



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE INGENIERÍA ELECTRÓNICA

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

“INGENIERO ELECTRÓNICO”

TEMA:

**“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE CONTROL
NUMÉRICO COMPUTARIZADO PARA LA ELABORACIÓN DE CIRCUITOS
IMPRESOS EN BAQUELITA”**

AUTORES:

**MARCO ANTONIO VALDIVIEZO MILLÁN
TEDDY JHENNSE NEGRETE DURAZNO**

TUTOR:

MSc. LUIS CÓRDOVA

GUAYAQUIL – ECUADOR

2018

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA

Los conceptos desarrollados en el presente proyecto de titulación, el desarrollo del tema, el análisis y respectivas conclusiones de este, corresponden exclusivamente a **Marco Antonio Valdiviezo Millán** con cédula de identificación 0925337131 y **Teddy Jhennse Negrete Durazno** con cédula de identificación 0951133099 y el patrimonio intelectual del mismo a la **Universidad Politécnica Salesiana**.

Guayaquil, julio del 2018

(f) Marco Antonio Valdiviezo Millán

C.I: 0925337131

(f) Teddy Jhennse Negrete Durazno

C.I: 0951133099

DOCUMENTO DE DERECHOS DE AUTOR

La responsabilidad y la autoría del contenido de este trabajo de titulación, corresponde exclusivamente a los autores; los mismos que damos nuestro consentimiento para que la UPS-GYE, realice la publicación de la obra por cualquier medio que considere conveniente con el fin de promover la consulta y difusión pública de la producción intelectual.

Guayaquil, julio del 2018

(f) Marco Antonio Valdiviezo Millán

C.I: 0925337131

(f) Teddy Jhennse Negrete Durazno

C.I: 0951133099

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo Msc. Luis Córdova certifico que bajo mi dirección y asesoría fue desarrollado el proyecto técnico de titulación “**Diseño e implementación de un prototipo de sistema de control numérico computarizado para la elaboración de circuitos impresos en baquelita**”, realizado por los señores **Marco Antonio Valdiviezo Millán** y **Teddy Jhennse Negrete Durazno** para la obtención del título de **Ingeniero Electrónico**.

Por medio de la presente certifico que el documento cumple con los requisitos establecidos en el Instructivo para la Estructura y Desarrollo de Trabajos de Titulación para pregrado de la Universidad Politécnica Salesiana.

En virtud de lo anterior, autorizo su presentación y aceptación como una obra auténtica y de alto valor académico.

Guayaquil, julio del 2018

(f) Luis Córdova

DEDICATORIA

A Dios por haberme dado salud para lograr mis objetivos, además de su infinito amor y haberme permitido llegar hasta este punto. A mi madre por sus consejos, sus valores, su motivación constante pero más que nada, por su amor que me ha permitido ser una persona de bien. A mi padre por los ejemplos de constancia, esfuerzo, perseverancia y sus valores que lo caracterizan que me ha infundado siempre, por su amor y confianza para salir adelante. En memoria de mi tía Flor Durazno por haberme inculcado valores y por haberme brindado todo su amor ya que en vida fue como una segunda madre.

Teddy Negrete

Dedico este proyecto de titulación en primer lugar a Dios, por la fortaleza y sabiduría que me ha dado para llevar a cabo cada una de mis metas y en particular por concluir el presente trabajo.

A mi madre Mariana Millán quien ha sido pilar fundamental durante mi vida, con su apoyo incondicional ha sido partícipe para llegar a esta meta y en memoria de mi hermana

María Fernanda Valdiviezo.

Marco Valdiviezo

AGRADECIMIENTOS

Agradezco a Dios por darme el coraje para concluir este proyecto, por haberme guiado y acompañado a lo largo de mi carrera. Le agradezco a mi familia, por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida. Le doy gracias a mis profesores y amigos que ayudaron de muchas maneras para que pudiera llevar a cabo este proyecto.

Teddy Negrete

Agradezco a Dios por ser mi fortaleza en los momentos de debilidad y por brindarme una vida llena de aprendizajes. Le doy gracias a mi familia por su apoyo moral, económico y a mis amigos que contribuyeron para este logro que mucho ansié.

A mi tutor Msc. Luis Córdova quien siempre nos brindó su conocimiento para culminar con éxito nuestro proyecto de titulación.

Marco Valdiviezo

RESUMEN

El presente proyecto consiste en el diseño y construcción de una máquina CNC dedicada a la elaboración de circuitos impresos a través de la fabricación por fresado. Una máquina CNC con interfaz por computador e interfaz manual para cargar los archivos desde un PC o desde una memoria extraíble. Se muestra el procedimiento de ejecución del proyecto, los algoritmos utilizados para la elaboración del firmware, los materiales empleados. Las pruebas del prototipo consisten en la operación de trazos geométricos simples y el trabajo de elaboración de una muestra de circuito impreso. Se detallan los problemas y las soluciones encontradas luego de las primeras pruebas y los resultados al resolver los problemas.

El trabajo que se presenta en este documento pretende ser un complemento en la asistencia a los trabajos de los estudiantes que utilizan los equipos del Laboratorio de Diseño y Control en la Universidad Politécnica Salesiana.

GLOSARIO

- Baquelita

Es un plástico sintético, el cual no conduce la electricidad, sobre ella usualmente viene una capa de cobre (conductor metálico). Baekeland lo creó en 1907, aunque hubo algunos experimentos previos realizados por el alemán Adolf von Baeyer.

- Boardfile Layout

El Boardfile es el archivo en donde se realiza el diseño físico de la tarjeta de circuito impreso. El diseño de esta se le conoce como PCB o Layout.

- CAM

Fabricación asistida por ordenador.

- CAD/ CAM

Significa diseño asistido por computadora (Computer Aided Design) y fabricación asistida por computadora (Computer Aided Manufacturing).

El software CAD / CAM es utilizado para diseñar y fabricar prototipos, productos terminados y series de producción.

- CNC

El control numérico por computadora (Computer Numeric Control) es la automatización de máquinas herramienta mediante computadoras que ejecutan secuencias preprogramadas de comandos de control de máquina. Esto está en contraste con las máquinas que se controlan manualmente con ruedas manuales o palancas, o mecánicamente automatizadas solo con levas.

- Diagrama esquelético

Es una representación visual de las interconexiones de los elementos y partes de un circuito a realizar.

- Huella – Footprint

Para el diseño de los pads de un componente en particular existen huellas estándar y huellas específicas de cada componente.

En las hojas de datos técnicos (datasheet) de los componentes o dispositivos electrónicos, los fabricantes adjuntan las dimensiones de los componentes con lo que se puede generar un footprint.

- Pad

Zona de contacto en la PCB de un pin o terminal de un componente.

- Pad Through - hole

Zona de soldadura, son los agujeros circulares que constan de área de cobre para la inserción del terminal (drill).

- Pad smd Rectangulares

Es el área rectangular de cobre, se visualiza en la parte superior (cara TOP).

ÍNDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA.....	ii
DOCUMENTO DE DERECHOS DE AUTOR.....	iii
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.....	iv
DEDICATORIA.....	v
AGRADECIMIENTOS.....	vi
RESUMEN.....	vii
GLOSARIO.....	viii
ÍNDICE GENERAL.....	x
ÍNDICE DE FIGURAS.....	xiii
ÍNDICE DE TABLAS.....	xvi
1. INTRODUCCIÓN.....	1
EL ENTORNO.....	1
1.1. Descripción del problema.....	3
1.1. Delimitación.....	4
1.1.1. Espacial.....	4
1.1.2. Temporal.....	4
1.1.3. Académica.....	4
1.2. Importancia.....	7
1.3. Alcance.....	8
1.4. Justificación.....	9
1.5. Objetivos.....	9
1.5.1. Objetivo general.....	9
1.5.2. Objetivos específicos.....	10
2. MARCO TEÓRICO REFERENCIAL.....	11
2.1. Control numérico.....	11
2.1.1. Conceptos.....	11
2.1.2. Componentes del Control Numérico.....	11
2.1.3. Principio de funcionamiento.....	13
2.1.4. Control de movimiento.....	13
2.1.5. Accesorios programables.....	16
2.1.6. Programa CNC.....	17
2.1.7. Comandos CNC.....	19
2.1.8. Grupo de comandos de movimiento:.....	19
2.1.9. Grupo de comandos de selección de plano.....	20
2.1.10. Comandos de referencia de distancias.....	21
2.1.11. Comandos de velocidad de trabajo.....	21
2.1.12. Comandos de sistema de unidades:.....	21
2.1.13. Comandos de compensación de radio en corte.....	22
2.1.14. Comandos de desplazamiento de herramienta:.....	22
2.1.15. Sistema de selección de coordenadas:.....	23
2.1.16. Control de trayectoria.....	23
2.2. Controlador CNC.....	23
2.3. Programa CAM.....	26
2.4. Sistema DNC.....	27
2.5. Clasificación de los controles numéricos.....	29
2.5.1. Según el sistema de referencia.....	29
2.5.2. Según el control de las trayectorias.....	30
2.5.3. Según el tipo de accionamiento.....	32

2.5.4.	Según el bucle de control	33
2.5.5.	Según la tecnología de control.	34
2.6.	Ventajas y desventajas del CN	37
2.6.1.	Ventajas.....	37
2.6.2.	Desventajas	38
2.7.	Circuitos impresos	38
2.7.1.	Concepto	38
2.7.2.	Tipos de circuitos.....	40
2.7.3.	Elaboración de PCB mediante Ataque Químico.....	45
3.	MARCO METODOLÓGICO	48
3.1.	Descripción de la propuesta	48
3.1.1.	Firmware.....	49
3.1.2.	Modelado 3D	51
3.2.	Metodología del diseño.....	54
3.2.1.	Procedimiento de diseño	54
3.2.2.	Estructura mecánica	55
3.2.3.	Códigos G a actuadores	55
3.2.4.	Driver herramienta	56
3.2.5.	Carcasa	56
3.2.6.	Comunicación interfaz manual	56
3.2.7.	Calibración y ajuste	56
3.3.	Pruebas.....	57
3.4.	Construcción de la máquina CNC.....	57
3.4.1.	Estructura mecánica	57
4.	ANÁLISIS DE RESULTADOS	64
4.1.	Construcción	64
4.2.	Procedimiento de trabajo de la máquina	65
4.3.	Diagramas de flujo, algoritmo de trabajo.	66
4.4.	Driver de motores a paso.....	73
4.5.	Ensamble de tarjetas electrónicas del controlador	75
4.6.	Generación del código G	78
4.7.	Software envío Gcode	80
4.7.1.	Grbl Control	80
4.8.	Carcasa plástica	83
4.9.	Armado de carcasa.....	86
4.10.	Validación del prototipo	88
4.10.1.	Comunicación entre Gcode sender y el board de la máquina CNC.....	89
4.10.2.	Pruebas en la interfaz manual	91
4.10.3.	Pruebas de mecanizado.....	95
4.10.4.	Proceso de mecanizado.....	96
4.10.5.	Prueba de elaboración board:.....	98
	CONCLUSIONES.....	105
	PROBLEMAS PRESENTADOS.....	107
	Calibración y ajuste	107
	Prueba final para comprobar los resultados de la calibración	109
	Prototipo circuito estable	113
	RECOMENDACIONES	119
	REFERENCIAS BIBLIOGRÁFICAS.....	120
	ANEXOS	125
	Cronograma de actividades a desarrollar.....	125

Presupuesto.....	126
Código	127
Código de la interfaz manual	146
Esquemático del circuito controlador.....	155

ÍNDICE DE FIGURAS

Figura 1 - Ubicación de la Universidad Politécnica Salesiana. Google Maps.....	4
Figura 2 - Máquina CNC - Mecanizados (2015).....	12
Figura 3 - Funcionamiento Máquina CNC.....	13
Figura 4 - Accionamiento regulable (husillo) - Heidenhain (2012)	14
Figura 5 - Posiciones programadas	14
Figura 6 - Coordenadas absolutas. Universidad del País Vasco (2013).....	15
Figura 7 - Coordenadas incrementales. Universidad del País Vasco (2013).....	15
Figura 8 - Trayectoria y posición de ejes. Universidad del País Vasco (2013)	16
Figura 9 - Mapa Accesorios programables. Universidad del País Vasco (2013)	16
Figura 10 - Accesorios programables. Universidad del País Vasco (2013)	17
Figura 11 - Estructura bloque programa CNC. Universidad Nacional de la Plata (2014)	18
Figura 12 - Ejemplo programa CNC. Universidad Nacional de la Plata (2014)	19
Figura 13 - Controlador CNC (Peters, 2013).....	24
Figura 14 - RS-232C. Adrcó (2017)	28
Figura 15 - Sistema DNC. Adrcó (2017)	29
Figura 16 - CN punto a punto. Rolón (2011)	30
Figura 17 - CN paraxial. Rolón (2011)	31
Figura 18 - Máquina Control Numérico (CN). Rolón (2011).....	34
Figura 19 - Máquina CNC. Rolón (2011).....	36
Figura 20 - Circuito. (Electrosoft Ingeniería, 2010)	39
Figura 21 - Circuito orificios doble cara. RS (2010)	40
Figura 22 - Circuito impreso multicapa. (Cipsa Circuits © , 2016)	41
Figura 23 - Circuito impreso un lado. (Mouser electronics, 2017).....	42
Figura 24 - Circuito impreso CAD. (Google Sites, 2017)	46
Figura 25 - Componentes del sistema	49
Figura 26 - Modelado 3D realizado en AUTOCAD. (Radoff, 2008)	51
Figura 27 - Curva de corrientes.....	52
Figura 28 - Procedimiento de diseño	55
Figura 29 - Motor de paso.....	59
Figura 30 - Motor de herramienta.....	60
Figura 31 - Armado de la estructura.....	61
Figura 32 - Voltímetro.....	61
Figura 33 - Fresas para mecanizado	62
Figura 34 - Fuente de alimentación.....	62
Figura 35 - Tarjeta controladora Grbl 0.9.....	63
Figura 36 - Ensamble de máquina CNC (Paso 1).....	64
Figura 37 - Ensamble de máquina CNC (Paso 2).....	64
Figura 38 - Ensamble de máquina CNC (Paso 3).....	65
Figura 39 - Procedimiento para el trabajo de la máquina CNC.....	66
Figura 40 - Algoritmo de trabajo de la máquina	67
Figura 41 - Algoritmo de trabajo basado en Grbl	69
Figura 42 - Identificación de cada instrucción	71
Figura 43 - Diagrama de conexiones microcontrolador –drivers.....	73
Figura 44 - Diagrama de conexiones microcontrolador -drivers.	73
Figura 45 - Conexiones entre motor y drivers.	74
Figura 46 - Controlador del motor	75

Figura 47 - Controlador del motor	76
Figura 48 - Controlador del motor	77
Figura 49 - Tarjeta controladora ensamblada	77
Figura 50 - Controlador del motor	78
Figura 51 - Parámetros de calibración	79
Figura 52 - Exportando Gcode	80
Figura 53 - Interfaz de usuario software GrblControl	81
Figura 54 - Posicionamiento de la herramienta y desplazamiento en los ejes.	81
Figura 55 - Archivo de prueba para calibración cargado en el software Grbl controller.	82
Figura 56 - Transferencia del Archivo.	82
Figura 57 - Vista lateral	83
Figura 58 - Cara superior	84
Figura 59 - Cara anterior	84
Figura 60 - Cara inferior	85
Figura 61 - Cara frontal	85
Figura 62 - Tapa.....	86
Figura 63 - Toma de medidas	86
Figura 64 - Corte de piezas.....	87
Figura 65 - Armado de la carcasa	87
Figura 66 - Carcasa montada.....	88
Figura 67 - Interfaz de usuario	89
Figura 68 - Conexión al puerto COM	90
Figura 69 - Conexión establecida tarjeta-interfaz.....	90
Figura 70 - Movimientos en eje x, eje y, eje z	91
Figura 71 - Menú de interfaz manual	92
Figura 72 - Opción info del menú principal.....	92
Figura 73 - Pruebas de desplazamiento de los ejes.	93
Figura 74 - Información enviada por la tarjeta controladora.....	94
Figura 75 - Lectura de archivos en la interfaz manual	94
Figura 76 - Archivo de prueba de mecanizado	95
Figura 77 - Vista previa del trabajo	95
Figura 78 - Ajuste del punto de origen	96
Figura 79 - Inicio de trabajo.....	96
Figura 80 - Calado en forma circular	97
Figura 81 - Medición de trabajo final	97
Figura 82 - Interfaz gráfica, culminación de primera prueba.....	98
Figura 83 - Interfaz gráfica, culminación de primera prueba.....	99
Figura 84 - Interfaz gráfica, culminación de primera prueba.....	99
Figura 85 - Trayectorias para elaborar código G	100
Figura 86 - Trayectorias para corte en el borde	100
Figura 87 - Configuración punto origen.....	101
Figura 88 - Trayectorias para corte en el borde	101
Figura 89 - Trayectorias para corte en el borde	102
Figura 90 - Inicio de mecanizado	102
Figura 91 - Mecanizado mitad de trabajo, corte en el borde.....	103
Figura 92 - Corte final del prototipo.....	103
Figura 93 - Corte final.....	104
Figura 94 - Trabajo finalizado.....	104
Figura 95 - Trayectorias para corte en el borde	107
Figura 96 - Ajuste de altura de rieles del eje X	108
Figura 97 - Ajuste de altura en el lado derecho	108

Figura 98 - Ajuste de altura en el lado izquierdo.....	109
Figura 99 - Segunda Prueba.....	109
Figura 100 - Generación código G.....	110
Figura 101 - Muestra las trayectorias generadas por el software CAM para el código G....	110
Figura 102 - Vista previa del trabajo a mecanizar.....	111
Figura 103 - Inicio de mecanizado.....	111
Figura 104 - Trabajo de mecanizado a mitad del proceso.....	112
Figura 105 - Trabajo final luego de calibración.....	112
Figura 106 - Comparación de trabajos.....	113
Figura 107 - Esquemático en Eagle CAD.....	114
Figura 108 - PCboard diseñado en Eagle CAD.....	114
Figura 109 - Exporte de archivos Gerber.....	115
Figura 110 - Trayectorias de código G.....	116
Figura 111 - Prototipado final. Circuito estable.....	116
Figura 112 - Máscara antisoldante.....	117
Figura 113 - Trayectorias para corte en el borde.....	117
Figura 114 - Trayectorias para corte en el borde.....	118

ÍNDICE DE TABLAS

Tabla 1 - Materiales para Máquina CNC.....	57
Tabla 2 - Valores de revolución según voltaje.....	59
Tabla 3 - Cronograma de actividades (Diagrama de Gantt).....	125
Tabla 4 - Presupuesto para el proyecto de titulación.....	126

1. INTRODUCCIÓN

EL ENTORNO

A lo largo de la historia, los sistemas económicos han evolucionado de la mano de la elaboración de productos o servicios, para los que se han utilizado procesos de transformación que involucran una secuencia de operaciones ya sea en la fabricación o en el montaje o en el embalaje de estos, etc., desarrollándolos a través del uso de maquinarias, equipos o insumos.

Parte importante de los sistemas productivos modernos está basado en sus características de flexibilidad y alta productividad, unidas por el uso de una automatización programable que hace posible el desarrollo de ambas características. Entiéndase por flexibilidad, la posibilidad de realizar una diversidad de piezas ya sea por responder al mercado, a la necesidad o al producto propiamente dicho y, por otro lado, la parte productiva se encasilla en el proceso estricto de la planificación, gestión y el control simultáneo de todo (González & Rosado, 2015).

Las máquinas de control numérico son la base fundamental para que un sistema de producción moderno sea flexible, ya que se basan en una programación y son de controles universales, las que presentan distintos grados de automatización desde el proceso de automatización hasta las que incluyen otros automatismos de carga y descarga de piezas, haciendo a la máquina de funcionamiento autónomo.

Pero estos sistemas no solo se utilizan en la industria, sino que vienen siendo operadas desde la Universidad, en las Carreras de Ingeniería que es donde se desarrollan las habilidades de uso de este tipo de sistemas para ser puesto en práctica

con gran habilidad ya en el campo industrial. En ese proceso, la Universidad Politécnica Salesiana se ha especializado, a través de la Carrera de Ingeniería.

El ámbito de profesionalización que tienen los ingenieros graduados en la Universidad Politécnica Salesiana se desarrolla con asistencia técnica en el sector energético, como ingenieros electrónicos para diseñar aplicaciones biomédicas que permiten mejorar la calidad de vida de las personas y optimizar los recursos en la construcción de equipos biomédicos, con alta calidad a fin de aportar significativamente y garantizar el cumplimiento de los estándares de seguridad industrial y ambiental, mediante la propuesta de líneas de investigación con lo que se busca la innovación y la transferencia de tecnología en búsqueda del mejoramiento de procesos productivos, procesos ampliamente complejos y especializados en el sector electrónico, por lo que su complejidad requiere ser tratada con mayor certeza.

La malla curricular donde se detalla las materias que deben ver los estudiantes a lo largo de la Carrera de Ingeniería Electrónica se considera la materia denominada Teoría del Diseño, que se encarga de impartir todo lo relacionado con constantes localizadas, resistores de todos tipos y tamaños, así como capacitores, inductores además de fuentes de corriente y de tensión y relacionados a la elaboración de circuitos impresos en varios tamaños. Sin embargo, la Universidad no cuenta con el sistema de Control Numérico Computarizado para que los estudiantes de la universidad realicen las prácticas en el laboratorio y puedan, en tiempo real, identificar claramente el uso de estas desde sus experiencias personales, causando una deficiencia en el aprendizaje de la materia Teoría del Diseño.

1.1. Descripción del problema

El laboratorio de la UPS tiene equipos costosos y necesitan supervisión de personal técnico encargado para realizar la operación respectiva, mientras que la máquina CNC que se propone es de fácil uso y menos costosa, permitiendo así la fácil operación de la máquina por parte de los estudiantes.

Una de las características diferenciadoras de nuestro proyecto es el bajo coste para su implementación en comparación a otros equipos mucho más costosos, a los cuáles los estudiantes no tienen acceso.

En el proceso de elaboración se tiene riesgos, causar daños para la salud como la piel, porque se utiliza cloruro férrico para eliminar el cobre sobrante de la baquelita, este proceso se debe realizar con mucha precaución.

Por estos motivos los estudiantes de los semestres básicos de Ingeniería Electrónica y Carreras afines realizan circuitos impresos de manera artesanal muy rústicos y poco precisos.

Los materiales para utilizarse en la elaboración de los circuitos impresos, como el acetato implican un alto costo, poseen diferentes características de fabricación y no se optimiza el trabajo final al ser el proceso casero.

En una de las etapas finales se genera otro inconveniente, al momento de realizar las perforaciones, resultan imprecisas y varía su tamaño lo que desencadena en una falla al hacer el encaje de los elementos electrónicos en la baquelita.

1.1. Delimitación

1.1.1. Espacial

El siguiente proyecto se destinará al Laboratorio de Diseño y Control para que pueda ser usado por los estudiantes de la Universidad Politécnica Salesiana sede Guayaquil que se encuentre ubicada en la Avenida Domingo Comín, tal como se muestra en la Figura 1.

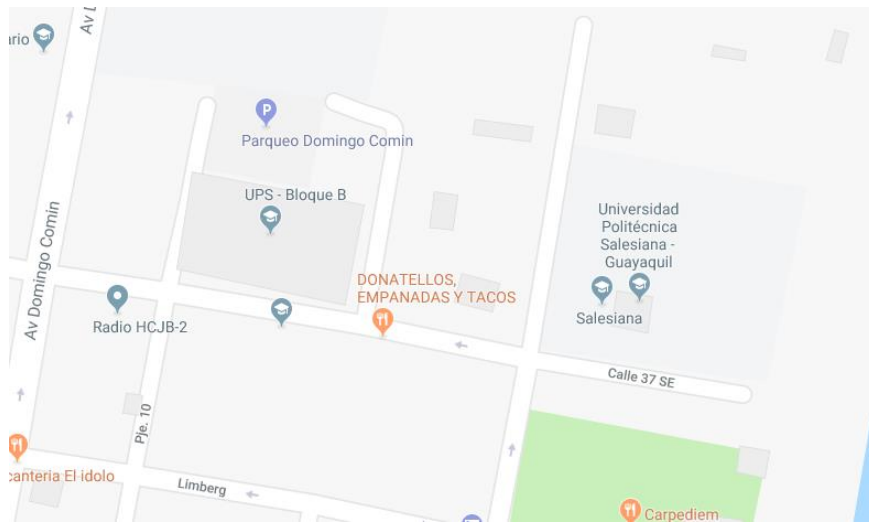


Figura 1 - Ubicación de la Universidad Politécnica Salesiana. Google Maps

1.1.2. Temporal

EL presente proyecto de titulación será efectuado en el período de septiembre del 2017 a diciembre del 2018.

1.1.3. Académica

El presente proyecto de titulación permitirá elaborar circuitos impresos de hasta 20 x 15 cm a una sola capa, con el fin de que los estudiantes de próximas generaciones puedan tener mayor variedad de instrumentos para realizar sus prácticas universitarias. Se usaron los conocimientos aprendidos en Diseño de Control para elaborar el

esquemático de los circuitos eléctricos que queremos implementar, empleando el CAD de diseño de circuitos impresos más apropiado por cada usuario.

Antecedentes

En la actualidad las máquinas de control numérico tienen un inmenso campo de acción y un gran abanico de aplicaciones. Usualmente en el mercado se encuentran máquinas de uso industrial que van desde máquinas para tallar madera, máquinas de corte láser, máquinas de corte por chorro de agua, de corte por plasma, entre otras, y es que el principio de todas es el mismo, el Control Numérico Computarizado (CNC), solo difieren en el tipo de herramienta. En el campo de los circuitos impresos, las máquinas de control numérico han tenido gran acogida en los laboratorios de investigación por su versatilidad en cuanto al prototipado rápido que implica su uso, evitando la necesidad de elementos químicos que se emplean en otros procesos de producción de circuitos impresos. La empresa LPKF Láser&Electronics dispone de varios modelos de máquinas CNC usadas para prototipar circuitos impresos a través del fresado, y otras máquinas que complementan esta tarea, como las máquinas para trabajos con los anillos metalizados de los PCBS, así también dispone de máquinas de corte láser usando este tipo de control.

Por su utilidad muy amplia, los sistemas de Control Numérico Computarizado son tema de investigación y constante trabajo, no solo en la industria sino también en centros de académicos. En la Universidad Politécnica Salesiana, Arpi y Cabrera (2013) realizaron la implementación de un sistema de control para una máquina CNC láser en el cual se basa en un generador de trayectorias para así elegir la ruta más corta para el desplazamiento de la herramienta, además consideran los aspectos de inercia que se

deben tener en cuenta para los actuadores que en un instante determinado deben cambiar su estado de movimiento, del reposo al desplazamiento. Exponen que el diseño de un control para máquinas CNC cuya estructura se basa en un robot cartesiano es más sencillo que otras configuraciones robóticas, ya que el cálculo cinemático es más simple debido a que el movimiento de un eje no está en dependencia lineal con el movimiento de otro eje.

Andino (2016) se encargó de realizar el diseño e implementación de un alimentador automático de barras para un torno CNC el cual es alimentado con barras circulares de acero inoxidable. Consiste en una técnica de sustento que marcha de forma vinculada con el controlador del torno CNC. En ese hardware se ha efectuado el sistema de control del alimentador. Para este objetivo Andino selecciona el tipo de alimentador y mecanismo que más se ajusta a los requerimientos de diseño de la máquina, espacio disponible, funcionalidad, y costos. El centro del presente proyecto es el desarrollo de un sistema que trabaja de manera conjunta con el torno CNC, el desarrollo de este tipo de accesorios es de gran importancia para la automatización de los procesos industriales y volver más eficientes los procesos de producción. En el proyecto mencionado, Andino (2016) desarrolla uno de estos accesorios, algunas máquinas de mecanizado CNC tienen accesorios de tambor giratorio, lo que la vuelve más competente en trabajos que requieren desarrollarse en más de 3 ejes.

Para desarrollar sistemas más eficientes en cuanto al control de los actuadores y mecanismos, Abril y Pacheco (2012), realizaron un control de velocidad difuso aplicado al robot móvil Robotito, este proyecto muestra que un control basado en lógica difusa ayuda a reducir la inestabilidad de un sistema móvil respecto a los controladores

convencionales. Obteniendo con el controlador difuso eficiencias del 69.8% respecto al controlador convencional.

Por otro lado, en la Escuela Superior Politécnica del Litoral, la Facultad de Ingeniería Mecánica y Ciencias de la Producción publicó los resultados de un proyecto dedicado a la implementación de una máquina fresadora CNC de grandes dimensiones para la elaboración de prototipos de barcos. El control utilizado en esta máquina es de tipo estándar, mediante el uso de motores de paso a paso en lugar de motores DC convencionales, esto debido a la facilidad de control de posición que tienen y de operación. Hacen énfasis en que el límite de 3 ejes restringe muchas otras opciones de mecanizado que la pudieran volver más versátil para los trabajos en 3D. Están de acuerdo en usar programas licenciados en lugar de programas libres, esto debido a que los licenciados poseen una amplia gama de herramientas la generación de código G más eficiente. Sánchez y Villavicencio (2004) en su proyecto detallan el manejo de los microcontroladores sus características, la arquitectura interna, los tipos de registros, puertos y el convertidor analógico digital.

1.2. Importancia

El diseño de PCB en la actualidad es de vital importancia ya que todo producto que lleva algún componente electrónico necesita de un circuito impreso donde se conecten los dispositivos.

Para poder producir en masa los circuitos anteriormente mencionados es necesario que el trabajo sea automatizado mediante alguna máquina especializada, con el fin de evitar errores humanos, reducir el coste de fabricación de los PCB y sobre todo poder producirlos en grandes cantidades.

1.3. Alcance

En el presente proyecto de titulación se va a diseñar e implementar un prototipo de sistema de control numérico el cual permitirá diseñar circuitos impresos, cuyo alcance es:

- Armar la estructura mecánica de una CNC.
- Instalación del firmware de la CNC en un microcontrolador.
- Convertir el modelo en 3D a Gcode entendible por la CNC.
- Comunicar la CNC con una computadora vía comunicación serial.
- Calibrar y ajustar los componentes de la CNC.

1.4. Justificación

Los sistemas de Control Numérico Computarizado no son tan frecuentes en instituciones educativas o empresas, debido a los altos costos de sus equipos y su implementación.

Las industrias en nuestro país buscan trabajar con la tecnología que supla la demanda de producción nacional y apuntarse al mercado internacional, en este medio surge la fabricación de productos asistidos por un computador (CNC) y automatización de procesos.

Elaborar una máquina asistida por un ordenador para su fabricación es posible y es un impulso para que nuestro país mejore su producción, utilice tecnología actual y se genere la capacitación, mantenimiento y conocimiento de dicha tecnología.

En la Carrera de Ingeniería Electrónica con mención en Sistemas Industriales de la Universidad Politécnica Salesiana Sede Guayaquil se ha realizado la propuesta de un proyecto, el cual consiste en generar un prototipo de sistema de control numérico para la elaboración de circuitos impresos, tecnología escasamente desarrollada en el país, para que los estudiantes puedan realizar sus proyectos con pocos recursos.

1.5. Objetivos

1.5.1. Objetivo general

Diseñar e implementar un prototipo de sistema de Control Numérico Computarizado para la elaboración de circuitos impresos en baquelita.

1.5.2. Objetivos específicos

- Recolectar información sobre los sistemas de control numérico, su funcionamiento y velocidad según la herramienta.
- Diseñar una tarjeta maestro – control CNC cuyas características cumplan los requerimientos entre el computador y el prototipo de la CNC.
- Elegir el sistema de movimiento lineal, motores que realizan el posicionamiento de la herramienta de trabajo y proporcionar las velocidades de cortes ideales.
- Elegir un software de fácil acceso que puedan importar archivos con extensiones estándar de los programas de diseño electrónico.
- Crear una interfaz gráfica para visualizar el tallado final, del diseño que se está elaborando.
- Diseñar un control manual que permita operar la máquina y enviar un código G desde una memoria extraíble.

2. MARCO TEÓRICO REFERENCIAL

2.1. Control numérico

2.1.1. Conceptos

El control numérico está fundamentado en la organización lógica de elementos que combinados a través de instrucciones pueden lograr un objetivo específico. Entre las varias definiciones se puede citar las siguientes:

- Es todo dispositivo que tiene la capacidad de realizar posicionamientos de una herramienta en un mecanismo, en el que las instrucciones de desplazamiento se ejecutan desde un programa que contiene secuencialmente las instrucciones. (Tecnología mecánica , 2017).
- Es un dispositivo que ejecuta un mando mediante líneas de código que contienen palabras compuestas por letras y números, logrando que las máquinas realicen su trabajo de manera automática.

NC son las siglas que refieren a la tecnología de Control Numérico mientras que las siglas CNC por significado en inglés como Computer Numerical Control. CNC es la abreviatura que más se utiliza hoy en día (Medina, 2011).

2.1.2. Componentes del Control Numérico

Para inicios de los años 90s, se inició la tecnología de control numérico abierto que permitía personalizar e incorporar conocimientos propios, presentación gráfica de las trayectorias, subprogramas y ciclos fijos, comunicación e integración con redes, que fueron posibles a través de la revolución de los microprocesadores en el mundo del control numérico dando como producto las máquinas CNC que son máquinas-

herramientas asistidas por computadoras (Mecanizados, 2015), y se compone de 6 partes vitales:

1. Terminal de entrada
2. Dispositivo de control
3. Máquina herramienta
4. Método de accionamiento
5. Módulos de realimentación (sólo en sistemas con servomotores)
6. Ordenador

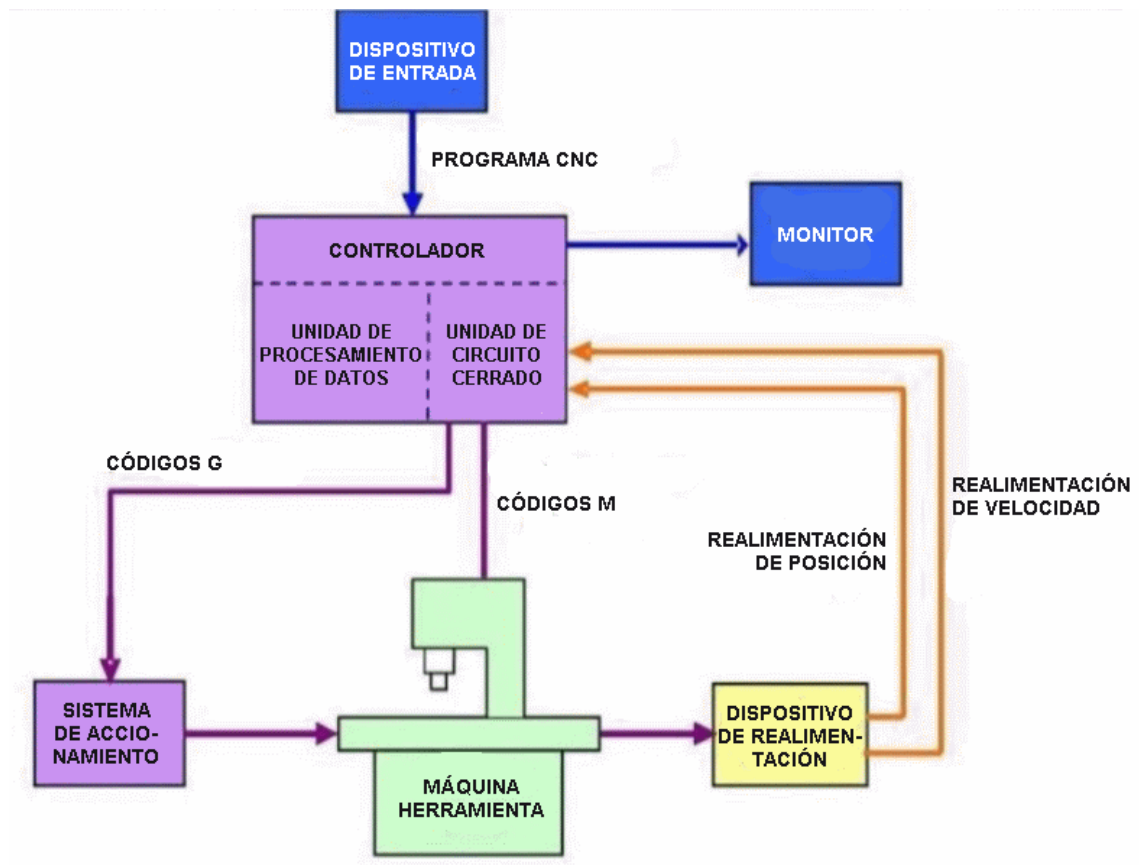


Figura 2 - Máquina CNC - Mecanizados (2015)

2.1.3. Principio de funcionamiento

Se puede determinar que para el funcionamiento de una máquina CNC existen algunas funciones necesarias como por ejemplo: programa CAM, programa CNC, el control del movimiento, accesorios que son reprogramables, controlador CNC, sistema CNC, los cuales se encuentran relacionados con el resultado que se pretende obtener.



Figura 3 - Funcionamiento Máquina CNC

2.1.4. Control de movimiento

Los ejes de una máquina NC disponen de un accionamiento regulable y de sistemas de medición de recorrido: la sección muestra los diferentes métodos de la medición del recorrido y demuestra el principio de regulación. Los ejes se nombran con letras. Los nombres más comunes para estos ejes son X, Y y Z, No obstante, los más comunes en los ejes giratorios son A, B y C (Heidenhain, 2012; Mecanizados, 2015).

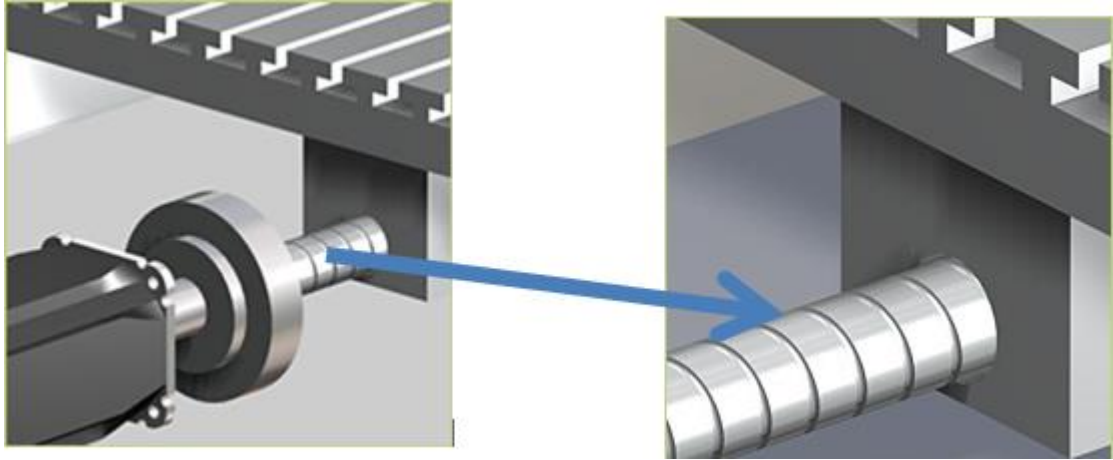


Figura 4 - Accionamiento regulable (husillo) - Heidenhain (2012)

En un eje NC un electromotor mueve los carros de la máquina mediante un husillo o una cremallera. El control traslada un programa de mecanizado a una secuencia corriente de datos de posición. Un circuito de regulación supervisa el movimiento del eje, para que las posiciones programadas puedan ser recorridas de forma precisa y con la velocidad adecuada, en la Figura 5 se muestra el espacio cartesiano y el volumen ocupado por un trabajo de mecanizado a realizarse. (Heidenhain, 2012).

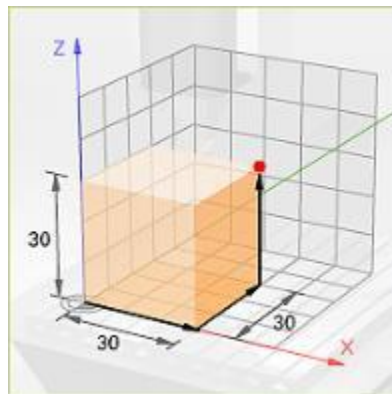


Figura 5 - Posiciones programadas. (Heidenhain, 2012).

El control de movimiento puede ejecutarse a través dos métodos, que funcionan particularmente o mezclados entre sí:

En la Figura 6 se muestra detalladamente los valores absolutos (código G90), donde los ejes del punto son alusivos al punto de inicio de la máquina. Se emplean las variables X y Z.

Coordenadas Absolutas

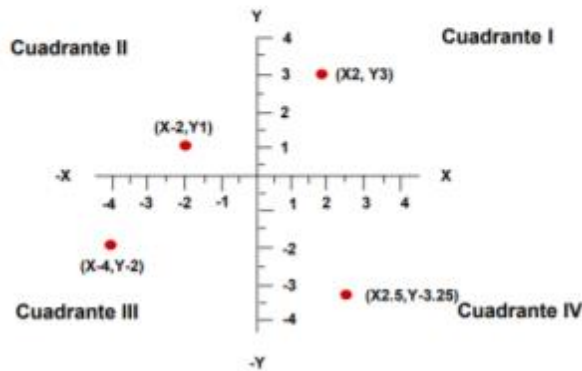


Figura 6 - Coordenadas absolutas. Universidad del País Vasco (2013)

En la Figura 7 se muestra claramente los valores incrementales (código G91), donde los ejes del punto de puesto son referentes al punto presente. Se utilizan las variables U (distancia radial) y W (medida en dirección paralela al eje de giro del husillo).

Coordenadas Incrementales

Utiliza a la posición actual como punto de referencia para el siguiente movimiento

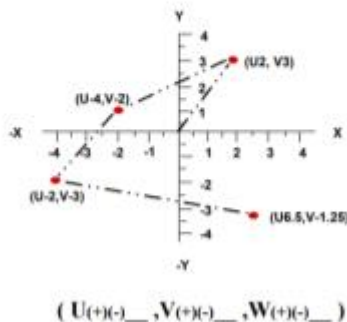


Figura 7 - Coordenadas incrementales. Universidad del País Vasco (2013)

En las máquinas de control numérico que utilizan el robot cartesiano como estructura, los ejes de movimientos no tienen dependencia lineal, lo que facilita su diseño. En la Figura 8 se observa el desplazamiento de cada eje, y el movimiento tridimensional que realiza la herramienta.

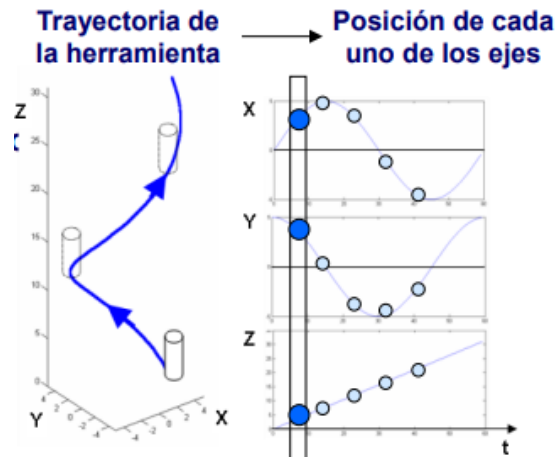


Figura 8 - Trayectoria y posición de ejes. Universidad del País Vasco (2013)

2.1.5. Accesorios programables

En la Figura 9 encontramos un gráfico que muestra un mapa de accesorios programables, los cuales se conectan a la unidad central procesadora, mediante un controlador de interfaz, en este caso un PLC.

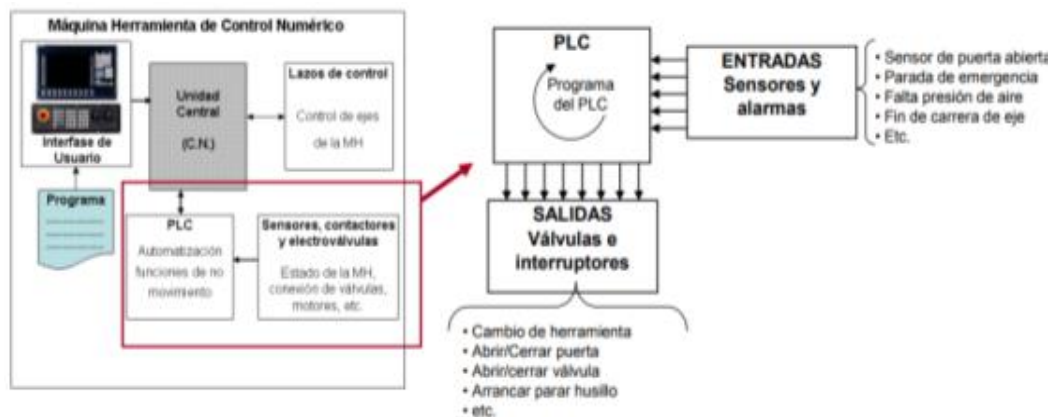


Figura 9 - Mapa Accesorios programables. Universidad del País Vasco (2013)

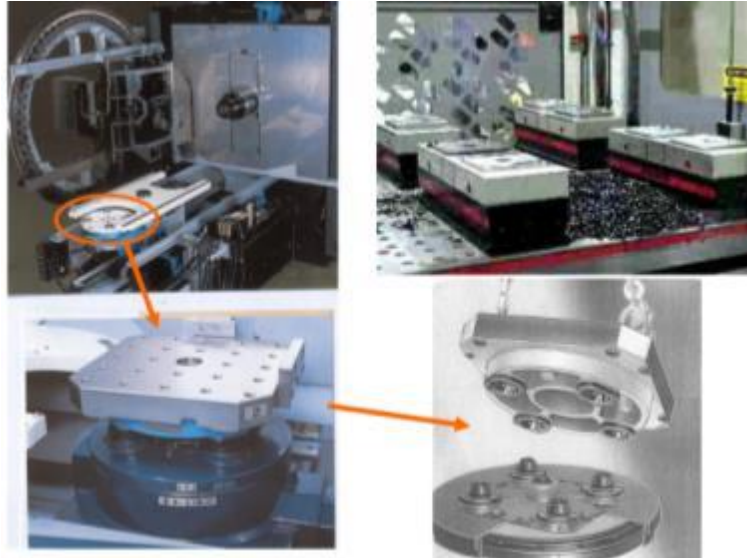


Figura 10 - Accesorios programables. Universidad del País Vasco (2013)

La Figura 10 muestra un ejemplo de accesorio programable, en este el cabezal de la herramienta se mueve para lograr un ajuste de esta.

2.1.6. Programa CNC

Los fabricantes utilizan la programación CNC (Computer Numerical Control Programming) para crear instrucciones de programa para que las computadoras controlen una máquina herramienta. CNC está muy involucrado en el proceso de fabricación y mejora la automatización y la flexibilidad, los grupos de los códigos G se muestran gráficamente en la Figura 11, en donde se identifica cada acción de código con un color.



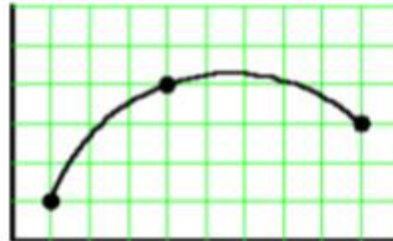
Figura 11 - Estructura bloque programa CNC. Universidad Nacional de la Plata (2014)

No todas tienen la misma estructura como una plegadora de chapas no tiene husillo ni requiere de refrigerante. En la Figura 12 se muestra un ejemplo de código G, una trayectoria sencilla, operación que corresponde a las dos líneas n10 y n20 mostradas en la misma.

**G09
TRAYECTORIA
CIRCULAR DEFINIDA
POR 3 PUNTOS**

Permite programar un arco definiendo el punto final y un punto intermedio. El punto inicial es el definido por el bloque anterior. No es modal. No permite círculos completos.

```
N10 G01 X10 Y10  
N20 G09 X90 Y30 I40 J40
```



El punto intermedio se programa usando I y J.

Figura 12 - Ejemplo programa CNC. Universidad Nacional de la Plata (2014)

2.1.7. Comandos CNC

Los comandos CNC usualmente se agrupan en modos, que se muestran a continuación:

2.1.8. Grupo de comandos de movimiento:

G1 = {G00, G01, G02, G03, G38.2, G80, G81, G82, G84, G85, G86, G87, G88, G89}

2.1.8.1. G00: Movimiento lineal rápido

Para conseguir un movimiento lineal simple, programe: G0 (X ó Y o Z o A según sea el eje en el cual quiere desplazarse), no es necesario que se desplace en todos los ejes, pero si en al menos uno. El G00 no es obligatorio si el modo de movimiento en el instante es G0. Esto causará un movimiento lineal coordinado al punto de destino con la velocidad de desplazamiento instantánea. Se espera que el corte no se efectúe cuando se está ejecutando un comando G00.

2.1.8.2. G01: movimiento lineal a velocidad de avance

Este comando se emplea para realizar un movimiento lineal a una velocidad de avance definida, programa: G01 X ~ Y ~ Z ~ A ~, donde el uso de todas las palabras del eje no es necesario, pero excepto que se debe usar al menos una.

2.1.8.3. G02: Arco a velocidad de avance definida

Se define un arco circular o helicoidal usando G02 (sentido horario) o G03 (sentido antihorario). El eje del círculo o hélice debe ser paralelo a alguno de los ejes de la máquina CNC. El eje (o, de forma equivalente, el plano perpendicular al eje) se selecciona con G17 (eje Z, plano XY), G18 (eje Y, plano XZ) o G19 (eje X, plano YZ). Si el arco es circular, se encuentra en un plano paralelo al plano seleccionado.

2.1.8.4. G04: Pausa

Este comando permite introducir una pausa, hará que los ejes de la máquina estén inmóviles durante el intervalo de tiempo en segundos designado por el número P.

2.1.8.5. G80: Cancelar movimientos

Este comando permite bloquear los movimientos de cualquiera de los ejes. Debe restablecerse la habilitación para poder mover los ejes nuevamente.

2.1.8.6. G81-G89

Los comandos G81-G89 son todos los ciclos fijos que se realizan con respecto al plano actualmente seleccionado. Se puede seleccionar cualquiera de los tres planos (XY, YZ y ZX). El comportamiento siempre es similar si se selecciona el plano YZ o XZ.

2.1.9. Grupo de comandos de selección de plano

G 2 = {G17, G18, G19}

2.1.9.1. G17, G18, G19: Comandos de selección de plano

Estos comandos nos permiten seleccionar el plano de trabajo, G17 para seleccionar el plano XY, G18 para el plano XZ o el comando G19 para seleccionar el plano YZ.

2.1.10. Comandos de referencia de distancias

G3 = {G90, G91}

Para entrar en el modo de distancia absoluta, use el comando: G90. En el modo de distancia absoluta, los números de los ejes (X, Y, Z, A) generalmente representan las posiciones en términos del sistema de coordenada. Cualquier excepción a esa regla se describe explícitamente en esta sección que describe los códigos G.

Para trabajar en el modo de distancia incremental, use el comando: G91. En el modo de distancia incremental, los números de los ejes (X, Y, Z, A) generalmente representan incrementos a partir de los valores actuales de los números.

2.1.11. Comandos de velocidad de trabajo

G5 = {G93, G94}

Se tienen tres modos de velocidad de avance: tiempo inverso, unidades por minuto y unidades por revolución del husillo. Programa: G93 para iniciar el modo de tiempo inverso (esto se emplea con poca frecuencia). Programa: G94 para iniciar el modo unidades por minuto.

2.1.12. Comandos de sistema de unidades:

G6 = {G20, G21}

Usando el comando G20 podemos usar el sistema de unidades en pulgadas y el comando G21 se emplea para establecer las unidades en milímetros.

2.1.13. Comandos de compensación de radio en corte

$G7 = \{G40, G41, G42\}$

Para desactivar la compensación del radio de corte, use el comando: G40. Está bien desactivar la compensación cuando ya está apagada.

La compensación del radio de corte solo se puede realizar si el plano XY está activo.

Para girar la compensación del radio de corte hacia la izquierda (es decir, la cuchilla queda a la izquierda de la ruta programada cuando el radio de la herramienta es positivo), programe: G41 D ~. Para girar la compensación del radio de corte a la derecha (es decir, la cuchilla queda a la derecha de la trayectoria programada cuando el radio de la herramienta es positivo), programe: G42 D ~. La palabra D es opcional; si no hay palabra D, se usará el radio de la herramienta actualmente en el eje. Si se usa, el número D normalmente debería ser el número de ranura de la herramienta en el eje, aunque esto no es necesario. Está bien que el número D sea cero; se usará un valor de radio de cero.

2.1.14. Comandos de desplazamiento de herramienta:

$G8 = \{G43, G49\}$

Este comando es usado para obtener un desplazamiento de longitud de herramienta, G43 H ~, en donde H es el desplazamiento deseado en la herramienta.

2.1.15. Sistema de selección de coordenadas:

G12 = {G54, G55, G56, G57, G58, G59}

Para seleccionar el sistema de coordenadas de desplazamiento de trabajo, use los comandos como se describen a continuación:

- G54 sistema de coordenadas 1
- G55 sistema de coordenadas 2
- G56 sistema de coordenadas 3
- G57 sistema de coordenadas 4
- G58 sistema de coordenadas 5
- G59 sistema de coordenadas 6
- G59.1 sistema de coordenadas 7
- G59.2 sistema de coordenadas 8
- G59.3 sistema de coordenadas 9

2.1.16. Control de trayectoria

G13 = {G61, G64}

G61 - Establece el modo de control de ruta exacta

G64 - Establece el modo de control de ruta mezclada

2.2. Controlador CNC

En la Figura 13 se muestra una plataforma de hardware libre, el PIC18F4550, usado comúnmente como controlador de máquinas CNC open source, ya que tiene

características que le permiten ejercer esta función; pines de entrada salida, fácil programación.

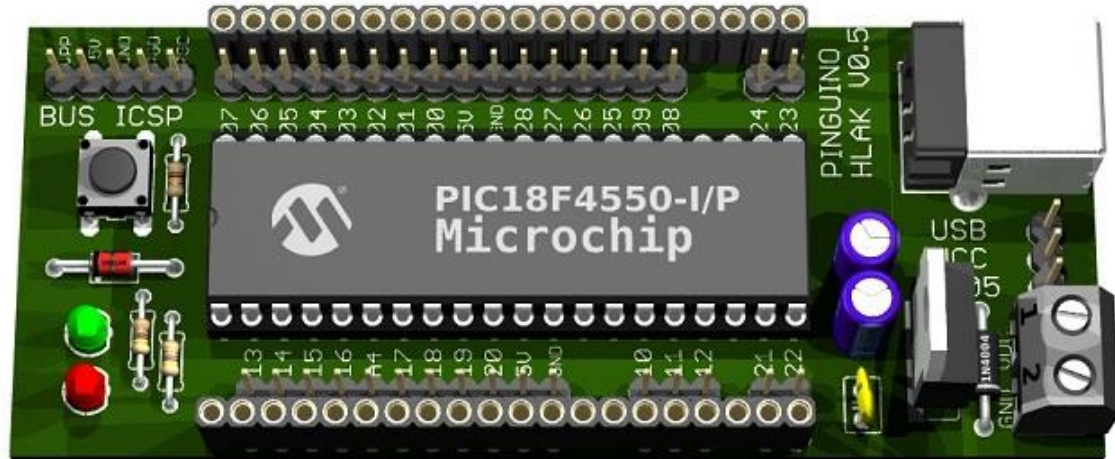


Figura 13 - Controlador CNC (Peters, 2013)

Principales características del Pic 18F4550

Microcontrolador con módulo USB 2.0. Soporta Low speed 1.5Mb/s y full speed 12Mb/s.

1kB de memoria de doble acceso vía USB.

35 pines I/O disponibles.

Memoria de programa flash de 32 kB.

RAM de 2048 Bytes.

EEPROM de datos de 256 Bytes.

Velocidad de la CPU 12 MIPS.

Oscilador externo de dos modos hasta 48 MHz.

Oscilador interno seleccionable entre 8 frecuencias desde 31kHz hasta 8MHz.

Oscilador secundario con Timer 1 de hasta 32kHz.

Opciones de oscilador dual permiten que la velocidad de la CPU y del módulo USB sean diferentes.

ADC de 10 bits y 13 canales.

Tecnología nanoWatt que brinda características y funciones de bajo consumo y ahorro de energía.

Voltaje de operación 4.2V a 5.5V.

4 Timer(desde Timer0 a Timer3). Uno de 8 bits y 3 de 16 bits.

2 módulos de captura/comparación/PWM.

EUSART, SPP, SPI, I²C.

20 fuentes de interrupciones (3 externas).

Resistencias de pull-ups en el puerto B programables.

Función del pin MCLR opcional.

Brown-out Reset de valor programable.

Power-on Reset.

Power-up Timer y Oscillator Start-up Timer.

Soporta 100,000 ciclos de borrado/escritura en memoria flash.

Soporta 1,000,000 ciclos de borrado/escritura en memoria EEPROM.

Retención de datos mayor a 40 años.

Protección de código y datos programable.

Encapsulado DIP de 40 pines.

Además de interpretar el programa CNC, el controlador tiene la responsabilidad de realizar varias tareas extra.

- Modificar los programas (de ser posible) si se detectan errores.
- Realizar funciones de verificación especial (como el funcionamiento en vacío). para confirmar que la exactitud del programa es suficientemente buena.
- Especificar ciertas entradas importantes del operador, por ejemplo, como los valores de longitud de las herramientas, entre otros.

2.3. Programa CAM

La fabricación asistida por computadora (CAM) consiste en el uso de software especializado para controlar máquinas, herramienta y otras relacionadas en la fabricación de piezas de trabajo.

Esta no es la única definición que existe de CAM, pero es la definición más común, además CAM también se puede referir al uso de una computadora para ayudar en todas y cada una de las operaciones de una planta de fabricación, esto incluye:

- Planificación
- Gestión
- Transporte
- Almacenamiento

Su principal objetivo es que el proceso de producción sea más ágil y rápido, además de que sus componentes y herramientas tengan dimensiones más pequeñas y precisas.

Esto minimiza la cantidad requerida de materia prima (minimizando así el desperdicio), al tiempo que reduce el consumo de energía.

CAM es ahora un sistema ampliamente utilizado tanto en escuelas y con fines educativos más bajos.

CAM es un proceso posterior asistido por computadora después del diseño asistido por computadora (CAD) y en ocasiones ingeniería asistida por computadora (CAE), ya que el modelo generado en CAD y verificado en CAE puede ingresarse en el software CAM que luego controla la máquina herramienta. CAM se utiliza en muchas escuelas junto con el diseño asistido por computadora (CAD) para crear objetos.

2.4. Sistema DNC

El control numérico directo (DNC), también conocido como control numérico distribuido (también DNC), es un término de fabricación común para máquinas herramienta CNC de red. En algunos controladores de máquina CNC, la memoria

disponible es demasiado pequeña para contener el programa, por lo que en este caso el programa se almacena en una computadora separada y se envía directamente a la máquina, un bloque a la vez. Si la computadora está conectada a varias máquinas, puede distribuir programas a diferentes máquinas según sea necesario. Por lo general, el fabricante del control proporciona el software DNC adecuado. Sin embargo, si esta disposición no es posible, algunas compañías de software proporcionan aplicaciones DNC que cumplen con el propósito. Siempre se requiere la comunicación DNC cuando los programas CAM se ejecutan en algún control de máquina CNC, vea la Figura 14.

Sin embargo, la tecnología ha avanzado de forma considerable para dotar a los controladores actuales con mejores capacidades de comunicación, de manera que puedan conectarse en red de maneras más convencionales y convenientes, por ejemplo, mediante Ethernet.



Figura 14 - RS-232C. Adrco (2017)



Figura 15 - Sistema DNC. Adrco (2017)

Podemos observar en la Figura 15 un sistema de máquinas conectadas a un computador común que actúa como DNC, el equipo que distribuye códigos de trabajo CNC a cada una de las máquinas conectadas.

2.5. Clasificación de los controles numéricos

Los controles numéricos, se han ido desarrollando y acoplado a las necesidades de la industria y del producto a desarrollar, por lo que hay diversos conectores, para diferentes máquinas en las cuales pueden trabajar los controles numéricos, por lo que su clasificación podría estar dada por:

2.5.1. Según el sistema de referencia

Mediante su sistema de referencia se logra determinar de forma clara las posiciones en el plano de mecanizado y en el área de trabajo de la máquina.

Los datos de una posición se refieren siempre a un punto fijo en el área de trabajo, el cual, se describe mediante coordenadas.

La máquina dispone de un sistema de referencia fijo que lo determina el fabricante.

Sin embargo, el usuario puede seleccionar su propio sistema de referencia para la pieza, el control conoce el origen y la situación de este sistema de referencia en relación con el sistema de referencia original de la máquina.

Por ello el sistema de control puede transmitir de forma correcta los datos de posición desde un programa NC a la pieza (Heidenhain, 2012).

2.5.2. Según el control de las trayectorias

Los CN pueden ser: paraxial, continuo o de contorneado, punto a punto.

2.5.2.1. Control numérico punto a punto

Este tipo de control es utilizado para determinar la posición de la herramienta en los sucesivos puntos donde se deben realizar una o varias mecanizaciones.

La trayectoria usada para pasar desde un punto a otro no es controlada, ya que las funciones de posición y mecanizado son diferentes (Rolón, 2011).

En la siguiente Figura 16, se muestra la ubicación de la herramienta para este tipo de control punto a punto.

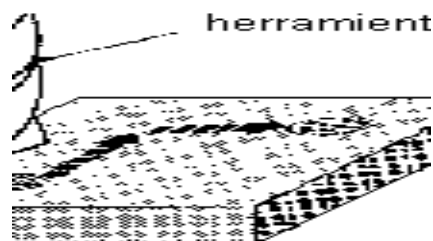


Figura 16 - CN punto a punto. Rolón (2011)

El CN punto a punto solamente controla el posicionado de la herramienta en los puntos en los cuales debe ser realizada una operación de mecanizado, esto se logra

realizando los desplazamientos en vacío según trayectorias paralelas a los ejes o a 45 grados sin ninguna coordinación entre los sistemas de mando de cada uno.

Se utiliza fundamentalmente en máquinas taladradoras, punzadoras, punteadoras y en algunas mandriladoras y no se requiere una coordinación entre ejes (Escalona, 2014).

2.5.2.2. Control numérico paraxial

El sistema CN, en el que la herramienta trabaja en línea recta en la dirección de un eje principal de coordenadas, como a lo largo de la dirección de alimentación durante el torneado, aburrado la operación de fresado a una velocidad controlada, se conocen como sistema de control de línea recta. (Escalona, 2014). En la siguiente Figura 17, se muestra gráficamente este control, que mejora punto a punto, ya que se puede controlar la trayectoria seguida.

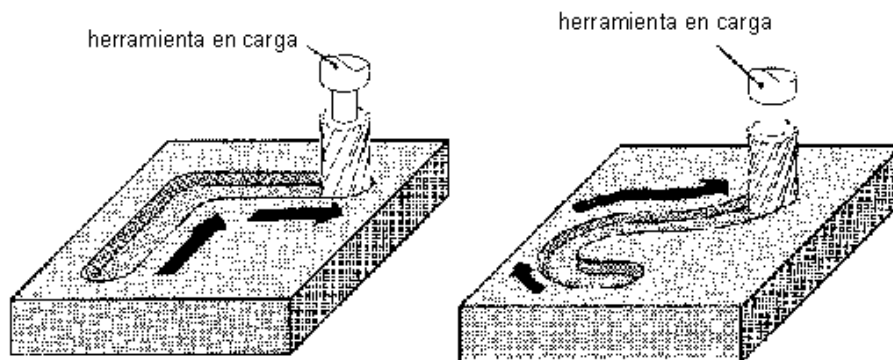


Figura 17 - CN paraxial. Rolón (2011)

2.5.2.3. Control numérico de contorno

El CN de contorno o continuo fue la primera opción en aparecer para después quedar relegado a un segundo plano frente a los sistemas punto a punto y paraxiales.

Posteriormente, con los grandes avances en la tecnología electrónica e informática, logró desplazar a los otros dos sistemas siendo el más utilizado en la mayor parte de las máquinas-herramienta.

Estos sistemas no solo controlan la posición final en la que queda la herramienta, sino que también controlan el movimiento en cada instante del eje mediante técnicas de interpolación lineal, circular y parabólica.

Además, esta tiene la capacidad de realizar un control continuo de la trayectoria de la herramienta durante el mecanizado, estas son definidas matemáticamente de forma cualquiera y obtenidas por aproximación (Escalona, 2014).

Este tipo de control de contorneado se aplica a:

- Tornos
- Fresadoras
- Centros de mecanizado

Y en general, a cualquier tipo de máquina que deba realizar mecanizados según una trayectoria más o menos compleja.

2.5.3. Según el tipo de accionamiento

Pueden ser: hidráulicos, eléctricos o neumáticos

2.5.3.1. Accionamiento hidráulico

Este tipo de accionamientos en las máquinas de control numérico permiten realizar más esfuerzos, mediante el uso de cilindros hidráulicos, pudiendo ser usados en otro

tipo de mecanismos diferentes al robot cartesiano, por ejemplo, mecanismo de brazo mecánico.

2.5.3.2. Accionamiento eléctrico

Son los accionamientos que comúnmente empleamos en las aplicaciones eléctricas, estos accionamientos se realizan por medio de motores eléctricos, tales como: motores de paso que son los más usados, motores AC, motores DC.

2.5.3.3. Accionamiento neumático

Al igual que los accionamientos hidráulicos emplean cilindros, pero por el tipo de fluido, su aplicación según esfuerzos está orientada a los trabajos que demandan menos potencia que los hidráulicos. Pero con la ventaja de que son más rápidos que los sistemas hidráulicos.

2.5.4. Según el bucle de control

El control del sistema se puede realizar de dos formas:

- En bucle cerrado
- En bucle abierto

2.5.4.1. En bucle cerrado

En el bucle cerrado mediante sensores se mide el valor a la salida, y se compara de forma continua con un valor de referencia proporcionando una adecuada señal de control.

2.5.4.2. En bucle abierto

En el bucle abierto no existe retroalimentación y la precisión de la trayectoria depende de la precisión de los actuadores.

2.5.5. Según la tecnología de control.

Se puede clasificar las CN por la forma física de realizar el control:

- Control Numérico (CN)
- Control Numérico Computarizado (CNC)
- Control Numérico Adaptativo (CNA)

2.5.5.1. Control Numérico (CN)

EL término Control Numérico (CN) es usado para todos aquellos controles donde cada una de las funciones que es realizada por el control la realiza usa un circuito electrónico que fue creado de forma específica para este fin, se realiza la interconexión entre ellos mediante lógica cableada.

Estos sistemas se caracterizan por trabajar sin memoria, utilizan una cinta perforada como medio de introducción del programa que se ejecuta de forma secuencial.

Los armarios de control son de gran volumen y difícil mantenimiento (Escalona, 2014; Peters, 2013). La Figura 18 muestra una computadora utilizada netamente para trabajar con códigos G, para enviar los comandos al actuador.



Figura 18 - Máquina Control Numérico (CN). Rolón (2011)

Control Numérico Computarizado (CNC)

Los controles basados en circuitos específicos y lógica cableada (CN) se han dejado de usar en la industria por la aparición de los Controles Numéricos Computarizados (CNC), estos usan uno o varios microprocesadores los cuales sustituyen a los circuitos de lógica cableada de los sistemas CN, que son poco fiables y de gran tamaño.

Estos sistemas de control incluyen una memoria interna de semiconductores la cual permite el almacenamiento del programa, además de las compensaciones de las herramientas y de los datos de la máquina, además incorpora además un teclado el cual facilita la comunicación y el grado de interactividad con el operario además permiten la ruptura de la secuencia de los programas, la incorporación de subrutinas, los saltos condicionales y la programación paramétrica, facilitando la programación estructurada y sencilla de aprender.

El tamaño compacto de los equipos con circuitos integrados aumenta el grado de fiabilidad del control ya que se usa un solo dispositivo especializado para realizar tareas programadas por él usuario y permite su instalación en espacios reducidos y con un nivel de ruido elevado (Escalona, 2014; Jiménez, 2013). La Figura 19 expone un ejemplo de máquina CNC.



Figura 19 - Máquina CNC. Rolón (2011)

2.5.5.2. Control Numérico Adaptativo (CNA)

El sistema de control adaptativo es una extensión lógica del mecanismo CNC. En CNC mecanismo la velocidad de corte y las velocidades de avance son prescritas por el programador de piezas. La determinación de estos parámetros de operación depende del conocimiento y la experiencia con respecto a la pieza de trabajo, los materiales de la herramienta, las condiciones del refrigerante y otros factores. Por contraste en mecanizado de control adaptativo, hay una mejora en la tasa de producción y la reducción en el costo de mecanizado como resultado del cálculo y configuración de los parámetros óptimos durante el mecanizado. Este cálculo se basa en mediciones de variables de proceso en tiempo real y seguido de unos ajustes posteriores en línea de los parámetros sometidos a restricciones de mecanizado para optimizar el rendimiento del sistema en general. (Escalona, 2014).

2.6. Ventajas y desventajas del CN

2.6.1. Ventajas

- La reducción de los tiempos muertos gracias a la rapidez de los movimientos que son realizados en el vacío, además de las velocidades del cabezal.
- Los tiempos de reglaje se reducen considerablemente ya que se disminuyen el número de reglajes de la máquina.
- Los tiempos de control y medida son mucho menores gracias a la elevada precisión de cada uno de los mecanizados.
- Los tiempos que se toman en cambios de pieza también se ven reducidos considerablemente.
- Los tiempos de espera que existe entre máquinas también se ve reducido, debido a que son máquinas polifuncionales.
- Se reduce el número de verificaciones que son realizadas entre operaciones.
- Se reduce del tamaño del lote económico y nivel de almacenes.
- Aumento de la flexibilidad y fácil adaptabilidad a distintos tipos de fabricados.
- Disminución de rechazos de piezas.
- La duración de las herramientas aumenta, debido a su mejor aprovechamiento.
- Ahorro de las herramientas y el utillaje ya que se utilizan herramientas universales.
- Manera económica de piezas con geometría complicada.
- Mejora de la seguridad ya que se reduce el grado de interactividad entre la máquina y el operario durante los procesos de mecanizado.
- Se necesita una menor cantidad de operarios y estos a su vez necesitan una menor cualificación, para manejo de las máquinas.

- Ofrece además una atracción y motivación por parte del personal hacia las nuevas tecnologías.

2.6.2. Desventajas

- Existe un coste de elevada inversión que es necesaria para la adquisición de una MHCN, además de los elementos auxiliares los cuales son necesarios para su funcionamiento. Con el fin de asegurar un alto nivel de ocupación de la máquina y la puesta a varios turnos del equipo para conseguir una muy buena amortización.
- Se necesita de un personal mejor cualificado en programación y mantenimiento, lo que eleva los costes en formación del personal y los salarios.
- Con máquinas más grandes y complejas se necesita un mayor coste de mantenimiento. Se estima que el coste de mantenimiento de una MHCN es un 50% más elevado que en las máquinas convencionales.
- Se necesita de un tiempo de adaptación además de un cambio en la estructura organizativa de la institución además de la gestión de la fabricación. No es fácil que los empleados se adapten a las nuevas técnicas que son exigidas por el CN.
- Se requiere de un alto coste de inversión, adquisición, mantenimiento y reposición de herramientas.

2.7. Circuitos impresos

2.7.1. Concepto

Un circuito impreso o PCB, es una tarjeta o placa que se utiliza para realizar las conexiones de los distintos elementos que conforman el circuito.

Antiguamente los circuitos impresos se los realizaba con técnicas caseras para el diseño de sistemas, pero a medida que ha pasado el tiempo esta práctica se ha reducido, así como el tamaño de los componentes electrónicos, teniendo implicación debido a la menor separación que existe entre los pines para circuitos integrados de alta densidad, las actuales frecuencias de operación de los dispositivos que son mucho más elevadas que las que se usaban en antaño, ahora se requiere una excelente precisión en el proceso de impresión de la placa con la finalidad de garantizar tolerancias mínimas. En la Figura 20 se muestra un ejemplo de circuito impreso con mascarilla antisolder. (Electrosoft Ingeniería, 2010).

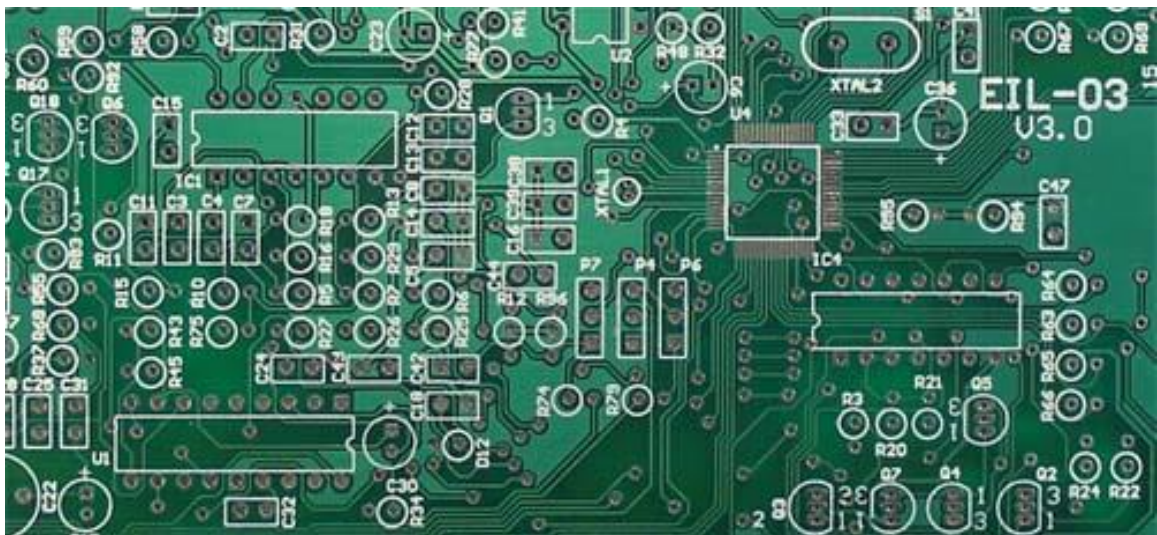


Figura 20 - Circuito. (Electrosoft Ingeniería, 2010)

Los circuitos impresos (PCB) más sencillos son aquellos que contienen caminos de cobre solamente en una de las superficies de la placa, a estas placas se les conoce como circuitos impresos de una capa.

Los PCB más comunes de la actualidad son los de 2 capas, es decir una placa con circuito impreso de lado y lado.

Sin embargo, dependiendo de la complejidad que se requiera en el diseño del circuito, pueden llegar a fabricarse hasta 8 o más capas.

2.7.2. Tipos de circuitos

En la actualidad, se pueden encontrar diferentes y diversos tipos de circuitos impresos dependiendo de su estructura y material, como la división según su placa, son:

2.7.2.1. Sided Plated Holes

Este circuito es de bajo coste, pero su diseño es mucho más complejo y complicado, aunque permite hacer pasos de cara a cada ya que dispone de unos taladros metalizados, los permiten tener continuidad entre la capa superior e inferior, se muestra un ejemplo de este tipo de circuitos impresos en la Figura 21. (Arqhys, 2012).

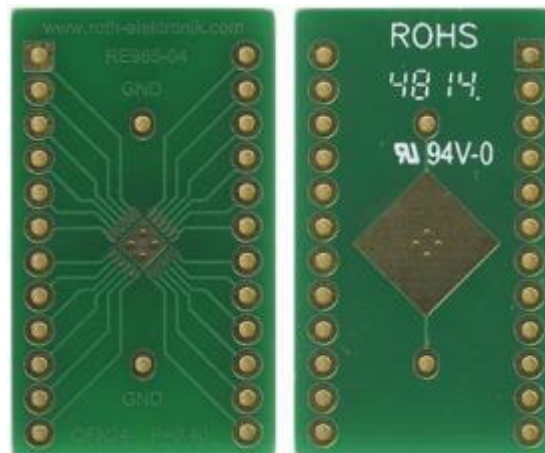


Figura 21 - Circuito orificios doble cara. RS (2010)

2.7.2.2. Multicapa

Este tipo de PCB es el más utilizado a nivel comercial debido a su gran flexibilidad.

Generalmente disponen de 8 a 10 capas y algunas de estas capas suelen encontrarse dentro del sustrato (Arqhys, 2012).

La Figura 22 muestra un lote de circuitos impresos de fabricación industrial, circuitos impresos de varias capas necesitan procesos más complejos para su correcta elaboración.



Figura 22 - Circuito impreso multicapa. (Cipsa Circuits ©, 2016)

2.7.2.3. 2-Sided Non-Plated Holes

Este tipo de circuito posee taladros, pero sin metalizar.

Para lograr una continuidad en este tipo de PCB la persona debe de soldar el circuito por los dos lados, usualmente se le denomina circuito perforado y se lo muestra en la Figura 23. (Arqhys, 2012).

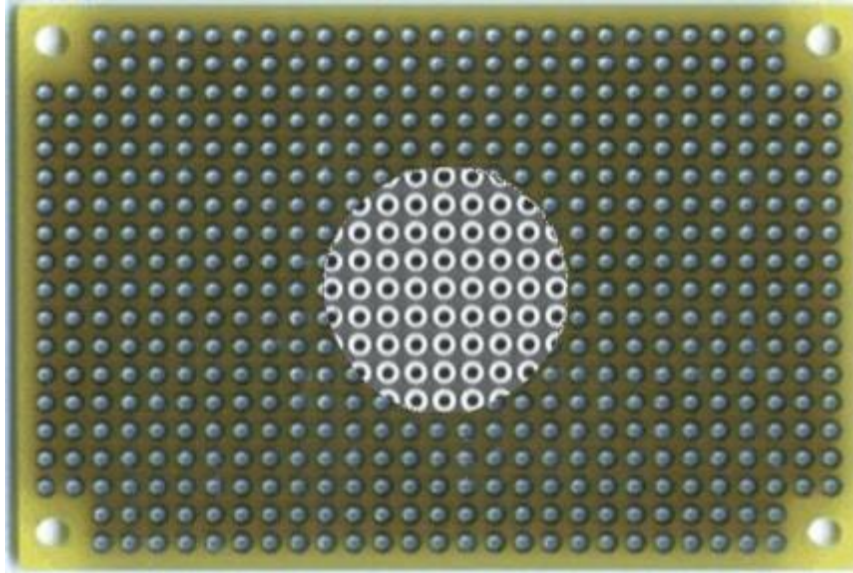


Figura 23 - Circuito impreso un lado. (Mouser electronics, 2017)

2.7.2.4. Single-Sided Non-Plated Holes

Habitualmente este tipo de PCB es usado en diseños más sencillos y de bajo coste.

Es de tipo de PBC y posee agujeros los cuales no están metalizados.

Estos están formados por capas conductoras, los que se encuentran separadas una de las otras además que estas están sostenidas por el sustrato (capa de material aislante).

Cada una de estas capas pueden ser conectadas por medio de sus vías o bien por medio de orificios los cuales suelen ser electro recubiertos, o por pequeños remaches (Arqhys, 2012).

2.7.2.5. Placas de circuito impreso (PCB's)

Una placa de circuito impreso (PCBS) es una plancha de material rígido aislante, que posee una cubierta por de cobre en una de sus caras o en ambas caras de ser

necesario, estas pistas sirven como conductor o de interconexión eléctrica entre los componentes que serán montados sobre la placa.

Esta es la forma más perfeccionada y que ofrece uno de los mejores acabados de todos, pero por el contrario, este proceso exige un proceso más laborioso.

Existen placas a simple cara y a doble cara, dependiendo del material de que está hecha la placa, podemos distinguir tres tipos fundamentales:

2.7.2.6. Baquelita

La baquelita es el material aislante, sobre el cual se va a diseñar el circuito impreso. Este no conduce la electricidad por sí mismo, sobre ella se debe dibujar la capa de material electro conductor.

Se encuentran recubiertos por una capa de conductor, el cual es comúnmente cobre, este debe ser removido de forma que solo queden dibujadas las pistas de nuestro circuito.

2.7.2.7. Fibra de vidrio

Es el material más usado gracias a su calidad y economía, si bien son más caras que las placas tradicionales, estas ofrecen un acabado mucho mejor, lo que las convierte en la alternativa más comúnmente usada.

2.7.2.8. Placa normal

La placa normal es aquella placa sobre la que se dibuja directamente la pista, podemos dibujar con rotulador indeleble, o bien mediante pegatinas adecuadas y luego mediante un ácido remover el material conductor innecesario.

2.7.2.9. Placa fotosensible

Las placas fotosensibles disponen de un barniz el cual es sensible a la luz, son sensibles a los rayos UVA, el principal problema de estas placas es su imperfección.

2.7.2.10. Teflón

Estas placas son mucho más caras que las placas tradicionales, pero tiene una resistencia mecánica muy superior, permite crear placas de mejor calidad y mayor resistencia ya que no absorben el agua, evita la corrosión sobre la pista.

2.7.2.11. Técnica de trazado de circuitos impresos

Para realizar el diseño de un circuito impreso debemos:

- Realizar el diseño mediante algún software computacional.
- Una vez se tenga el diseño del circuito se procede a realizar un corte en la placa para que tenga el tamaño adecuado.
- Luego se realiza la impresión del circuito impreso sobre la placa previamente cortada.
- Se sumerge la placa previamente impresa en una solución que elimine el material conductor sobrante.
- Se realiza el taladrado de los orificios necesarios de la placa, para que los elementos puedan ser soldados.
- Se realizan pruebas de funcionamiento de la placa para asegurarse que las conexiones sean las adecuadas.

2.7.3. Elaboración de PCB mediante ataque Químico

2.7.3.1. Manual o Artesanal

En circuitos de baja complejidad, se sumerge la placa en una solución la cual va a eliminar el material conductor excesivo.

2.7.3.2. Marcador con Tinta Permanente

Se usa un marcador permanente para realizar el diseño del circuito, este método es el más barato y sencillo para circuitos de baja complejidad ya que el diseño de la posta debe ser realizado a mano.

2.7.3.3. Cintas plásticas adheribles y/o Transferibles de Pistas

Se imprime la pista en formato espejo (invertido) sobre papel fotográfico, luego con calor se realiza la transferencia del diseño a la placa.

2.7.3.4. Diseño Asistido por Computadora (CAD)

Las CNC permiten usar una baquelita virgen o fibra de vidrio para diseñar un PCB, esta CNC mediante su punta remueve el material conductor sobrante y solo deja el necesario para las pistas del circuito, adicionalmente la CNC realiza los orificios necesarios para que los componentes puedan ser soldados posteriormente.

Este método permite crear circuitos impresos 2 capas de forma sencilla ya que simplemente se da la vuelta a la placa para que la CNC vuelva a realizar el diseño del circuito del otro lado.

La Figura 24 muestra un circuito impreso por CNC a partir del diseño en el CAD. (Google Sites, 2017).



Figura 24 - Circuito impreso CAD. (Google Sites, 2017)

2.7.3.5. Serigrafía

La técnica de la serigrafía consiste en proteger nuestro circuito impreso mediante una capa de pintura acrílica sobre todas las pistas, esto con el fin de evitar el óxido de nuestro circuito con el pasar del tiempo

Requiere ciertos materiales de serigrafía, como lo son:

- Pantalla.
- Foto emulsión.
- Insoladora (Fuente de luz visible).
- Paleta plástica.
- Removedor de pantalla.
- Limpiador Universal.
- Pinturas acrílicas para serigrafía.
- Estopa.

2.7.3.6. Papel para transferencia térmica

Este método consiste en imprimir el circuito impreso sobre algún papel el cual permita realizar la transferencia sobre nuestra placa, usualmente se usa papel satinado.

Mediante la aplicación de calor se transfiere el circuito de nuestro papel a la baquelita (este circuito debió ser impreso en formato espejo), una vez el circuito fue transferido de forma completa se utiliza un ácido que elimine el material conductor que no está cubierto por la tinta trasferida mediante calor.

Este método reduce de forma significativamente el tiempo necesario para realizar los PCBS, este método permite hasta realizar circuitos impresos de dos caras.

2.7.3.7. Tarjeta Sensibilizada

Esta técnica consiste en utilizar una placa la cual haya sido sensibilizada de fábrica mediante soluciones especiales (KPR aerosol o solución foto resistiva positiva).

Mediante la exposición a luz ultravioleta se remueve el material conductos sobrante al circuito impreso.

3. MARCO METODOLÓGICO

3.1. Descripción de la propuesta

El enfoque de la solución ante esta problemática consiste principalmente en el diseño e implementación de un prototipo de sistema de Control Numérico Computarizado para la elaboración de circuitos impresos, el cual servirá de gran ayuda a los estudiantes de ciclos básicos de la Carrera de Ingeniería Electrónica para la realización de sus proyectos en PCB hay que tener en cuenta que el prototipo de sistema de control numérico solo trabaja con una cara.

Con la implementación de este proyecto se planea tecnificar el proceso de elaboración de PCB mediante el empleo de una herramienta de control numérico y de esta forma reducir considerablemente el tiempo que se toma elaborar estos circuitos impresos de forma artesanal, además de evitar daños producidos por los efectos corrosivos que tiene el cloruro férrico.

Se diseñó un sistema embebido de control para realizar la programación en base al funcionamiento del prototipo que elabora estos PCB para ejecutar de una manera factible y más segura la realización de estos proyectos.

El prototipo CNC puede usarse con un programa para diseñar circuitos electrónicos como el EAGLE PCB y otros programas parecidos que cumple con la misma función, los archivos son formato GERBER y se necesita un programa para transformar de formato GERBER a NGC, el formato NGC es necesario para el reconocimiento de los datos al usar el programa Gcode sender que es el encargado de enviar la información al sistema de embebidos de control. La Figura 25 muestra el procedimiento de elaboración de un circuito impreso, donde a partir del diseño se obtienen los archivos Gerber que

son vectoriales para a partir de ellos generar el código G. Luego la máquina recibe los códigos Generados y a través de sus actuadores realiza los movimientos correspondientes para mecanizar tal como se requiere.

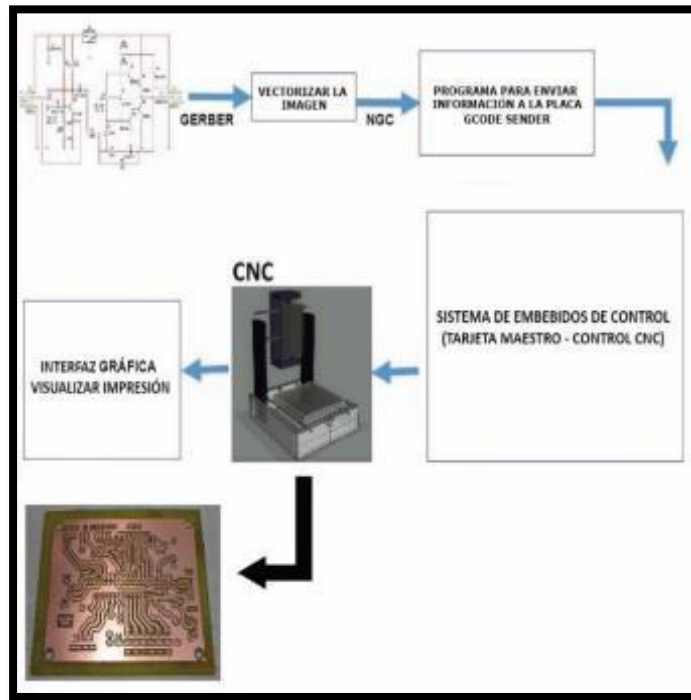


Figura 25 - Componentes del sistema

El sistema embebido de control posee dos áreas de funcionamiento, (control maestro y control CNC), el control maestro recibe datos del Gcode sender donde está la información vectorizada de la imagen, luego transfiere los datos mediante un firmware que es el software que permite entender las órdenes que envía el programa al área de control CNC la cual es la encargada de variar el voltaje y la corriente que permite mover los motores PAP de la máquina CNC.

3.1.1. Firmware

El programa está relacionado con cuatro versiones diferentes del firmware, a continuación, están sus características.

3.1.1.1. Txapuzas Firmware

El firmware Txapuzas es el más básico, y además se encuentra disponible para su uso.

3.1.1.2. CNC Controller V1.0.9

Basado en el firmware Txapuzas, esta versión tiene mejor funcionamiento y aceleración.

3.1.1.3. CNC Controller V1.0.12

En esta nueva versión presenta nuevas funcionalidades cómo, por ejemplo

- Un botón de paro de emergencia.
- Una sonda de contacto.

3.1.1.4. GRBL

Es el firmware más actual y recomendado para manejar la máquina y en el cual se fundamenta el controlador que se desarrolla en este proyecto.

Una vez obtenida la comunicación con la máquina CNC, empieza a realizar el tallado de los PCB y de manera paralela podemos encontrar el mismo tallado mediante una interfaz gráfica, con los datos obtenidos por el encoder que presentan cada uno de los motores PAP que se mueven de forma milimétrica.

El sistema de embebidos de control y la interfaz gráfica será diseñado por los autores con habilidades y conocimientos adquiridos a lo largo de toda la Carrera de Ingeniería Electrónica.

3.1.2. Modelado 3D

Para poder visualizar como quedará el prototipo CNC, se diseñó en AutoCAD el boceto dándole forma y la estructura adecuada teniendo en cuenta que el prototipo solo usará una cara para luego poder realizarlo ya con los componentes que son parte del prototipo CNC. La Figura 26 muestra la vista previa de la máquina CNC. De cama móvil en un solo eje y herramienta móvil en los ejes restantes.

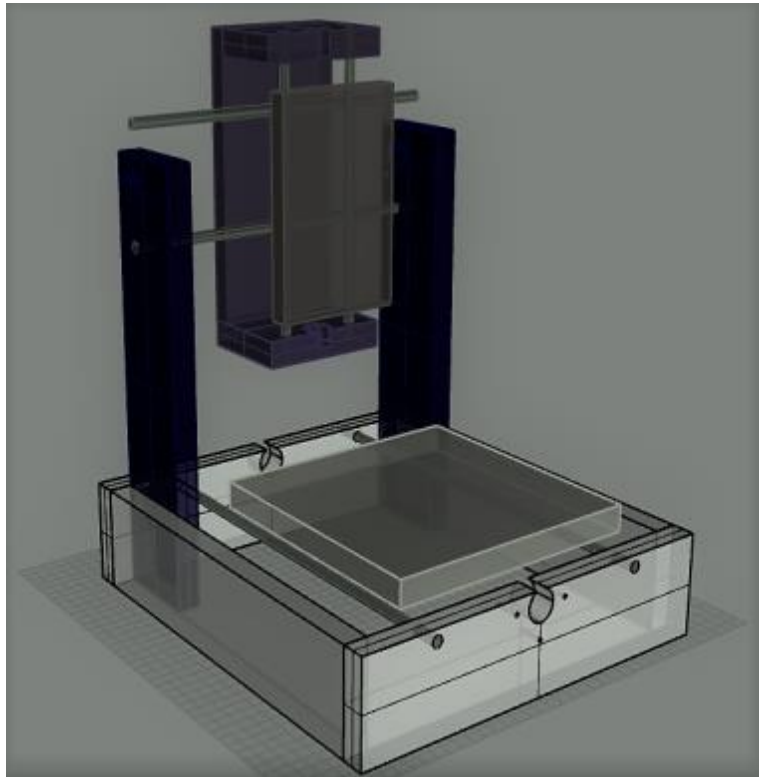


Figura 26 - Modelado 3D realizado en AUTOCAD. (Radoff, 2008)

La selección de los dispositivos semiconductores de potencia para los motores se va a realizar teniendo en cuenta las especificaciones nominales de los motores paso a paso.

- Corriente nominal de 1.4 Amperios
- Voltaje de alimentación de 3.5 Voltios.

Se usará un circuito integrado A4998 que permite controlar motores con estas características. El A4998

- Corriente de colector de 5 Amperios
- Voltaje colector emisor de 100 Voltios

El transistor opera en las regiones de corte y saturación, las cuales poseen las siguientes especificaciones en cuanto a corriente y voltaje.

3.1.2.1. Región de saturación

- $I_c = 1.4$ Amperios
- $V_{ce} = 0.2$ Voltios.

3.1.2.2. Región de corte

- $I_c = 0$ Amperios
- $V_{ce} = 3.5$ Voltios.

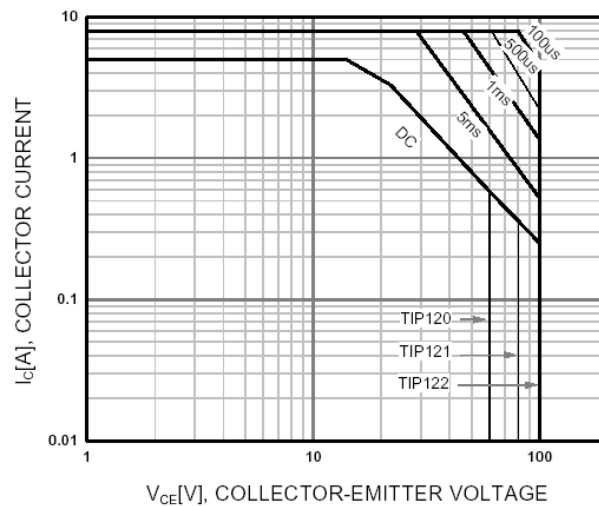


Figura 27 - Curva de corrientes

Para la selección del motor se usó la recomendación en textos guías donde ya han sido implementados en proyectos similares por lo cual se escogió el motor paso a paso S4H56B04L.

Con parámetros nominales de:

- UNIPOLAR (4 FASES, 6 CABLES)
- DIÁMETRO: 57mm
- LARGO MOTOR: 50mm
- LARGO TOTAL: 71mm
- DIÁMETRO EJE: 1/4"
- MARCA: SHIBAURA
- MODELO: S4H56B04L
- TORQUE: 35 OZ-IN HOLD
- PASOS: 200 POR VUELTA 1,8 GRADOS POR PASO
- ALIMENTACIÓN: 3.5 VDC
- CONSUMO: 1,4A POR FASE

Para el diseño de la tarjeta se utilizó:

- PIC 18F4550
- Driver ftt232
- Display LCD 20x4
- Módulo SD Card
- Arduino nano (para la interfaz manual)
- 4 pulsadores
- Baquelita sin perforar 20cmx10cm

- 16 resistencias 10k
- 3 borneras 3 terminales
- Im7805
- 4 capacitores electrolíticos 10uF
- Cristal 16Mhz
- 2 capacitores 22uF
- Conectores en espadín
- Cable No. 26

Para el diseño de la carcasa se empleó:

- Plancha de acrílico transparente de 3 mm
- Tornillos y tuercas 1/16"

3.2. Metodología del diseño

Para llevar a cabo el proyecto se emplea el método inductivo. Así partiendo desde las aplicaciones muy particulares completaremos todo el sistema en general todo el procedimiento a continuación descrito muestra la metodología mencionada. A continuación se describen las acciones a desarrollar, tanto el proceso como las pruebas de laboratorio.

3.2.1. Procedimiento de diseño

En la siguiente Figura 28 se muestra el diagrama de flujo que representa el proceso de diseño a seguir para este trabajo.

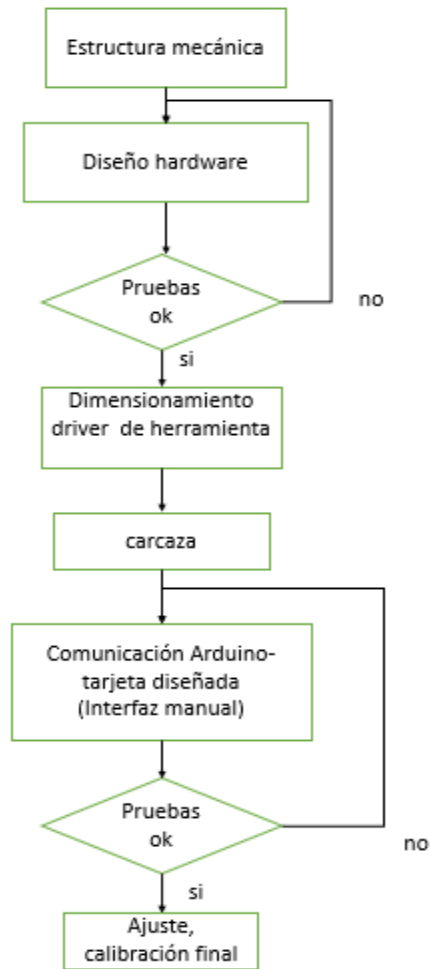


Figura 28 - Procedimiento de diseño

3.2.2. Estructura mecánica

Para la estructura mecánica se emplea un kit comercial CNC el cual elimina la necesidad de hacer un análisis mecánico y de esfuerzos para así centrar la atención al diseño de la electrónica. La estructura de un metal ligero para reducir la necesidad de motores de gran torque.

3.2.3. Códigos G a actuadores

Los códigos G deben ser traducidos a instrucciones para los actuadores, para ello se debe conocer las acciones que representan cada código. El código G01 que es un

movimiento lineal deberá traducirse en un movimiento del motor de pasos en ese eje las unidades de distancia que correspondan.

3.2.4. Driver herramienta

La herramienta de fresado, un motor DC, necesita un controlador que soporte los parámetros eléctricos de corriente y potencia consumidos por el motor. Para ello una vez establecido el motor se dimensiona el controlador, definiendo así si se emplea un módulo que tenga esas características o se implementa una tarjeta como controlador de este motor de herramienta.

3.2.5. Carcasa

La carcasa de acrílico una vez ya ensamblada la estructura y teniendo los movimientos definidos, se procede a diseñar, dejando espacio para el desplazamiento de las piezas móviles de la estructura en cada eje.

3.2.6. Comunicación interfaz manual

Para la interfaz manual es conveniente tener lista la tarjeta controladora diseñada. Así lo que se pretende en esta parte es implementar un mensajero de código G; que realice las tareas cargadas al computador, como son el movimiento y configuración de punto de inicio y el envío de las instrucciones de código G.

3.2.7. Calibración y ajuste

Para un correcto fresado de los trabajos como es recomendado en varios proyectos analizados en el estado del arte, la calibración de los ejes es muy importante para que el desnivel sea mínimo y la operación de la máquina no se vea afectada. Se muestra más adelante una manera de calibrar y ajuste final de la máquina. Se detalla más sobre esta parte en el capítulo 4.

3.3. Pruebas

Las pruebas a la primera parte del proyecto, con la tarjeta controladora implementada, y las pruebas a la segunda parte ya con la interfaz manual.

Pruebas para realizar:

- Trayectorias
- Fresado de circuito impreso

3.4. Construcción de la máquina CNC

3.4.1. Estructura mecánica

La estructura de la máquina CNC está elaborada en aluminio y algunas piezas prototipadas en impresión 3D.

3.4.1.1. Piezas principales

Tabla 1 – Materiales para Máquina CNC

<ul style="list-style-type: none">• Perfil de aluminio	
<ul style="list-style-type: none">• Tornillo sin fin, para dar movimiento en los ejes.	

<ul style="list-style-type: none">• Rieles Plataforma de trabajo	
<ul style="list-style-type: none">• Junturas para los perfiles y armado de estructura	
<ul style="list-style-type: none">• Soportes para rieles	

3.4.1.2. Motores de paso

Los motores empleados para la movilidad en los ejes de la máquina CNC, son los motores nema 17. Un motor de 2 fases y 200 pasos por vuelta. La Figura 29 se muestra un motor de pasos nema 17 ensamblado con la junta para colocar el tornillo sin fin a emplear para el desplazamiento.



Figura 29 - Motor de paso

3.4.1.3. Motor de la herramienta

El motor de la herramienta es un motor DC cuyo voltaje de operación se encuentra entre (12-36V), la siguiente tabla muestra los valores de las revoluciones según el voltaje en el cual opere.

Tabla 2 – Valores de revolución según voltaje

Voltaje	Corriente	Velocidad (RPM)
12	0.14	3500
18	0.15	4500
24	0.16	7000
30	0.17	8100

36	0.20	9000
----	------	------

La Figura 30 muestra el tipo de motor para la herramienta de fresado, un motor DC con las características descritas en la Tabla 2.



Figura 30 - Motor de herramienta

3.4.1.4. Armado de la estructura

En la Figura 31 se muestra la plataforma de trabajo de la máquina CNC junto con los rieles de desplazamiento y el tornillo sin fin que transfiere el movimiento. Se muestra una vista inferior de la plataforma de trabajo. Con dimensiones de 20cmx15cm.



Figura 31 - Armado de la estructura

3.4.1.5. Láser

El equipo láser que se incluye en el kit, se muestra en la Figura 31, tiene una potencia de 5.5 W y su longitud de onda se encuentra comprendido entre 445 y 450nm.

Voltaje de operación: 12V



Figura 32 - Voltímetro

3.4.1.6. Fresas para mecanizado



Figura 33 - Fresas para mecanizado

Las fresas para realizar el trabajo se muestran en la Figura 33, son fresas de 1/8" con terminación Vee a 60 grados.

3.4.1.7. Fuente de alimentación

La Fuente de alimentación corresponde a una potencia de 500w. Con salida de voltaje de 24 V. Esta fuente DC se muestra en la Figura 34.



Figura 34 - Fuente de alimentación

3.4.1.8. Tarjeta controladora

La tarjeta controladora, una Grbl 0.9 que se rediseñara.



Figura 35 - Tarjeta controladora Grbl 0.9

4. ANÁLISIS DE RESULTADOS

4.1. Construcción

Paso 1. Se estableció un cuadrante de aluminio de (30cmx45cm).



Figura 36 - Ensamble de máquina CNC (Paso 1)

Tal como se muestra en la Figura 36 se ensambla la base de la máquina, un cuadrante de las dimensiones establecidas anteriormente, 30cmx45cm.

Paso 2. Se incorpora la estructura armada previamente en la base de aluminio.



Figura 37 - Ensamble de máquina CNC (Paso 2)

En la Figura 37 se muestra la plataforma de trabajo de la máquina colocándose en la base de la máquina para fijar así su punto de referencia para el desplazamiento. Corresponde al eje Y.

Paso 3: Ensamble completo Máquina CNC

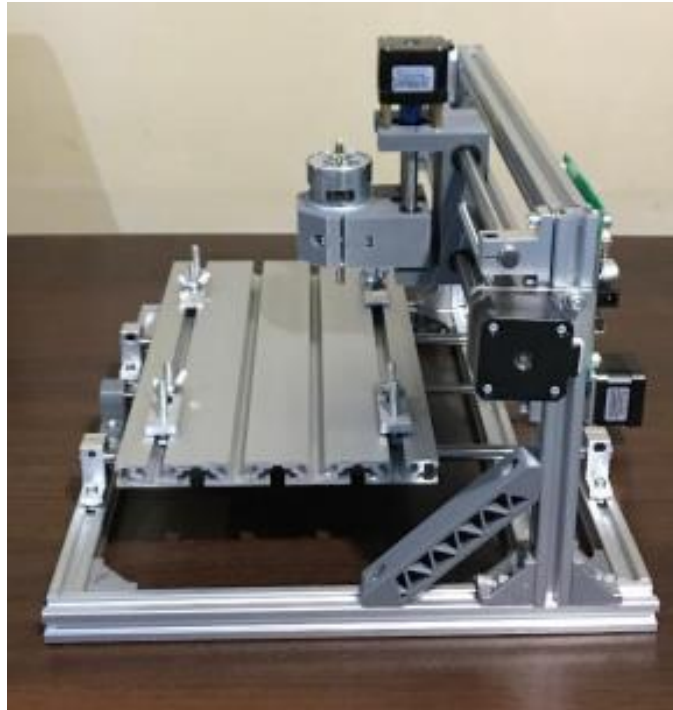


Figura 38 - Ensamble de máquina CNC (Paso 3)

La máquina ya ensamblada totalmente se muestra en la Figura 38, con todos los rieles y tornillos sin fin para cada eje.

4.2. Procedimiento de trabajo de la máquina

El procedimiento para el trabajo de la máquina CNC es el que se muestra a continuación en el siguiente diagrama de la Figura 39. La generación de código G o Gcode, corresponde a la parte del trabajo que se desarrolla en el software CAM, y en él se generan todas las instrucciones de trabajo para la máquina CNC. El intérprete de los

comandos en este caso el controlador general recibe cada una de las instrucciones que se incluyen en el código G, las recibe una por una, y así mismo las ejecuta, para ello debe recibir una apenas termina con la que está trabajando. Este decide si moverse en X, Y o Z según sea la instrucción del código G y convierte estas en señales de control para los drivers de los actuadores de la máquina.

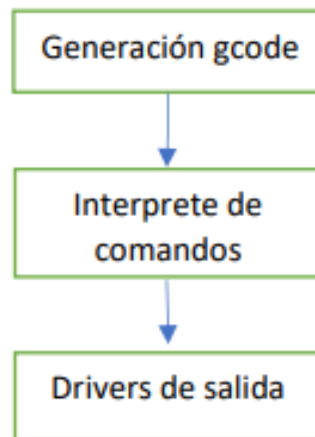


Figura 39 - Procedimiento para el trabajo de la máquina CNC

Los drivers de salida son los módulos controladores de los motores de paso y herramienta de fresado. Estos reciben la señal de control de parte del microcontrolador de la tarjeta controladora.

4.3. Diagramas de flujo, algoritmo de trabajo.

Los siguientes diagramas corresponden a los diagramas de flujo que muestran el algoritmo de trabajo de la máquina a implementar.

La primera opción analizada para implementar el hardware controlador es la que se muestra en la Figura 40, y se implementa tanto para el controlador interfaz con el computador y para la interfaz con la SD Card.

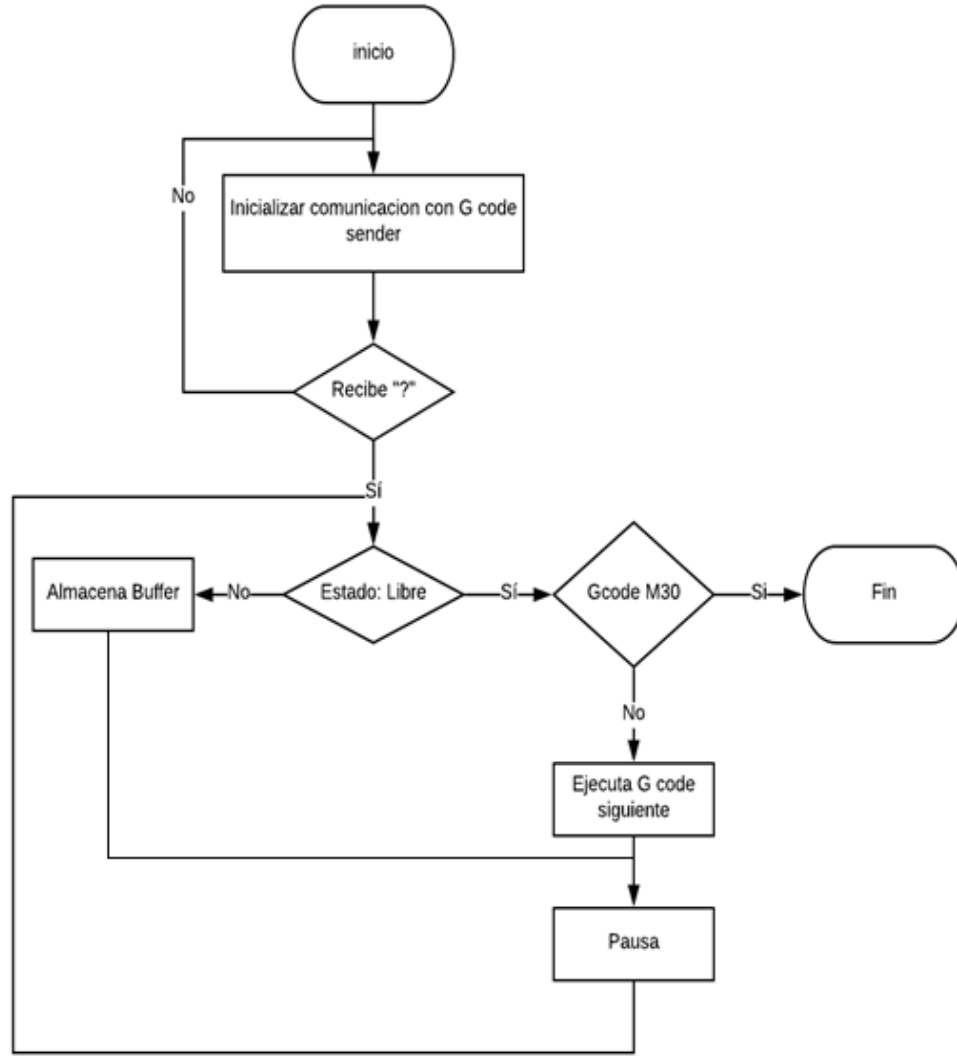


Figura 40 - Algoritmo de trabajo de la máquina

Implementación pseudocódigo

Proceso ()

{

Serial.send (?);

Serial.recieve (estate);

Serial.send (instruccion);

```
Pause ();  
}
```

La Figura 41, muestra el diagrama general del trabajo. Al inicializar la máquina, el hardware controlador espera la pregunta del estado que le hace el software (Gcode sender). Estando libre la máquina, puede desarrollar la instrucción de Gcode que ha recibido. Luego de un tiempo enviar la siguiente línea de código, este tiempo de pausa hasta enviar la siguiente línea de código debe calcularse en función de las operaciones cargadas en el Gcode total. Este método implica mucha complejidad en vano, ya que, si se tiene una pausa constante, podría ser que existan operaciones que demande demasiado tiempo de operación y provoque que el buffer se sature y ocasione pérdidas de información. Por ejemplo, si se tiene una pausa de 200 ms y existe una operación que demanda más de un minuto, podría ser que en este tiempo se llene el buffer con las instrucciones pendientes y producto de ello se pierdan instrucciones; pudiendo generar así problemas en el trabajo CNC. Por otro lado, se podría solucionar eso calculando el tiempo estimado de operación, usando las coordenadas de la instrucción que está siendo implementada y la siguiente. Pero existe un modo más eficiente y es el que usa GRBL y que se muestra en el diagrama de la figura 39. Esperar la respuesta de tarea terminada. Es decir, el hardware controlador envía una respuesta confirmando que ha terminado la tarea y está listo para ejecutar la siguiente tarea.

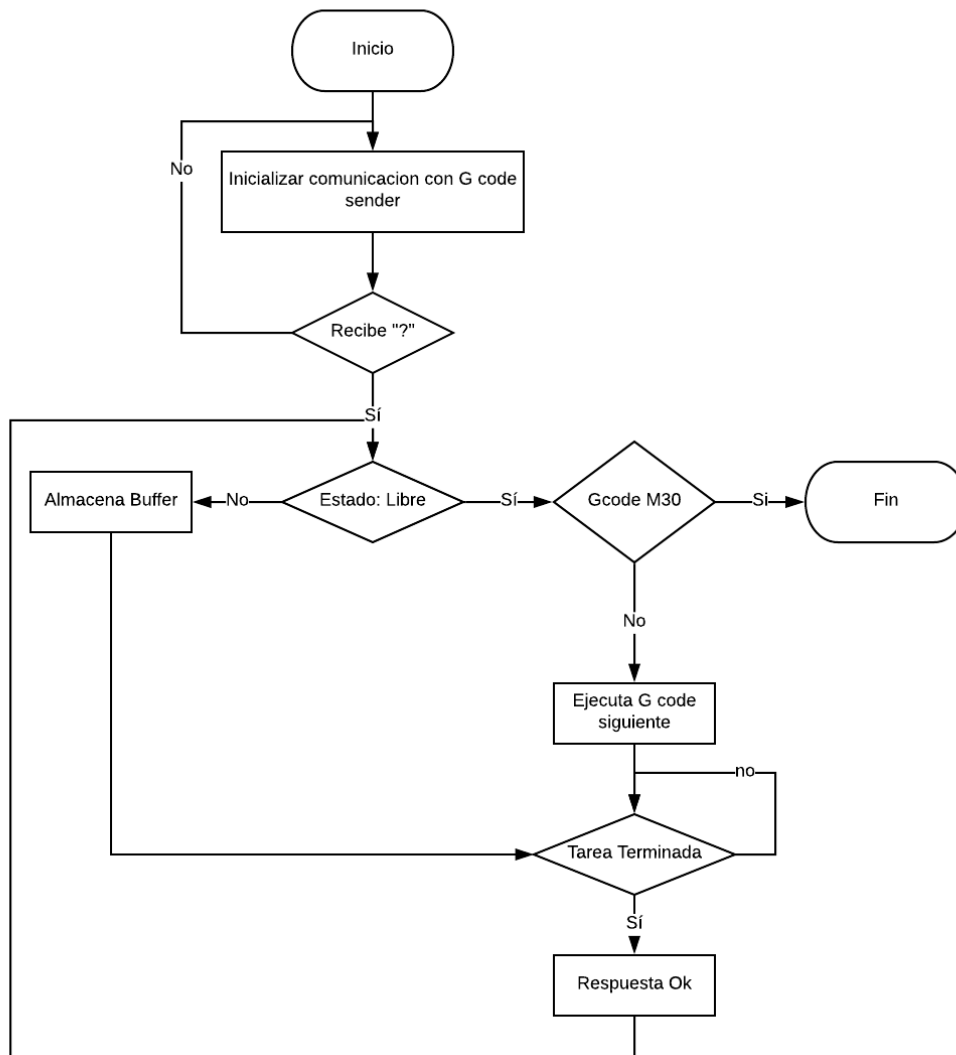


Figura 41 - Algoritmo de trabajo basado en Grbl

Implementación pseudocódigo, la línea en rojo es la implementación de la espera por la respuesta ok.

Proceso ()

{

Instruccion = "";

Serial.send (instruccion);

While (Serial.recieve () !=ok)

```
    {}  
    LCD.setCursor (0, 2);  
    LCD.print (get_ok);  
    LCD.setCursor (0,3);  
    LCD.write (Serial.read ());  
  }  
}
```

Una vez establecida la comunicación como se describió en el diagrama anterior, el hardware espera un Gcode a ejecutar, si está ocupado haciendo una tarea, almacena la línea de trabajo en el buffer hasta terminar la tarea instantánea. Cuando la tarea que me mantenía en ejecución ha sido terminada, el hardware responde con un ok y está listo para realizar la siguiente tarea. Se sigue este proceso hasta que una línea de Gcode corresponda al código M30 que indica la culminación del trabajo, fin del programa. Este proceso resulta más eficiente ya que espera una respuesta de confirmación, evitando así, usar recursos de memoria y tiempos de instrucción del microcontrolador en complejos cálculos de tiempos de espera.

El siguiente diagrama de la Figura 42, corresponde al proceso que se realiza para identificar y ejecutar cada instrucción.

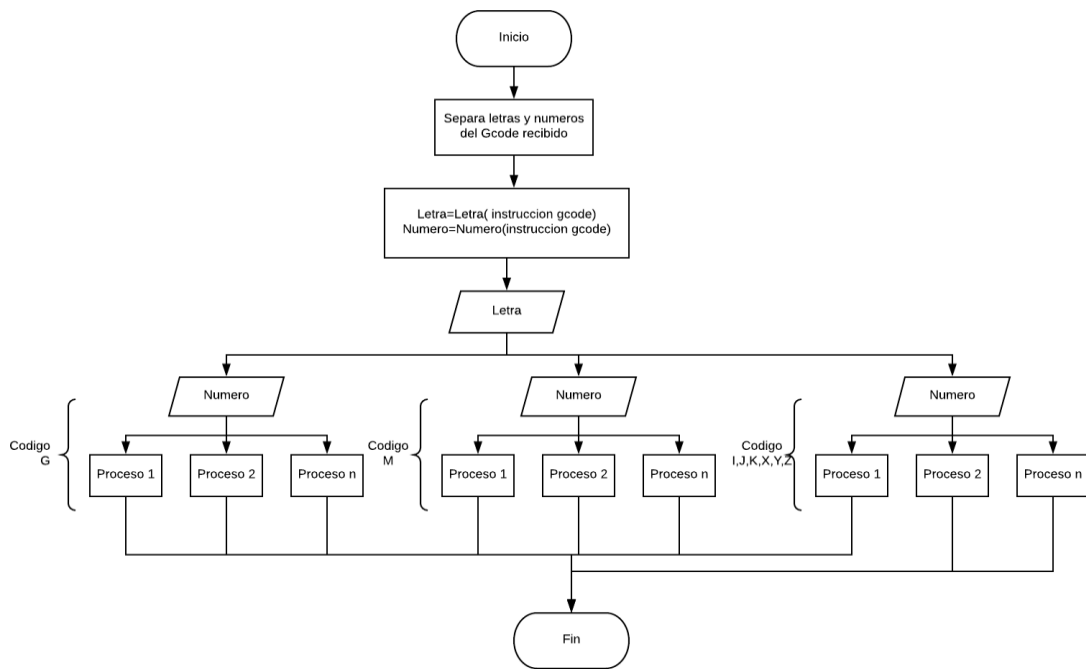


Figura 42 - Identificación de cada instrucción

Para ejecutar cada instrucción es necesario identificar que código corresponde, para lo cual se realiza el proceso y se describe a continuación; en la línea de Gcode recibida se separan letras de números, esto permite identificar si corresponde a un código, G, M, I, o las coordenadas X, Y, Z. A través de un switch case se dirige al proceso que corresponde, en el cual así mismo se realiza una nueva validación para identificar que instrucción es, según el código identificado. Este procedimiento es el equivalente a la subrutina “ejecuta Gcode” que se muestra en el diagrama anterior.

Implementación en pseudocódigo:

```

Proceso (){
    Letra=obtener_letra (instrucción)
    Numero=obtener_numero (instrucción)
    Switch (letra)
  
```

```
Case G:
    Switch (numero)
        Case 1: funcion1
        Case2: funcion2
    Case M
        ...
}
```

Los códigos se adjuntan en el anexo.

El controlador es representado en el bloque que muestra el intérprete de comandos.

Para el diseño del hardware controlador se necesita un microcontrolador de gama alta para un correcto procesamiento. El microcontrolador seleccionado es el PIC18f4550.

- Se necesita que el controlador sea capaz de interpretar los comandos Gcode. Esto es, un programa que procesa el Gcode generados en el software CAD que se haya empleado para la generación del Gcode y genere las señales de control.
- Las señales de control que se generan se envían a los drivers que permiten controlar los actuadores (motores).

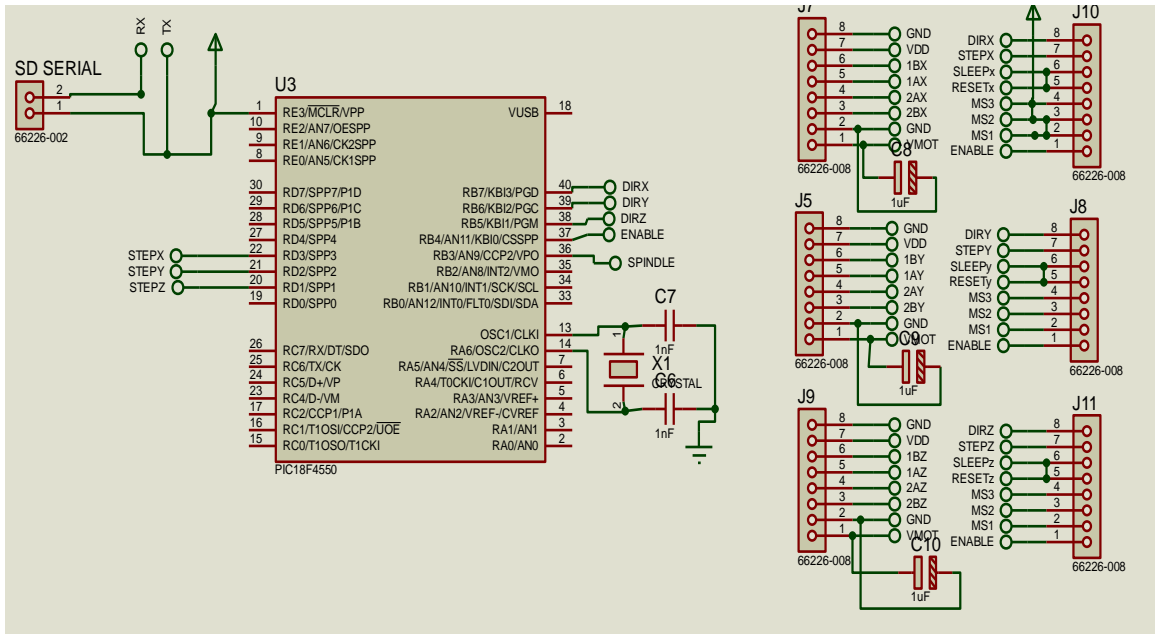


Figura 43 - Diagrama de conexiones microcontrolador –drivers

La Figura 43 muestra el diagrama esquemático que corresponde a las conexiones entre el microcontrolador y los drivers de salida.

4.4. Driver de motores a paso

El A4988 que se muestra en la Figura 44, es un controlador de motor de pasos y permite trabajar con micro pasos. Está diseñado para operar motores paso a paso bipolares en pleno, medio, cuarto, octavo y modos de dieciséis pasos, con una capacidad de salida de hasta 35 V y ± 2 A.



Figura 44 - Diagrama de conexiones microcontrolador - drivers

Los motores de paso se conectan a 12V y para ello necesitan una fuente que provea ese nivel de voltaje y permita manipular la corriente que consuma.

Las conexiones entre el motor y el driver se muestran en la siguiente, Figura 45. Se muestran las señales DIRX, STEPX que van desde el controlador al driver de los motores. La conexión es la misma para los motores de los otros ejes.

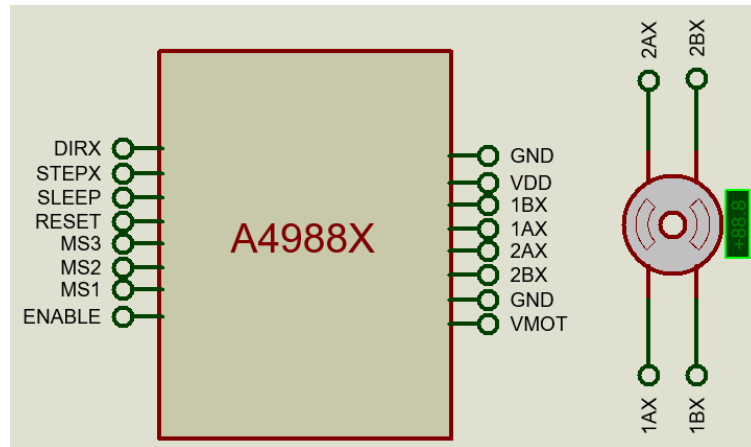


Figura 45 - Conexiones entre motor y drivers.

El control del motor de la herramienta es un módulo que consta de un circuito integrado L298 empleado para manipular corrientes de hasta 3 amperios en el manejo de motores DC. Las conexiones que realizar en el módulo son las que se muestran en la Figura 46.

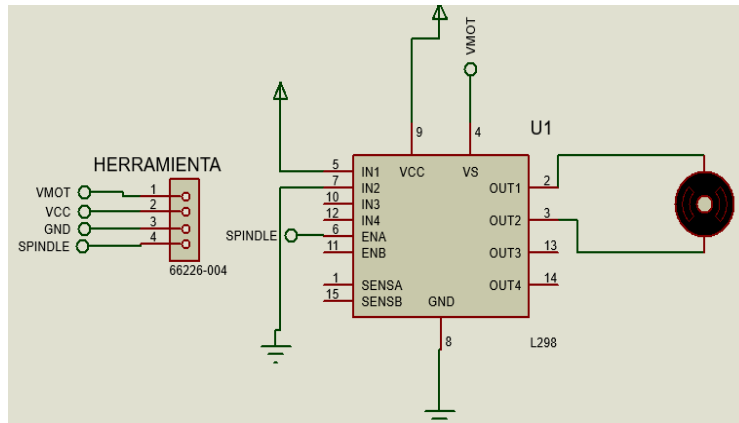


Figura 46 - Controlador del motor

4.5. Ensamble de tarjetas electrónicas del controlador

Paso 1: Dentro de la caja, que se usara de soporte para los microcontroladores que irán conectados a los drivers del motor de paso. En la Figura 47 se muestra la caja y la ubicación de la tarjeta controladora, y cableado las conexiones haciendo uso de ductos negros para mejorar la distribución del espacio.

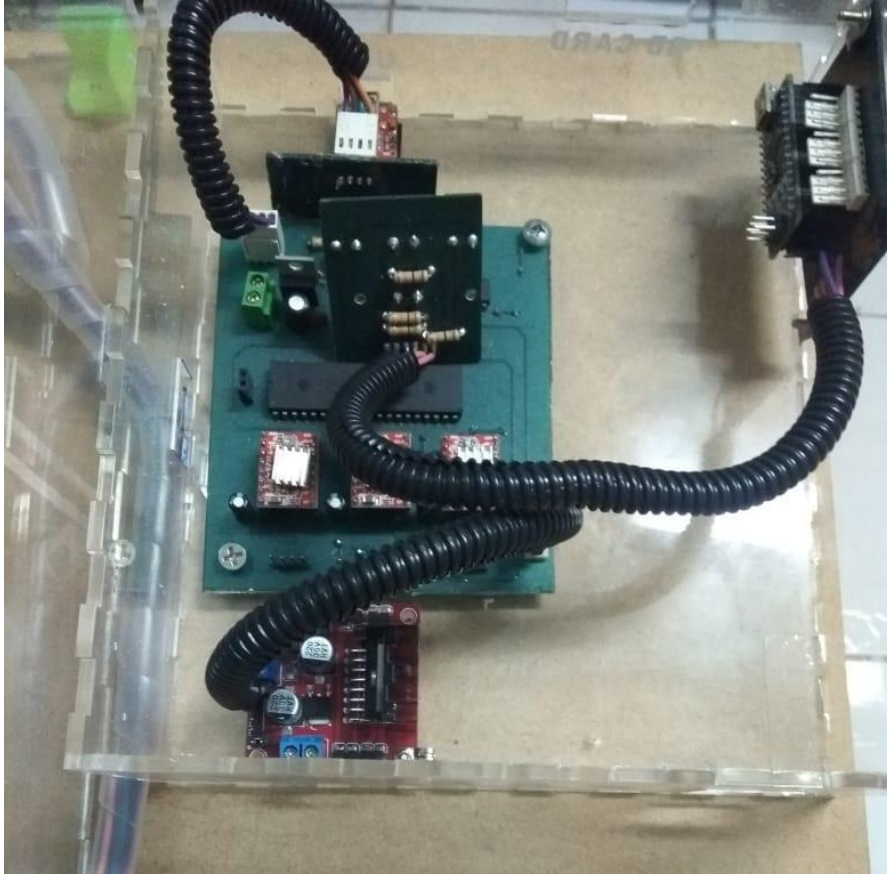


Figura 47 - Controlador del motor

Paso 2: Acople de tarjetas y medición de continuidad en las conexiones. En la Figura 48 se muestra el desarrollo del trabajo de asegurar la existencia de continuidad en los puntos que deben mantener conexión.



Figura 48 - Controlador del motor

Paso 3: Tarjeta ensamblada, lista para usarse.

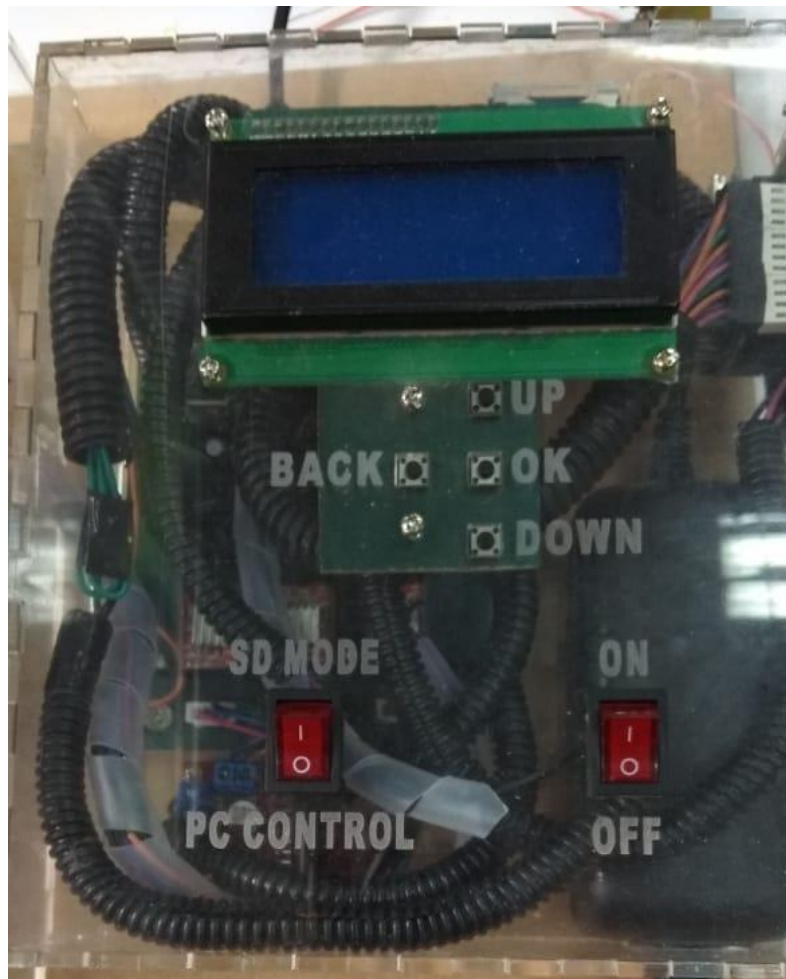


Figura 49 - Tarjeta controladora ensamblada

La Figura 49 expone la tarjeta controlada ya ensamblada con sus periféricos de la interfaz manual.

4.6. Generación del código G

Para la generación del código G y realizar pruebas usamos el software Carbide Create ya que es un software fácil de usar y su extensión para generación de código G, es la convencional (.nc). La Figura 50 muestra un trazo de prueba realizado para verificar los movimientos de la máquina. Pero aquí solo se muestra el proceso de generación de G code.

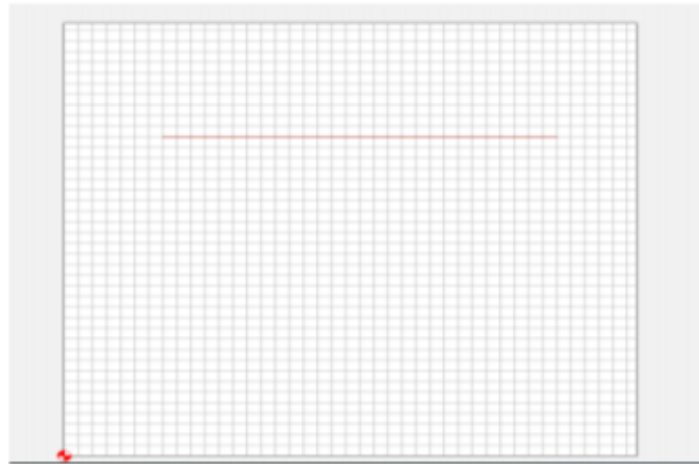


Figura 50 - Controlador del motor

La herramienta de trazado es de dimensión 0.03 mm, que corresponde a las fresas de desbaste que fueron descritas anteriormente. La Figura 51 muestra el proceso de selección de herramienta. Este proceso se realiza en todo software CAM.

The image shows a software interface with two main sections. The top section is titled "Tool" and contains a dropdown menu with the text "#122 - 0.031 in End". Below the dropdown is a green button labeled "Edit Library". Underneath that is a checkbox labeled "Set speeds automatically" which is currently unchecked. The bottom section is titled "Cutting Parameters" and contains two input fields. The first is labeled "Depth per Pass" and has the value "0.1" entered, followed by the unit "mm". The second is labeled "Stepover" and has the value "0.357" entered, followed by the unit "mm".

Figura 51 - Parámetros de calibración

En la figura anterior también se muestran los parámetros tales como Depth Pass que corresponden a la profundidad en cada paso, está configurado a 0.1 mm, eso quiere decir que, si se requiere penetrar 1 mm, realizara 10 veces la trayectoria para alcanzar esa profundidad. En la Figura 52 se muestra el Toolpath generado. Para tenerlo disponible se procede a guardar el código G.



Figura 52 - Exportando Gcode

4.7. Software envío Gcode

4.7.1. Grbl Control

El software usado para enviar el código G a la máquina CNC es el de licencia libre, GRBL controller, este nos permite controlar el movimiento de los ejes de la máquina y enviar el código G, a la vez de tener una vista previa del mecanizado. La Figura 53 muestra la interfaz de usuario del software que se menciona, el Grbl control. Existen muchos otros, pero este es el que se emplea en las pruebas del proyecto, al ser un software muy versátil y de licencia libre.

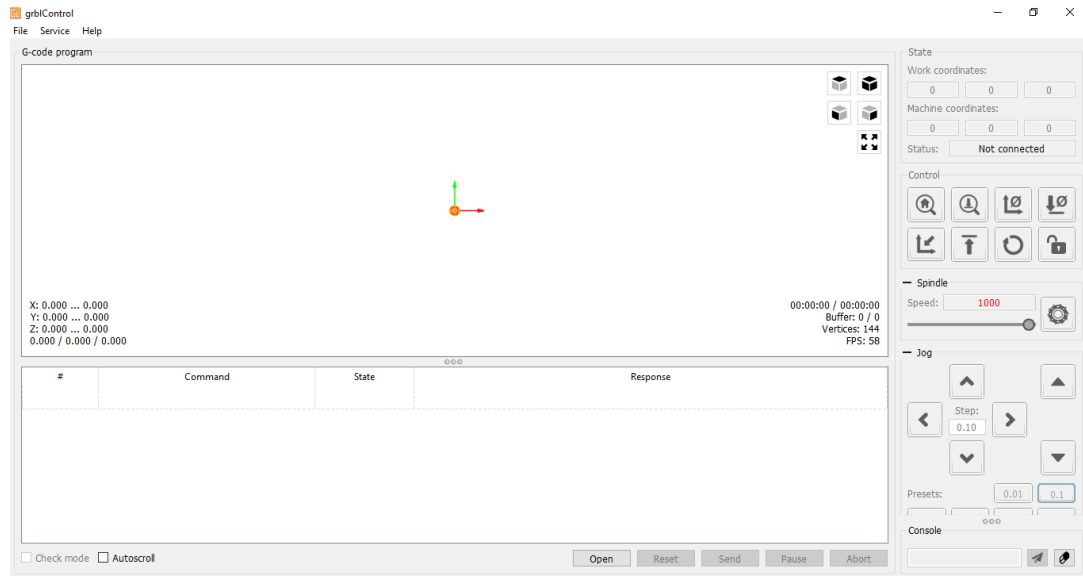


Figura 53 - Interfaz de usuario software GrblControl

Los parámetros de la máquina al encender o iniciar la máquina están a cero, como se indica en la siguiente imagen, Figura 54, dependiendo del tipo de máquina algunas suelen tener sensores que indican la posición a la cual se encuentra al momento de encender respecto al punto de origen de la máquina.

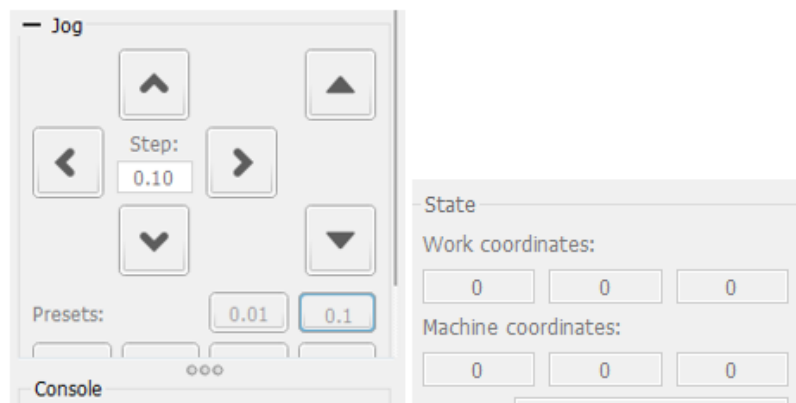


Figura 54 - Posicionamiento de la herramienta y desplazamiento en los ejes.

El software que se use para enviar el código G permite cargar y visualizar el archivo previamente. En la Figura 55 me expone un claro ejemplo de lo que permiten hacer

estos programas, visualizar las trayectorias y el punto de inicio donde se ubica la herramienta.

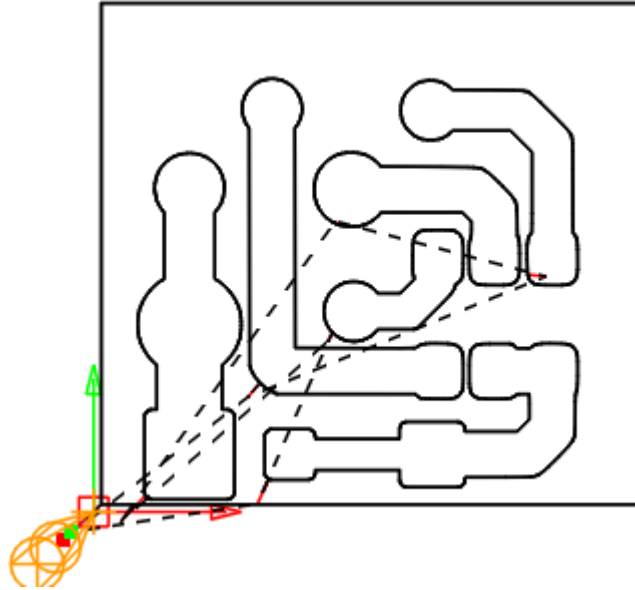


Figura 55 - Archivo de prueba para calibración cargado en el software Grbl controller. Posteriormente, se hace la transferencia del archivo.

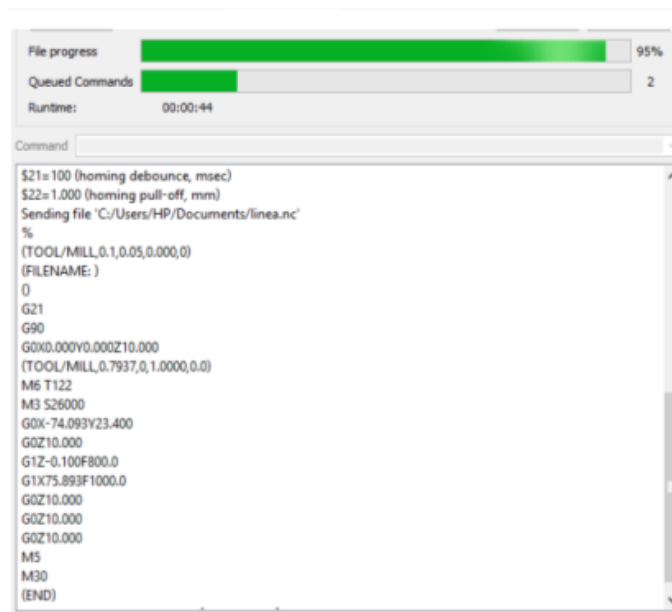


Figura 56 - Transferencia del Archivo.

En la Figura 56 se muestra el proceso de transferencia de un código G. La barra verde indica gráficamente el porcentaje del archivo que ha sido enviado y en la ventana inferior se muestran los códigos enviados.

4.8. Carcasa plástica

La carcasa se la hará de un material plástico, acrílico transparente, con unas dimensiones de 45 cm de largo, 30 cm de alto, pendiente trapezoide de 31,24 cm de longitud, 10 cm en la cota mínima y 30 cm en la cota máxima (Vista lateral).

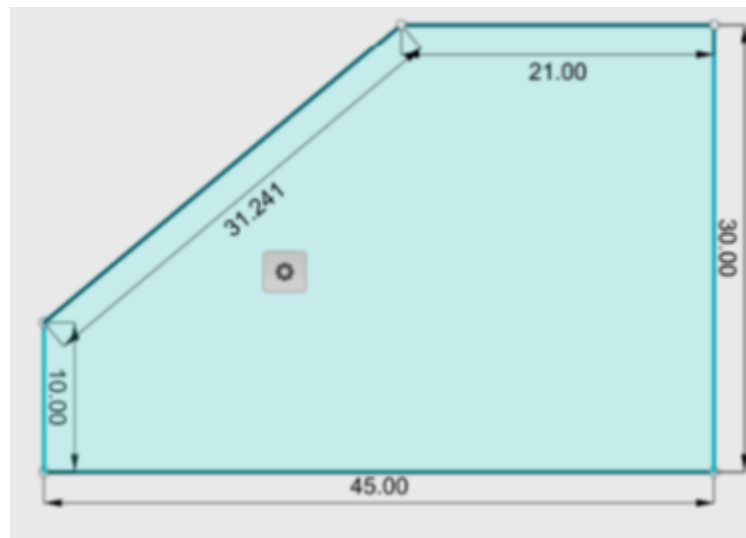


Figura 57 - Vista lateral

La Figura 57 muestra la vista lateral izquierda y derecha de la carcasa a elaborar. La cara superior, cuya vista se muestra en la figura 54, tiene 45 cm de largo y 21 cm de ancho.

El espesor del material empleado es de 3 mm.

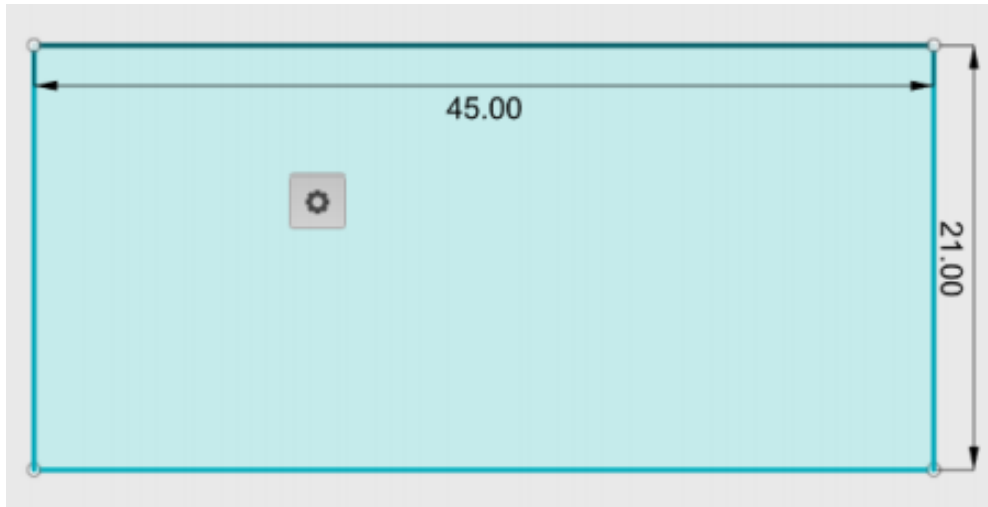


Figura 58 - Cara superior

La cara posterior de la carcasa tiene 45 cm de largo y 30 cm de alto, la Figura 58 muestra las dimensiones de esta.

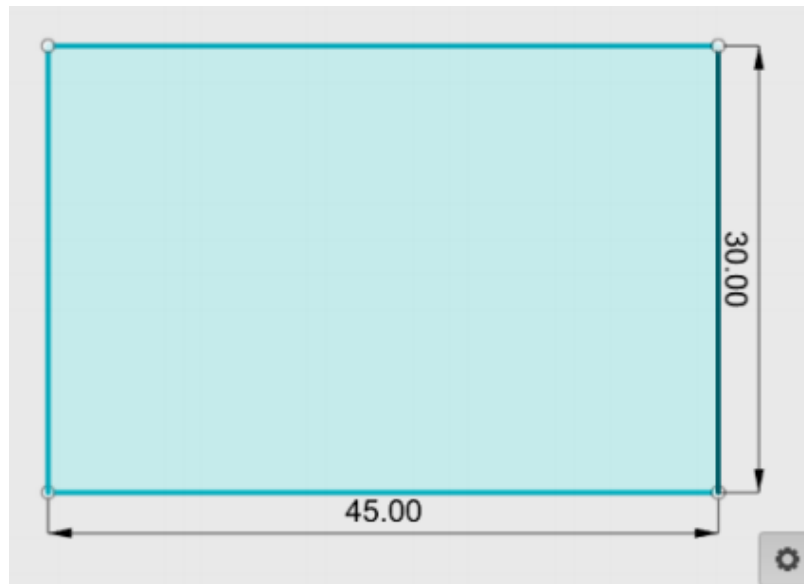


Figura 59 - Cara anterior

La cara inferior, cuya vista se muestra en la Figura 59, tiene 45 cm de largo y 45 cm de alto, es la base de la carcasa sobre la cual se posa la estructura de la máquina CNC.

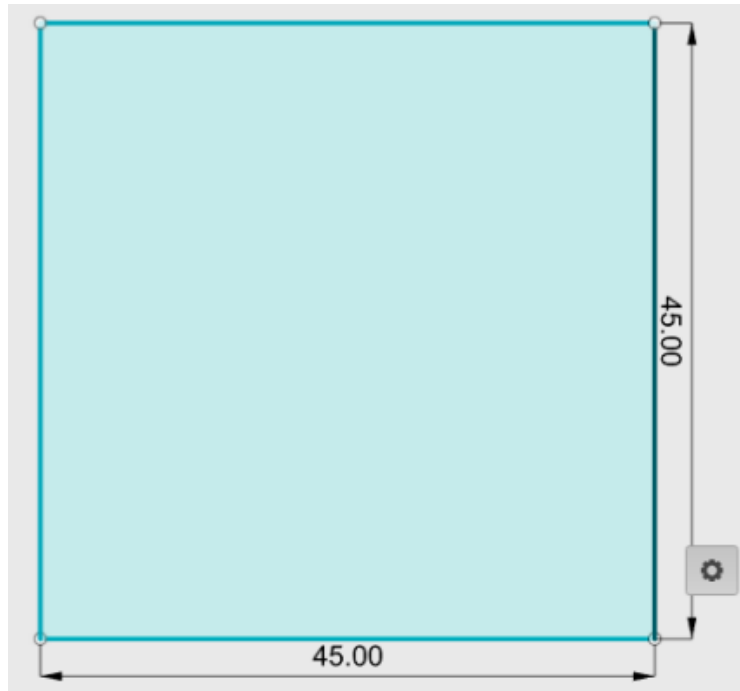


Figura 60 - Cara inferior

Las dimensiones de la vista frontal, Figura 60, de la carcasa corresponden a 45 cm de largo y 10 cm de alto.

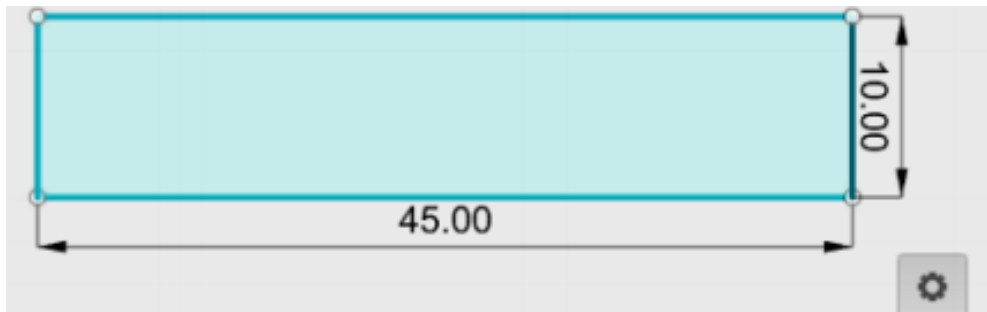


Figura 61 - Cara frontal

La tapa de la carcasa tiene las medidas que se muestran en la Figura 61 y corresponde a 45 cm largo y 31.24 cm de ancho.

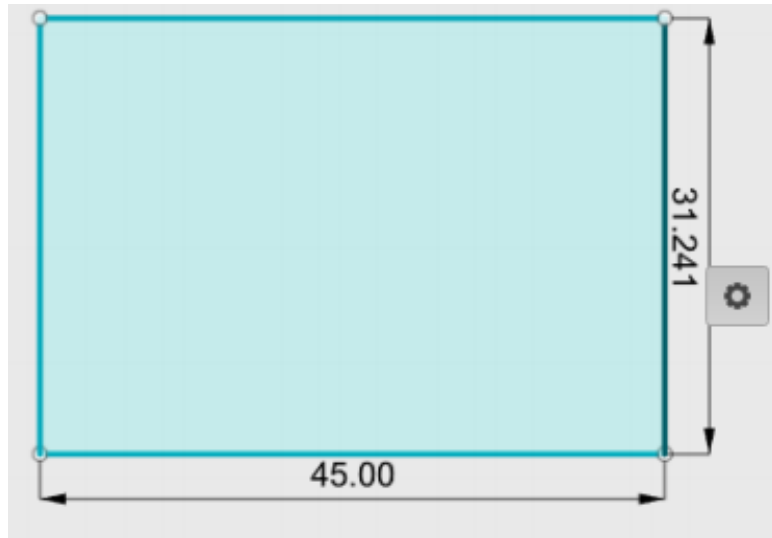


Figura 62 - Tapa

4.9. Armado de carcasa

Se toman las medidas de las dimensiones de la carcasa, para proceder a realizar los cortes. En la Figura 63 se muestra el proceso de toma de medidas y corte del material.

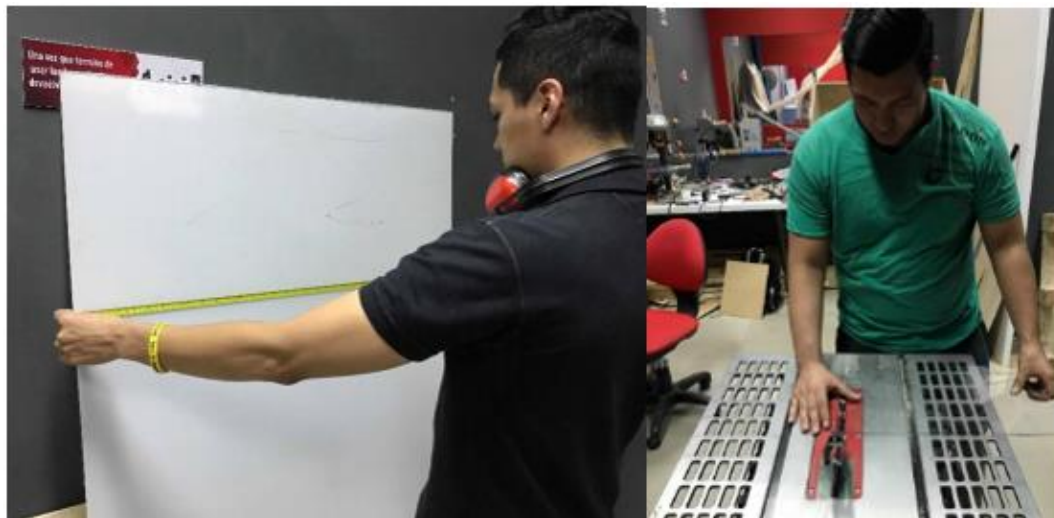


Figura 63 - Toma de medidas

Luego del corte de retazos, se procede a realizar cortes más pequeños para elaborar las caras de la carcasa, y una a una ensamblarlas, tal como se muestra en la Figura 64.



Figura 64 - Corte de piezas



Figura 65 - Armado de la carcasa

La Figura 65, la figura anterior, muestra el proceso de armado y sujeción de la carcasa con tornillos para evitar que se separen los lados de la carcasa.

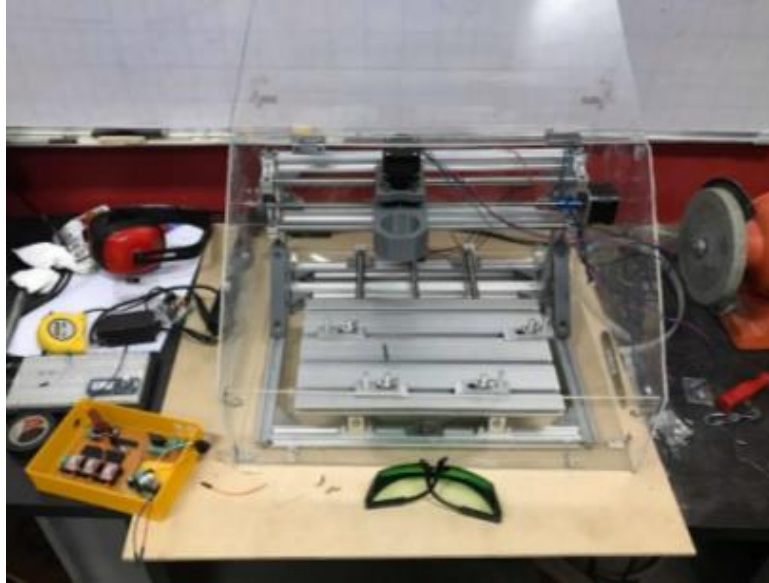


Figura 66 - Carcasa montada

En la Figura 66 se muestra la carcasa ya ensamblada con todas sus partes. Tal como fue descrita, con material acrílico transparente de 3 mm.

4.10. Validación del prototipo

Luego de seguir los procedimientos de implementación y montaje descritos en el capítulo anterior el equipo es sometido a pruebas de funcionamiento.

Las características que se evaluarán son:

Comunicación entre Gcode Sender y el board de la máquina CNC.

Lectura y trabajo de códigos G generales.

Pruebas de mecanizado CNC.

4.10.1. Comunicación entre Gcode sender y el board de la máquina CNC.

Abriendo el programa grblControl, la Figura 67 muestra la interfaz de usuario del programa que envía el código G a la tarjeta controladora. En la imagen se visualiza el estado de la máquina, e indica, no conectado.

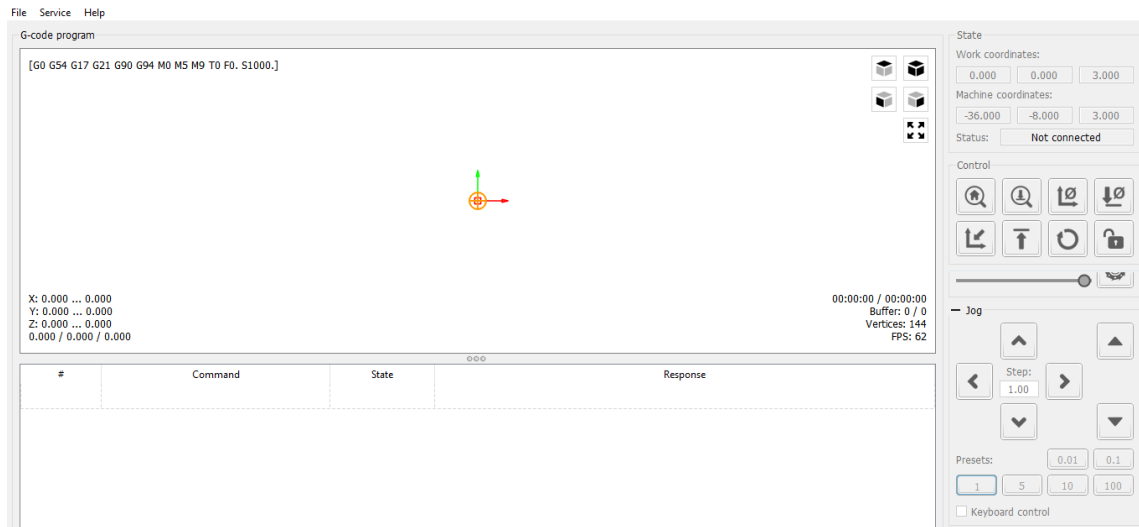


Figura 67 - Interfaz de usuario

La Figura 68, muestra la ventana de configuraciones en la cual seleccionando la opción de conexión se puede verificar el puerto COM al cual está conectado la tarjeta, para establecer la comunicación.

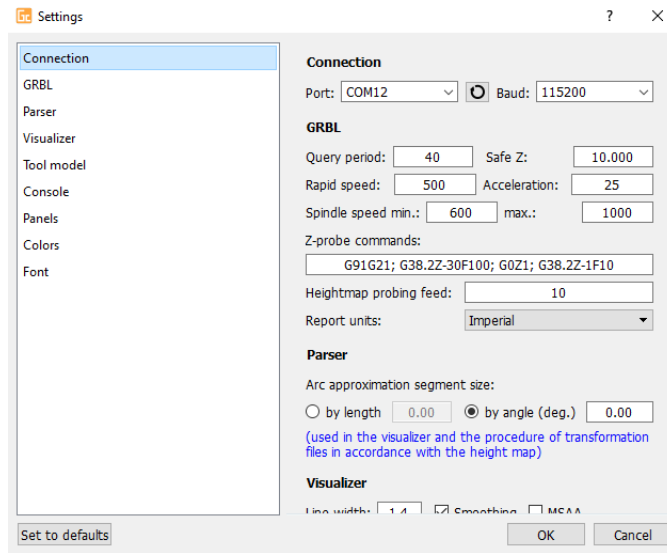


Figura 68 - Conexión al puerto COM



Figura 69 - Conexión establecida tarjeta-interfaz

En la Figura 69 se puede observar que la comunicación ha sido establecida, ya que el estado indica Idle, que es la respuesta de la máquina indicando que está libre.

La prueba de movimiento en los ejes se realiza a continuación, moviendo uno a uno los ejes, en la Figura 70 se muestra el estado “run” de la máquina cuando se trabaja sobre cada uno de ellos.

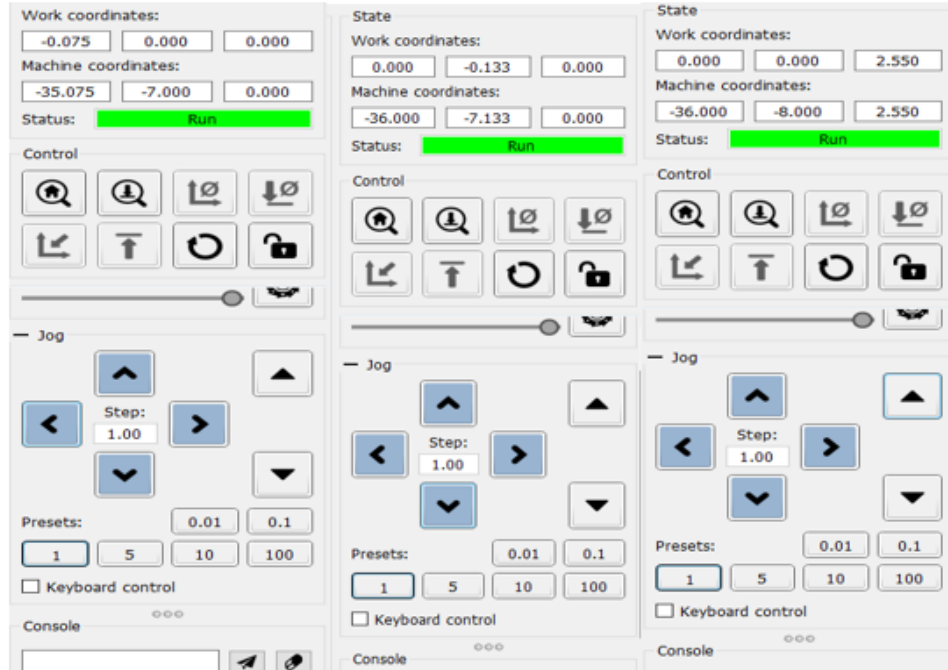


Figura 70 - Movimientos en eje x, eje y, eje z

4.10.2. Pruebas en la interfaz manual

Las pruebas de la interfaz manual se realizan probando en menú principal, las opciones de menú y el movimiento de los ejes de la máquina. En la Figura 71 se muestra el menú principal de la interfaz manual, con sus botones para acceder a las opciones.



Figura 71 - Menú de interfaz manual

En la Figura 72, se muestra el mensaje de la opción info del menú principal, donde indica que se debe establecer el punto de inicio para el trabajo si se va a enviar desde la tarjeta de memoria.



Figura 72 - Opción info del menú principal

En la Figura 73 se visualiza la opción de desplazamiento de los ejes. Indica que con el botón up se desplaza en sentido positivo y con el botón down se desplaza en sentido negativo, con el botón ok se configura el punto establecido como punto de coordenada

(0, y, z). Cuando se realice la acción son los demás ejes, en el caso del eje Y se configura (x, 0, z) y en el caso del eje Z se configura (x, y, 0); resultando al final la configuración del punto de origen para iniciar el trabajo, coordenada (0, 0,0).



Figura 73 - Pruebas de desplazamiento de los ejes.

En la Figura 74 se muestra la respuesta enviada por la tarjeta controladora a la interfaz manual, indica las posiciones que ocupa al desplazarse y el estado, si está libre o está ocupada.



Figura 74 - Información enviada por la tarjeta controladora



Figura 75 - Lectura de archivos en la interfaz manual

Desde la interfaz manual se puede leer una tarjeta SD y visualizar los archivos guardados en ella, para seleccionar un archivo de código G y posteriormente enviar al controlador de la máquina CNC para trabajarlo. En la Figura 75 se muestra la lectura de archivos .nc que corresponden a archivos de código G.

4.10.3. Pruebas de mecanizado CNC

Las pruebas para verificar la resolución del trabajo se realizan con trazos geométricos y con archivos de circuitos impresos. En la Figura 76 se muestra el trazo de una circunferencia para generar el código G, y mecanizar, para luego obtener el error entre el diseño CAD y el mecanizado de la máquina CNC.

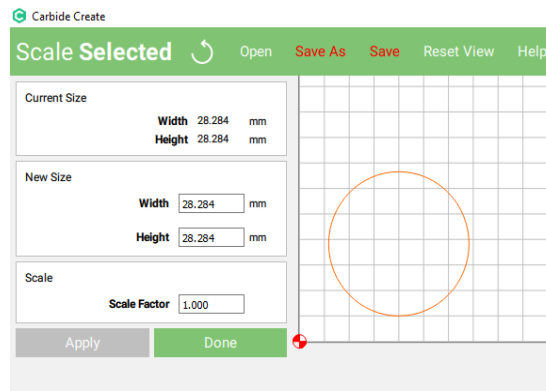


Figura 76 - Archivo de prueba de mecanizado

La siguiente Figura 77, muestra la vista previa del trabajo a realizarse. La vista previa que permite el software GrblControl que se mencionó en el capítulo anterior.

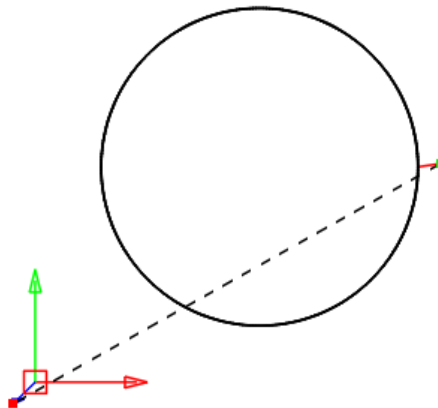


Figura 77 - Vista previa del trabajo

4.10.4. Proceso de mecanizado

Como se describió en el capítulo anterior, debe configurarse el punto de inicio antes de cada mecanizado. En la Figura 78 se muestra el ajuste del punto de origen. Es conveniente que se ubique cerca del punto de origen en el material a mecanizar.

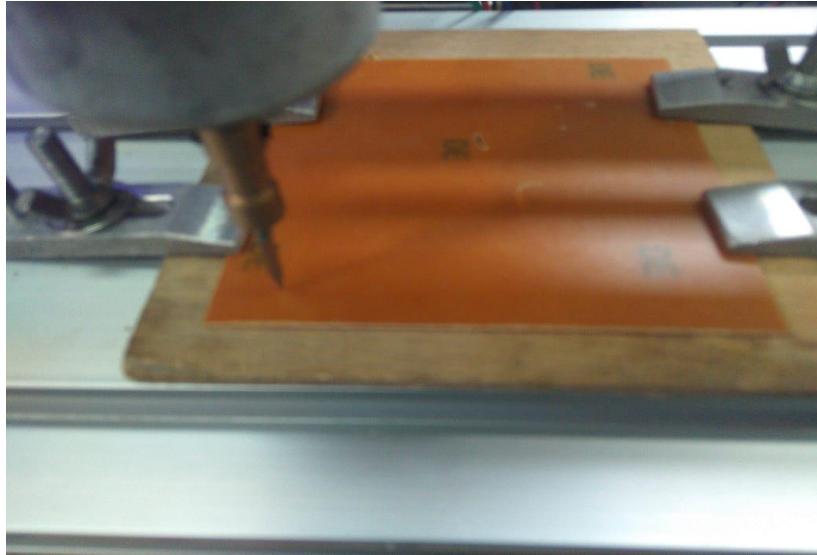


Figura 78 - Ajuste del punto de origen

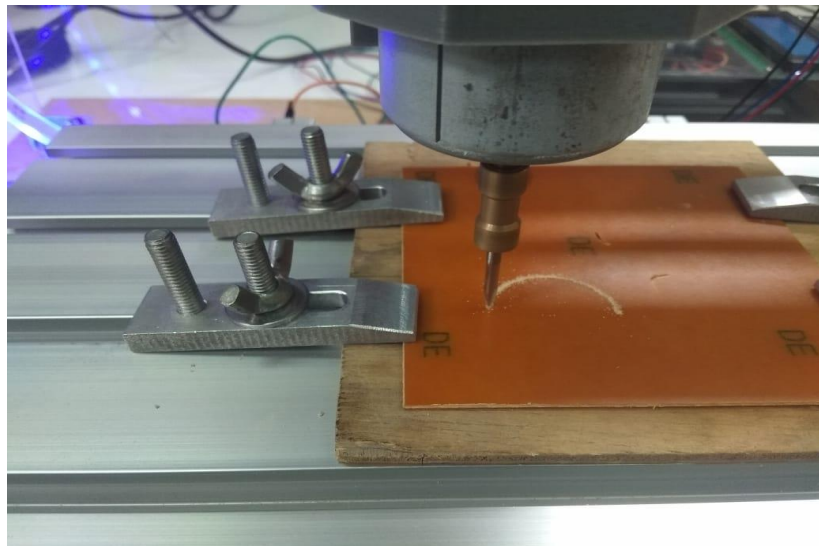


Figura 79 - Inicio de trabajo

En las Figura 80 y Figura 81 se muestra el procedimiento de mecanizado según el trazo circular diseñado en el software CAM.

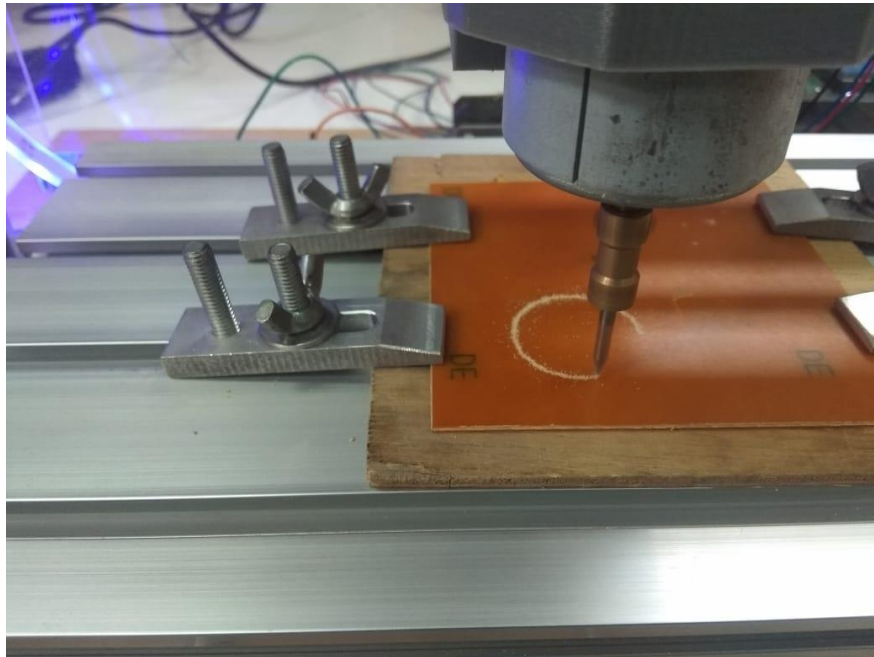


Figura 80 - Calado en forma circular

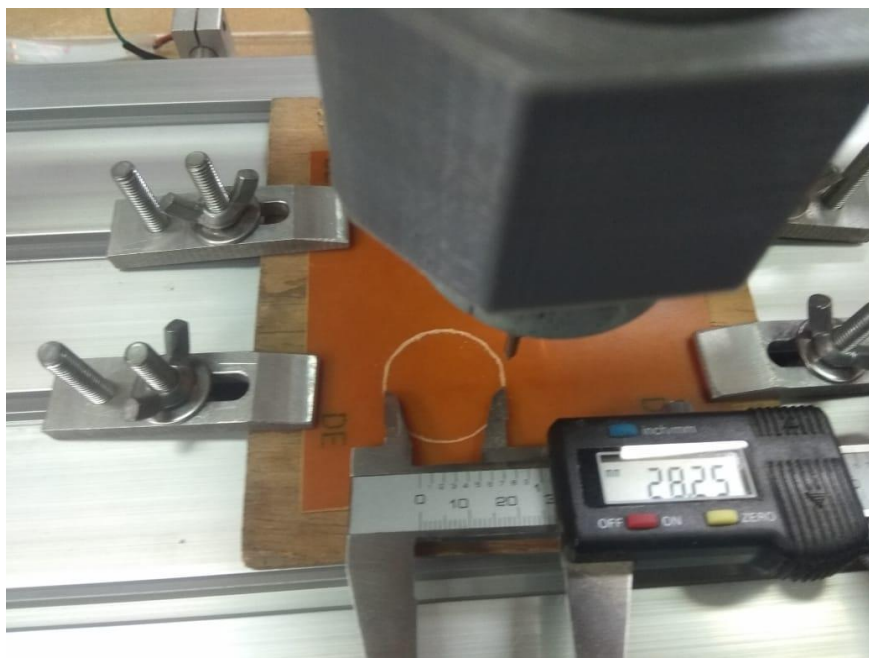


Figura 81 - Medición de trabajo final

El trabajo terminado como se muestra en la Figura 81, al medirse con un calibrador digital, tiene una longitud de 28.25 mm y que la longitud del diseño es de 28.284 mm.

El error existente es entonces:

$$Error = \frac{teorico - real}{real} = \frac{28.284 - 28.15}{28.25} = 0.0012$$

Un error del 0.12% que nos permite tener una buena resolución en el trabajo realizado.

La Figura 82 muestra el mensaje de culminación de la tarea terminada y el tiempo aproximado del trabajo.

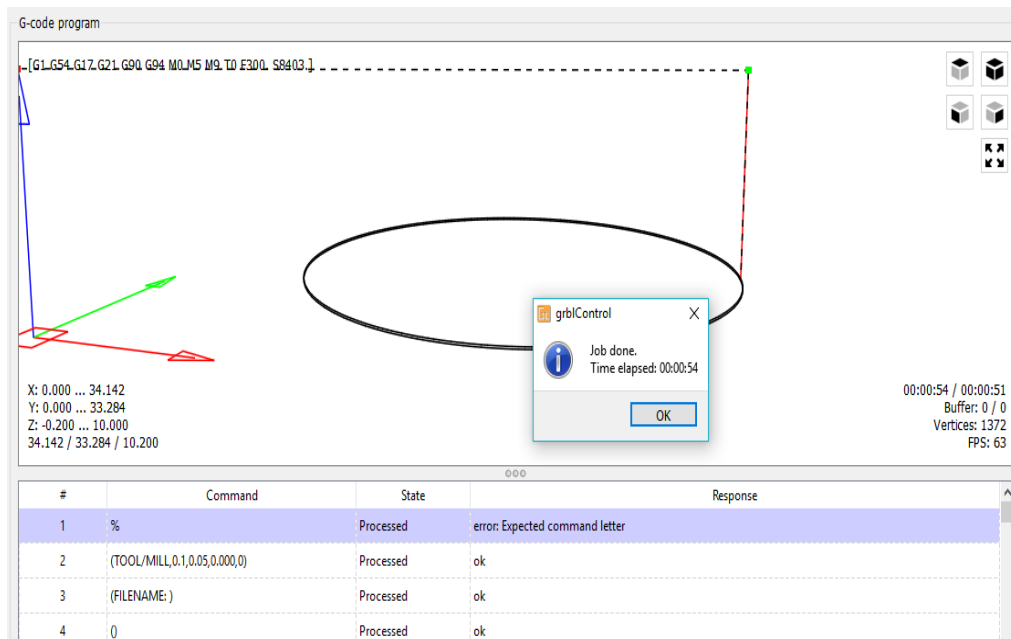


Figura 82 - Interfaz gráfica, culminación de primera prueba

4.10.5. Prueba de elaboración board:

Diseño en el software Proteus 8.1. PCB sencillo para la prueba.

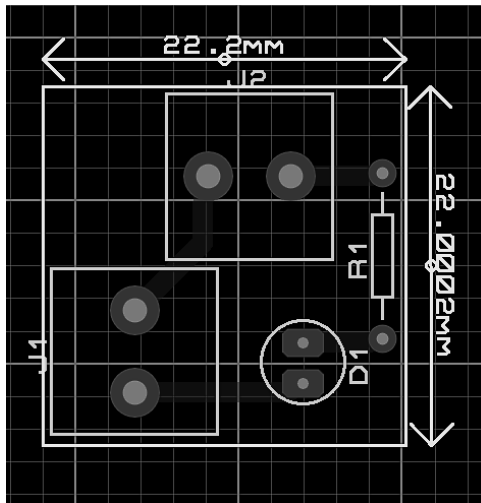


Figura 83 - Interfaz gráfica, culminación de primera prueba

Con la imagen generada en el diseño del circuito impreso, imagen que se muestra en la Figura 83, vectorizamos para obtener las trayectorias de las pistas, tal como se observa en la Figura 84.

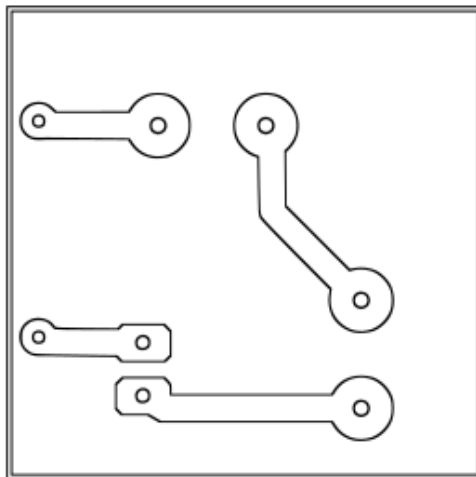


Figura 84 - Interfaz gráfica, culminación de primera prueba

Con las trayectorias vectoriales se puede generar el Gcode para el circuito impreso de prueba. Usando la herramienta de software, Carbide create, a partir de las trayectorias vectoriales se genera el código G, como puede verse en la Figura 85.

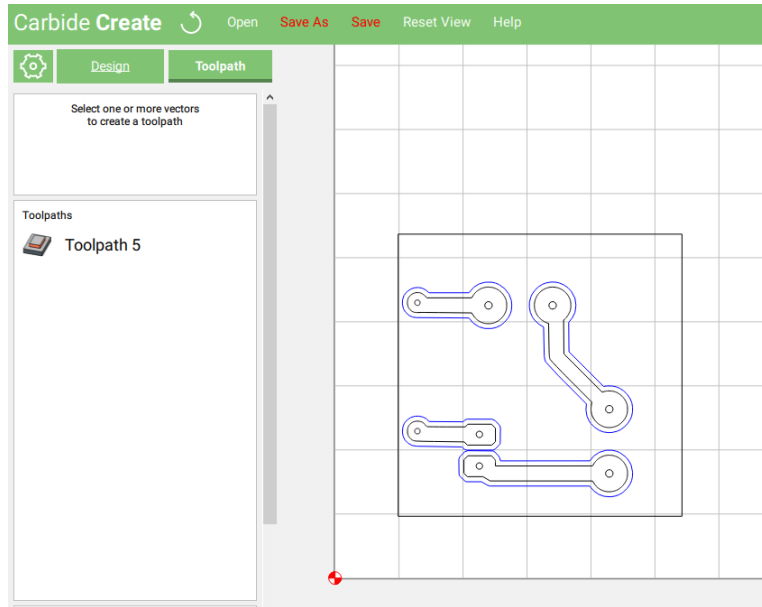


Figura 85 - Trayectorias para elaborar código G

Fresado del corte que corresponde al borde del circuito impreso y es otra trayectoria, puesto que se debe cortar más profundo. En la Figura 86 se muestra la trayectoria del corte en el borde.

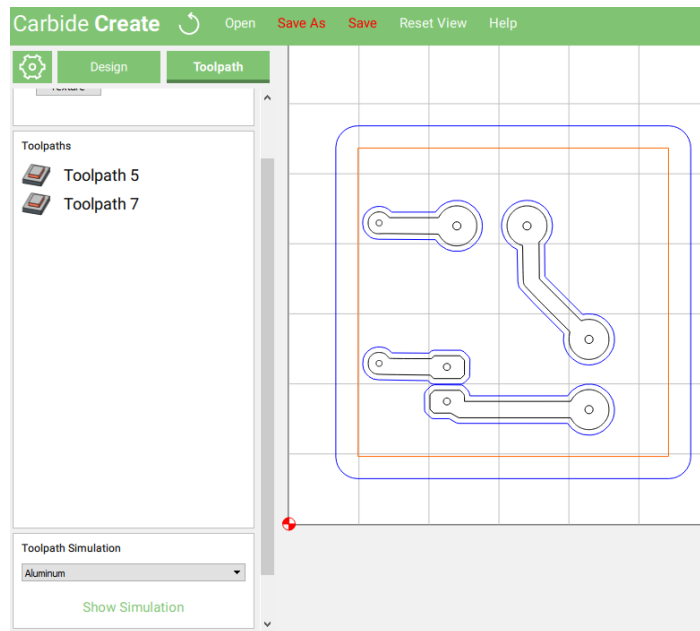


Figura 86 - Trayectorias para corte en el borde

Envío del código G a través del software GrblControl, el Gcode sender.

Una vez que se ha colocado el punto de inicio para el trabajo, en la máquina, se procede a configurar como punto de origen al punto deseado, como se describió previamente y como se muestra en la Figura 87, a través del código G92 con las coordenadas X0Y0Z0 correspondientes.

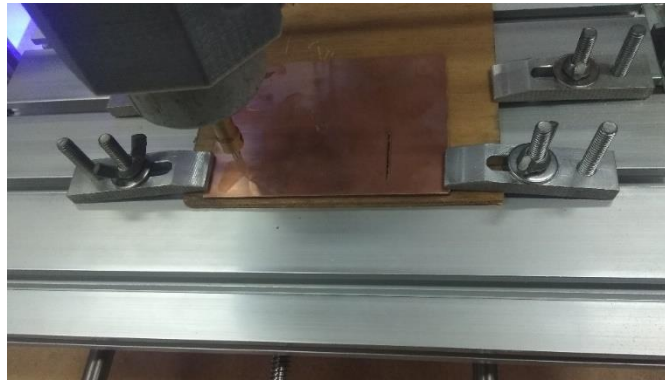


Figura 87 - Configuración punto origen

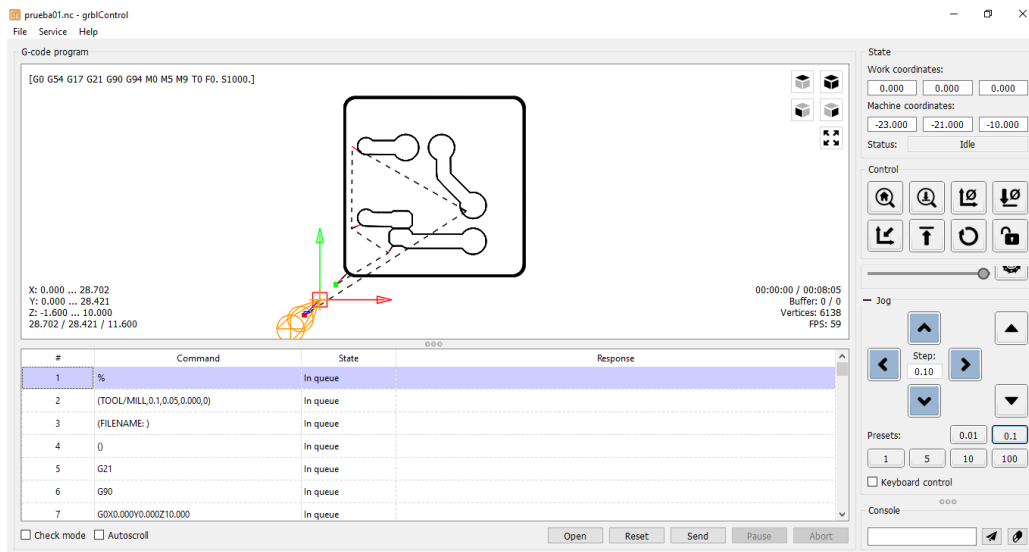


Figura 88 - Trayectorias para corte en el borde

En la Figura 89 se muestra la vista previa del archivo a mecanizar, la vista previa desde el software que envía el código G.

En la ventana del software que envía el Gcode, vemos el estado de la máquina y las líneas de código enviadas y procesadas.

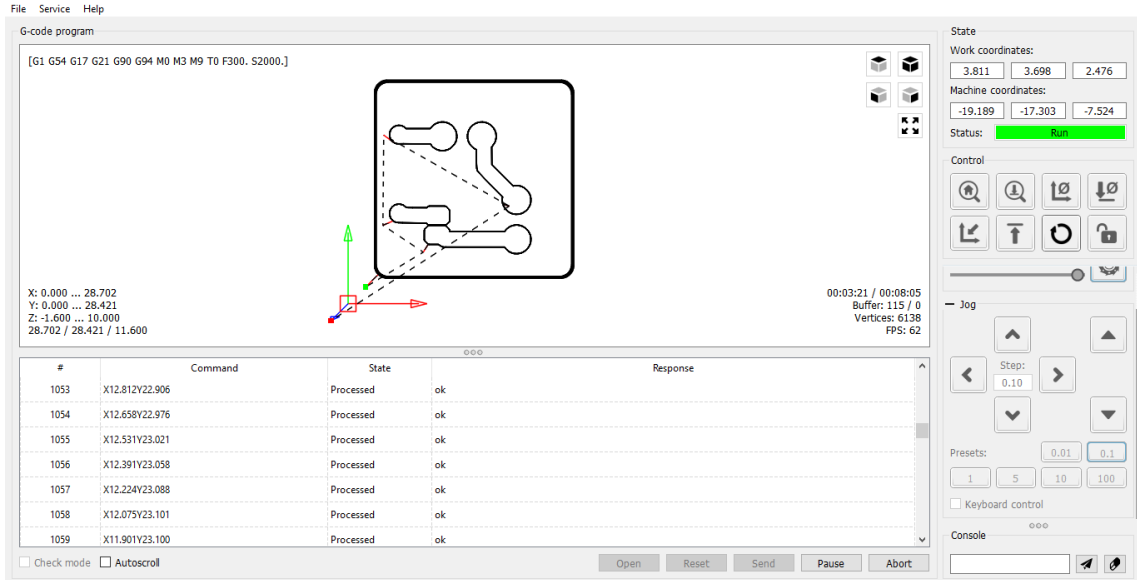


Figura 89 - Trayectorias para corte en el borde

Las Figura 90, Figura 91 y Figura 92 muestran el proceso de mecanizado desde el inicio, mitad de trabajo y trabajo culminado respectivamente.

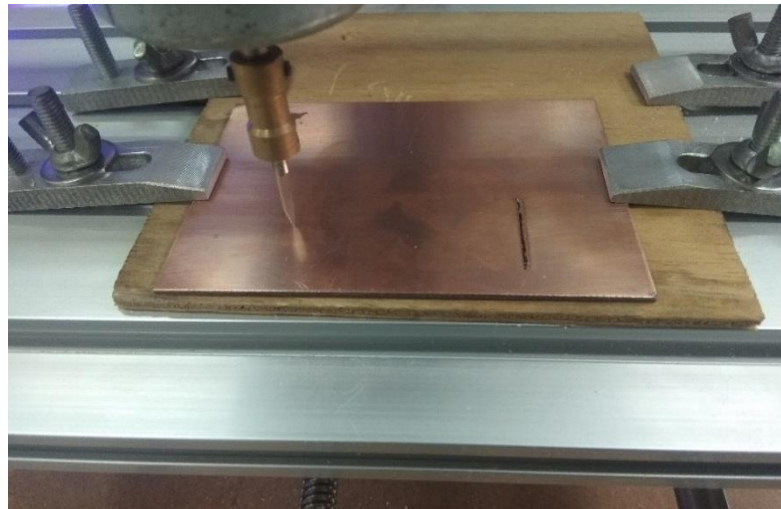


Figura 90 - Inicio de mecanizado

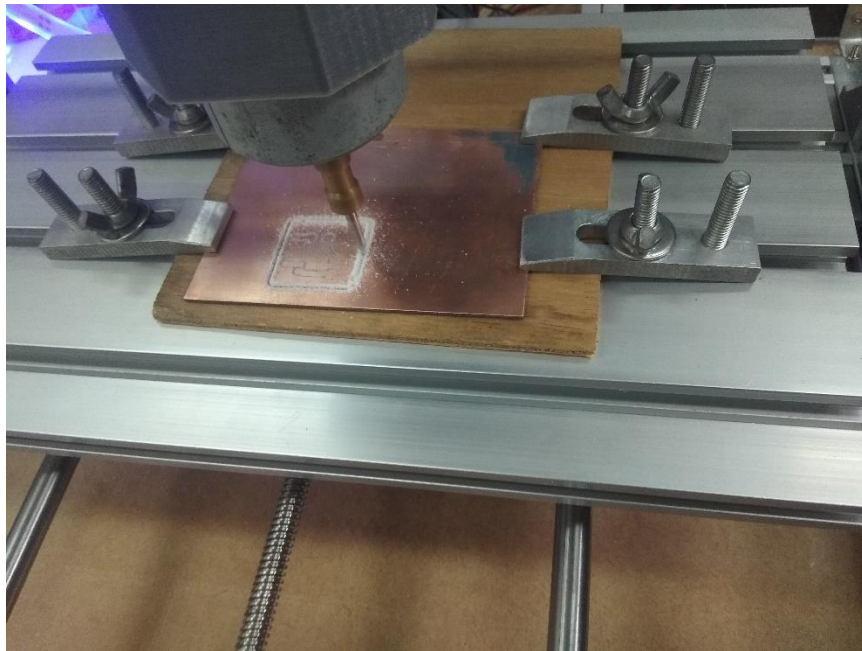


Figura 91 - Mecanizado mitad de trabajo, corte en el borde



Figura 92 - Corte final del prototipo



Figura 93 - Corte final

En la Figura 93 se observa al prototipo del PCB finalizado, se nota cierto error en los trazos eso corresponde a la calibración de la máquina, problema que será resuelto luego.



Figura 94 - Trabajo finalizado

CONCLUSIONES

El controlador principal al ser usado con programas de uso libre como GrblControl, es capaz de desarrollar una comunicación continua con el Gcode sender que puede ser la computadora de escritorio o la tarjeta de la interfaz manual implementada. Intercambiando información (estado) aun cuando no está trabajando.

El detalle del prototipado final depende mucho de la calibración de la máquina, ya que, las vibraciones causadas por el desplazamiento de la máquina sumado a los desajustes causan que se la trayectoria deseada sufra alteraciones. Así mismo, la velocidad de mecanizado que depende el material que se vaya a trabajar junto con el tipo de herramienta, son otros aspectos que influyen en el detalle que se obtiene en el trabajo terminado.

El uso del tipo de robot cartesiano en este proyecto y el uso de tornillo sin fin hace que la velocidad de trabajo para el uso del láser no sea optimo ya que con este tipo de mecanismo el desplazamiento es lento, aunque la tracción es directa al no usar cajas reductoras y tener el eje de desplazamiento directamente conectado al motor.

La interfaz manual fue implementada con un módulo comercial Arduino Nano, debido a la sencillez de las librerías para trabajar con periféricos como LCD y SD Card.

La interfaz manual se implementó el tipo de comunicación SPI, para reducir el número de pines en el microcontrolador de la interfaz manual y así aprovechar en base a la necesidad para el proyecto y por los demás periféricos que se conectan como la pantalla LCD. La tarjeta SD empleada es de tipo SDHC, probando con una tarjeta

SDXC no funcionó correctamente debido a la gran capacidad que estas tienen por lo cual le demandan gran trabajo al microcontrolador.

PROBLEMAS PRESENTADOS

Los archivos inicialmente mecanizados tienen un error en cuanto a algunas líneas, es un problema de calibración de los ejes mecánicos, en la Figura 95 se observa en más detalle estos problemas, como trazos no tan rectos.



Figura 95 - Trayectorias para corte en el borde

Calibración y ajuste

Algunos de los ejes de la máquina presentaban problemas lo que causaba errores en el desplazamiento y resultados no deseados, se realiza la respectiva calibración de los ejes y plataforma para luego hacer una nueva prueba y comprar los resultados. En la Figura 96 se muestra el ajuste de la altura de los rieles del eje X para que no haya desnivel y eso cause sobreesfuerzo en la herramienta, provocando así la curvatura forzada de los trazos lineales.

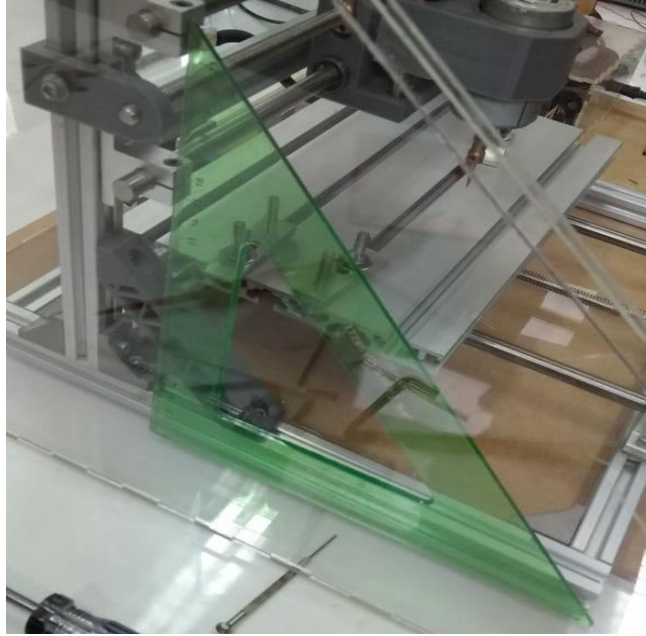


Figura 96 - Ajuste de altura de rieles del eje X

Para nivelar en el desplazamiento se fija la punta de la herramienta en cada extremo de la plataforma y así notar el desnivel y poder igualar. Tal como se muestra en la Figura 97 que se fija en el lado derecho y como se muestra en la Figura 98 en el lado izquierdo.

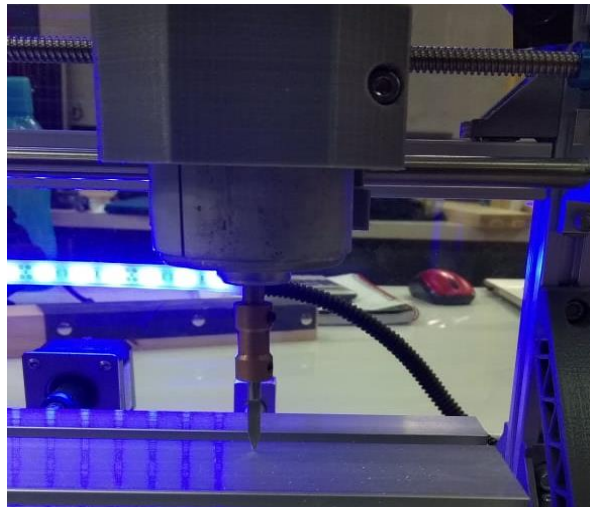


Figura 97 - Ajuste de altura en el lado derecho

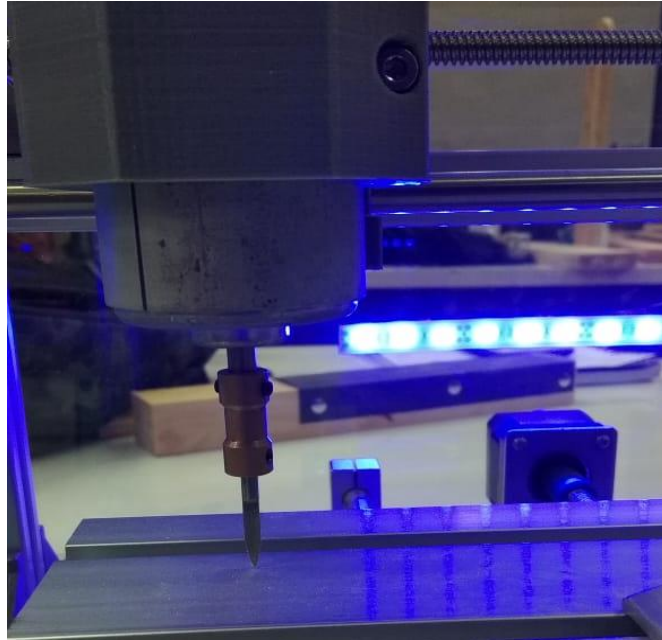


Figura 98 - Ajuste de altura en el lado izquierdo

Prueba final para comprobar los resultados de la calibración

Se ha realizado el diseño de un circuito impreso con diferentes dimensiones en los pads, y las pistas, como se muestra en la Figura 99.

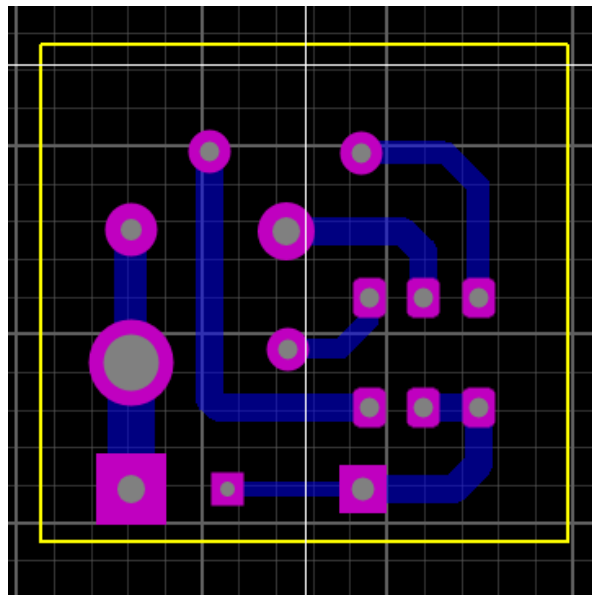


Figura 99 - Segunda Prueba

La generación del Gcode para la prueba se la realiza en el entorno Carbide con los parámetros que se muestran en la siguiente figura, penetrando hasta 0.1m en el desbaste del material, los parámetros descritos se muestran en la Figura 100, que representa la ventana de trabajo del software CAM.

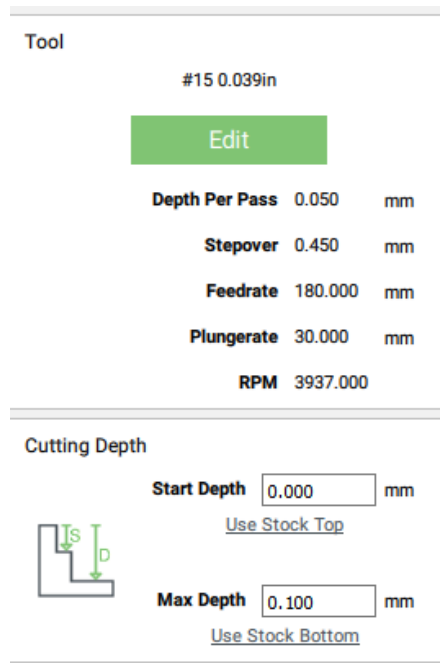


Figura 100 - Generación código G

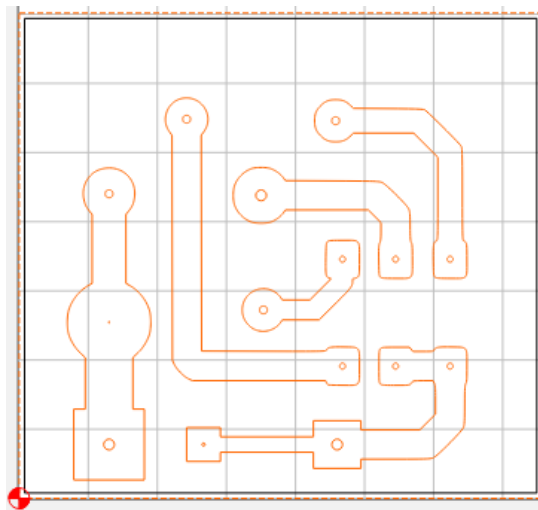


Figura 101 - Muestra las trayectorias generadas por el software CAM para el código G.

[G1 G54 G17 G21 G90 G94 M0 M3 M9 T0 F180. S3937.]

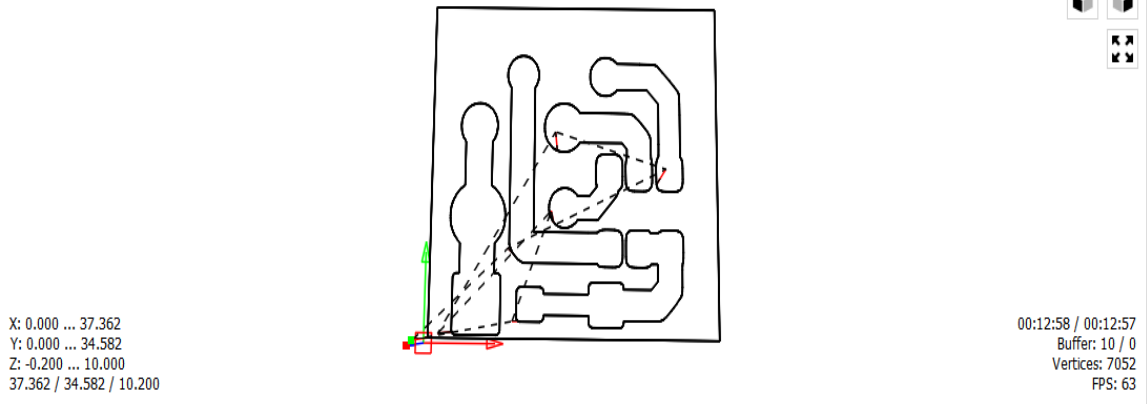


Figura 102 - Vista previa del trabajo a mecanizar

En la Figura 102 se muestra la vista previa desde el software mensajero de código G, y en la Figura 103 y Figura 104 se muestran respectivamente el inicio y el trabajo a mitad del proceso.

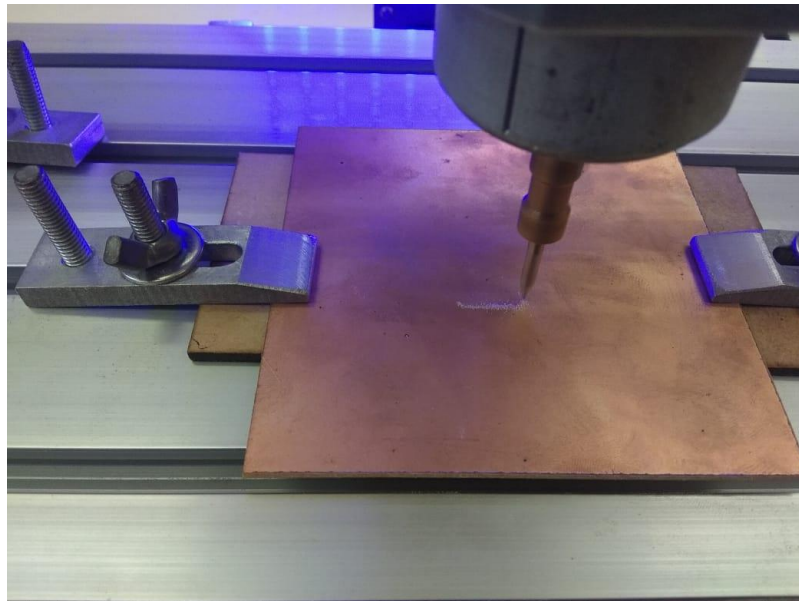


Figura 103 - Inicio de mecanizado

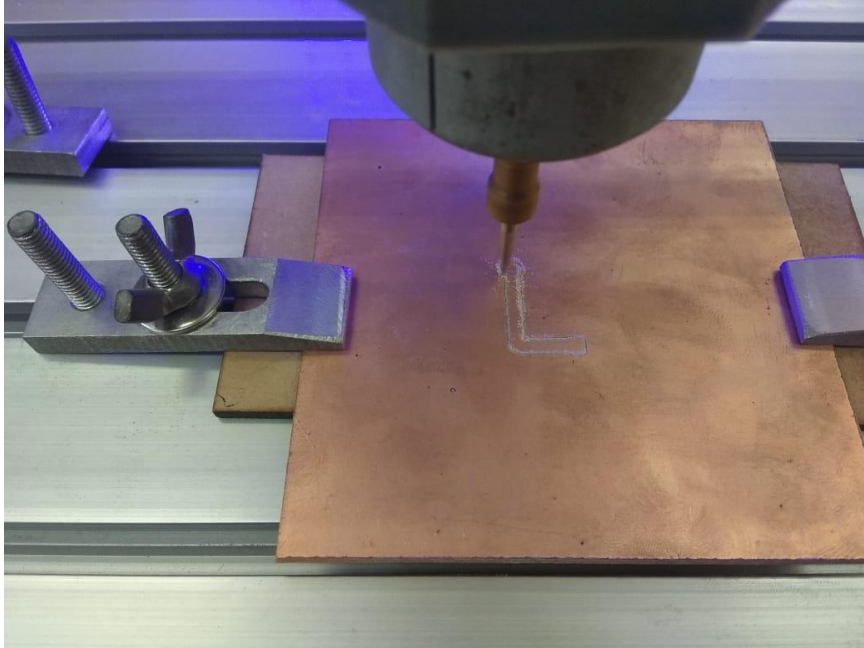


Figura 104 - Trabajo de mecanizado a mitad del proceso

Luego de mecanizar se tiene el trabajo final, luego de realizada la calibración, el trabajo final se muestra en la Figura 105.

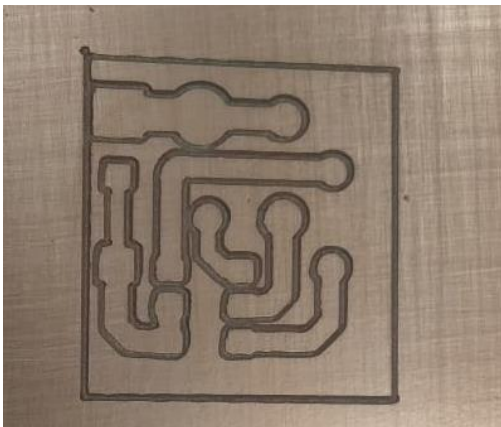


Figura 105 - Trabajo final luego de calibración

Una vez finalizado el trabajo final, comparamos los prototipos de cada caso y notamos una diferencia en el mecanizado, el prototipo de la derecha es de la primera prueba y el de la izquierda es luego de la calibración.

En la Figura 106 se muestra la comparación de los trabajos, a la derecha el trabajo de mecanizado sin calibrar los ejes de la máquina y en el lado izquierdo el trabajo luego de configurar dichos ejes.

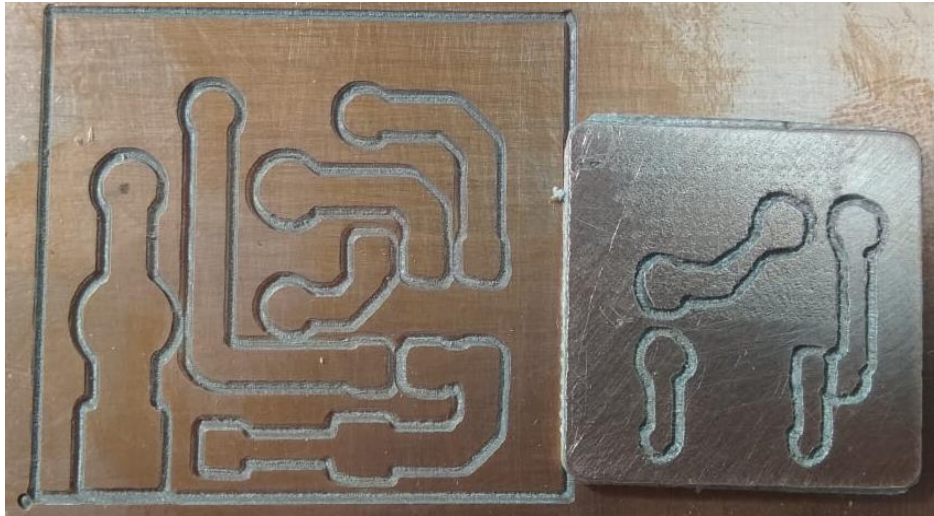


Figura 106 - Comparación de trabajos

Prototipo circuito estable

Para probar la funcionalidad de los circuitos impresos prototipados, se desarrolla un ejemplo simple, un oscilador estable. Tal como se muestra en el siguiente diagrama. Además de probar otras herramientas de generación de Gcode.

En la Figura 107 se muestra el diagrama esquemático que corresponde al circuito estable que se pretende elaborar, añadiendo máscara antisolder.

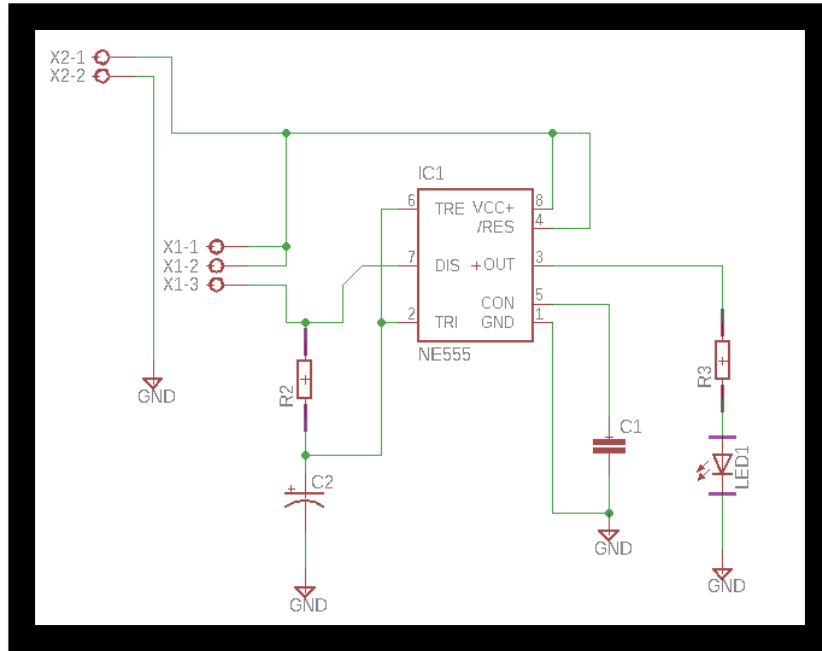


Figura 107 - Esquemático en Eagle CAD

Diseño del circuito impreso usando CAD Eagle.

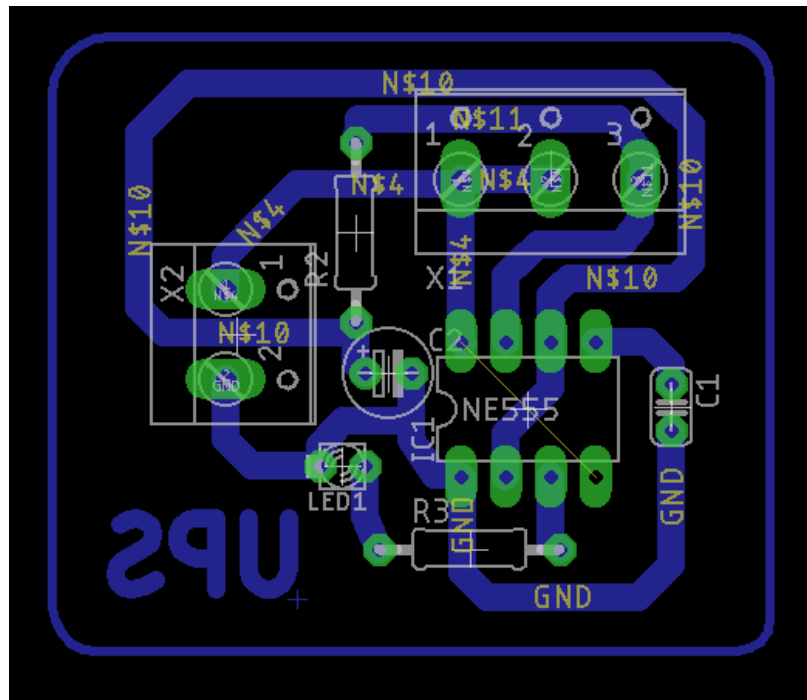


Figura 108 - PCboard diseñado en Eagle CAD

Se exportan el archivo Gerber, el vectorizado para uso de la herramienta CAM carbide copper, una herramienta libre. En la Figura 109 se muestra la ventana que permite exportar los archivos Gerber, que son los archivos vectoriales para luego generar el código G.

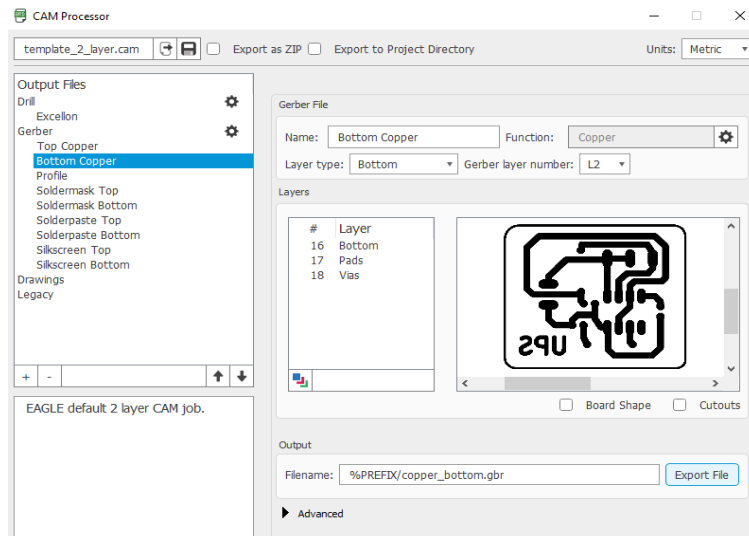


Figura 109 - Exporte de archivos Gerber

En la Figura 110 se muestra las trayectorias para generación de código G realizadas en el carbide copper

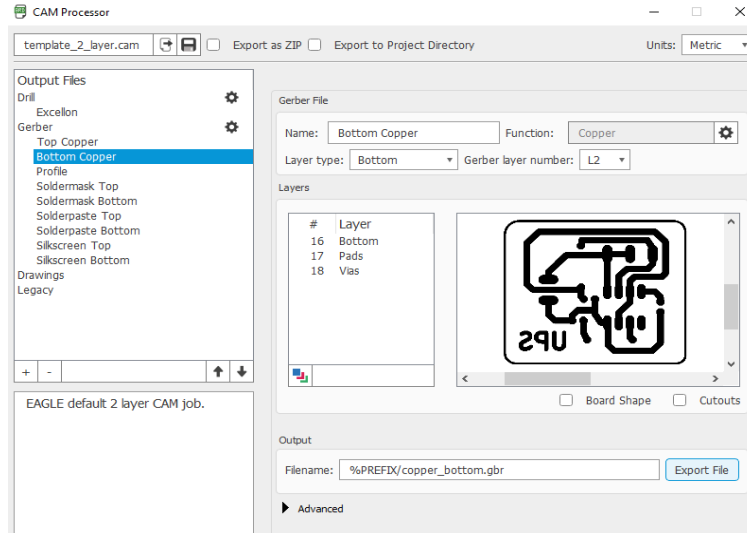


Figura 110 - Trayectorias de código G

En la Figura 111 se muestra el prototipado final del circuito diseñado.



Figura 111 - Prototipado final. Circuito estable.

Para darle un mejor acabado y presentación al circuito impreso, se le puede añadir una capa de máscara antisoldante, se muestra en la Figura 112, este polímero se solidifica usando luz UV, al no tener una cámara de exposición a luz UV, se ha optado por usar la radiación UV procedente del sol en el día, esto solo para mostrar la presentación que se puede dar al mejorar el prototipado final de los circuitos impresos

mecanizados ya que el proyecto no se centra en la elaboración de PCBS con acabado antisoldante.



Figura 112 - Máscara antisoldante

Se unta el circuito mecanizado con el polímero antisoldante y los puntos que deben soldarse deben ser untados con pintura acrílica de modo que bloquee el paso de luz UV hacia esas partes del antisolder y evitar solidificarse. Así al limpiarse quedarán expuestos esos pines a soldar, llevándose así de manera más eficaz el proceso de soldadura, tal como se muestra en la Figura 113.

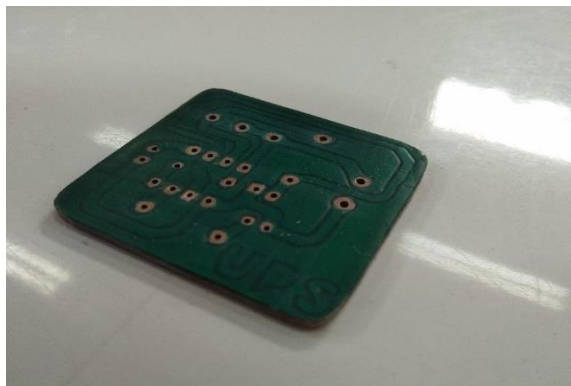


Figura 113 - Trayectorias para corte en el borde

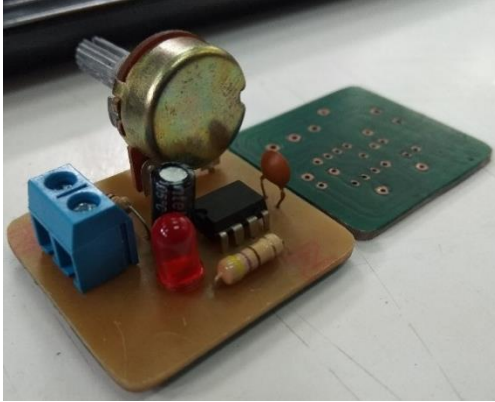


Figura 114 - Trayectorias para corte en el borde

En las Figura 114 se tiene el circuito impreso luego de solidificada la máscara antisolder y el circuito prototipado con los elementos soldados, el producto terminado.

RECOMENDACIONES

Sería conveniente que se desarrolle el proyecto, ampliando el área de trabajo o con algún nuevo mecanismo de desplazamiento que pudiera ser más rápido que de tornillo sin fin implementado en este proyecto.

Trabajar en el desarrollo de un sistema que pueda almacenar en la memoria el Gcode y evitar que el trabajo de la máquina se tire a perder por algún apagón u otra calamidad a la cual están expuestas las máquinas que trabajan autónomamente. Así también, hacer eso para que el dispositivo de almacenamiento masivo pueda retirarse.

Trabajar en el uso de otro tipo de dispositivo de almacenamiento para cargar los archivos desde la interfaz manual para reducir la carga de trabajo que el uso de la tarjeta de memoria demanda.

Acoplarle un sistema de extracción de residuos del material sería apropiado, ya que así se podría inspeccionar en diferentes instantes de tiempo el desarrollo del trabajo y así evitar manipular algún instrumento de limpieza cerca de la herramienta en pleno trabajo.

REFERENCIAS BIBLIOGRÁFICAS

Abril Carrillo, É., & Pacheco Pineda, A. (2012). *Análisis, diseño e implementación de un control de velocidad difuso aplicado al robot móvil Robotino*. Universidad Politécnica Salesiana. [En línea]. Available: <https://dspace.ups.edu.ec/handle/123456789/3587> [Último acceso: 08 13 2018].

Adrco. (2017). *Sistema DNC*. Obtenido de <https://www.adrco.com>.

Andino Páez, G. (2016). *Diseño y construcción de un alimentador automático de barras para un torno de Control Numérico Computarizado*. Universidad Politécnica Salesiana. [En línea]. Available: <https://dspace.ups.edu.ec/handle/123456789/13178>. [Último acceso: 08 13 2018].

Arpi Trujillo, J., & Cabrera Mendieta, D. (2013). *Implementación de un sistema de control para una máquina CNC láser*. Cuenca: Universidad Politécnica Salesiana. [En línea]. Available: <http://dspace.ups.edu.ec/handle/123456789/3351>. [Último acceso: 08 13 2018].

Arqhys. (2012). *Circuitos impresos*. Obtenido de <http://www.arqhys.com/contenidos/circuitos-impresos.html>.

Cipsa Circuits © . (2016). *CIRCUITOS MULTICAPA*. Obtenido de <http://www.cipsacircuits.com/portfolio-item/circuitos-multicapa/>.

Ecured. (2017). *Baquelita*. Obtenido de <https://www.ecured.cu/Baquelita>.

Electrosoft Ingeniería. (2010). *¿Qué es un circuito impreso?* Obtenido de <http://www.PCB.electrosoft.cl/04-articulos-circuitos-impresos-desarrollo-sistemas/01-conceptos-circuitos-impresos/conceptos-circuitos-impresos-PCB.html>.

Gonzalez, F., & Rosado, P. (2015). *Control numérico: marco y fundamentos*. Obtenido de https://gdocu.upv.es/alfresco/service/api/node/content/workspace/SpacesStore/e767993c-219f-4ed9-8925-2c8da327b5fe/TOC_0465_06_02.pdf?guest=true.

Google Sites. (2017). *Fabricación de PCB, con fresadora CNC*. Obtenido de <https://sites.google.com/site/controlofmachines/fabricacion-de-placas-de-circuito-impreso-con-una-fresadora-CNC>.

Gordon Sánchez, B., & Tacurí Romero, E. (2016). *Diseño e implementación de un módulo didáctico de domótica por medio de láminas con la tarjeta raspberry PI y el programa QT en la Universidad Politécnica Salesiana Sede Guayaquil*. Universidad Politécnica Salesiana.

Heidenhain. (2012). *TNC-Training*. Obtenido de <http://content.heidenhain.de/presentation/elearning/ES/index/1271254391253/1271254391255/1271254391256/1271254391256.html>.

Inca Valle, H., & Rengifo Muñoz, P. (2015). *Diseño e implementación de un sistema hmi para un prototipo de máquina fresadora utilizando control numérico computacional para elaborar placas de circuitos electrónicos*. UPS.

Jiménez, R. (2013). *Control Numérico por Computadora*. Obtenido de <http://materias.fi.uba.ar/7565/U4-control-numericopor-computadora.pdf>.

Jurado Valencia, E. A., & Salas Bohórquez, G. X. (2011). *Diseño y construcción de un módulo orientado al desarrollo de prácticas para la materia de control automático bajo la plataforma de matlab*. Universidad Politécnica Salesiana.

Mecanizados. (2015). *Introducción a la tecnología CNC*. Obtenido de <http://www.demaquinasyherramientas.com/mecanizado/introduccion-a-la-tecnologia-CNC>.

Medina, N. (2011). *OPTIMIZACIÓN DE TRAYECTORIAS PARA MÁQUINAS DE CONTROL NUMÉRICO MEDIANTE COLONIA DE HORMIGAS*. Obtenido de <http://www.repositoriodigital.ipn.mx/bitstream/123456789/15598/1/Medina%20Rodriguez%20Nataly.pdf>.

Mouser electronics. (2017). *Through Hole Placas de circuitos impresos (PCB) y tarjetas de prueba*. Obtenido de http://www.mouser.ec/Tools-Supplies/Prototyping-Products/PCBS-Breadboards/_/N-b11pq?P=1yzn9itZ1yyfiz9.

Peters, J. J. (2013). *Introducción al Control Numérico Computarizado (CNC)*. Obtenido de [http://wiki.ead.pucv.cl/index.php/Introducci%C3%B3n_al_control_num%C3%A9rico_computarizado_\(CNC\)](http://wiki.ead.pucv.cl/index.php/Introducci%C3%B3n_al_control_num%C3%A9rico_computarizado_(CNC)).

Rolón, P. (2011). *Tipos de CNC*. Obtenido de <https://sites.google.com/site/634mantenimiento/tipos-de-CNC>.

RS. (2010). *Adaptador RE965-04 QFN24*. Obtenido de es.rs-online.com.

Salas, R. (2008). *TÉCNICAS DE DISEÑO, DESARROLLO Y MONTAJE DE CIRCUITOS IMPRESOS*. Obtenido de http://www.saber.ula.ve/bitstream/123456789/26205/1/tecnicas_diseno.pdf.

Sánchez, S., & Villavicencio, H. (2004). *Diseño e implementación de una balanza electrónica en base al microcontrolador pic 16lf877*. Escuela Politécnica del Litoral. Facultad de Ingeniería en Electricidad y Computación. [En línea]. Available: <https://www.dspace.espol.edu.ec/handle/123456789/3005>. [Último acceso: 08 13 2018].

Sitenordeste. (2017). *Clasificación según el sistema de referencia*. Obtenido de http://www.sitenordeste.com/mecanica/control_numerico.htm#class.

Tecnología mecánica. (2017). *Control numérico*. Obtenido de http://www.sitenordeste.com/mecanica/control_numerico.htm#definicion.

Tene Asimbaya, P., & Ortiz Vaca, B. (2016). *Diseño e implementación de un módulo electrónico para mantenimiento preventivo y predictivo de los motores eléctricos de los Laboratorios de Electrónica de la Universidad Politécnica Salesiana - Sede Quito, Campus Sur, mediante análisis de vibración mecánica*. Quito: UPS.

Universidad del País Vasco. (2013). *Control Numérico y CAM*. Obtenido de http://www.ehu.eus/manufacturing/docencia/737_ca.pdf.

Universidad Nacional de la Plata. (2014). *Programación de control numérico*. Obtenido de <http://www.gulmi.com.ar/iso.pdf>.

Universidad Politécnica Salesiana. (2017). *Electrónica y Automatización*. Obtenido de <http://www.ups.edu.ec/electronica-y-automatizacion-quito>.

ANEXOS

Cronograma de actividades a desarrollar

Se estima un plazo de 8 meses para el desarrollo del proyecto de titulación.

Tabla 3 - Cronograma de actividades (Diagrama de Gantt)

Actividades	Meses							
	1	2	3	4	5	6	7	8
Recopilación de información	■							
Análisis de la información y selección de software		■						
Diseño y construcción de la estructura de la máquina			■	■				
Diseño y elaboración de sistemas embebidos de control			■	■	■			
Integración de software y hardware					■	■	■	
Pruebas de funcionamiento y correcciones						■	■	■
Elaboración del manual de operaciones	■							
Análisis, evaluación de resultados y pruebas finales							■	■
Presentación del proyecto								■

Presupuesto

El presupuesto destinado para este proyecto se detalla en la Tabla 4, será cubierto en su totalidad por los autores.

Tabla 4 - Presupuesto para el proyecto de titulación

Material	Cantidad	Precio	Subtotal
Pinza Amperimétrica	1	\$ 30.00	\$ 30.00
Estructura de la máquina	1	\$800.00	\$ 800.00
Fuente de poder	2	\$ 20.00	\$ 40.00
Motors PAP	4	\$100.00	\$ 400.00
Impresión de las tarjetas del sistema de control	1	\$200.00	\$ 200.00
Componentes electrónicos varios	1	\$320.00	\$ 320.00
Papelería	1	\$ 60.00	\$ 60.00
Material para imprimir la PCB	1	\$ 75.00	\$ 75.00
			\$ 1,925.00

Código

```
#define MAX_LINEAa_NUMBER 9999999

#define COMAND_EJES_NONE 0
#define COMAND_EJES_NN 1
#define COMAND_EJES_MODO_MOV 2
#define COMAND_EJES_DESP_HERRAMIENTA 3 //

#define NN_NO_ACTION 0 //
#define NN_delay 1 // G4
#define NN_coor_data 2 //
#define NN_GO_HOME_0 3 //
#define NN_SET_HOME_0 4 // .1
#define NN_GO_HOME_1 5 //
#define NN_SET_HOME_1 6 // .1
#define NN_ABSOLUTE_OVER_RIDE 7 //
#define NN_SET_COORDINATE_DESPLAZAMIENTO 8 //
#define NN_RESET_COORDINATE_DESPLAZAMIENTO 9 //

#define MODO_MOV_SEEK 0 /
#define MODO_MOV_LINEAAL 1 //
#define MODO_MOV_HORARIO_ARC 2 //
#define MODO_MOV_ANTIHORARIO_ARC 3 //
#define MODO_MOV_NONE 8 //
#define SELECC_PLANO_XY 0 //
#define SELECC_PLANO_ZX 1 //
#define SELECC_PLANO_YZ 2 //

#define MODO_ABSOLUTE 0 //
#define MODO_INCREMENTAL 1 //

#define MODO_ARCO_INCREMENTAL 0

#define SECUENCIA_PROG_RUNNING 0 // (
#define SECUENCIA_PROG_PAUSED 1 //
#define SECUENCIA_PROG_COMPLETED 2 //

#define FEEDRATE_MODE_UNIDADES_PER_MIN 0
#define FEEDRATE_MODE_INVERSE_TIME 1 /

//
#define UNIDADES_MM 0 //
#define UNIDADES_PULGADAS 1 //

#define COMPESACION_CORT_DISABLE 0 //

#define CONTROL_MOD 0 //
```

```

#define GIRO_OFF 0
#define GIRO_ON_HORARIO 1 //
#define GIRO_ON_ANTIHORARIO 2 //

```

```

#define DESP_HERRAMIENTA_CANCEL 0 //
#define DESP_HERRAMIENTA_ENABLE_DYNAMIC 1 //

```

```

Typedef struct {
    Uint8_t movimiento; // {
    Uint8_t FEEDRATE; //
    Uint8_t distance; //
    uint8_t SELECC_PLANO; //
    uint8_t SELECC_COORD;
    uint8_t SECUENCIA_PROG;
    Uint8_t GIRO; //

```

```

Typedef struct {
    Float f; //
    Float ijk [3]; //
    Uint8_t l; //
    Int32_t n; //
    float p; //
    // float q; /
    float r; //
    float s; // GIRO velocidad
    uint8_t t; //
    float xyz[3]; //
} valores_Gcode_t;

```

```

typedef struct {
    modo_Gcode_t MODO;

    float GIRO_velocidad; //
    uint8_t herramienta; //
    int32_t LINEAa_number; //

    float position[N_EJES]; //

    float sist_coordenadas[N_EJES]; //
    //
    float coord_desplazamiento[N_EJES]; //
    //
    float DESP_HERRAMIENTA; // T
} analisis_estado_t;

```



```

extern analisis_estado_t estado_Gcode;

typedef struct {

    uint8_t NN_COMANDO;
    modo_Gcode_t MODO;
    valores_Gcode_t NUMS;

} analisis_bloque;

extern analisis_bloque bloque_Gcode;

void init_Gcode();

uint8_t EJECUTAR_LINEAa_gc(char *LINEAa);

void sincr_posicion_gc();

// Declaracion Gcode estructuras
analisis_estado_t estado_Gcode;
analisis_bloque bloque_Gcode;

//0 es estado de LINEAa G code
void init_Gcode()
{
    memset(&estado_Gcode, 0, sizeof(analisis_estado_t));

    if
(!ajust_datos_leidos(estado_Gcode.MODO.SELECC_COORD,estado_Gcode.sist_coor
denadas))) {
        report_ESTADO_message(ESTADO_SETTING_READ_FAIL);
    }
}

static uint8_t check_pos_igual(float *pos_a, float *pos_b)
{
    uint8_t difx;
    for (difx=0; difx<N_EJES; difx++) {
        if (pos_a[difx] != pos_b[difx]) { return(false); }
    }
    return(true);
}

```

```

uint8_t EJECUTAR_LINEA_gc(char *LINEA)
{
    /* -----*/

    memset(&bloque_Gcode, 0, sizeof( analisis_bloque)); /

    memcpy(&bloque_Gcode.MODO,&estado_Gcode.MODO,sizeof(modos_Gcode_t));
    uint8_t COMAND_EJES = COMAND_EJES_NONE;
    uint8_t EJES_0, EJES_1, EJES_LINEAAL;
    uint8_t SELECC_COORD = 0; //
    float COORD_DATOS[N_EJES];
    float PARAMETROS_DATOS[N_EJES]; //
    uint8_t EJES_INSTRUCT = 0; //
    uint8_t ijk_INSTRUCT = 0; //

    uint16_t COMANDO_INSTRUCT = 0;
    uint16_t NUMS_INSTRUCT = 0;

    uint8_t INSTRUCT_BIT; //
    uint8_t CONT_CHARACTER = 0;
    char LETRA_Gcode;
    float NUMS;
    uint8_t NUMS_ENTERO = 0;
    uint16_t SUB_INSTRUCT = 0;

    while (LINEA[CONT_CHARACTER] != 0) {

        LETRA_Gcode = LINEA[CONT_CHARACTER];
        if((LETRA_Gcode < 'A') || (LETRA_Gcode > 'Z')) {
            FAIL(STATUS_EXPECTED_COMMAND_LETRA_Gcode); }
        CONT_CHARACTER++;
        if (!read_float(LINEA, &CONT_CHARACTER, &NUMS)) {
            FAIL(STATUS_BAD_NUMBER_FORMAT); }

        switch(LETRA_Gcode) {
            case 'G':
                // ETERMINA GRUPO G
                switch(NUMS_ENTERO) {
                    case 10: case 28: case 30: case 92:

                        if (SUB_INSTRUCT == 0) { // Ignore G28.1, G30.1, and G92.1
                            if (COMAND_EJES) {
                                FAIL(ESTADO_Gcode_COMAND_EJES_CONFLICT); } // [EJES
                                INSTRUCT/COMANDO conflict]
                                COMAND_EJES = COMAND_EJES_NN;

```

```

    }
    //
case 4: case 53:
    CODIG_BIT = GRUPO_G0;
    switch(NUMS_ENTERO) {
        case 4: bloque_Gcode.NN_COMANDO = NN_delay; break; // G4
        case 10: bloque_Gcode.NN_COMANDO = NN_coor_data; break; // G10
        case 28:
            switch(SUB_INSTRUCT) {
                case 0: bloque_Gcode.NN_COMANDO = NN_GO_HOME_0; break;
// G28
                case 10: bloque_Gcode.NN_COMANDO = NN_SET_HOME_0;
break; // G28.1
                default: FAIL(ESTADO_Gcode_NO_SOPORTA_COMANDO); //
[NO_SOPORTA G28.x COMANDO]
            }
            SUB_INSTRUCT = 0; //
            break;
        case 30:
            switch(SUB_INSTRUCT) {
                case 0: bloque_Gcode.NN_COMANDO = NN_GO_HOME_1; break;
// G30
                case 10: bloque_Gcode.NN_COMANDO = NN_SET_HOME_1;
break; // G30.1
                default: FAIL(ESTADO_Gcode_NO_SOPORTA_COMANDO); //
[NO_SOPORTA G30.x COMANDO]
            }
            SUB_INSTRUCT = 0; // cero para especificar que es un comandi no
entero G COMANDO.
            break;
        case 53: bloque_Gcode.NN_COMANDO =
NN_ABSOLUTE_OVER_RIDE; break; // G53
        case 92:
            switch(SUB_INSTRUCT) {
                case 0: bloque_Gcode.NN_COMANDO =
NN_SET_COORDINATE_DESPLAZAMIENTO; break; // G92
                case 10: bloque_Gcode.NN_COMANDO =
NN_RESET_COORDINATE_DESPLAZAMIENTO; break; // G92.1
                default: FAIL(ESTADO_Gcode_NO_SOPORTA_COMANDO); //
[NO_SOPORTA G92.x COMANDO]
            }
            SUB_INSTRUCT = 0;
            break;
        }
        break;
    case 0: case 1: case 2: case 3: case 38:

        if (COMAND_EJES) {
            FAIL(ESTADO_Gcode_COMAND_EJES_CONFLICT); } //
            COMAND_EJES = COMAND_EJES_MODO_MOV;

```

```

        case 80:
            CODIG_BIT = GRUPO_G1;
            switch(NUMS_ENTERO) {
                case 0: bloque_Gcode.MODO.movimiento = MODO_MOV_SEEK;
break; // G0
                case 1: bloque_Gcode.MODO.movimiento = MODO_MOV_LINEAAL;
break; // G1
                case 2: bloque_Gcode.MODO.movimiento =
MODO_MOV_HORARIO_ARC; break; // G2
                case 3: bloque_Gcode.MODO.movimiento =
MODO_MOV_ANTIHORARIO_ARC; break; // G3
                case 38:
                    switch(SUB_INSTRUCT) {
                        case 20: bloque_Gcode.MODO.movimiento =
MODO_MOV_PROBE_TOWARD; break; // G38.2
                        case 30: bloque_Gcode.MODO.movimiento =
MODO_MOV_PROBE_TOWARD_NO_ERROR; break; // G38.3
                        case 40: bloque_Gcode.MODO.movimiento =
MODO_MOV_PROBE_AWAY; break; // G38.4
                        case 50: bloque_Gcode.MODO.movimiento =
MODO_MOV_PROBE_AWAY_NO_ERROR; break; // G38.5
                        default: FAIL(ESTADO_Gcode_NO_SOPORTA_COMANDO); //
[NO_SOPORTA G38.x COMANDO]
                    }
                    SUB_INSTRUCT = 0; //
                    break;
                case 80: bloque_Gcode.MODO.movimiento = MODO_MOV_NONE;
break; // G80
            }
            break;
        case 17: case 18: case 19:
            CODIG_BIT = GRUPO_G2;
            switch(NUMS_ENTERO) {
                case 17: bloque_Gcode.MODO.SELECC_PLANO =
SELECC_PLANO_XY; break;
                case 18: bloque_Gcode.MODO.SELECC_PLANO =
SELECC_PLANO_ZX; break;
                case 19: bloque_Gcode.MODO.SELECC_PLANO =
SELECC_PLANO_YZ; break;
            }
            break;
        case 90: case 91:
            if (SUB_INSTRUCT == 0) {
                CODIG_BIT = GRUPO_G3;
                if (NUMS_ENTERO == 90) { bloque_Gcode.MODO.distance =
MODO_ABSOLUTE; } // G90
                else { bloque_Gcode.MODO.distance = MODO_INCREMENTAL; } //
G91
            } else {
                CODIG_BIT = GRUPO_G4;

```

```

        if ((SUB_INSTRUCT != 10) || (NUMS_ENTERO == 90)) {
FAIL(ESTADO_Gcode_NO_SOPORTA_COMANDO); } //
        SUB_INSTRUCT = 0;
    }
    break;
    case 93: case 94:
        CODIG_BIT = GRUPO_G5;
        if (NUMS_ENTERO == 93) { bloque_Gcode.MODO.FEEDRATE =
FEEDRATE_MODE_INVERSE_TIME; } // G93
        else { bloque_Gcode.MODO.FEEDRATE =
FEEDRATE_MODE_unidades_PER_MIN; } // G94
        break;
    case 20: case 21:
        CODIG_BIT = GRUPO_G6;
        if (NUMS_ENTERO == 20) { bloque_Gcode.MODO.unidades =
UNIDADES_PULGADAS; } // G20
        else { bloque_Gcode.MODO.unidades = UNIDADES_MM; } // G21
        break;
    case 40:
        CODIG_BIT = GRUPO_G7;
        break;
    case 43: case 49:
        CODIG_BIT = GRUPO_G8;
        if (COMAND_EJES) { FAIL(ESTADO_Gcode_COMAND_EJES_CONFLICT); } //
        COMAND_EJES = COMAND_EJES_DESP_HERRAMIENTA;
        if (NUMS_ENTERO == 49) { // G49
            bloque_Gcode.MODO.despl_herramienta =
DESP_HERRAMIENTA_CANCEL;
        } else if (SUB_INSTRUCT == 10) { // G43.1
            bloque_Gcode.MODO.despl_herramienta =
DESP_HERRAMIENTA_ENABLE_DYNAMIC;
        } else { FAIL(ESTADO_Gcode_NO_SOPORTA_COMANDO); } //
        SUB_INSTRUCT = 0; //
        break;
    case 54: case 55: case 56: case 57: case 58: case 59:
        /
        CODIG_BIT = GRUPO_G12;
        bloque_Gcode.MODO.SELECC_COORD = NUMS_ENTERO-54; //
        break;
    case 61:
        CODIG_BIT = GRUPO_G13;
        if (SUB_INSTRUCT != 0) {
FAIL(ESTADO_Gcode_NO_SOPORTA_COMANDO); } //

        break;
        default: FAIL(ESTADO_Gcode_NO_SOPORTA_COMANDO); //
    }
    if (SUB_INSTRUCT > 0) {
FAIL(ESTADO_Gcode_COMANDO_NUMS_NOT_INTEGER); } //

```

```

        if ( BIT_istrue(COMANDO_INSTRUCT,BIT(CODIG_BIT)) ) {
FAIL(ESTADO_Gcode_GRUPO_VIOLATION); }
        COMANDO_INSTRUCT |= BIT(CODIG_BIT);
        break;

        case 'M':

                if (SUB_INSTRUCT > 0) {
FAIL(ESTADO_Gcode_COMANDO_NUMS_NOT_INTEGER); } //
                switch(NUMS_ENTERO) {
                        case 0: case 1: case 2: case 30:
                                CODIG_BIT = GRUPO_M4;
                                switch(NUMS_ENTERO) {
                                        case 0: bloque_Gcode.MODO.SECUENCIA_PROG =
SECUENCIA_PROG_PAUSED; break; //
                                        case 1: break; //
                                        case 2: case 30: bloque_Gcode.MODO.SECUENCIA_PROG =
SECUENCIA_PROG_COMPLETED; break; //
                                }
                                break;
                                #ifndef USE_GIRO_DIR_AS_ENABLE_PIN
                                        case 4:
                                #endif
                                case 3: case 5:
                                        CODIG_BIT = GRUPO_M7;
                                        switch(NUMS_ENTERO) {
                                                case 3: bloque_Gcode.MODO.GIRO = GIRO_ON_HORARIO; break;
                                                #ifndef USE_GIRO_DIR_AS_ENABLE_PIN
                                                case 4: bloque_Gcode.MODO.GIRO = GIRO_ON_ANTIHORARIO;
break;
                                                #endif
                                                case 5: bloque_Gcode.MODO.GIRO = GIRO_OFF; break;
                                        }
                                break;
                                #ifdef ENABLE_M7
                                        case 7:
                                #endif
                                        case 8: case 9:

                                                break;
                                                default: FAIL(ESTADO_Gcode_NO_SOPORTA_COMANDO); //
[NO_SOPORTA M COMANDO]
                                }

                if ( BIT_istrue(COMANDO_INSTRUCT,BIT(CODIG_BIT)) ) {
FAIL(ESTADO_Gcode_GRUPO_VIOLATION); }
                COMANDO_INSTRUCT |= BIT(CODIG_BIT);
                break;

```

```

default:

switch(LETRA_Gcode){

    case 'F': CODIG_BIT = CODIG_F; bloque_Gcode.NUMS.f = NUMS; break;

    case 'I': CODIG_BIT = CODIG_I; bloque_Gcode.NUMS.ijk[X_EJES] =
NUMS; ijk_INSTRUCT |= (1<<X_EJES); break;
    case 'J': CODIG_BIT = CODIG_J; bloque_Gcode.NUMS.ijk[Y_EJES] =
NUMS; ijk_INSTRUCT |= (1<<Y_EJES); break;
    case 'K': CODIG_BIT = CODIG_K; bloque_Gcode.NUMS.ijk[Z_EJES] =
NUMS; ijk_INSTRUCT |= (1<<Z_EJES); break;
    case 'L': CODIG_BIT = CODIG_L; bloque_Gcode.NUMS.l =
NUMS_ENTERO; break;
    case 'N': CODIG_BIT = CODIG_N; bloque_Gcode.NUMS.n =
trunc(NUMS); break;
    case 'P': CODIG_BIT = CODIG_P; bloque_Gcode.NUMS.p = NUMS;
break;

    case 'R': CODIG_BIT = CODIG_R; bloque_Gcode.NUMS.r = NUMS;
break;
    case 'S': CODIG_BIT = CODIG_S; bloque_Gcode.NUMS.s = NUMS;
break;
    case 'T': CODIG_BIT = CODIG_T; break; // gc.NUMS.t =
NUMS_ENTERO;
    case 'X': CODIG_BIT = CODIG_X; bloque_Gcode.NUMS.xyz[X_EJES] =
NUMS; EJES_INSTRUCT |= (1<<X_EJES); break;
    case 'Y': CODIG_BIT = CODIG_Y; bloque_Gcode.NUMS.xyz[Y_EJES] =
NUMS; EJES_INSTRUCT |= (1<<Y_EJES); break;
    case 'Z': CODIG_BIT = CODIG_Z; bloque_Gcode.NUMS.xyz[Z_EJES] =
NUMS; EJES_INSTRUCT |= (1<<Z_EJES); break;
    default: FAIL(ESTADO_Gcode_NO_SOPORTA_COMANDO);
}

if (BIT_istrue(NUMS_INSTRUCT,BIT(CODIG_BIT))) {
FAIL(ESTADO_Gcode_CODIG_REPETIDO); } // [INSTRUCT REPETIDO]

if ( ( BIT(CODIG_BIT) &
(BIT(CODIG_F)|BIT(CODIG_N)|BIT(CODIG_P)|BIT(CODIG_T)|BIT(CODIG_S)) ) {
if (NUMS < 0.0) { FAIL(ESTADO_NEGATIVO_NUMS); }
NUMS_INSTRUCT |= BIT(CODIG_BIT);
}
}

if (EJES_INSTRUCT) {
if (!COMAND_EJES) { COMAND_EJES = COMAND_EJES_MODALIDAD; } //

```

```

    }

    if (BIT_istrue(NUMS_INSTRUCT,BIT(CODIG_N))) {

        if (bloque_Gcode.NUMS.n > MAX_LINEAa_NUMBER) {
            FAIL(ESTADO_Gcode_INVALID_LINEAa_NUMBER); } //
        }

        if (bloque_Gcode.MODO.FEEDRATE == FEEDRATE_MODE_INVERSE_TIME)
        { // = G93
            if (COMAND_EJES == COMAND_EJES_MODO_MOV) {
                if ((bloque_Gcode.MODO.movimiento != MODO_MOV_NONE) ||
                    (bloque_Gcode.MODO.movimiento != MODO_MOV_SEEK)) {
                    if (BIT_isfalse(NUMS_INSTRUCT,BIT(CODIG_F))) {
                        FAIL(ESTADO_Gcode_UNDEFINED_FEEDRATE); } // [F INSTRUCT missing]
                    }
                }
            } else { // = G94
                if (estado_Gcode.MODO.FEEDRATE ==
                    FEEDRATE_MODE_unidades_PER_MIN) { // Last state is also G94
                    if (BIT_istrue(NUMS_INSTRUCT,BIT(CODIG_F))) {
                        if (bloque_Gcode.MODO.unidades == UNIDADES_PULGADAS) {
                            bloque_Gcode.NUMS.f *= MM_PER_INCH; }
                        } else {
                            bloque_Gcode.NUMS.f = estado_Gcode.FEEDRATE; // Push last state feed
                            rate
                        }
                    }
                }
            }

            if (BIT_isfalse(NUMS_INSTRUCT,BIT(CODIG_S))) { bloque_Gcode.NUMS.s =
                estado_Gcode.GIRO_velocidad; }

            if (bloque_Gcode.NN_COMANDO == NN_delay) {
                if (BIT_isfalse(NUMS_INSTRUCT,BIT(CODIG_P))) {
                    FAIL(ESTADO_Gcode_NUMS_CODIG_MISSING); } // [P INSTRUCT missing]
                    BIT_false(NUMS_INSTRUCT,BIT(CODIG_P));
                }
            }

            switch (bloque_Gcode.MODO.SELECC_PLANO) {
                case SELECC_PLANO_XY:
                    EJES_0 = X_EJES;
                    EJES_1 = Y_EJES;
                    EJES_LINEAAL = Z_EJES;
                    break;
                case SELECC_PLANO_ZX:
                    EJES_0 = Z_EJES;
                    EJES_1 = X_EJES;
            }

```



```

    EJES_LINEAAL = Y_EJES;
    break;
default: // case SELECC_PLANO_YZ:
    EJES_0 = Y_EJES;
    EJES_1 = Z_EJES;
    EJES_LINEAAL = X_EJES;
}

uint8_t difx;
if (bloque_Gcode.MODO.unidades == UNIDADES_PULGADAS) {
    for (difx=0; difx<N_EJES; difx++) { // Axes indices are consistent, so loop may
be used.
        if (BIT_istrue(EJES_INSTRUCT,BIT(difx)) ) {
            bloque_Gcode.NUMS.xyz[difx] *= MM_PER_INCH;
        }
    }
}

    if (COMAND_EJES == COMAND_EJES_DESP_HERRAMIENTA ) { // Indicates
called in block.
        if (bloque_Gcode.MODO.despl_herramienta ==
DESP_HERRAMIENTA_ENABLE_DYNAMIC) {
            if (EJES_INSTRUCT ^ (1<<DESP_HERRAMIENTA_EJES)) {
FAIL(ESTADO_Gcode_G43_DYNAMIC_EJES_ERROR); }
        }
    }

memcpy(COORD_DATOS,estado_Gcode.sist_coordenadas,sizeof(estado_Gcode.sist_
coordenadas));
    if ( BIT_istrue(COMANDO_INSTRUCT,BIT(GRUPO_G12)) ) { //
        if (bloque_Gcode.MODO.SELECC_COORD > N_COORDINATE_SYSTEM) {
FAIL(ESTADO_Gcode_NO_SOPORTA_COORD_SYS); } // [Gr
            if (estado_Gcode.MODO.SELECC_COORD !=
bloque_Gcode.MODO.SELECC_COORD) {
                if
(!ajust_datos_leidos(bloque_Gcode.MODO.SELECC_COORD,COORD_DATOS))) {
FAIL(ESTADO_SETTING_READ_FAIL); }
            }
        }

    switch (bloque_Gcode.NN_COMANDO) {
        case NN_coor_data:
            if (!EJES_INSTRUCT) { FAIL(ESTADO_Gcode_NO_EJES_INSTRUCT) }; //
[No EJES INSTRUCT]
            if (BIT_isfalse(NUMS_INSTRUCT,((1<<CODIG_P)|(1<<CODIG_L)))) {
FAIL(ESTADO_Gcode_NUMS_CODIG_MISSING); } // [P/L INS
                SELECC_COORD = trunc(bloque_Gcode.NUMS.p); // Convert p NUMS to
int.

```

```

        if (SELECC_COORD > N_COORDINATE_SYSTEM) {
FAIL(ESTADO_Gcode_NO_SOPORTA_COORD_SYS); } // [Gr
        if (bloque_Gcode.NUMS.l != 20) {
            if (bloque_Gcode.NUMS.l == 2) {
                if (BIT_istrue(NUMS_INSTRUCT,BIT(CODIG_R))) {
FAIL(ESTADO_Gcode_NO_SOPORTA_COMANDO); } // [G10 L2 R not supported]
                } else { FAIL(ESTADO_Gcode_NO_SOPORTA_COMANDO); } //
[NO_SOPORTA L]
            }
            BIT_false(NUMS_INSTRUCT,(BIT(CODIG_L)|BIT(CODIG_P)));

        if (SELECC_COORD > 0) { SELECC_COORD--; } //
        else { SELECC_COORD = bloque_Gcode.MODO.SELECC_COORD; } //

        for (difx=0; difx<N_EJES; difx++) { //
            PARAMETROS_DATOS[difx] = estado_Gcode.position[difx]-
estado_Gcode.coord_desplazamiento[difx]-bloque_Gcode.NUMS.xyz[difx];
            if (difx == DESP_HERRAMIENTA_EJES) { PARAMETROS_DATOS[difx]
-= estado_Gcode.DESP_HERRAMIENTA; }
            } else {
                PARAMETROS_DATOS[difx] = bloque_Gcode.NUMS.xyz[difx];
            }
        }
        }
        break;
    case NN_SET_COORDINATE_DESPLAZAMIENTO:
        if (!EJES_INSTRUCT) { FAIL(ESTADO_Gcode_NO_EJES_INSTRUCT); } //
[No EJES INSTRUCT]

        for (difx=0; difx<N_EJES; difx++) { // Axes indices are consistent, so loop may
be used.
            if (BIT_istrue(EJES_INSTRUCT,BIT(difx)) ) {
                bloque_Gcode.NUMS.xyz[difx] = estado_Gcode.position[difx]-
COORD_DATOS[difx]-bloque_Gcode.NUMS.xyz[difx];
                if (difx == DESP_HERRAMIENTA_EJES) { bloque_Gcode.NUMS.xyz[difx]
-= estado_Gcode.DESP_HERRAMIENTA; }
                } else {
                    bloque_Gcode.NUMS.xyz[difx] =
estado_Gcode.coord_desplazamiento[difx];
                }
            }
        }
        break;

    default:
        if (COMAND_EJES != COMAND_EJES_DESP_HERRAMIENTA ) { //
            if (EJES_INSTRUCT) {
                for (difx=0; difx<N_EJES; difx++) { //
                    if ( BIT_istrue(EJES_INSTRUCT,BIT(difx)) ) {
                        bloque_Gcode.NUMS.xyz[difx] = estado_Gcode.position[difx]; //
                    } else {
                        if (bloque_Gcode.NN_COMANDO != NN_ABSOLUTE_OVER_RIDE) {

```

```

        if (bloque_Gcode.MODO.distance == MODO_ABSOLUTE) {
            bloque_Gcode.NUMS.xyz[difx] += COORD_DATOS[difx] +
estado_Gcode.coord_desplazamiento[difx];
            if (difx == DESP_HERRAMIENTA_EJES) {
bloque_Gcode.NUMS.xyz[difx] += estado_Gcode.DESP_HERRAMIENTA; }
            } else { // Incremental mode
                bloque_Gcode.NUMS.xyz[difx] += estado_Gcode.position[difx];
            }
        }
    }
}
}
}
}

switch (bloque_Gcode.NN_COMANDO) {
    case NN_GO_HOME_0:

        if (!EJES_INSTRUCT) { COMAND_EJES = COMAND_EJES_NONE; } //
        if (!ajust_datos_leidos(SETTING_INDEX_G28,PARAMETROS_DATOS)) {
FAIL(ESTADO_SETTING_READ_FAIL); }
        break;
    case NN_GO_HOME_1:

        if (!EJES_INSTRUCT) { COMAND_EJES = COMAND_EJES_NONE; } //
        if (!ajust_datos_leidos(SETTING_INDEX_G30,PARAMETROS_DATOS)) {
FAIL(ESTADO_SETTING_READ_FAIL); }
        break;
    case NN_SET_HOME_0: case NN_SET_HOME_1:

        break;
    case NN_RESET_COORDINATE_DESPLAZAMIENTO:

        break;
    case NN_ABSOLUTE_OVER_RIDE:

        if (!(bloque_Gcode.MODO.movimiento == MODO_MOV_SEEK ||
bloque_Gcode.MODO.movimiento == MODO_MOV_LINEAAL)) {
            FAIL(ESTADO_Gcode_G53_INVALID_MODAL);
        }
        break;
    }
}

if (bloque_Gcode.MODO.movimiento == MODO_MOV_NONE) {

    if ((EJES_INSTRUCT) && (COMAND_EJES != COMAND_EJES_NN)) {
        FAIL(ESTADO_Gcode_EJES_INSTRUCT_EXISTENTE); //
    }
}

```

```

} else if ( COMAND_EJES == COMAND_EJES_MODO_MOV ) {
    if (bloque_Gcode.MODO.movimiento == MODO_MOV_SEEK) {
        if (!EJES_INSTRUCT) { COMAND_EJES = COMAND_EJES_NONE; }

    } else {
        if (bloque_Gcode.NUMS.f == 0.0) {
            FAIL(ESTADO_Gcode_UNDEFINED_FEEDRATE); }

        switch (bloque_Gcode.MODO.movimiento) {
            case MODO_MOV_LINEAAL:
                if (!EJES_INSTRUCT) { COMAND_EJES = COMAND_EJES_NONE; }

                break;
            case MODO_MOV_HORARIO_ARC:
            case MODO_MOV_ANTIHORARIO_ARC:
                if (!EJES_INSTRUCT) { FAIL(ESTADO_Gcode_NO_EJES_INSTRUCT); }
                if (!(EJES_INSTRUCT & (BIT(EJES_0)|BIT(EJES_1)))) {
                    FAIL(ESTADO_Gcode_NO_EJES_INSTRUCT_IN_PLANE); } // [No EJES INSTRUCT in
                    plane]

                float x,y;
                x = bloque_Gcode.NUMS.xyz[EJES_0]-estado_Gcode.position[EJES_0];
                y = bloque_Gcode.NUMS.xyz[EJES_1]-estado_Gcode.position[EJES_1];

                if (NUMS_INSTRUCT & BIT(CODIG_R)) {
                    BIT_false(NUMS_INSTRUCT,BIT(CODIG_R));
                    if (check_pos_igual(estado_Gcode.position, bloque_Gcode.NUMS.xyz)) {
                        FAIL(ESTADO_Gcode_INVALID_TARGET); } //

                if (bloque_Gcode.MODO.unidades == UNIDADES_PULGADAS) {
                    bloque_Gcode.NUMS.r *= MM_PER_INCH; }

                float h_x2_div_d = 4.0 * bloque_Gcode.NUMS.r*bloque_Gcode.NUMS.r -
                x*x - y*y;

                if (h_x2_div_d < 0) { FAIL(ESTADO_Gcode_ARC_RADIUS_ERROR); }

                h_x2_div_d = -sqrt(h_x2_div_d)/hypot_f(x,y);
                if (bloque_Gcode.MODO.movimiento ==
                    MODO_MOV_ANTIHORARIO_ARC) { h_x2_div_d = -h_x2_div_d; }

```

```
estado_Gcode.LINEAa_number = bloque_Gcode.NUMS.n;
```

```
estado_Gcode.MODO.FEEDRATE = bloque_Gcode.MODO.FEEDRATE;
```

Always copy this NUMS. See feed rate error-checking.

```
if (estado_Gcode.GIRO_velocidad != bloque_Gcode.NUMS.s) {  
  if (estado_Gcode.MODO.GIRO != GIRO_OFF) {  
    GIRO_run(estado_Gcode.MODO.GIRO, bloque_Gcode.NUMS.s); }  
    estado_Gcode.GIRO_velocidad = bloque_Gcode.NUMS.s;  
  }  
}
```

```
estado_Gcode.herramienta = bloque_Gcode.NUMS.t;
```

```
if (estado_Gcode.MODO.GIRO != bloque_Gcode.MODO.GIRO) {  
  GIRO_run (bloque_Gcode.MODO.GIRO, estado_Gcode.GIRO_velocidad);  
  estado_Gcode.MODO.GIRO = bloque_Gcode.MODO.GIRO;  
}
```

```
if (bloque_Gcode.NN_COMANDO == NN_delay) {  
  mc_delay(bloque_Gcode.NUMS.p); }  
}
```

```
estado_Gcode.MODO.SELECC_PLANO =  
bloque_Gcode.MODO.SELECC_PLANO;
```

```
estado_Gcode.MODO.unidades = bloque_Gcode.MODO.unidades;
```

```
If (COMAND_EJES == COMAND_EJES_DESP_HERRAMIENTA) { // I  
  estado_Gcode.MODO.despl_herramienta =  
bloque_Gcode.MODO.despl_herramienta;  
  If (estado_Gcode.MODO.despl_herramienta ==  
DESP_HERRAMIENTA_ENABLE_DYNAMIC) { // G43.1  
    estado_Gcode.DESP_HERRAMIENTA = bloque_Gcode.NUMS.xyz  
[DESP_HERRAMIENTA_EJES];  
  } Else { // G49  
    estado_Gcode.DESP_HERRAMIENTA = 0.0;  
  }  
}
```

```

        if (estado_Gcode.MODO.SELECC_COORD!=
bloque_Gcode.MODO.SELECC_COORD) {
            estado_Gcode.MODO.SELECC_COORD =
bloque_Gcode.MODO.SELECC_COORD;
            memcpy (estado_Gcode. Sist_coordenadas, COORD_DATOS, sizeof
(COORD_DATOS));
        }

        estado_Gcode.MODO.distance = bloque_Gcode.MODO.distance;

        switch(bloque_Gcode.NN_COMANDO) {
            case NN_coor_data:
                settings_write_coord_data(SELECC_COORD,PARAMETROS_DATOS);

                if (estado_Gcode.MODO.SELECC_COORD == SELECC_COORD) {
                    memcpy(estado_Gcode.sist_coordenadas,PARAMETROS_DATOS,sizeof(PARAMETR
OS_DATOS)); }
                    break;
                Case NN_GO_HOME_0: case NN_GO_HOME_1:

                    if (COMAND_EJES) {
                        #ifdef USE_LINEAa_NUMBERS
                            mc_LINEAa(bloque_Gcode.NUMS.xyz, -1.0, false,
estado_Gcode.LINEAa_number);
                        #else
                            mc_LINEAa (bloque_Gcode.NUMS.xyz, -1.0, false);
                        #endif
                    }
                    #ifdef USE_LINEAa_NUMBERS
                        mc_LINEAa(PARAMETROS_DATOS, -1.0, false,
estado_Gcode.LINEAa_number);
                    #else
                        mc_LINEAa (PARAMETROS_DATOS, -1.0, false);
                    #endif
                    memcpy(estado_Gcode.position, PARAMETROS_DATOS,
sizeof(PARAMETROS_DATOS));
                    break;
                case NN_SET_HOME_0:
                    settings_write_coord_data(SETTING_INDEX_G28,estado_Gcode.position);
                    break;
                case NN_SET_HOME_1:
                    settings_write_coord_data (SETTING_INDEX_G30, estado_Gcode.position);
                    break;
                case NN_SET_COORDINATE_DESPLAZAMIENTO:
                    memcpy (estado_Gcode.coord_desplazamiento, bloque_Gcode.NUMS.xyz,
sizeof (bloque_Gcode.NUMS.xyz));
                    Break;
                Case NN_RESET_COORDINATE_DESPLAZAMIENTO:

```

```

        LIMPIAR_vector (estado_Gcode.coord_desplazamiento); // Disable G92
        DESPLAZAMIENTOs by zeroing DESPLAZAMIENTO vector.
        Break;
    }

```

```

        estado_Gcode.MODO.movimiento = bloque_Gcode.MODO.movimiento;
    If (estado_Gcode.MODO.movimiento!= MODO_MOV_NONE) {
        If (COMAND_EJES == COMAND_EJES_MODO_MOV) {
            switch (estado_Gcode.MODO.movimiento) {
                case MODO_MOV_SEEK:
                    #ifdef USE_LINEAa_NUMBERS
                        mc_LINEAa (bloque_Gcode.NUMS.xyz, -1.0, false,
estado_Gcode.LINEAa_number);
                    #else
                        mc_LINEAa (bloque_Gcode.NUMS.xyz, -1.0, false);
                    #endif
                    Break;
                Case MODO_MOV_LINEAAL:
                    #ifdef USE_LINEAa_NUMBERS
                        mc_LINEAa (bloque_Gcode.NUMS.xyz, estado_Gcode.FEEDRATE,
estado_Gcode.MODO.FEEDRATE, estado_Gcode.LINEAa_number);
                    #else
                        mc_LINEAa (bloque_Gcode.NUMS.xyz, estado_Gcode.FEEDRATE,
estado_Gcode.MODO.FEEDRATE);
                    #endif
                    Break;
                Case MODO_MOV_HORARIO_ARC:
                    #ifdef USE_LINEAa_NUMBERS
                        mc_arc (estado_Gcode.position, bloque_Gcode.NUMS.xyz,
bloque_Gcode.NUMS.ijk, bloque_Gcode.NUMS.r,
estado_Gcode.FEEDRATE, estado_Gcode.MODO.FEEDRATE,
EJES_0, EJES_1, EJES_LINEAAL, true, estado_Gcode.LINEAa_number);
                    #else
                        mc_arc (estado_Gcode.position, bloque_Gcode.NUMS.xyz,
bloque_Gcode.NUMS.ijk, bloque_Gcode.NUMS.r,
estado_Gcode.FEEDRATE, estado_Gcode.MODO.FEEDRATE,
EJES_0, EJES_1, EJES_LINEAAL, true);
                    #endif
                    Break;
                Case MODO_MOV_ANTIHORARIO_ARC:
                    #ifdef USE_LINEAa_NUMBERS
                        mc_arc (estado_Gcode.position, bloque_Gcode.NUMS.xyz,
bloque_Gcode.NUMS.ijk, bloque_Gcode.NUMS.r,
estado_Gcode.FEEDRATE, estado_Gcode.MODO.FEEDRATE,
EJES_0, EJES_1, EJES_LINEAAL, false, estado_Gcode.LINEAa_number);
                    #else
                        mc_arc (estado_Gcode.position, bloque_Gcode.NUMS.xyz,
bloque_Gcode.NUMS.ijk, bloque_Gcode.NUMS.r,

```

```

        estado_Gcode.FEEDRATE,          estado_Gcode.MODO.FEEDRATE,
EJES_0, EJES_1, EJES_LINEAAL, false);
    #endif
    Break;
    Case MODO_MOV_PROBE_TOWARD:

        #ifdef USE_LINEAa_NUMBERS
            mc_probe_cycle (bloque_Gcode.NUMS.xyz, estado_Gcode.FEEDRATE,
estado_Gcode.MODO.FEEDRATE, false, false, estado_Gcode.LINEAa_number);
        #else
            mc_probe_cycle (bloque_Gcode.NUMS.xyz, estado_Gcode.FEEDRATE,
estado_Gcode.MODO.FEEDRATE, false, false);
        #endif
        Break;
    Case MODO_MOV_PROBE_TOWARD_NO_ERROR:
        #ifdef USE_LINEAa_NUMBERS
            mc_probe_cycle (bloque_Gcode.NUMS.xyz, estado_Gcode.FEEDRATE,
estado_Gcode.MODO.FEEDRATE, false, true, estado_Gcode.LINEAa_number);
        #else
            mc_probe_cycle (bloque_Gcode.NUMS.xyz, estado_Gcode.FEEDRATE,
estado_Gcode.MODO.FEEDRATE, false, true);
        #endif
        Break;
    Case MODO_MOV_PROBE_AWAY:
        #ifdef USE_LINEAa_NUMBERS
            mc_probe_cycle (bloque_Gcode.NUMS.xyz, estado_Gcode.FEEDRATE,
estado_Gcode.MODO.FEEDRATE, true, false, estado_Gcode.LINEAa_number);
        #else
            mc_probe_cycle (bloque_Gcode.NUMS.xyz, estado_Gcode.FEEDRATE,
estado_Gcode.MODO.FEEDRATE, true, false);
        #endif
        Break;
    Case MODO_MOV_PROBE_AWAY_NO_ERROR:
        #ifdef USE_LINEAa_NUMBERS
            mc_probe_cycle (bloque_Gcode.NUMS.xyz, estado_Gcode.FEEDRATE,
estado_Gcode.MODO.FEEDRATE, true, true, estado_Gcode.LINEAa_number);
        #else
            mc_probe_cycle (bloque_Gcode.NUMS.xyz, estado_Gcode.FEEDRATE,
estado_Gcode.MODO.FEEDRATE, true, true);
        #endif
    }

    Malloc (estado_Gcode.position,      bloque_Gcode.NUMS.xyz,      sizeof
(bloque_Gcode.NUMS.xyz)); //
}
}

    estado_Gcode.MODO.SECUENCIA_PROG          =
bloque_Gcode.MODO.SECUENCIA_PROG;

```



```

    If (estado_Gcode.MODO.SECUENCIA_PROG) {
    protocol_buffer_synchronize (); //
    If (estado_Gcode.MODO.SECUENCIA_PROG ==
SECUENCIA_PROG_PAUSED) {
    If (sys.state! = STATE_CHECK_MODE) {
        BIT_true_atomic (sys_rt_exec_state, EXEC_FEED_HOLD); //
        protocol_execute_realtime (); //
    }
    } Else {/
    estado_Gcode.MODO.movimiento = MODO_MOV_LINEAAL;
    estado_Gcode.MODO.SELECC_PLANO = SELECC_PLANO_XY;
    estado_Gcode.MODO.distance = MODO_ABSOLUTE;
    estado_Gcode.MODO.FEEDRATE = FEEDRATE_MODE_unidades_PER_MIN;

    estado_Gcode.MODO.SELECC_COORD = 0; //
    estado_Gcode.MODO.GIRO = GIRO_OFF;

    If (sys.state! = STATE_CHECK_MODE) {
        If
        (! (ajust_datos_leidos (estado_Gcode.MODO.SELECC_COORD, COORD_DATOS)))
        {FAIL (ESTADO_SETTING_READ_FAIL) ;}
        Memcpy (estado_Gcode. Sist_coordenadas, COORD_DATOS, sizeof
        (COORD_DATOS));
        GIRO_stop ();

        }

        report_feedback_message (MESSAGE_PROGRAM_END);
    }
    estado_Gcode.MODO.SECUENCIA_PROG =
SECUENCIA_PROG_RUNNING;
    }

    Return (ESTADO_OK);
}

```

Código de la interfaz manual

```
#include <LiquidCrystal.h>
#include <SPI.h>
#include <SD.h>

LiquidCrystal LCD (10, 9, 8, 7, 6, 5);

Int select=0; //seleccionar opcion;
Int sel_prev=0;
Int sel_prev2=0;
Int select2=0;
String x_mov, y_mov, z_mov, Zero, Gcode_instruct;
String home_pos;
String get_ok;

Int i;
Int j;
//variables para modo jog
Int x_jog=0;
Int y_jog=0;
Int z_jog=0;
File root;
File entry;
File dataFile;
//////////FUNCIONES IMPLEMENTADAS//////////
Void jog_x ();
Void jog_y ();
Void jog_z ();
Void enviar ();
Void printDirectory (File dir, int numTabs);
////////// FIN FUNCIONES IMPLEMENTADAS //////////

////////// CONFIGURACIONES //////////
Void setup ()
{
  // set up the LCD's number of columns and rows:
  LCD.begin (20, 4);
  Serial.begin (115200);
  x_mov = String ("G91X5");
  y_mov = String ("G0Y");
  z_mov = String ("G91Z2");
  Zero= String ("G92X0Y0Z0");
  home_pos= String ("G0X0Y0Z0");
  String get_ok= String ();
  //root = SD.open ("/");
}
```

```

//////////////////////////////////FIN CONFIGURACIONES ////////////////////////////////////

//programa principal
Void loop ()
{
LCD.clear ();

////////////////////////////////// MENU PRINCIPAL//////////////////////////////////
While (analogRead (4) <500)
{
  //LCD.clear ();
  If (analogRead (0)>500) //Boton up
  {While (analogRead (0)>500) {}// antirrebote
    If (select>0)
    {sel_prev=select; select-- ;}
    Else if (select==0)
    {sel_prev=select; select=2 ;}
  }
  If (analogRead (1)>500) // boton down
  {While (analogRead (1)>500) {}// antirrebote
    If (select<2)
    {sel_prev=select; select++ ;}
    Else if (select==2)
    {sel_prev=select; select=0 ;}
  }

  LCD.setCursor (0, select);
  LCD.print (>"); //1
  LCD.setCursor (0, sel_prev);
  LCD.print (" "); //1
  LCD.setCursor (2,0);
  LCD.print ("LEER SD"); //1
  LCD.setCursor (2, 1);
  LCD.print ("JOG ");
  LCD.setCursor (2, 2); //2
  LCD.print ("INFO");
  LCD.setCursor (2, 3); //3
  LCD.print (" PULSE OK ");
  Delay (200);
}
//////////////////////////////////

LCD.clear ();

// SIGUIENTE MENU
//////////////////////////////////
Switch (select)
{
  Case 0:
    LCD.clear ();
    LCD.setCursor (0, 1); //3

```

```

LCD.print ("Enviando");
// see if the card is present and can be initialized:
If (! SD.begin (4)) {
  // nananana
  LCD.setCursor (0, 1); //3
LCD.print ("inserte SD card");
  Return;
}
LCD.setCursor (0, 1); //3
LCD.print ("SD card ok");
  Delay (800);
LCD.clear ();
  Root = SD.open ("/");
    PrintDirectory (root, 0);

Enviar ();

```

Break;

Case 1:

```

////////////////////MODO JOG AJUSTE (0, 0,0)////////////////////

```

```

Delay (500);
//Serial.println ("G91");
LCD.clear ();

```

```

  While (1)
{
  If (analogRead (0)>500) //Boton up
  {While (analogRead (0)>500) {}// antirrebote
    If (select2>0)
      {sel_prev2=select2; select2-- ;}
    Else if (select2==0)
      {sel_prev2=select2; select2=2 ;}
  }
  If (analogRead (1)>500) // boton down
  {While (analogRead (1)>500) {}// antirrebote
    If (select2<2)
      {sel_prev2=select2; select2++ ;}
    Else if (select2==2)
      {sel_prev2=select2; select2=0 ;}
  }
}

```

```

LCD.setCursor (0, select2);
LCD.print (">"); //1
LCD.setCursor (0, sel_prev2);
LCD.print (" "); //1
LCD.setCursor (2,0);
LCD.print ("EJE X"); //1

```

```

    LCD.setCursor (2, 1);
    LCD.print ("EJE Y ");
    LCD.setCursor (2, 2); //2
    LCD.print ("EJE Z");
    LCD.setCursor (2, 3); //3
    LCD.print (" PULSE OK ");

    If (analogRead (4)>500) {break ;}
    Delay (200);
    }

    Switch (select2)
    {
    Case 0:
    LCD.clear ();
    LCD.setCursor (8,0);
    LCD.print ("+X"); //1
    LCD.setCursor (
2, 1);
    LCD.print ("OK PARA X=0");
    LCD.setCursor (8, 2); //2
    LCD.print ("-X");

    Delay
(500);
    While (1)
    {
    jog_x ();
    If
(AnalogRead (4)>500)
    {
    //x_jog--;
    //Gcode_instruct =x_mov;
    Serial.println ("G92 X0Y0Z0");
    Delay (250); break;
    }
    }
    Break;
    Case 1:
    LCD.clear ();
    LCD.setCursor (8,0);
    LCD.print ("+Y"); //1
    LCD.setCursor (2, 1);
    LCD.print ("OK PARA Y=0");
    LCD.setCursor (8, 2); //2
    LCD.print ("-Y");
    Delay (500);
    While (1)
    {
    jog_y ();
    If (analogRead (4)>500)

```

```

{
  //x_jog--;
  //Gcode_instruct =x_mov;
  Serial.println ("G92 X0Y0Z0");
  Delay (250); break;
}
}
Break;
Case
2:
  LCD.clear ();
  LCD.setCursor (8, 0);
  LCD.print ("+Z"); //1
  LCD.setCursor (2, 1);
  LCD.print ("OK PARA Z=0");
  LCD.setCursor (8, 2); //2
  LCD.print (
"-Z");
  Delay (500);
  While (1)
  {
    jog_z (
);
    If (analogRead (4)>500)
  {
    //x_jog--;
    //Gcode_instruct =x_mov;
    Serial.println ("G92 X0Y0Z0");
    Delay (250); break;
  }
  }
  Break;
}

```

Break;

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

Case 2: //mensaje de informacion e intrucciones de manejo con SD Card

```

LCD.clear ();
LCD.setCursor (0,0);
LCD.print ("Aviso"); //1
LCD.setCursor (0,1);
LCD.print ("Configure punto de"); //1
  LCD.setCursor (0, 2);
LCD.print ("coordenadas (0, 0,0)");
  LCD.setCursor (0, 3); //2
LCD.print ("en el modo Jog");
Delay (5000);

```

```

    If (analogRead (4)>500) {break ;}
    LCD.clear ();
    LCD.setCursor (0,0);
    LCD.print ("Valdiviezo"); //1
    LCD.setCursor (0, 1);
    LCD.print ("Negrette"); //1
    LCD.setCursor (0, 2);
    Delay (1000);
    ///////////////////////////////////////////////////////////////////

}

}

//funcion para modo jog en x
Void jog_x ()
{
Serial.write (63);
Delay (100);
If (Serial.available ()) {
// wait a bit for the entire message to arrive
Delay (100);
// clear the screen
LCD.clear ();
// read all the available characters
While (Serial.available () > 0) {
// display each character to the LCD
LCD.write (Serial.read ());
}
}
}

If (analogRead (0)>500)
{
//x_jog++;
// Gcode_instruct =x_mov;
Serial.println ("G91");

Serial.println ("G0X2");

Delay (250);
}
If (analogRead (1)>500)
{
//x_jog--;
//Gcode_instruct =x_mov;
Serial.println ("G91");

0 Serial.println ("G0X-2");

Delay (250);
}

```

```

}

}
//fin movimiento en x

//funcion para modo jog en y
Void jog_y ()
{
Serial.println ('?');
Delay (100);
If (Serial.available ()) {
// wait a bit for the entire message to arrive
Delay (100);
// clear the screen
LCD.clear ();
// read all the available characters
While (Serial.available () > 0) {
// display each character to the LCD
LCD.write (Serial.read ());
}
}

If (analogRead (0)>500)
{
//x_jog++;
// Gcode_instruct =x_mov;
Serial.println ("G91");

Serial.println ("G0Y2");

Delay (250);
}
If (analogRead (1)>500)
{
//x_jog--;
//Gcode_instruct =x_mov;
Serial.println ("G91");

Serial.println ("G0Y-2");

Delay (250);
}

}
//fin movimiento en y

// Funcion para modo jog en z
Void jog_z ()
{
Serial.println ('?');

```



```

Delay (100);
If (Serial.available ()) {
  // wait a bit for the entire message to arrive
  Delay (100);
  // clear the screen
  LCD.clear ();
  // read all the available characters
  While (Serial.available () > 0) {
    // display each character to the LCD
    LCD.write (Serial.read ());
  }
}

If (analogRead (0)>500)
{
  //x_jog++;
  // Gcode_instruct =x_mov;
  Serial.println ("G91");

  Serial.println ("G0Z2");

  Delay (250);
}
If (analogRead (1)>500)
{
  //x_jog--;
  //Gcode_instruct =x_mov;
  Serial.println ("G91");

  Serial.println ("G0Z-2");

  Delay (250);
}

}
//fin movimiento en z

Void enviar ()
{

  LCD.print ("enviano");
  String l_line = "";

  File data2 = SD.open (dataFile.name ());
  While (data2.available () != 0)
  {

    LCD.setCursor (0, 0);
    LCD.print (l_line);
    //espera por la respuesta de tarea ejecutada

```

```

While (Serial.read () != 111)
{

    get_ok = Serial.read ();
    LCD.setCursor (0, 2);
    LCD.print (get_ok);
    LCD.setCursor (0,3);
    LCD.write (Serial.read ( ));

    //for (j=0; j<6; j++)
    // {delay (100) ;}
    //get_ok="Run idlejs ok 9";

    Delay (300);

}
data2.close ();
}

```

```

Void printDirectory (File dir, int numTabs) {
    LCD.clear ();
    While (1) {

        File entry = dir.openNextFile ();
        If (! entry) {
            // no more files
            // return to the first file in the directory
            dir.rewindDirectory ();
            Break;
        }

        LCD.write (entry.name ());
        DataFile=entry;
        LCD.write (dataFile.name ());

        entry.close ();
        while(analogRead(0)<500){if(analogRead(4)>500){break;}}// select
    }

    If (analogRead (4)>500) {break ;}
    While (analogRead (0)>500) {}
    LCD.clear ();

}
}
}

```

Esquemático del circuito controlador

