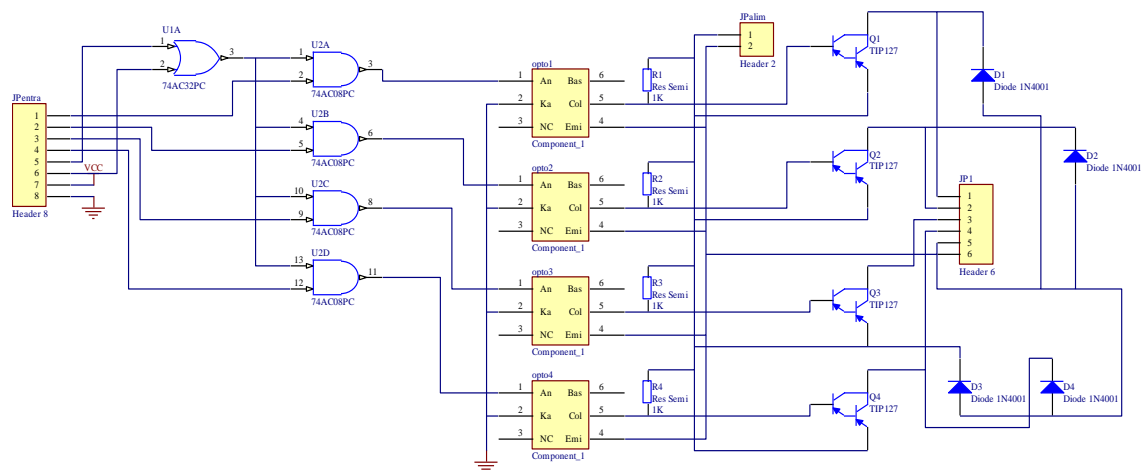


Figura 2.2.6.1 Circuito de Potencia

La placa de circuito impreso se diseñó en Protel DXP 2004. De igual manera que en el circuito de control se la diseñó de manera modular, es decir, una placa para cada

motor, por lo que se colocaron borneras para el control en los extremos de la misma, como se observa en la figura

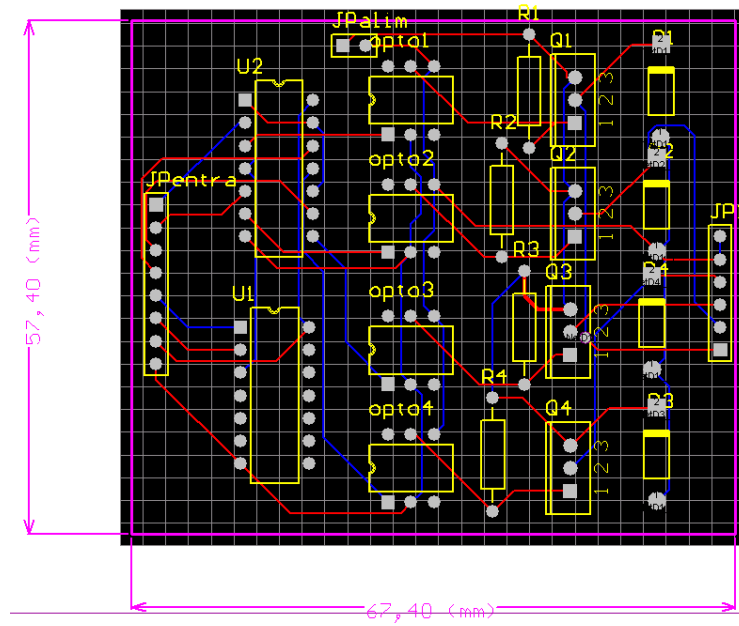


Figura 2.2.6.2 Circuito de Potencia, placa de circuito impreso

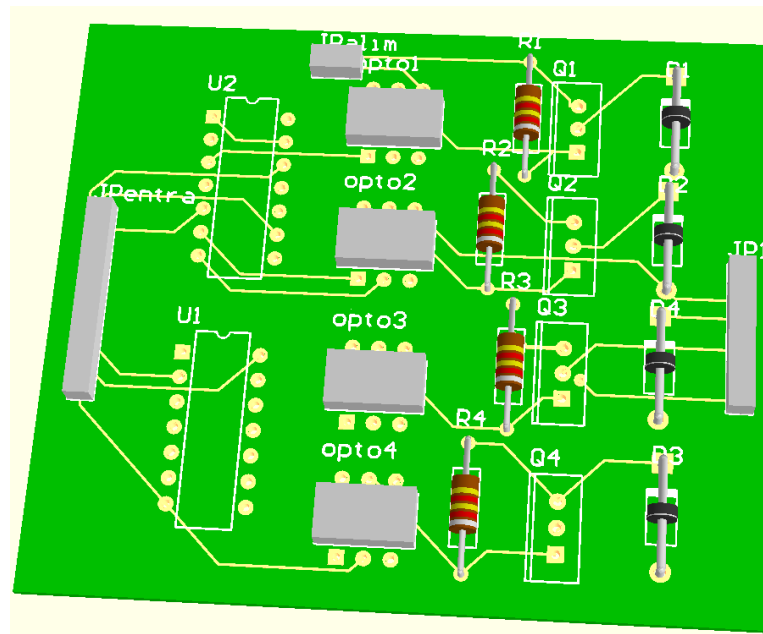


Figura 2.2.6.3 Circuito de Potencia, placa en 3D

Capítulo 3. Programación de la pantalla gráfica de control mediante LabView

3.1 Control numérico por computador

Existe un estándar para este lenguaje, que en realidad apunta a las máquinas industriales. A continuación se tratará de describir algunos de los códigos más utilizados.

3.1.1 Código NC-Drill

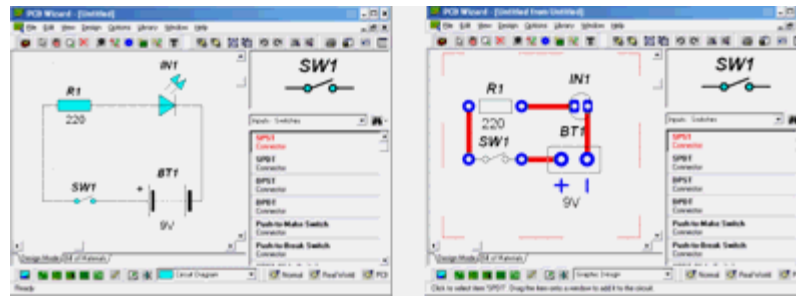
Estos archivos por lo general se los puede obtener de los programas de diseño de PCB, el cual posee información únicamente sobre los agujeros a realizarse. Todos los programas que permiten diseños de PCB, generan el archivo NC-Drill respectivo, pero en nuestro caso se trabajará con Altium Designer Protel y con PcbWizard. En la tabla 3.1.1.1 se muestra el Código de control numérico para routers y taladros, es decir los códigos que se obtienen en un archivo de texto plano.

Tabla 3.1.1.1 Significado de código NC-Drill

Código	Significado
%	reiniciar y detener
X#Y#	Mover y Perforar
T#	Selección de Herramienta
M30	Fin de programa
M00	Fin de programa
M25	Inicio de Modelo
M31	Inicio de Modelo
M01	Fin de Modelo
M02 X#Y#	Repetir Modelo

R#M02X#Y#	Repetición Múltiple del Modelo
M02 X#Y#	
M70	Cambio de Eje
M02 X#Y#	
M80	Reflejar Imagen en el Eje X
M02 X#Y#	
M90	Reflejar Imagen en el Eje Y
M08	Fin de paso y Repetir
N#	Bloquear Secuencia de Números
/	Eliminar Bloque
R#X#Y#	Repetir orificio
G05, G81	Seleccionar modo de Perforación
G04 X#	Variable Local (ignorar)
G90	Modo Absoluto
G91	Modo Incremental
G92 X#Y#	Poner a cero
G93 X#Y#	Poner a cero
M48	Encabezado de Programa antes de "% "
M47	Operador de Mensajes CRT en Pantalla
M71	Modo Métrico
M72	Modo Inglés-Imperial
Snn	Velocidad de husillo (RPM)
Fnn	Mantener velocidad en Eje z (IPM)

A continuación se muestra un circuito básico en PcbWizard, la placa de circuito impreso que se desea y el archivo NC-Drill generado:



M48
 %
 T01
 X+01400Y+01600
 X+01800Y+01600
 X+02200Y+01600
 X+02300Y+01600
 X+01400Y+02000
 X+01800Y+02000
 T02
 X+02100Y+02000
 X+02300Y+02000
 M30

Figura 3.1.1.2 Circuito realizado en PcbWizard y código NC-Drill generado

Al inicio y al final del archivo se observa lo que se conoce como código de funciones de la máquina, los cuales definen lo que debe hacer la máquina, por ejemplo, rociar el enfriador, iniciar el giro del mandril, detenerlo, etc, dependiendo de cada tipo de máquina, pero en general se puede decir que marcan el inicio y el final del programa.

T01 y T02, es el tipo de herramienta que se debe utilizar, para nosotros es el diámetro de la herramienta de perforación. Si se observa la cantidad de coordenadas X,Y podemos notar que nos indica 8 perforaciones y eso es correcto, aunque dos de ellas están separadas, y esto se debe a que se trata de dos perforaciones de diámetro distinto, es decir, son las perforaciones para la bornera de la fuente de alimentación del circuito.

Si se observa ahora las primeras 4 coordenadas, se puede notar que están alineadas en el eje **Y**, lo único que cambia es la ubicación respecto del eje **X**, bien, en nuestro PCB serían las perforaciones para la Resistencia y el LED, y las otras dos que le siguen son para el interruptor. En la tabla se encuentran los tipos de comandos existentes en control numérico:

Tabla 3.1.1.3 Comandos generalidades de código obtenido de programas de CAD

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

Número de Secuencia (N)

Es el número de secuencia que identifica una línea de código de trazado, en el código anterior se definirían de la siguiente manera:

N0000 M48

N0010 %
N0020 T01
N0030 X+01400Y+01600
N0040 X+01800Y+01600
N0050 X+02200Y+01600
N0060 X+02300Y+01600
N0070 X+01400Y+02000
N0080 X+01800Y+02000
N0090 T02
N0100 X+02100Y+02000
N0110 X+02300Y+02000
N0120 M30

Suelen utilizarse como un puntero, es decir, una vez termine el trazado puede volver a repetir una parte de éste, sólo indicándole en que línea de código debe comenzar, y hasta donde debe hacerlo.

Comandos para los Ejes X,Y y Z

Por ejemplo:

X50 Y40

Se indica primero el eje en el cual se desplaza y luego un valor numérico, que puede indicar la cantidad de puntos a desplazarse, en este caso, la orden sería desplazarse 50 puntos hacia el lado positivo de las X, y 40 puntos hacia el lado positivo de las Y.

El comando para cada eje, va separado por un espacio, otros suelen utilizar una coma (,) y en otros directamente se toma como separador la letra que identifica a la instrucción.

En la segunda línea del ejemplo anterior, se indica el comando para el eje X, pero nada para el eje Y, y es por que el eje Y debe permanecer donde está (es decir, no habrá desplazamiento en ese eje), algo muy similar ocurre en la última línea, sólo que en este caso es respecto al eje Y. Igual que antes, algunas máquinas requieren obligatoriamente, que se le indique el par de comandos, y en otras no es necesario.

Tasa de Alimentación feedrate (F)

Algunos comandos, que luego veremos más adelante, lo utilizan sin necesidad de especificarlo, indica que la máquina tiene un registro en su memoria en donde se almacena este valor, y cuando el comando se ejecuta, automáticamente toma este valor de la memoria, y cambia o mantiene la velocidad indicada en ese registro.

Esta velocidad se refiere al desplazamiento, es la velocidad con que la máquina recorrerá la pieza que está grabando, sería más que nada respecto a los desplazamientos en los tres ejes, y no con el giro de la herramienta

Velocidad de Giro (S)

Está referido al Husillo, la velocidad de giro de la fresadora, dependerá por lo general del material de la pieza que se esté trabajando, no es lo mismo calar madera que cobre o acero por ejemplo, la primera puede hacerse más rápido que la segunda y la tercera.

Número de Herramienta (T)

Las máquinas industriales poseen un cabezal que les permite cambiar de herramienta automáticamente, según la instrucción o comando que se ejecute, el valor que

acompañe a T apuntará a una herramienta en especial. Estos valores están dados en Pulgadas o unidades Inglesas.

Funciones Misceláneas o Funciones de la Maquina (M)

Estos código, son utilizados para todo aquello que antes no se había tenido en cuenta, algunos códigos controlan el flujo del programa, otros sin embargo, tienen funciones muy especiales, por ejemplo, el encendido de la máquina, el calibrado cuando ésta se enciende, el sentido de giro del mandril, el inicio o la repetición de un bloque de códigos, el control del rociador para el enfriamiento de la herramienta y la pieza que se está trabajando, etc. Las más comunes se pueden ver en la tabla

Tabla 3.1.1.4 Funciones misceláneas

Comando	Descripción
M03	Inicio de la rotación del mandril en la dirección de las agujas del reloj.
M04	Inicio de la rotación del mandril en la dirección contraria a las agujas del reloj.
M05	Detención de la rotación del mandril.
M07	Conexión del aporte de rocío del enfriador.
M30	Paro general
M99	Retorno desde la subrutina al programa principal

3.1.2 Funciones Preparatorias (G) Código Gerber

Es un formato de impresión utilizado en la producción de circuitos impresos y fue creado por Gerber Systems Corporation. Que en 1970 se convirtió en el mayor

fabricante de impresoras utilizadas en este trabajo. El formato se formalizó en 1980 por la Electronic Industries Association, que de manera similar a RS-232, recibió una descripción técnica de RS-274-D.

El archivo en cuestión está en texto plano con el listado de movimientos de la fuente de luz abierta o cerrada (códigos G) y utilizando una de las formas / aberturas (códigos D) disponible, entre muchos otros comandos. En 1991, el formato se amplificó para incluir los códigos de "apertura" que permite a un expediente con la descripción de Gerber en un circuito completo e independiente, sin necesidad de archivos adicionales con información acerca de la forma y el tamaño de cada apertura, una lista de los agujeros, tolerancias, etc.

Las funciones preparatorias, también conocidas como G-Codes o Códigos G, son las más importantes en la programación CNC, ya que controlan el modo en que la máquina va a realizar un trazado, o el modo en que va a desplazarse sobre la superficie de la pieza que está trabajando.

Los posibles valores que acompañan a este comando, van de 00 a 99, y cada uno tiene una función determinada, en la tabla se muestran los más importantes en la elaboración de circuitos impresos

Tabla 3.1.2.1 Código G resumido

Comando	Descripción
G00	Interpolación Lineal Rápida.
G01	Interpolación lineal a la velocidad programada en el registro F .
G02	Movimiento Circular en el sentido horario Feedrate.
G03	Movimiento Circular en el sentido anti-horario Feedrate.
G04	Es una demora o una pausa con un tiempo específico.
G17	Selección del Plano X-Y
G18	Selección del Plano X-Z

G19	Selección del Plano Y-Z
G40	Compensación anulada, o al centro de la línea de desplazamiento.
G41	Compensación a la Izquierda de la línea de desplazamiento.
G42	Compensación a la Derecha de la línea de desplazamiento.
G70	Unidad de Datos expresados en Pulgadas.
G71	Unidad de Datos expresados en Milímetros.
G90	Desplazamiento en Modo Absoluto.
G91	Desplazamiento en Modo Incremental o Relativo.

Interpolaciones Lineales (G00 y G01)

Interpolación es el proceso mediante el cual, conocidos los valores que toma una función en dos puntos (A,B), se determina con cierto grado de exactitud los valores de un tercer punto (C) comprendido entre A y B, es decir si se conoce la ubicación del extremo (A) de una línea, y la ubicación del otro extremo (B), entonces es posible conocer cada punto (C) por los cuales se debe pasar para llegar desde A hasta B.

G00 es un movimiento rápido, en este caso no existe contacto entre la herramienta y la pieza de trabajo, es decir se desplaza sin realizar corte alguno. G01 es un movimiento lineal pero cortando el material, es decir que se está graficando, para ello utiliza la velocidad programada en el registro F, el cual ya describimos anteriormente.

Interpolaciones Circulares o Movimientos Circulares (G02 y G03)

Lo único que indican estos comando es que el movimiento será circular, G02 en sentido Horario, y G03 en sentido Anti-horario, y que el movimiento debe mantenerse constante a la velocidad programada en el registro F. Hay dos formas de

realizar un Arco o un círculo, una es utilizando el Radio (R) y otra es indicando el centro u origen de la curva por las coordenadas (I,J,K)

Trazado de Arcos Utilizando el Radio (R):

Si el comando es G03 significa que el arco se trazará en sentido anti-horario, y si el centro de la curva está dado por el Radio (R), hay dos posibilidades, -R o +R.

Si el Radio es Negativo, el centro del Radio se encuentra del lado Izquierdo de la línea imaginaria que une los puntos Inicial y Final de la curva, como se ve en la figura 3.1.1.2:

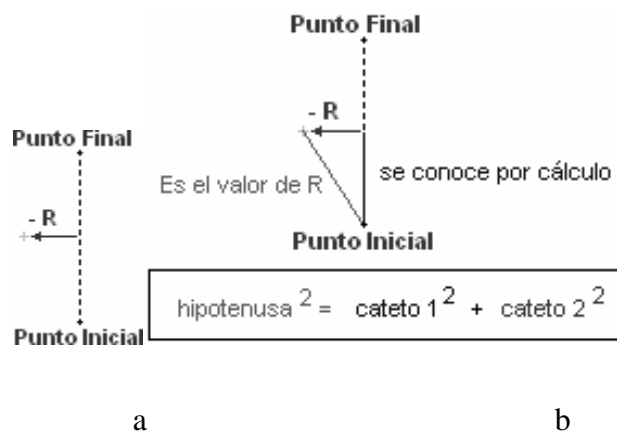


Figura3.1.2.2 (a) El centro del arco está del lado izquierdo (b) La ubicación exacta se obtiene por Pitágoras

Si el Radio es Positivo, el centro del Radio se encuentra del lado Derecho de la línea imaginaria que une los puntos Inicial y Final de la curva.

Trazado de Arcos Utilizando los comandos I, J y K:

El comando K se utiliza cuando la máquina trabaja en 3D, si se conoce la ubicación del punto inicial y el punto final, lo único que se necesita para trazar una curva es la

ubicación del centro del radio. En este caso, la ubicación del radio está dada por las coordenadas I,J. El valor numérico que acompañe a la letra I será la ubicación respecto del eje X, mientras que el valor que acompañe a J será la ubicación respecto del eje Y.

Selección del Plano (G17, G18, G19)

Cuando la máquina trabaja en 3D, se tendrá tres vistas o caras de trabajo, es decir, se está trabajando en los tres ejes, como se observa en la figura

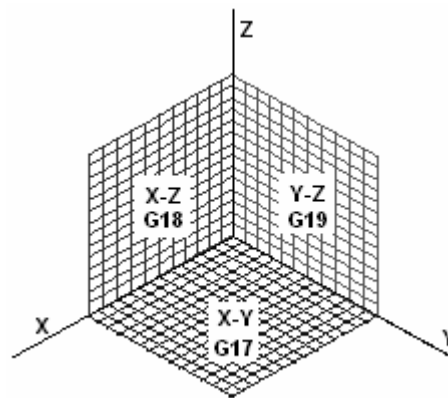


Figura 3.1.2.3 Planos referentes al código G [5]

Compensaciones (G40, G41, G42)

La compensación, no es otra cosa que un pequeño desplazamiento de la herramienta sobre la línea de corte, es decir, se puede cortar justo por el centro de la línea (G40), o bien a un lado (Derecha (G42) o Izquierda (G41)). En la figura se puede observar las tres compensaciones posibles.



Figura 3.1.2.4 Compensaciones [5]

A estos tres comandos, no les acompaña ningún valor que le indique de cuanto será la compensación, sin embargo la máquina debe tener la capacidad suficiente para reconocer el diámetro de la herramienta con la que está trabajando en el momento de aplicarla y así calcular, de cuanto será la compensación.

Unidad de Datos (G70, G71)

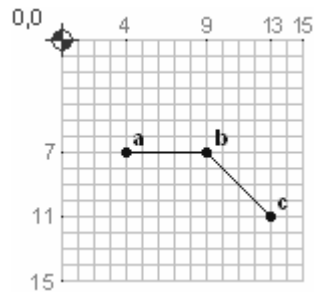
Internamente una máquina trabaja con puntos, y la resolución de la misma estará dada en Puntos por Pulgada (G70), o bien Puntos por Milímetros (G71), de allí surgirá una equivalencia que luego será traducida en las dimensiones de las piezas que se estén trabajando.

Modos de Desplazamiento (G90, G91)

El comando G90 indica Modo Absoluto, es decir que todos los valores de las coordenadas X,Y serán referidos a un punto de origen.

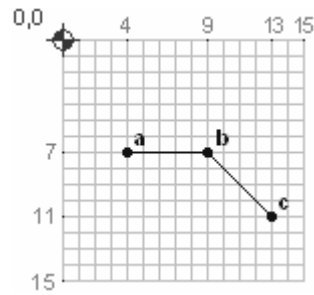
El comando G91 indica Modo Incremental o Relativo, o sea, utilizará el punto cero sólo cuando comience el trazado, de allí en más, el último punto, se convertirá en punto de origen para el próximo desplazamiento.

Se puede observar la diferencia con mayor claridad en la figura. En un caso se utiliza el Modo Absoluto (G90), y en el otro el Modo Incremental o Relativo (G91).



Desplazamiento Absoluto

```
N0000 G90 G00 X4 Y7
N0010 G01 X9
N0020 X13 Y11
N0030 G00 X0 Y0
```



Desplazamiento Incremental

```
N0000 G91 G00 X4 Y7
N0010 G01 X5
N0020 X4 Y4
N0030 G00 X-13 Y-11
```

Figura 3.1.2.5 Tipos de desplazamientos [5]

Desplazamiento en Modo Absoluto:

N0000 G90 (los desplazamientos que siguen serán referidos al punto 0,0); **G00** (el desplazamiento será rápido, sin realizar cortes); se debe dirigir al punto X=4,Y=7 (a).

N0010 G01 (el movimiento es realizando corte); se debe dirigir hasta el punto X=9 (b).

N0020: Continuar cortando hasta el punto X=13,Y=11 (c).

N0030: G00 (Movimiento sin corte y rápido); desplazarse hasta el punto X=0,Y=0 (Regresar al punto de origen 0,0)

Desplazamiento en Modo Incremental o Relativo:

N0000: G91 (los desplazamiento que siguen serán en modo incremental); **G00** (el desplazamiento será rápido, sin realizar cortes); se debe mover 4 puntos en el Eje X, y 7 puntos en el Eje Y, (llega al punto a)

N0010: G01 (el movimiento es realizando corte); debe moverse 5 puntos en el Eje X, (alcanza el punto **b**)

N0020: Desplazarse 4 puntos en el Eje X, y 4 puntos en el Eje Y (llega al punto **c**)

N0030: G00 (el desplazamiento será rápido, sin realizar cortes); Desplazarse 13 puntos hacia atrás en el Eje X, y 11 puntos hacia atrás en el Eje Y (Regresar al punto de origen 0,0)

3.2 Identificación de los archivos generados por software de circuitos impresos

Los archivos Gerber pueden identificarse de diferentes maneras, dependiendo del software que se usa. En programas de diseño de placas de circuito impreso se tienen los siguientes tipos de archivos

- *. GBL - Gerber capa inferior
- *. GTL - Gerber capa superior
- *. APG - Gerber parte inferior de soldadura resistir
- *. Gts - Gerber superior soldadura resistir
- *. GM1 - Gerber mecánica 1
- *. Xln - Excellon taladro archivo
- *. Cmp - Top-capa de cobre (componente de lado)
- *. Sol - Bajo la capa de cobre (lado de soldadura)
- *. STC - Top-capa soldermask (parada máscara)
- *. STS - soldermask capa de abajo (sbop máscara)
- *. DRD - Excellon taladro archivo

Y específicamente para archivos gerber:

- top.gbr - Gerber pistas cara componentes
- bot.gbr - Gerber pistas cara soldaduras
- masktop.gbr - Gerber componentes de la máscara
- maskbot.gbr - Gerber máscara de cara soldaduras

- mechanical.gbr - Gerber de control o el tamaño
- drill.ncd - Perforación ncd

3.3 Comunicación con LabView

LabView es una herramienta gráfica para pruebas, control y diseño mediante programación. El lenguaje que usa se llama lenguaje G o lenguaje Gráfico. Este programa fue creado por National Instruments (1976) para funcionar sobre máquinas MAC, salió al mercado por primera vez en 1986. Ahora está disponible para las plataformas Windows, UNIX, MAC y Linux.

Los programas desarrollados con LabView se llaman Instrumentos Virtuales, o VIs, y su origen provenía del control de instrumentos, aunque hoy en día se ha expandido ampliamente no sólo al control de todo tipo de electrónica (Instrumentación electrónica) sino también a su programación embebida. Un lema tradicional de LabVIEW es: "La potencia está en el Software", que con la aparición de los sistemas multinúcleo se ha hecho aún más potente. Entre sus objetivos están el reducir el tiempo de desarrollo de aplicaciones de todo tipo (no sólo en ámbitos de Pruebas, Control y Diseño) y el permitir la entrada a la informática a profesionales de cualquier otro campo. LabVIEW consigue combinarse con todo tipo de software y hardware, tanto del propio fabricante -tarjetas de adquisición de datos, PAC, Visión, instrumentos y otro Hardware de otros fabricantes.

Principales usos

Es usado principalmente por ingenieros y científicos para tareas como:

- Adquisición de datos y análisis matemático
- Comunicación y control de instrumentos de cualquier fabricante

- Automatización industrial y programación de PACs (Controlador de Automatización Programable)
- Diseño de controladores: simulación, prototipaje rápido, hardware-en-el-ciclo (HIL) y validación
- Diseño embebido de micros y chips
- Control y supervisión de procesos
- Visión artificial y control de movimiento
- Robótica
- Domótica y redes de sensores inalámbricos

En 2008 el programa fue utilizado para controlar el LHC, el acelerador de partículas más grande construido hasta la fecha. Pero también juguetes como el Lego Mindstorms o el WeDo lo utilizan, llevando la programación gráfica a niños de todas las edades.

Principales características

Su principal característica es la facilidad de uso, válido para programadores profesionales como para personas con pocos conocimientos en programación pueden hacer (programas) relativamente complejos, imposibles para ellos de hacer con lenguajes tradicionales. También es muy rápido hacer programas con LabView y cualquier programador, por experimentado que sea, puede beneficiarse de él.

Presenta facilidades para el manejo de:

- Interfaces de comunicaciones:
 - Puerto serie
 - Puerto paralelo
 - GPIB
 - PXI
 - VXI

- TCP/IP, UDP, DataSocket
- Irda
- Bluetooth
- USB
- OPC

Capacidad de interactuar con otros lenguajes y aplicaciones:

- DLL: librerías de funciones
- .NET
- ActiveX
- Multisim
- Matlab/Simulink
- AutoCAD, SolidWorks, etc
- Herramientas gráficas y textuales para el procesado digital de señales.
- Visualización y manejo de gráficas con datos dinámicos.
- Adquisición y tratamiento de imágenes.
- Control de movimiento (combinado incluso con todo lo anterior).
- Tiempo Real estrictamente hablando.
- Programación de FPGAs para control o validación.
- Sincronización entre dispositivos.

3.3.1 Programa en LabView

Como se ha dicho es una herramienta gráfica de programación, esto significa que los programas no se escriben, sino que se dibujan, facilitando su comprensión. Al tener ya pre-diseñados una gran cantidad de bloques, se le facilita al usuario la creación del proyecto, con lo cual en vez de estar una gran cantidad de tiempo en programar un dispositivo/bloque, se le permite invertir mucho menos tiempo y dedicarse un poco más en la interfaz gráfica y la interacción con el usuario final. Cada VI consta de dos partes diferenciadas:

Panel Frontal: El Panel Frontal es la interfaz con el usuario, la utilizamos para interactuar con el usuario cuando el programa se esta ejecutando. Los usuarios podrán observar los datos del programa actualizados en tiempo real (como van fluyendo los datos, un ejemplo sería una calculadora, donde tu le pones las entradas, y te pone el resultado en la salida). En esta interfaz se definen los controles (los usamos como entradas, pueden ser botones, marcadores etc..) e indicadores

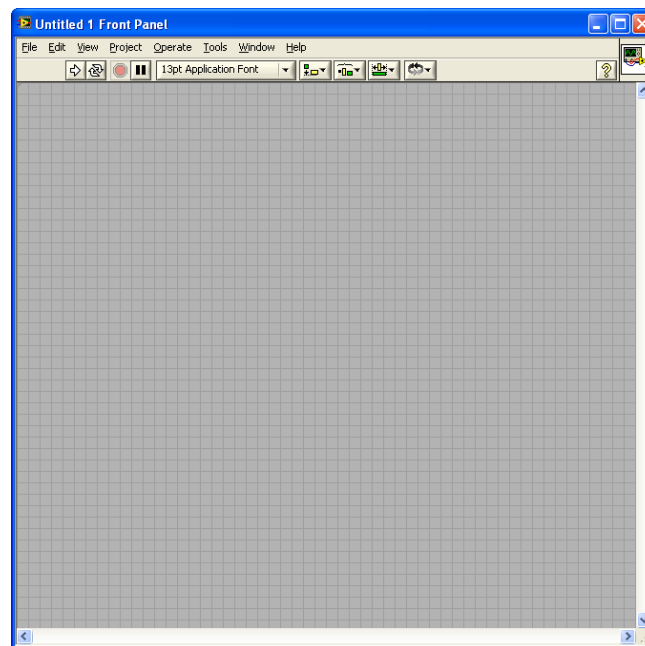


Figura 3.3.1.1. Panel Frontal de LabView

Diagrama de Bloques: es el programa propiamente dicho, donde se define su funcionalidad, aquí se colocan íconos que realizan una determinada función y se interconectan (el código que controla el programa --). Suele haber una tercera parte icono/conector que son los medios utilizados para conectar un VI con otros VIs.

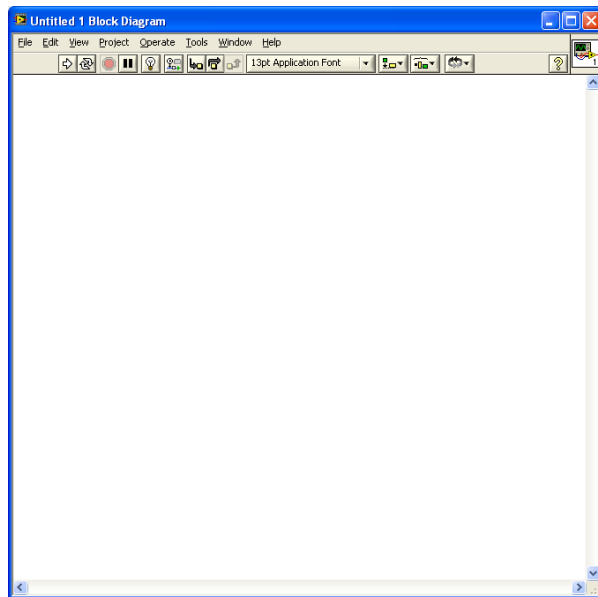


Figura 3.3.1.2. Diagrama de Bloques de LabView

En el panel frontal, encontraremos todo tipos de controles o indicadores, donde cada uno de estos elementos tiene asignado en el diagrama de bloques una terminal, es decir el usuario podrá diseñar un proyecto en el panel frontal con controles y indicadores, donde estos elementos serán las entradas y salidas que interactuaran con la terminal del VI. Podemos observar en el diagrama de bloques, todos los valores de los controles e indicadores, como van fluyendo entre ellos cuando se esta ejecutando un programa VI.

Controles a utilizar

En el programa de control necesario para comandar nuestro proyecto se utilizaron los siguientes controles:

Controles booleanos (botones).- Se utilizarán para comandar el inicio y el paro de la máquina



Figura 3.3.1.3. Controles booleanos

Indicadores booleanos (leds).- los cuales indicarán estados internos del programa y estados de máquina



Figura 3.3.1.4. Indicadores booleanos

Comandos de texto.- en los cuales se ingresarán datos como la dirección física del usb de la máquina,

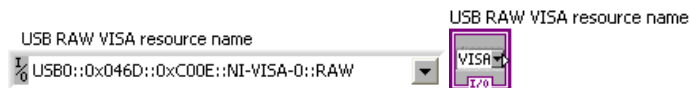


Figura 3.3.1.5. Comandos de texto

Indicadores de texto.- Servirán como información para el usuario sobre el software de diseño de circuito impreso del cual se cargó el archivo NC-Drill, el sistema de medida utilizado, y el agujero que se está perforando.



Figura 3.3.1.6. Indicadores de texto

Matrices de texto.- son conjuntos de indicadores de texto que servirán para visualizar varios datos del mismo tipo, como los diámetros de las fresas y las posiciones de agujeros en los subvi's

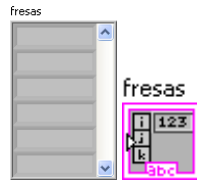


Figura 3.3.1.7. Matrices de texto

Indicadores tipo chart.- donde se visualizarán los agujeros a realizarse y los ya perforados



Figura 3.3.1.8. Indicador gráfico

Operadores lógicos.- se utilizarán para crear condiciones especiales para salir de bucles



Figura 3.3.1.9. Operador lógico and

Comparaciones.- sirven para comparar matrices de textos, específicamente para identificar el código G del NC-drill



Figura 3.3.1.10. Comparador

Bucles While.- tipos de bucle que se repiten hasta que se cumpla una condición, se utilizarán para que el programa corra indefinidamente y para comparar características específicas de la comunicación USB

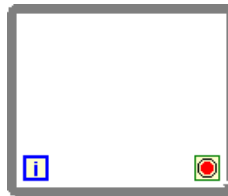


Figura 3.3.1.11. Bucle while

Estructuras secuenciales.- Como la comunicación de LabView es paralela, USB no resultaría compatible pues éste funciona de manera serial, entonces se usan este tipo de estructura para lograr que el programa se ejecute serialmente.

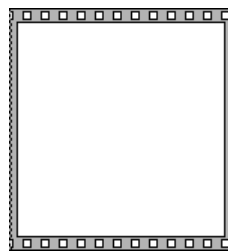


Figura 3.3.1.12. Estructura secuencial

Estructuras Case.- sirven para realizar acciones que necesitan se cumplan ciertas condiciones del programa.

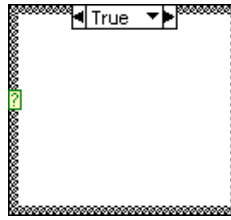


Figura 3.3.1.13. Estructura case

3.3.2 Comunicación USB mediante LabView

LabView consta con un módulo de comunicación denominado VISA, el cual configura el hardware del computador para permitir la comunicación serial o paralela con tarjetas de adquisición de datos o dispositivos especiales.

En este proyecto se utilizará el VISA para comunicarse con un módulo USB. Para esta configuración se debe seguir el siguiente procedimiento:

Configuración de VISA para USB

Se debe tener en cuenta que USB es una comunicación basada en mensajes, es decir que el computador y el dispositivo se comunican mediante envío de comandos y datos sobre el bus, estos pueden ser texto o datos binarios.

VISA se puede comunicar con todas las clases de USB, pero nuestro dispositivo corresponde a una clase denominada USB RAW por lo cual necesita una configuración previa.

Para que nuestro dispositivo funcione es necesario hacer lo siguiente:

1. Crear el archivo INF utilizando el Driver Development Wizard
2. Instalar el archivo INF y el dispositivo USB en el computador
3. Probar la comunicación del dispositivo con LabView

Para crear el archivo INF se debe ingresar en el Driver Development Wizard que viene instalado dentro de NI-VISA, en donde al abrir el acceso directo se observa la siguiente pantalla

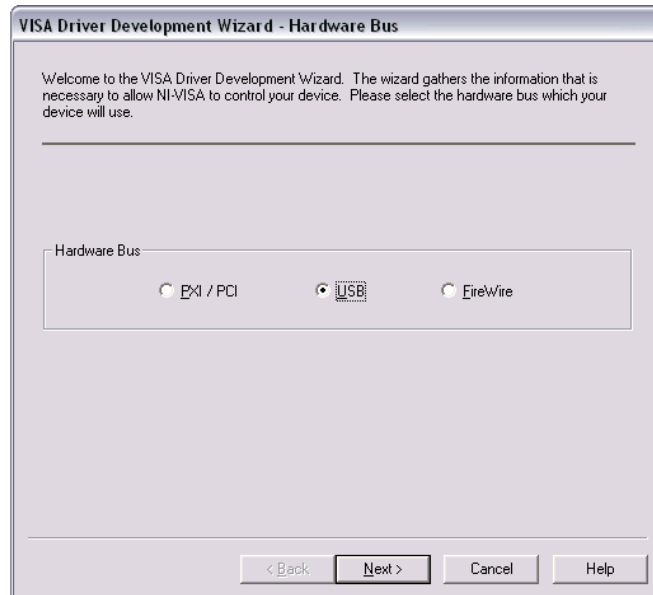


Figura 3.3.2.1. Ventada del Driver Development Wizard

En este Wizard se puede crear drivers para tarjetas PXI/PCI, FireWire y USB, por lo que se debe seleccionar el tipo de driver a crear. Al hacer click en siguiente, se desplegará la pantalla que se puede observar en la figura3.3.2.2. Aquí se debe indicar el Vendor ID y el Product ID del dispositivo.



Figura 3.3.2.2. Ingreso de ID del dispositivo

Cabe recalcar que el ID que se cargó en Protón no es el que se utilizará, sino el que se lo obtiene del computador, para lo cual se debe ingresar en las propiedades de Hardware de Windows y observar el ID del dispositivo. La pantalla se indica en la figura

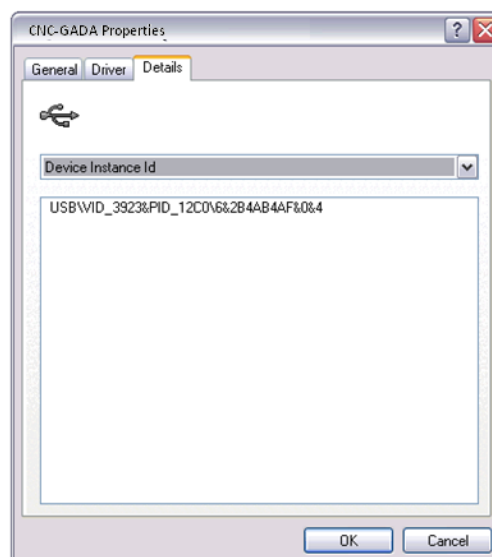


Figura 3.3.2.3. Pantalla donde se visualiza el ID real de nuestro dispositivo

Así se creará el driver del dispositivo para poder ser reconocido por LabView, pero no se instalará en la computadora. Para instalarlo se deben seguir los siguientes pasos:

1. Copiar el archivo INF en la carpeta C:\WINDOWS\INF.
2. Click derecho en el archivo INF y seleccionar “instalar” del menú desplegable
3. Conectar el dispositivo USB, para que el computador lo reconozca
4. Actualizar el driver instalado por defecto, por el creado para LabView

Al conectar el dispositivo al computador, Windows asignará un driver genérico, por lo cual se debe actualizar el driver desde el Administrador de dispositivos. Para esto se debe ingresar en la opción de actualizar el driver, como se indica en la figura y cambiarlo por el creado propiamente para LabView.

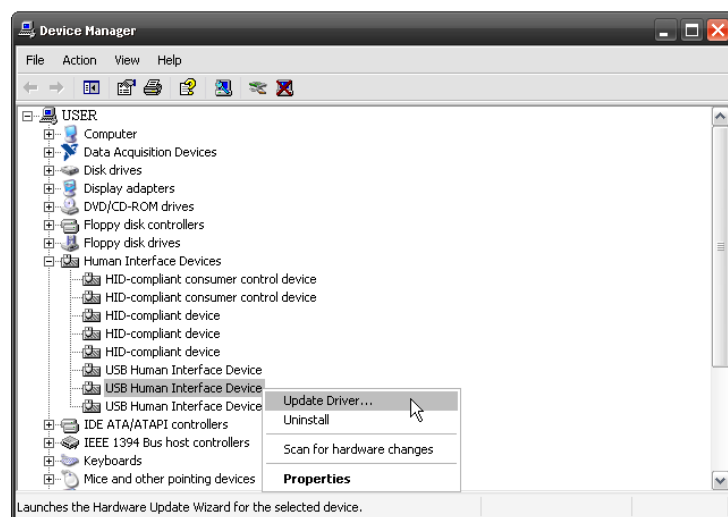


Figura 3.3.2.4. Actualización del driver

Por ultimo se debe comprobar que el NI-VISA reconoció satisfactoriamente a nuestro dispositivo, esto se hace ingresando al Measurement and Automation de National Instruments y comprobando la conexión USB del dispositivo RAW, como se muestra en la figura

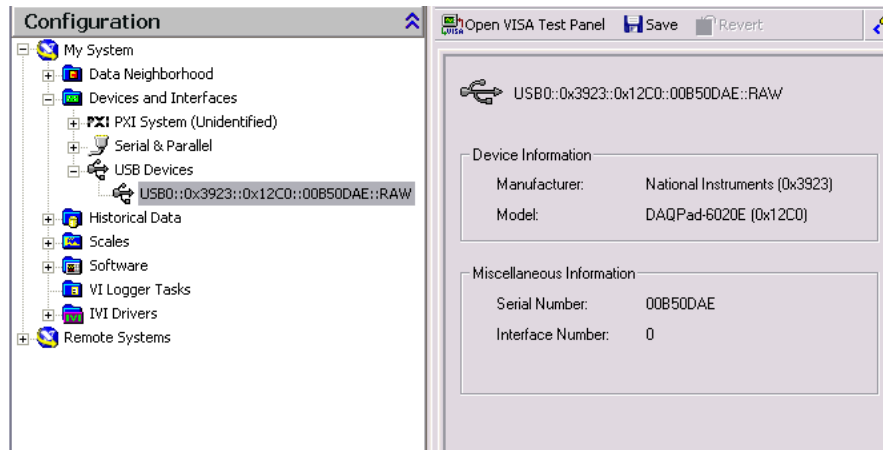


Figura 3.3.2.5. Pantalla donde comprobar la conexión del dispositivo con LabView

VI de conexión

En cuanto a la lectura en el Puerto se utilizó el esquema básico que se puede observar en la figura 3.3.2.6. En general funciona creando una interrupción pues el dispositivo HID transmite datos por interrupción, así se logra recibir los datos, en nuestro caso el confirmaciones de estado de la máquina.

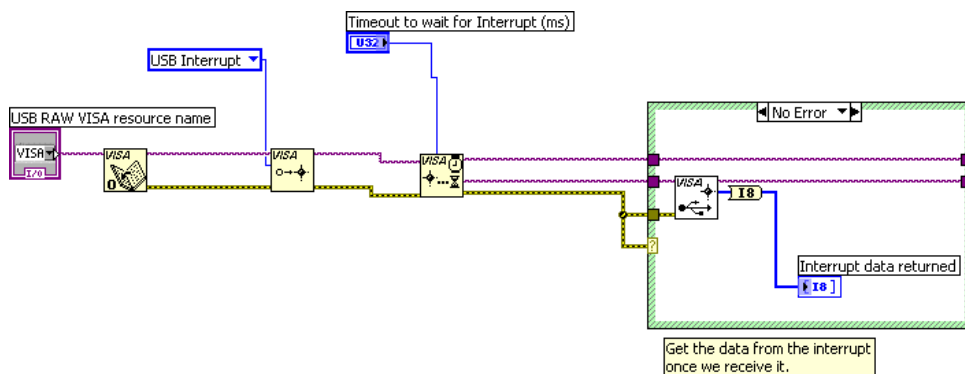


Figura 3.3.2.6. Lectura HID

En lo que se refiere a la escritura del puerto, no es necesario crear una interrupción, sino simplemente cargar el buffer. El esquema básico se puede observar en la figura 3.3.2.7.

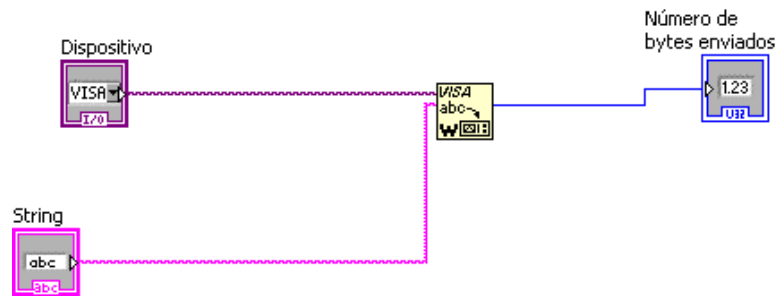


Figura3.3.2.7 Escritura HID

3.4 Decodificador del archivo NC-drill en LabView

La estructura del software de control se basa en cuatro VIs principales

- envioRecepcion
- ncd drill_lee
- pasos string
- pantalla

EnvíoRecepción:

En este VI, se configura la comunicación USB con el microcontrolador, tanto la escritura como la lectura del buffer. El programa consiste en modificar el funcionamiento paralelo de LabView para que sea compatible con el funcionamiento serial del USB. La pantalla del VI se muestra en la figura 3.4.1, y el programa se puede observar en el anexo.

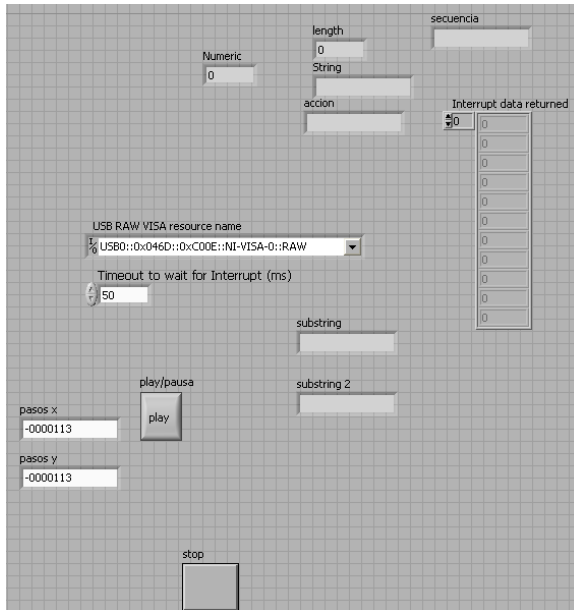


Figura 3.4.1 Pantalla del VI de EnvioRecepcion

NCdrill_lee:

Este VI es el encargado de interpretar el código NCdrill obtenido mediante el VI de envioRecepcion. El algoritmo interpreta el software del cual proviene el archivo, los grosores de broca, y las posiciones de los agujeros. La pantalla del VI se muestra en la figura 3.4.2 y el programa se puede observar en el anexo

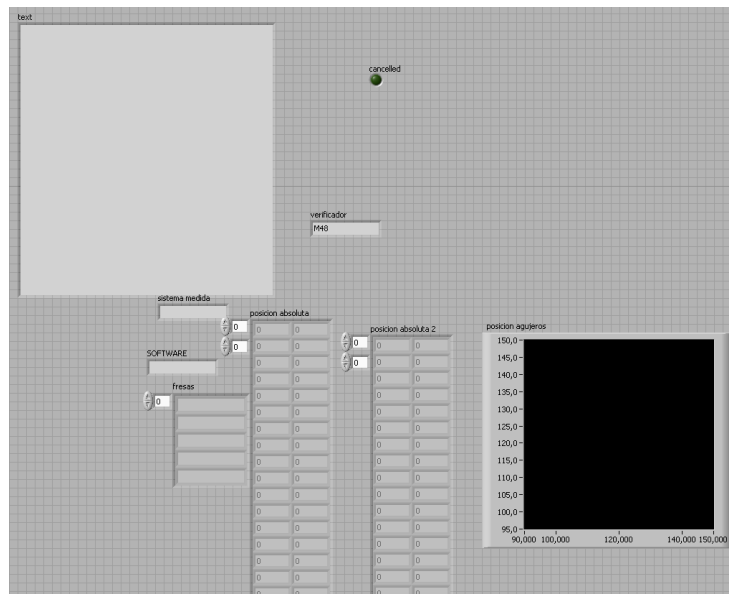


Figura 3.4.2 Pantalla del VI NCdrill_lee

Pasos String:

En este VI, el programa se encarga de convertir las coordenadas decodificadas por el VI NCdrill_lee y los convierte en strings equivalentes al número de pasos que deben realizar los motores. Este String es pasado al VI de envioRecepcion para su posterior envío por USB al microcontrolador. La pantalla del VI se muestra en la figura 3.4.3 y el programa se puede observar en el anexo.

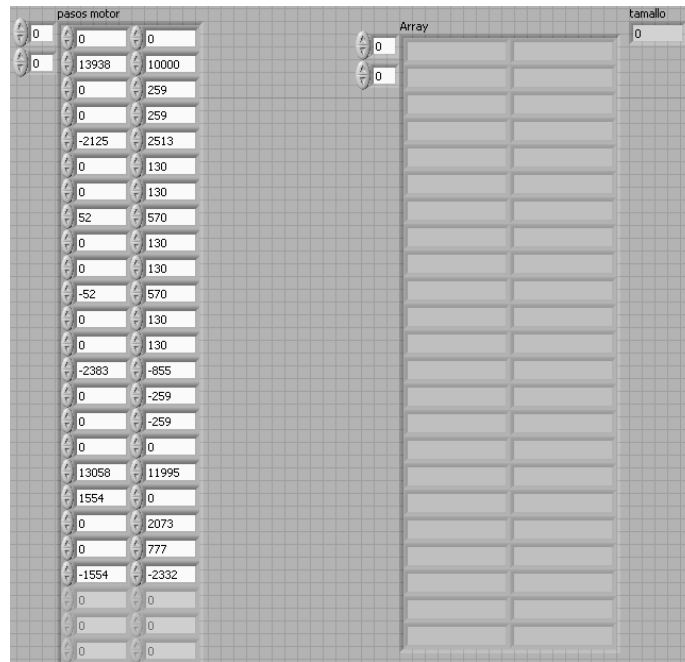


Figura 3.4.3 Pantalla del VI pasos string

Pantalla:

Es la pantalla principal del programa, en la cual se visualiza en los indicadores el programa generador del pcb, el grosor de las brocas a utilizar, la posición de los agujeros tanto de manera numérica como de manera gráfica, y los agujeros realizados y por realizar. Además tiene los controles de inicio y paro.

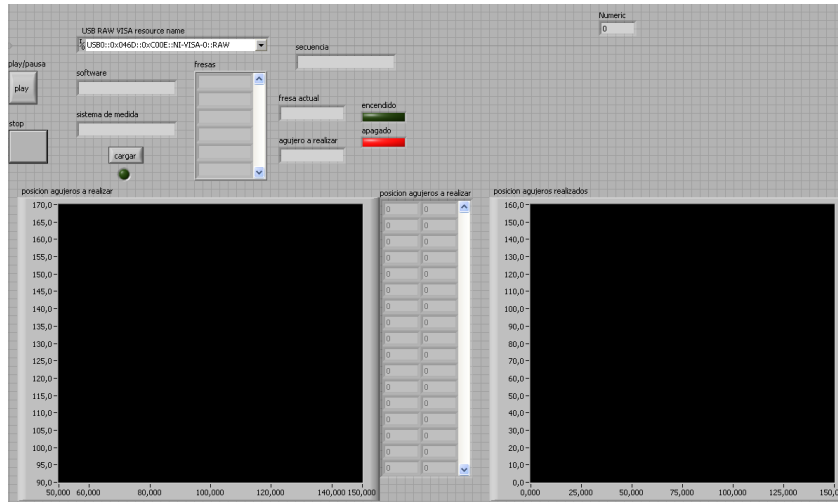


Figura 3.4.4 Pantalla del VI pantalla

3.5 Seguridad industrial

Las características operacionales de robots industriales pueden ser significativamente diferentes de otras máquinas y equipo. Los robots son capaces de movimientos (rápidos o de gran alcance) de gran energía a través de un volumen grande de espacio incluso más allá de la base del robot (véase la figura). El patrón y la iniciación del movimiento del robot es predecible si la pieza de trabajo y el entorno son constantes. Cualquier cambio al objeto que es trabajado (es decir, un cambio de modelo físico) o al entorno, puede afectar los movimientos programados.

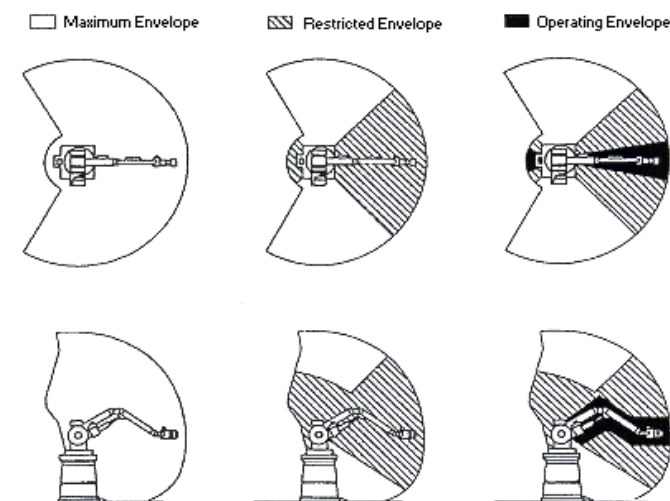


Figura3.3.2.8. Área de trabajo de un robot

Se necesita personal capacitado de mantenimiento y programación en el área restringida de la máquina mientras la misma se encuentra en funcionamiento. El área de peligro de la máquina debe estar marcada para evitar accidentes. El área de riesgo debe cubrir problemas tales como, que el robot posea dos o más programas en su memoria, y erróneamente escoja un programa equivocado en una acción específica, causando alteraciones de velocidad, aceleración y posición de la herramienta.

3.5.1 Tipos de accidentes

Los accidentes en robótica pueden ser agrupados en cuatro categorías: una herramienta del robot causa el accidente, la máquina pone al individuo en una circunstancia de riesgo, una falla mecánica o de la fuente de poder.

1. **Accidentes de impacto o colisión.-** Causados por movimientos impredecibles de la máquina, errores de componentes internos o herramientas
2. **Accidentes de estancamiento o ruptura.-** cuando una extremidad de un trabajador se queda atrapada en una parte de la máquina
3. **Accidentes por fallas mecánicas.-** se conoce como falla mecánica al rompimiento de un elemento, herramienta o falla eléctrica de un robot. Los accidentes pueden darse por desprendimientos de estas partes.
4. **Otros accidentes.-** como rupturas hidráulicas, accidentes por arco electric, polvo, etc

3.5.2 Fuentes de peligros en robots industriales

1. Errores humanos.- Causado por la mala utilización de la maquinaria
2. Errores de control.- Causado por errores de programación, o por falta de control en medidores(presión, temperatura, etc)
3. Acceso sin autorización.- Causado por ingresar al área de peligro del robot cuando éste se encuentra trabajando
4. Fallas mecánicas.- causado por cierta incapacidad del programa de control para sensor fallas mecánicas o desgaste

5. Fallas en el entorno.- como ruido electromagnético, interferencia de radiofrecuencia, etc
6. Sistemas de potencia.- fallas hidráulicas, neumáticas o eléctricas
7. Instalación errónea.- la cual se da cuando el equipo no ha sido instalado según las normas del fabricante

3.5.3 Tipos de control de seguridad

Existen ciertas consideraciones que se deben tomar en cuenta en el diseño de robots y maquinaria industriales, entre las cuales tenemos

1. **Indicadores de riesgo.**- Se debe contar con identificadores alrededor de la máquina, indicando el tipo de riesgo del área.
2. **Dispositivos de seguridad.**- en el diseño se debe contemplar restricciones para el usuario como, limitadores mecánicos, sensores de proximidad, barreras de sectores de trabajo,etc
3. **Dispositivos de advertencia.**- como luces o sirenas que indiquen estados de peligro para el operador
4. **Seguridad de manejo continuo.**- hace referencia a elementos de seguridad para el operador de la máquina
5. **Seguridad de mantenimiento.**- se debe limitar el acceso en el momento de mantenimiento a ciertas partes de la máquina o a controles de activación.

Conclusiones

Electrónicas:

1. Mediante el uso del lenguaje Basic, la programación de dispositivos microcontrolados se simplifica en gran medida, teniendo el programador sólo que preocuparse por el envío correcto de datos y no por la interrupción necesaria para el correcto establecimiento de la comunicación.
2. El uso de transistores resultó menos eficiente que el uso de drivers para los motores de pasos, pero debido a las limitaciones tecnológicas del país se tuvo que optar por los transistores, lo cual alargó el proceso de diseño, tanto del programa en el microcontrolador como el diseño de la fuente de poder.
3. La mejor opción en cuanto al uso eficiente de recursos sería la utilización de motores de corriente continua, lo cual complicaría el sistema de control pero resultaría en una mayor precisión y velocidad de la máquina.
4. Aislar la alimentación del circuito de control de la alimentación del circuito de potencia resultó la mejor opción, pues se evitó el ruido que puedan introducir los motores de pasos al microcontrolador, esto se logró mediante el uso de optoacopladores.
5. El diseño modular aunque resulte costoso, conlleva ventajas en cuanto que para la reparación del mismo se centra en cierta parte del circuito, lo cual facilita el trabajo en diseños complejos.
6. Para el control de la máquina para funciones avanzadas como movimiento circulares, elípticos, o rectas con no ortogonales se requiere de un sistema que trabaje en tiempo real, debido a que la velocidad de la herramienta debe permanecer constante durante el recorrido, lo cual no se puede lograr con microcontroladores de gama media como el utilizado, por lo tanto se debería programar un sistema operativo en tiempo real (RTOS) en un microcontrolador más rápido, o diseñar la lógica pertinente en un FPGA

Eléctricas:

1. Si se requiere fuentes de gran capacidad de corriente, y que soporten picos de voltaje elevados, es preferible utilizar una fuente rectificadora más que una fuente switching para evitar inconvenientes con el circuito.

2. Antes de construir un transformador de corriente elevada se debe modelar para establecer las pérdidas y disminuirlas al máximo, ya que en circuitos de bajo voltaje como es nuestro caso, las pérdidas resultaron mayores a las esperadas, lo que causó que la eficiencia del transformador sea relativamente baja
3. Si se utilizan drivers para los motores, se puede diseñar una fuente de voltaje mayor a la utilizada en este proyecto, lo que disminuiría la capacitancia del condensador para la rectificación, pero aumentaría el tamaño del transformador para poder mantener la misma corriente

Mecánicas:

1. Se debieron considerar varios factores para la selección correcta de los materiales, como son la densidad, la resistencia a la flexión, la disponibilidad en el mercado y el proceso de mecanización que se debe seguir, lo cual permitió un desempeño aceptable de la maquina.
2. El diseño se simplificaría en gran manera si se utilizaran rodamientos lineales para el sistema de traslación de la herramienta, tal y como se diseñó al inicio de esta tesis, lo cual evitaría el uso de sistemas más complejos como el utilizado. Pero estos rodamientos no se consiguen en el mercado nacional, lo que complicaría el mantenimiento de la mesa en caso de que alguno de estos rodamientos se averiara.
3. Se deben utilizar perfiles que permitan la inserción de mecanismos de sujeción más seguros y resistentes, lo cual tampoco se pudo encontrar en el mercado local, por lo que se optó por perfiles menos resistentes.
4. En el diseño se debe evitar agujeros roscados de diámetro muy pequeños, pues la dificultad de roscar puede llevar a la obstrucción del agujero y ruptura de la herramienta (machuelo) con lo cual la pieza resultaría inservible.
5. Se debe considerar el tipo de cargas (axiales o radiales) que van a soportar los rodamientos para seleccionarlos correctamente.
6. Con esta tesis nos hemos dado cuenta que el diseño y construcción de maquinaria en nuestra universidad es posible y rentable, por lo que consideramos la maquinaria existente se encuentra subutilizada.

Software:

1. El tipo de programación gráfica ayuda en aspectos visuales como indicadores gráficos, pero en cuanto a la comunicación USB presentó inconvenientes debido a la incompatibilidad del paralelismo de LabView con la comunicación serial, pero esto se logró solucionar mediante el uso de secuencias.
2. Los formatos dados por los dos programas de diseño de circuitos impresos tratados en esta tesis (Protel y PCBWizard) difieren por los encabezados y las funciones especiales de máquina que generan, por lo que nuestro software no es universal para todos los programas de diseño de PCB, pero se puede agregar más, mediante un previo análisis del encabezado y la programación respectiva.
3. Se podría implementar un control completo de CNC mediante el puerto paralelo, para así obviar el uso del microcontrolador, pero éste tipo de puertos no se encuentra en computadores actuales.
4. El software diseñado, por el momento sólo corre bajo la plataforma de Windows XP, debido a la incompatibilidad de drivers en Windows Vista.

Generales:

1. Para el desarrollo de proyectos con CNC resulta muy difícil encontrar información al respecto, debido a que en el mercado nacional existen pocas personas que se dedican a la automatización de este tipo de maquinaria. El código que se consiguió es cerrado, y de propiedad de Hewlett Packard, lo que encarecería la producción en serie de máquinas debido al pago de patentes.

Recomendaciones

1. Se debe consultar a personal con basta experiencia en mecanizado con arranque de viruta para conseguir alta precisión en las piezas, y disminuir el tiempo de mecanizado
2. Se debe tomar en cuenta el tiempo de construcción de la parte mecánica para elaborar un buen cronograma de trabajo, así se evita el desperdicio de horas-máquina y horas-hombre

Bibliografía

- [1] Barrientos, Peñin, Balaguer y Aracil, “Fundamentos de robotica”. Universidad Politécnica de Madrid. Editorial McGrawn Hill. Madrid 1997
- [2] “Stepping motors fundamental”, application note. Microchip
- [3] Axelson, Jan. “USB complete” tercera edición. Editorial Lakeview Research LLC. Madison 2005
- [4] “Gerber RS274X-Format”. Barco Graphics, Gent, Belgium, and Barco Gerber Systems. Estados Unidos 1998
- [5] American National Standards Institute (ANSI) American National Safety Standard ANSI/ RIA R15.06-1992. Industrial Robots and Robot Systems - Safety Requirements. American National Standards Institute, Inc., 1430 Broadway, New York, New York 10018
- [6] National Institute for Occupational Safety and Health (NIOSH) Alert Publication No. 85103. Request for Assistance in Preventing the Injury of Workers by Robots. National Institute for Occupational Safety and Health, Division of Safety Research, 944 Chestnut Ridge Road, Morgantown, West Virginia 26505
- [7] Naranjo, Jhon Jairo, “Electrónica & Computadores”, Cedit

Anexos

Programa para el microcontrolador:

* Name : Taladro_XYZ.BAS *

* Author : [Dávila,Galarza] *

* Notice : Copyright (c) 2009 *

* : All Rights Reserved *

* Date : 8/18/2009 *

* Version : 1.0 *

Device = 18F4550

XTAL = 48

ALL_DIGITAL=On

'declaracion de variables

Dim tiempo As Byte

tiempo = 50

'Establece el tiempo de retardo para el encendido de las bobinas

Dim auxa[4] As Byte

auxa[0] = %00010011

auxa[1] = %00010110

auxa[2] = %00011100

auxa[3] = %00011001

'secuencia de encendido de los motores en sentido positivo

Dim auxb[4] As Byte

auxb[0] = %00011001,

auxb[1] = %00011100,

auxb[2] = %00010110,

auxb[3] = %00010011

'secuencia de encendido de los motores en sentido negativo

Dim enerx As DWord

Dim enery As DWord

Dim enerz As DWord

'numero de pasos a recorrer

Dim g As Byte

Dim d As Byte

Dim e As Byte

Dim h As Byte

Dim t As Byte

Dim k As Byte

Dim a As Byte

Dim cont As Byte

Dim sus As Byte

Dim llego As Byte

Dim todo As Byte

Dim ayudaju As Byte

Dim ayudaj As Byte

Dim espera As Byte

'variables a utilizar

'o es la que avisa al PC del estado de la operacion o programa

Dim x As DWord

Dim y As DWord

Dim ayuda As DWord

Dim q As DWord

Dim ayudan As DWord

Dim i As DWord

'variables de los datos

Symbol finx PORTC.7

Symbol finy PORTC.6

Symbol finz PORTC.1

Symbol Taladro PORTC.0

PORTA = %00000000 ' Estable las salidas en 0 de X

PORTB = %00000000 ' Estable las salidas en 0 de Y

PORTD = %00000000 ' Estable las salidas en 0 de Z

TRISA = %00000000 ' Declara al puerto A como salida de X

TRISB = %00000000 ' Declara al puerto B como salida de Y

TRISD = %00000000 ' Declara al puerto D como salida de Z

TRISC = %11000100 ' entrada de los fin carrera y control del taladro

' RC7 fin carrera x

' RC6 fin carrera y

' RC1 fin carrera z

' RC0 activacion taladro

Declare CCP1_PIN PORTC.2

HPWM 1,127,10 'activo pwm para bobinas

DelayMS 2000

PORTA.0=1

DelayMS 2000

PORTA.0=0

' descriptor file, located in \inc\usb_18 - a copy

' is located in the same folder as this file

USB_DESCRIPTOR = "USBProjectDESC.inc"

' we are going to use an interrupt to keep the USB

' connection alive, so disable auto polling from within

' the USBIn and USBOut commands

USBIN_AUTO_POLL = OFF

USBOUT_AUTO_POLL = OFF

' USB buffer...

Symbol USBBufferSizeMax = 8

Symbol USBBufferSizeTX = 8

Symbol USBBufferSizeRX = 8

Dim USBBuffer[USBBufferSizeMax] As Byte

' some useful flags...

Dim PP0 As Byte SYSTEM ' USBPOLL status return

Symbol CARRY_FLAG = STATUS.0 ' high if microcontroller does not have control over the buffer

Symbol ATTACHED_STATE = 6 ' is USB attached

Symbol TRNIF = UIR.3 ' low if USB Busy

Dim TIMER1 As TMR1L.Word ' access 16 bits of TMR1

' use an interrupt to keep USB connection alive...

Symbol TimerPreload = \$A23F ' approx 2ms

ON_HARDWARE_INTERRUPT GoTo USBServiceInterrupt

GoTo ProgramStart

' *****

' * the USB interrupt service routine maintains the *

' * connection to the bus - without this routine, a *

' * call to USBPoll, USBIn or USBOut must be made *

' * every couple of milliseconds or so *

' *****

USBServiceInterrupt:

' don't poll USB if it's busy...

btfss TRNIF

bra ExitInterrupt

' poll the USB interface...

Call (Check@BusStatus)

Call (Driver@Service)

' clear interrupt flag and exit...

ExitInterrupt:

bcf PIR1,0

TIMER1 = TimerPreload

retfie fast

' *****

' * program starts here... *

' *****

ProgramStart:

GoSub AttachToUSB

' turn on interrupt timer...

TIMER1 = TimerPreload

T1CON = %10000001

PIE1 = %00000001 ' enable TMR1 overflow interrupt

INTCON = %11000000 ' enable global and peripheral interrupts

GoTo ProgramLoop

'subprocedimientos

'++++
++++

energi:

DelayMS tiempo

Return

'retardo para las bobinas

pauComu:

DelayMS 10

Return

valora:

If enerx = 3 Then

enerx = -1

EndIf

enerx = enerx + 1

Return

'para calcular la posicion siguiente del motor a pasos en X

valorb:

If enery = 3 Then

 enery = -1

EndIf

enery = enery + 1

Return

'para calcular la posicion siguiente del motor a pasos en Y

valorc:

If enerz = 3 Then

 enerz = -1

EndIf

enerz = enerz + 1

Return

'para calcular la posicion siguiente del motor a pasos en Z

llamaxp:

GoSub valora

PORTA = auxa [enerx]

Return

'para calcular la posicion siguiente del motor a pasos en X

llamayp:

GoSub valorb

PORTB = auxa[enery]

Return

'para calcular la posicion siguiente del motor a pasos en Y

llamazp:

GoSub valorc

PORTD = auxa[enerz]

Return

'para calcular la posicion siguiente del motor a pasos en Z

llamaxs:

GoSub valora

PORTA = auxb [enerx]

Return

'para calcular la posicion siguiente del motor a pasos en X

llamays:

GoSub valorb

PORTB = auxb[enery]

Return

'para calcular la posicion siguiente del motor a pasos en Y

llamazs:

GoSub valorc

PORTD = auxb[enerz]

Return

'para calcular la posicion siguiente del motor a pasos en Z

llegada:

Select USBBuffer[llego]

Case "0"

ayudaj = 0

Case "1"

ayudaj = 1

Case "2"

ayudaj = 2

Case "3"

ayudaj = 3

Case "4"

ayudaj = 4

Case "5"

ayudaj = 5

Case "6"

ayudaj = 6

Case "7"

ayudaj = 7

Case "8"

ayudaj = 9

Case "-"

ayudaju = 0

Case "+"

ayudaju = 1

EndSelect

If USBBuffer[llego] = "+" Or USBBuffer[llego] = "-"Then

llego=llego

Else

ayudan = ayudan /10

ayuda = ayuda + (ayudaj * ayudan)

EndIf

If ayudan = 1 And todo=5 Then

todo = 6

EndIf

If ayudan = 1 And todo = 7 Then

todo = 8

g = 3

EndIf

Return

'no identifica en recibimos

juancho:

Leyendo:

GoSub pauComu

USBBuffer[0]="o"

GoSub DoUSBOut

espera = 1

While espera = 1

GoSub pauComu

GoSub DoUSBIn

If USBBuffer[0]=="p" Or USBBuffer[0]=="d" Or USBBuffer[0]=="s" Then

espera = 2

EndIf

Wend

Select Case USBBuffer[0]

Case "p" 'play

g = 3

Case "d" 'pause

GoTo Leyendo

Case "s" 'stop

g = 1

q = 10000000

EndSelect

Return

'ordenes de la compu

pasosenx:

If d = 1 Then

 GoSub llamaxp

Else

 GoSub llamaxs

EndIf

'ESTOY DANDO LOS PASOS EN (X)

Return

pasoseny:

If e = 1 Then

 GoSub llamayp

Else

 GoSub llamays

EndIf

'ESTOY DANDO LOS PASOS EN (Y)

Return


```
'+++++
```

```
'*****
```

```
' * main program loop *
```

```
'*****
```

ProgramLoop:

```
' your code in here - to send data, load USBBuffer
```

```
' and then call DoUSBOut. To receive data, call DoUSBIn...
```

```
Taladro = 0
```

```
enerx = 0
```

```
enery = 0
```

```
enerz = 0
```

```
g = 1
```

```
sus = 1
```

```
While sus = 1
```

```
While g = 1
```

```
GoSub energi
```

```
' tiempo de espera para activar la siguiente bobina
```

If finx = 0 And finy = 0 And finz = 0 Then 'posicion inicial maquinas

g = 2

PORTA.4=0

PORTB.4=0

PORTD.4=0

USBBuffer[0]="g"

GoSub DoUSBOut

Else

If finx = 1 And finy = 1 And finz = 1 Then

GoSub llamaxs

GoSub llamays

GoSub llamazs

Else

If finx = 1 And finy = 1 Then

GoSub llamaxs

GoSub llamays

PORTD.4=0

Else

If finx = 1 And finz = 1 Then

GoSub llamaxs

GoSub llamazs

PORTB.4=0

Else

If finz = 1 And finy = 1 Then

GoSub llamazs

GoSub llamays

PORTA.4=0

Else

If finx = 1 Then

GoSub llamaxs

PORTB.4=0

PORTD.4=0

Else

If finy = 1 Then

GoSub llamays

PORTA.4=0

PORTD.4=0

Else

If finz = 1 Then

GoSub llamazs

PORTA.4=0

PORTB.4=0

EndIf

EndIf

EndIf

EndIf

EndIf

EndIf

EndIf

EndIf

Wend

'colocamos el taladro en el origen

While g = 2

ayuda = 0

todo = 1

ayudan = 10000000

While todo = 1

GoSub pauComu

GoSub DoUSBIn

If USBBuffer[0] == "a" Then

 todo=2

 Else

 EndIf

Wend

While todo=2

 GoSub pauComu

 USBBuffer[0]="b"

 GoSub DoUSBOut

 todo=3

Wend

While todo = 3

 GoSub pauComu

 GoSub DoUSBIn

 If USBBuffer[0]=="c" Then

 todo=4

 EndIf

Wend

While todo = 4

 GoSub pauComu

USBBuffer[0]="x"

GoSub DoUSBOut

todo = 5

Wend

While todo = 5

GoSub pauComu

GoSub DoUSBIn

llego=0

If USBBuffer[0]=="+" Or USBBuffer[0]=="-" Then

For i = 0 To 7 Step 1

llego = i

GoSub llegada

Next

EndIf

Wend

x=ayuda

d=ayudaju

ayuda = 0

ayudan = 10000000

While todo = 6

GoSub pauComu

USBBuffer[0]="y"

GoSub DoUSBOut

todo = 7

Wend

While todo = 7

GoSub pauComu

GoSub DoUSBIn

If USBBuffer[0]="+" Or USBBuffer[0]="-" Then

For i = 0 To 7 Step 1

llego = i

GoSub llegada

Next

EndIf

Wend

y=ayuda

e=ayudaju

Wend

While g = 3

 If x > y Then

 h = 1

 Else

 h = 0

 ayuda = y

 y = x

 x = ayuda

 EndIf

'ordenamos el numero de pasos

GoSub juancho

If g = 3 Then

 For q = 1 To y

 GoSub pasosenx

 GoSub pasoseny

 GoSub energi

 GoSub juancho

Next

y = x - y

'CUENTO LOS PASOS QUE FALTAN

EndIf

If g = 3 Then

If h = 1 Then

PORTB.4=0

For q = 1 To y

GoSub pasosenx

GoSub energi

GoSub juancho

Next

'si h=1 damos los pasos que faltan a X

Else

PORTA.4=0

For q = 1 To y

GoSub pasoseny

GoSub energi

GoSub juancho

Next

EndIf

'si h=0 damos los pasos que faltan a Y

EndIf

If g = 3 Then

PORTA.4=0

PORTB.4=0

Taladro=1

For q = 1 To 4 '100 hay que cambiarle

GoSub llamazp

GoSub energi

GoSub juancho

Next

EndIf

If g = 3 Then

For q = 1 To 4 ' hay que cambiar

GoSub llamazs

GoSub energi

GoSub juancho

If q=4 Then

Taladro=0

GoSub pauComu

USBBuffer[0]="t"

GoSub DoUSBOut

g = 2

EndIf

Next

'regresamos el taladro a la posicion inicial

EndIf

Wend

Wend

GoTo ProgramLoop

' *****

' * receive data from the USB bus *

' *****

DoUSBIn:

USBIn 1, USBBuffer, USBBufferSizeRX, DoUSBIn

Return

' *****

' * transmit data *

' *****

DoUSBOut:

USBOut 1, USBBuffer, USBBufferSizeTX, DoUSBOut

Return

' *****

' * wait for USB interface to attach *

' *****

AttachToUSB:

Repeat

USBPoll

Until PP0 = ATTACHED_STATE

Return

Valores normalizados cables A.W.G (American Wire Gauge Standard)

Conductores en buen estado deben presentar una resistencia muy baja. Los valores dependen del largo y del grosor de los hilos. Para hilos comunes hasta 20 m. de largo la resistencia debe ser siempre inferior a 1 O. Para hilos esmaltados la resistencia varia bastante en función del espesor. En la siguiente tabla de hilos se da información de resistencia por unidad de longitud para los diversos tipos

Número AWG	Diámetro (mm)	Sección (mm ²)	Espiras/cm	Kg/Km.	Resistencia (Ω/Km.)	Capacidad [A]
0000	11,86	107,2	-		0,158	319
000	10,40	85,3	-		0,197	240
00	9,226	67,43	-		0,252	190
0	8,252	53,48	-		0,317	150
1	7,348	42,41	-	375	1,40	120
2	6,544	33,63	-	295	1,50	96

3	5,827	26,67	-	237	1,63	78
4	5,189	21,15	-	188	0,80	60
5	4,621	16,77	-	149	1,01	48
6	4,115	13,30	-	118	1,27	38
7	3,665	10,55	-	94	1,70	30
8	3,264	8,36	-	74	2,03	24
9	2,906	6,63	-	58,9	2,56	19
10	2,588	5,26	-	46,8	3,23	15
11	2,305	4,17	-	32,1	4,07	12
12	2,053	3,31	-	29,4	5,13	9,5
13	1,828	2,63	-	23,3	6,49	7,5
14	1,628	2,08	5,6	18,5	8,17	6,0
15	1,450	1,65	6,4	14,7	10,3	4,8
16	1,291	1,31	7,2	11,6	12,9	3,7
17	1,150	1,04	8,4	9,26	16,34	3,2
18	1,024	0,82	9,2	7,3	20,73	2,5
19	0,9116	0,65	10,2	5,79	26,15	2,0
20	0,8118	0,52	11,6	4,61	32,69	1,6

21	0,7230	0,41	12,8	3,64	41,46	1,2
22	0,6438	0,33	14,4	2,89	51,5	0,92
23	0,5733	0,26	16,0	2,29	56,4	0,73
24	0,5106	0,20	18,0	1,82	85,0	0,58
25	0,4547	0,16	20,0	1,44	106,2	0,46
26	0,4049	0,13	22,8	1,14	130,7	0,37
27	0,3606	0,10	25,6	0,91	170,0	0,29
28	0,3211	0,08	28,4	0,72	212,5	0,23
29	0,2859	0,064	32,4	0,57	265,6	0,18
30	0,2546	0,051	35,6	0,45	333,3	0,15
31	0,2268	0,040	39,8	0,36	425,0	0,11
32	0,2019	0,032	44,5	0,28	531,2	0,09
33	0,1798	0,0254	56,0	0,23	669,3	0,072
34	0,1601	0,0201	56,0	0,18	845,8	0,057
35	0,1426	0,0159	62,3	0,14	1069,0	0,045
36	0,1270	0,0127	69,0	0,10	1338,0	0,036
37	0,1131	0,0100	78,0	0,089	1700,0	0,028
38	0,1007	0,0079	82,3	0,070	2152,0	0,022

39	0,0897	0,0063	97,5	0,056	2696,0	0,017
40	0,0799	0,0050	111,0	0,044	3400,0	0,014
41	0,0711	0,0040	126,8	0,035	4250,0	0,011
42	0,0633	0,0032	138,9	0,028	5312,0	0,009
43	0,0564	0,0025	156,4	0,022	6800,0	0,007
44	0,0503	0,0020	169,7	0,018	8500,0	0,005