

UNIVERSIDAD NACIONAL TECNOLÓGICA LIMA SUR
FACULTAD DE INGENIERÍA MECÁNICA
ELECTRÓNICA Y AMBIENTAL
INGENIERÍA ELECTRÓNICA Y
TELECOMUNICACIONES



DISEÑO E IMPLEMENTACIÓN DE UNA
FRESADORA CNC DE 3 EJES DE LIBERTAD
PARA FABRICACIÓN DE TARJETAS
ELECTRÓNICAS USANDO ARDUINO Y UBUNTU
LINUX

TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO
ELECTRÓNICO Y TELECOMUNICACIONES

PRESENTADO POR EL BACHILLER:

JHAN CHRISTIAN ORÉ DÍAZ

VILLA EL SALVADOR - PERÚ

2015

Dedico esta tesis a todas las personas que creyeron que era posible realizar este proyecto y me ayudaron a seguir adelante siempre.

AGRADECIMIENTOS

En esta presente Tesis, agradezco a mis padres Juan y Ángela, mi hermana Angie, mi novia Adriana y mis familiares porque me brindaron su apoyo tanto moral y económicamente para seguir estudiando y lograr el objetivo trazado para un futuro mejor y ser orgullo para ellos y de toda la familia.

A la Universidad Nacional Tecnológica de Lima Sur, por formarnos y brindarnos las herramientas para poder ser mejores profesionales cada día.

De igual manera a mis queridos formadores, en especial a mi asesor de Tesis Ing. Ricardo J. Palomares Orihuela, pues fueron ellos quienes me guiaron para culminar esta Tesis con éxito.

ÍNDICE

| | |
|---------------------------|-----|
| DEDICATORIA | II |
| AGRADECIMIENTOS | III |
| ÍNDICE | IV |
| LISTADO DE FIGURAS | X |
| LISTADO DE TABLAS | XIV |
| INTRODUCCIÓN | XV |

CAPÍTULO I

| | |
|--|----|
| PLANTEAMIENTO DEL PROBLEMA | 17 |
| 1.1. Descripción de la realidad problemática | 17 |
| 1.2. Formulación del problema | 19 |
| 1.3. Objetivos | 19 |
| 1.3.1. Objetivo general | 19 |
| 1.3.2. Objetivo Específico | 19 |
| 1.4. Justificación del proyecto | 20 |
| 1.5. Delimitación de la investigación | 20 |
| 1.5.1. Espacial | 20 |
| 1.5.2. Temporal | 20 |
| 1.6. Variables | 21 |
| 1.6.1. Independientes | 21 |
| 1.6.2. Dependientes | 21 |

| | |
|---|----|
| ataque químico | 54 |
| 2.2.2.7. Técnicas y consideraciones del arte del PCB | 55 |
| 2.2.2.7.1. Pasos generales para la elaboración de circuitos impresos utilizando los métodos de ataque químico | 55 |
| 2.2.2.7.2. Herramientas computacionales de diseño | 56 |
| 2.2.2.7.3. Elección adecuada de los componentes | 56 |
| 2.2.2.7.4. Efecto resistivo de las pistas | 57 |
| 2.2.2.7.5. Efecto térmico de las pistas | 58 |
| 2.2.2.7.6. Transferencia del arte a la placa | 60 |
| 2.2.2.7.7. Pruebas de funcionamiento | 60 |
| 2.2.2.7.8. Otras consideraciones | 61 |
| 2.2.3. Arduino | 62 |
| 2.2.3.1. Hardware | 63 |
| 2.2.3.1.1. Elementos de la placa | 63 |
| 2.2.3.1.2. Elementos del microcontrolador | 64 |
| 2.2.3.1.3. Entradas y salidas | 64 |
| 2.2.3.2. Software | 66 |
| 2.2.3.2.1. Estructura general | 68 |
| 2.2.3.2.2. Variables | 68 |
| 2.2.3.2.3. Estructura de control | 68 |
| 2.2.3.2.4. Temporizadores | 70 |
| 2.2.3.2.5. Pines digitales | 70 |
| 2.2.3.2.6. Pines analógicos | 70 |
| 2.2.3.2.7. Comunicación de puerto serie | 71 |
| 2.2.4. Linux | 72 |
| 2.2.4.1. Componentes de Linux | 72 |
| 2.2.4.2. Unix | 73 |
| 2.2.4.3. Diseño de Unix | 73 |
| 2.2.4.4. Proyecto GNU | 75 |
| 2.2.4.5. Ubuntu | 76 |
| 2.3. Marco conceptual | 79 |

CAPÍTULO III

| | |
|---|-----------|
| DISEÑO Y CONSTRUCCIÓN DE LA HERRAMIENTA | 80 |
| 3.1. Diseño de la herramienta | 80 |
| 3.1.1. Etapas del proyecto | 80 |
| 3.1.2. Control y monitoreo HMI | 81 |
| 3.1.2.1. Arduino Uno R3 | 81 |
| 3.1.2.2. Gambas V3 | 83 |
| 3.1.2.3. Proceso | 84 |
| 3.1.2.4. Estado del arte | 85 |
| 3.1.2.4.1. EMC2 | 85 |
| 3.1.2.4.2. REP-RAP | 86 |
| 3.1.2.4.3. Contraptor CNC | 86 |
| 3.1.2.4.4. Untecs CNC | 86 |
| 3.1.2.5. Untecs CNC transmisión TX | 87 |
| 3.1.2.5.1. Control | 88 |
| 3.1.2.5.2. Instalación | 88 |
| 3.1.2.5.3. Componentes | 88 |
| 3.1.2.5.4. Editor | 89 |
| 3.1.2.5.5. Simulador | 90 |
| 3.1.2.5.6. Controlador | 92 |
| 3.1.2.6. Untecs CNC recepción TX | 95 |
| 3.1.2.6.1. Instalación | 95 |
| 3.1.3. Etapa de alimentación | 99 |
| 3.1.4. Etapa de potencia | 101 |
| 3.1.4.1. Control de motores paso a paso bipolar | 101 |
| 3.1.4.1.1. Identificación del tipo de motor | 101 |
| 3.1.4.1.2. Descripción del controlador | 102 |
| 3.1.4.2. Activación del cabezal | 104 |
| 3.1.4.2.1. Descripción | 104 |
| 3.1.4.2.2. Características técnicas | 105 |
| 3.1.5. Etapa de adquisición y envío de datos | 105 |
| 3.1.6. Etapa mecánica | 107 |

| | | |
|----------|---|------------|
| 3.1.6.1. | Componentes | 107 |
| 3.1.6.2. | Diseño del eje X | 108 |
| 3.1.6.3. | Diseño del eje Y | 111 |
| 3.1.6.4. | Diseño del eje Z | 113 |
| 3.1.6.5. | Diseño de los acoples motor – varilla roscada | 117 |
| 3.2. | Construcción de la fresadora CNC | 118 |
| 3.2.1. | Consideraciones para el armado de los ejes de la fresadora | 118 |
| 3.2.2. | Instalación de varillas lisas y armado de ejes | 119 |
| 3.2.2.1. | Instalación de las varillas lisas del eje Y | 119 |
| 3.2.2.2. | Instalación de las varillas lisas del eje X | 120 |
| 3.2.2.3. | Cuadrado de eje X e Y | 121 |
| 3.2.2.4. | Ensamble de la mesa de fresado en el eje Y | 122 |
| 3.2.2.5. | Ensamblaje del soporte del eje Z y cabezal de la fresadora en el eje X | 123 |
| 3.2.2.6. | Instalación de las varillas en el eje Z | 124 |
| 3.2.2.7. | Ensamble del eje Z al eje X | 125 |
| 3.2.2.8. | Prueba de movimiento de los ejes | 125 |
| 3.2.3. | Instalación de varilla roscada y motores | 125 |
| 3.2.3.1. | Instalación de motor y varilla roscada en el eje Y | 125 |
| 3.2.3.2. | Instalación de motor y varilla roscada en el eje X | 127 |
| 3.2.3.3. | Instalación de motor y varilla roscada en el eje Z | 128 |
| 3.2.4. | Instalación de finales de carrera | 129 |
| 3.2.5. | Fabricación del tablero eléctrico para la fresadora CNC | 130 |
| 3.2.5.1. | Parte superior etapa de control y adquisición | 130 |
| 3.2.5.2. | Parte intermedia etapa de alimentación y activación del cabezal | 131 |
| 3.2.5.3. | Parte inferior etapa de potencia | 131 |
| 3.3. | Pruebas y resultados | 132 |
| 3.3.1. | Pruebas | 132 |
| 3.3.2. | Resultados | 142 |
| | CONCLUSIONES | 144 |

| | |
|--|-----|
| RECOMENDACIONES | 145 |
| BIBLIOGRAFÍA | 146 |
| ANEXOS | 147 |
| Anexo 1: Imagen proyecto completo | 147 |
| Anexo 2: Datasheet de microcontrolador de Arduino Uno | 148 |
| Anexo 3: Datasheet de chip L298 de etapa de potencia | 149 |
| Anexo 4: Descripción de la herramienta de mecanizado para placa electrónica | 150 |
| Anexo 5: Procedimiento para programación del trazado | 151 |
| Anexo 6: Programa Untecs CNC Recepción RX | 156 |
| Anexo 7: Circuito eléctrico del proyecto | 159 |
| Anexo 8: Planos de fresadora CNC | 160 |
| Anexo 9: Presupuesto | 161 |

LISTADO DE FIGURAS

| | |
|---|----|
| Figura N° 1: Sistema de control de lazo abierto | 27 |
| Figura N° 2: Sistema de control de lazo cerrado | 30 |
| Figura N° 3: Sistema de control punto a punto | 31 |
| Figura N° 4: Desplazamiento angular | 32 |
| Figura N° 5: Desplazamiento paraaxial | 33 |
| Figura N° 6: Trayectoria continua | 34 |
| Figura N° 7: Tornillo de cuerda triangular | 36 |
| Figura N° 8: Tuerca | 36 |
| Figura N° 9: Ballscrew o tornillo embalado | 37 |
| Figura N° 10: Tipos de orden de desplazamiento | 38 |
| Figura N° 11: Sistema de tuerca – tornillo 1 eje | 40 |
| Figura N° 12: Sistema de tuerca – tornillo 2 ejes | 41 |
| Figura N° 13: Sistema de tuerca tornillo de 2 ejes y tercer eje independiente | 41 |
| Figura N° 14: Sistema de tuerca – tornillo 3 ejes | 42 |
| Figura N° 15: Planteo de sistema para fresadora de 3 ejes | 42 |
| Figura N° 16: Movimiento combinado diagonal y recto | 44 |
| Figura N° 17: Movimiento diagonal | 45 |
| Figura N° 18: Movimiento en curva | 45 |
| Figura N° 19: Placa monocapa | 46 |
| Figura N° 20: Placa doble cara o bicapa | 47 |
| Figura N° 21: Creación de placa PCB por ataque químico | 54 |
| Figura N° 22: Diseño de circuito eléctrico | 62 |
| Figura N° 23: Diseño de circuito de placa en Orcad | 62 |
| Figura N° 24: Placa Arduino Uno | 63 |
| Figura N° 25: Estructura de tarjeta Arduino Uno | 64 |
| Figura N° 26: Diagrama de funcionamiento de Arduino | 65 |
| Figura N° 27: Circuito de salida de Arduino | 66 |
| Figura N° 28: Circuito de entrada para Arduino | 66 |
| Figura N° 29: Logo oficial de Linux | 72 |
| Figura N° 30: Diseño de Unix | 74 |
| Figura N° 31: Distribuciones más conocidas de Linux | 76 |

| | |
|--|-----|
| Figura N° 32: Etapas del proyecto | 80 |
| Figura N° 33: Estructura general de la etapa electrónica de la Fresadora CNC | 81 |
| Figura N° 34: Arduino Uno R3 | 82 |
| Figura N° 35: Logo de software Gambas V3 | 83 |
| Figura N° 36: Diagrama de bloques de la etapa electrónica de la Fresadora CNC | 85 |
| Figura N° 37: Partes del software de control | 89 |
| Figura N° 38: Editor | 89 |
| Figura N° 39: Simulador | 91 |
| Figura N° 40: Control | 94 |
| Figura N° 41: Valores de cableado por color de Fuente ATX | 100 |
| Figura N° 42: Fuente ATX | 101 |
| Figura N° 43: Transformación de motor unipolar a bipolar | 102 |
| Figura N° 44: Circuito de control de motor paso a paso bipolar | 102 |
| Figura N° 45: Control de funcionamiento de motor paso a paso bipolar | 103 |
| Figura N° 46: Control de funcionamiento de motor paso a paso con dos pines | 103 |
| Figura N° 47: Circuito de control con Relé | 104 |
| Figura N° 48: Circuito de control de finales de carrera | 106 |
| Figura N° 49: Plantillas de fabricación de fresadora CNC | 108 |
| Figura N° 50: Plantillas de diseño de fresadora en Autocad | 108 |
| Figura N° 51: Pieza lateral izquierda de eje X | 109 |
| Figura N° 52: Pieza lateral derecha de eje X | 109 |
| Figura N° 53: Varillas de acero guía y bocinas de eje X | 110 |
| Figura N° 54: Ensamble de eje X | 110 |
| Figura N° 55: Pieza posterior de eje Y | 111 |
| Figura N° 56: Pieza frontal de eje Y | 112 |
| Figura N° 57: Ensamble de eje Y | 112 |
| Figura N° 58: Pieza superior de eje Z | 113 |
| Figura N° 59: Pieza inferior de eje Z | 113 |
| Figura N° 60: Soporte de fresadora y eje Z parte A | 114 |
| Figura N° 61: Soporte de fresadora y eje Z parte B | 114 |

| | |
|--|-----|
| Figura N° 62: Ensamble de soportes de fresadora parte A y B | 115 |
| Figura N° 63: Ensamble de eje Z | 115 |
| Figura N° 64: Ensamble de eje Z con soporte | 116 |
| Figura N° 65: Ensamble general de la fresadora | 117 |
| Figura N° 66: Ensamble de motor con varilla roscada | 118 |
| Figura N° 67: Pieza de la fresadora | 118 |
| Figura N° 68: Ensamblaje de eje Y | 120 |
| Figura N° 69: Ensamble de eje X y soporte de fresadora | 121 |
| Figura N° 70: Instalación de mesa de fresado | 122 |
| Figura N° 71: Fijación de soporte de fresadora en eje X | 123 |
| Figura N° 72: Fijación de fresadora y eje Z en soporte | 124 |
| Figura N° 73: Instalación de varilla roscada en eje Y | 126 |
| Figura N° 74: Instalación de motor en eje Y | 126 |
| Figura N° 75: Instalación de varilla roscada en eje X | 127 |
| Figura N° 76: Instalación de motor en eje X | 128 |
| Figura N° 77: Instalación del motor y varilla roscada en eje Z | 129 |
| Figura N° 78: Instalación de finales de carrera | 129 |
| Figura N° 79: Etapa de control y adquisición de datos | 130 |
| Figura N° 80: Etapa de alimentación y activación del cabezal | 131 |
| Figura N° 81: Etapa de potencia | 132 |
| Figura N° 82: Circuito eléctrico para pruebas en software ISIS PROTEUS | 133 |
| Figura N° 83: Diseño de pistas en software ARES PROTEUS | 133 |
| Figura N° 84: Vista en 3D de la placa | 134 |
| Figura N° 85: Pistas en formato .SVG | 134 |
| Figura N° 86: Pistas importadas en software INKSCAPE | 135 |
| Figura N° 87: Diseño de pistas a trazar | 135 |
| Figura N° 88: Fijación de punto de inicio con orientation points | 136 |
| Figura N° 89: Selección de herramienta para fresado | 137 |
| Figura N° 90: Trazos a seguir por fresadora CNC | 137 |
| Figura N° 91: Creación del archivo de mecanizado | 138 |
| Figura N° 92: Modificación del archivo de mecanizado | 138 |
| Figura N° 93: Archivo de mecanizado en software de Fresadora | 139 |
| Figura N° 94: Ubicación de herramienta Fresadora en punto de origen | 139 |

| | |
|--|-----|
| Figura N° 95: Fabricación de las pistas | 140 |
| Figura N° 96: Resultado de la placa PCB | 141 |
| Figura N° 97: Fabricación de agujeros en placa PCB | 141 |
| Figura N° 98: Funcionamiento de la placa PCB | 142 |
| Figura N° 99: Pruebas de funcionamiento | 142 |

LISTADO DE TABLAS

| | |
|---|----|
| Tabla N° 1: Códigos G y funciones | 39 |
| Tabla N° 2: Número de agujeros por tipo de circuito impreso | 50 |
| Tabla N° 3: Dígitos según tipo de placa | 50 |
| Tabla N° 4: Límites dimensionales para placa sin agujeros metalizados | 51 |
| Tabla N° 5: Límites dimensionales para placa con agujeros metalizados | 52 |
| Tabla N° 6: Límites dimensionales para placa multicapa | 53 |
| Tabla N° 7: Comandos G reconocidos por UNTECS CNC | 94 |
| Tabla N° 8: Listado de funciones de acceso rápido | 95 |

INTRODUCCIÓN

El presente Tesis tiene como objetivo diseñar e implementar una fresadora CNC de 3 ejes de libertad para la fabricación de tarjetas electrónicas usando Arduino y Ubuntu Linux. La fresadora CNC traslada el dibujo de las pistas de un circuito impreso generado por Orcad y Proteus utilizando el software Inkscape a una placa de cobre. El prototipo electromecánico de la fresadora CNC consta de cinco etapas: adquisición de datos, control y HMI, etapa de potencia, etapa de alimentación, y etapa mecánica.

El diseño y la fabricación de circuitos impresos son procesos muy importantes para la industrias ya que influyen en la competitividad de éstas, por lo cual, en cuanto más rápido y bien hechas sean, es mejor, también en la calidad, portabilidad y la duración que puedan tener los dispositivos electrónicos. Sin embargo, no sólo para la industria son de gran importancia, si no, también para los proyectos académicos, concursos, prototipos, proyectos y otros.

Debido a la importancia que tienen las placas de circuito impreso es necesario mencionar la elaboración de éstas: inicialmente se tiene que hacer el diseño de las pistas, que puede ser de forma manual o por software. En la forma manual se analizan todas las conexiones de cada dispositivo y por medio de una regla o a mano, se trazan las pistas. Por software, se crea el esquema eléctrico del circuito y después se obtiene un dibujo en un plano que contempla la colocación y posición de los componentes, así como los puntos de interconexión, contactos, entradas y salidas del circuito.

Otra forma de realizar el PCB es utilizando máquinas por control numérico, sin embargo presenta ventajas y desventajas. La ventaja principal de este método es que ya no se necesita imprimir un modelo para después ser transferido a la placa de cobre, pues el fresado lo hace directamente sobre la placa, tomando la información del archivo obtenido por el software para hacer circuitos impresos; además también presenta ahorro de tiempo y precisión en los trazos, el sistema es automático. La desventaja principal consiste en los precios elevados y que por esta razón no sea posible su adquisición.

La estructura que se ha seguido en esta tesis se compone de 3 capítulos. El Primer capítulo comprende el Planteamiento del Problema, el Segundo Capítulo el Desarrollo del Marco Teórico y el tercer capítulo corresponde al Diseño y Construcción del Proyecto.

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 Descripción de la realidad problemática

El diseño y la fabricación de circuitos impresos son procesos muy importantes para la industrias ya que influyen en la competitividad de éstas, por lo cual, en cuanto más rápido y bien hechas sean, es mejor, también en la calidad, portabilidad y la duración que puedan tener los dispositivos electrónicos. Sin embargo, no sólo para la industria son de gran importancia, si no, también para los proyectos académicos, concursos, prototipos, proyectos y otros.

Las estadísticas que tienen las empresas de fabricantes de placas PCB, es difícil conocerlas, sin embargo existen diferentes equipos electrónicos en los que podemos encontrar una placa de circuito impreso, equipos como: celulares, computadoras, laptop, equipos de audio, y muchos más. Tal es el caso de equipos Marshall fabricantes de amplificadores para guitarras eléctricas, que venden 11,000 amplificadores por semana en países como Inglaterra, India, Corea, China y Vietnam. Otra empresa es Solter Solder dedicada a la fabricación de equipos electrónicos para soldadura en acero inoxidable y aluminio, la cual facturó en el año 2003, seis millones de euros, con estas cifras se nota la importancia que tiene la fabricación de los circuitos impresos, pues estos una vez terminados

son ensamblados con otros componentes para formar un equipo electrónico, el cual ya está listo para salir al mercado.

Debido a la importancia que tienen las placas de circuito impreso es necesario mencionar el procedimiento para la elaboración de éstas en la Universidad Nacional Tecnológica de Lima Sur, lo cual consiste en lo siguiente: inicialmente se tiene que hacer el diseño de las pistas, que puede ser de forma manual o por software. En la forma manual se analizan todas las conexiones de cada dispositivo y por medio de una regla o a mano, se trazan las pistas. Por software, se crea el esquema eléctrico del circuito y después se obtiene un dibujo en un plano que contempla la colocación y posición de los componentes, así como los puntos de interconexión, contactos, entradas y salidas del circuito.

Una vez se han diseñado las pistas y la forma en la que quedarían, se procede a trasladar el dibujo a la placa. La forma más utilizada para trasladar el dibujo a la placa, es mediante el calcado del impreso original hacia la placa. En este proceso se realiza el diseño del dibujo de las pistas, que comúnmente se llama *layout* (dibujo de circuito impreso), regularmente se utilizan software como Eagle y Ares, para realizarlo. Una vez se obtiene este dibujo se imprime en una hoja blanca papel bond para fotografía, o papel couche, cualquiera sea la elección se hace en impresora láser, si no se cuenta con una, se imprime y se obtiene una copia de la fotocopidora, después sobre una de las caras de la placa de cobre se coloca el esquemático, a esta colocación se le debe mantener fijo por medio de un adhesivo temporal o algunas pinzas, mientras se le incide calor con una plancha hasta que el dibujo quede marcado sobre la placa, de esta manera es como se concluye la transferencia de su dibujo sobre la placa, para posteriormente ponerla en ácido y terminar el proceso de la fabricación de la placa PCB.

Si bien existen mejores métodos para el diseño de las tarjetas PCB la Universidad Nacional Tecnológica de Lima Sur no cuenta con el equipamiento en sus laboratorios para tener máquinas especializadas en la fabricación de placas de circuito impreso.

1.2 Formulación del problema

De acuerdo a lo mencionado en la descripción del problema, la fabricación de tarjetas electrónicas PCB es de mucha importancia, tanto en su aplicación académica para estudiantes de ingeniería así como en su aplicación profesional en la industria nacional por lo tanto, con la finalidad de dar una solución se formula el siguiente problema, ¿Cómo podremos fabricar tarjetas electrónicas PCB de buena calidad, en un menor tiempo y disminuyendo sustancialmente sus costos, con la finalidad de brindar una herramienta para los estudiantes de la Universidad Nacional Tecnológica de Lima Sur?

1.3 Objetivos

1.3.1 Objetivo general

El objetivo general de la presente Tesis es diseñar e implementar una fresadora CNC de 3 ejes de libertad para fabricación de tarjetas electrónicas usando Arduino y Ubuntu Linux, con la finalidad de brindar una herramienta para los estudiantes de la Universidad Nacional Tecnológica de Lima Sur.

1.3.2 Objetivos específicos

La presente Tesis tienen los siguientes objetivos específicos:

- Determinar los dispositivos electrónicos y materiales para diseñar e implementar una maquina CNC de 3 ejes que tenga la capacidad para crear tarjetas electrónicas PCB.
- Desarrollar la etapa de potencia para controlar los motores de la maquina CNC.
- Utilizar un software libre basado en Linux como interface entre la computadora y la maquina CNC.
- Desarrollar la etapa de adquisición y envío de datos requerida para poder comunicar la maquina CNC con la computadora.

- Crear guías del procedimiento para la fabricación de tarjetas electrónicas.

1.4 Justificación del proyecto

La justificación de la presente Tesis, “Diseño e implementación de una fresadora CNC de 3 ejes de libertad para fabricación de tarjetas electrónicas usando Arduino y Ubuntu Linux”, es brindar una solución frente a la realización de placas de circuito impreso de manera artesanal, además de buscar reducir los tiempos de fabricación de las mismas, como también el de brindar una solución económica, que puede ser implementada en la Universidad Nacional Tecnológica de Lima Sur.

Además la finalidad de este proyecto es que pueda ser una herramienta que pueda ser desarrollada por profesionales y principalmente por los estudiantes de la carrera de ingeniería electrónica de la Universidad Nacional Tecnológica de Lima Sur.

El problema de investigación es vulnerable, es decir, puede ser investigado. El investigador tiene los conocimientos suficientes y los recursos necesarios para obtener resultados rigurosos.

Se justifica a nivel tecnológico, que el diseño de este proyecto implica realizar nuevos diseños y mejoras adecuados a las tecnologías de las máquinas herramienta.

1.5 Delimitación de la investigación

1.5.1 Espacial

La presente Tesis ha sido desarrollada en la Universidad Nacional Tecnológica de Lima Sur – UNTELS y en el domicilio del Tesista, Villa El Salvador, Lima, Perú.

1.5.2 Temporal

La duración del desarrollo profesional de la Tesis será de 6 meses.

1.6 Variables

1.6.1 Independientes

- Diseño e implementación de una fresadora CNC de 3 ejes de libertad.

INDICADORES

- Determinar materiales y dispositivos electrónicos.
- Control de motores.
- Uso de software libre.
- Desarrollo de adquisición de datos.

1.6.2 Dependientes

- Fabricación de tarjetas electrónicas usando Arduino y Ubuntu Linux.

INDICADORES

- Calidad
- Tiempo
- Costo

CAPITULO II

MARCO TEÓRICO

2.1 Antecedentes

A continuación se presentan algunos de los principales aportes e investigaciones desarrolladas y relacionadas con la Tesis, tomadas del ámbito internacional y nacional.

2.1.1 Internacionales

Herrera D. (2012). "Diseño y construcción de una máquina para la elaboración de circuitos impresos basado en control numérico computarizado". Escuela Politécnica Nacional, donde se realizó un estudio de la elaboración de circuitos impresos basados en control numérico. En dicho proyecto se diseña y construye un sistema alternativo de elaboración de PCB (Printed Circuit Board) mediante una máquina fresadora por Control Numérico Computarizado (CNC) que sustituye los métodos artesanales y sistemáticos utilizados por los estudiantes universitarios y de educación secundaria, permitiendo realizar circuitos impresos con la precisión y exactitud requeridas, sin uso de ácidos o corrosivos y efectuando adicionalmente perforaciones para la disposición y posterior soldadura (de manera manual) de elementos electrónicos en la PCB elaborada.

Burbano G. (2011). "Mesa Robótica XY para la perforación de placas PCB con conexión USB". Universidad San Francisco de Quito, donde se realizó un estudio de la perforación de placas PCB por medio de una mesa robótica XY. En dicho proyecto de investigación se indica que esta mesa robótica XY va a realizar las placas a escala normal, es decir, que va a tener tamaño y dimensiones reales de acuerdo al tamaño real que se encuentran en el mercado. Usando un microcontrolador PIC 18f4550, nos va a permitir controlar y comunicar con la PC mediante USB, ya que en todas las computadoras se encuentra un conector. Para realizar los movimientos, usó motores paso a paso, de esta forma se consigue tener precisión, y la facilidad de control de todos los movimientos en diferentes ejes del robot. La interface gráfica, la cual permite enviar datos al microcontrolador para realizar los movimientos deseados para que este sitúe y taladre, según el circuito impreso realizado en Proteus – Ares o Livewire – PCB Wizard.

2.1.2 Nacionales

Nakamura J. (2009). "Diseño e implementación de un brazo robot de dos grados de libertad para el trazado de diagramas en un plano". Pontificia Universidad Católica del Perú, donde se realizó un estudio de un brazo robot para la aplicación de diagramas de trabajo en planos A3. En dicho proyecto. El diseño del sistema de control para el brazo robot de dos grados de libertad se dividió en tres partes: una etapa de interfaz de usuario, encargada de interactuar con el usuario mediante un computador personal y comunicarse con la etapa del controlador; la cual, se encarga de recibir del computador personal los valores de trayectoria ingresados por el usuario y generar las señales de control necesarias para accionar los actuadores encargados de realizar el movimiento, así como de recibir información de los sensores acoplados al sistema; y por último una etapa de interfaz de potencia, encargada de recibir las señales de

control generadas por el microcontrolador, llevándolas a los niveles de potencia adecuados para los actuadores.

Flores J. (2012) "Diseño Mecánico del Cabezal y el Bastidor de una fresadora CNC de 500x500x300 mm con mesa de trabajo tipo plataforma de Gough", Pontificia Universidad Católica del Perú, En el presente proyecto se diseña una fresadora CNC con la variante que llevará como mesa de trabajo una plataforma móvil de 6 GDL, cuya dinámica se regirá a partir de una correcta acción de control. Así, la mesa de trabajo permitirá 3 desplazamientos y 3 orientaciones. La presente tesis contempla el diseño mecánico del cabezal y del cuerpo de la fresadora CNC, así como la selección de componentes internos.

2.2 Bases teóricas

Uno de los elementos que ha jugado un papel fundamental en el desarrollo tecnológico en el mundo ha sido la máquina herramienta y podemos decir sin exagerar que ha afectado directamente en el desarrollo industrial.

Gracias a la utilización de la máquina herramienta se ha podido realizar de forma práctica, maquinaria de todo tipo que, aunque concebida y realizada, no podía ser comercializada por no existir medios adecuados para su construcción industrial por ejemplo, si la mecanización total de un número de piezas fuera necesario realizar las operaciones de fresado, mandrilado y perforado, es lógico que se alcanzaría la mayor eficacia si este grupo de máquinas herramientas estuvieran agrupadas, pero se lograría una mayor eficacia aún si todas estas operaciones se realizaran en una misma máquina.

Esta necesidad, sumada a numerosos y nuevos requerimientos que día a día aparecieron forzó a la utilización de nuevas técnicas que reemplazaran al operador humano. De esta forma se introdujo el control numérico en los procesos de fabricación, impuesto por varias razones:

- Necesidad de fabricar productos que no se podían conseguir en cantidad y calidad suficientes sin recurrir a la automatización del proceso de fabricación.
- Necesidad de obtener productos hasta entonces muy difíciles de fabricar, por ser excesivamente complejos para ser controlados por un operador humano.
- Necesidad de bajar costos de producción para ser competitivos y así proporcionar productos a precios bajos.

El factor primordial que condicionó todo automatismo fue el aumento de productividad. Posteriormente, debido a las nuevas necesidades de la industria aparecieron otros factores como la precisión, la rapidez y la flexibilidad.

En 1801 Jacquard Loom ideó una máquina textil que podía realizar distintos tipos de tejidos sin más que variar los programas de fabricación, que se introducían en la unidad de control de la máquina a través de unas tarjetas perforadas.

En 1942 surgió lo que podría llamar el primer control numérico (CN) verdadero, derivado a una necesidad impuestas por la industria aeronáutica para la realización de hélices de helicópteros de diferentes configuraciones, no fue hasta la década de los 70's que se conoce como CNC por la aplicación de las computadoras.

Un CN significa mando mediante números, con este sistema se consigue que las máquinas realicen trabajo de forma automática, mediante la introducción en la memoria del CN de un programa en el que se encuentran definidas en clave todas las operaciones del proceso. Originalmente la denominación CN se aplicaba a todas las máquinas programables que no iban equipadas con computador.

Con la miniaturización y abaratamiento de los microprocesadores, se ha podido generalizar su instalación en todas las máquinas CN. A estas máquinas se les denomina CNC (Control Numérico Computarizado).

Actualmente la fabricación de máquinas de CNC ha ido creciendo debido a la reducción de los costos y a la simplificación de la programación. Ahora, las máquinas se programan directamente, en lugar de hacerlo a través de una cinta perforada como habitualmente se hacía antes de los noventa.

2.2.1 Control numérico computarizado - CNC

El control numérico se puede definir como un dispositivo de automatización de una máquina que, mediante una serie de instrucciones codificadas (programa) controla su funcionamiento.

Cada programa establece un determinado proceso a realizar por la máquina, con lo que una misma máquina puede efectuar automáticamente procesos distintos sin más que sustituir su programa de trabajo. Esto permite una elevada flexibilidad de funcionamiento con respecto a las máquinas automáticas convencionales, en las que los automatismos se conseguían mediante sistemas mecánicos o eléctricos difíciles y a veces casi imposible de modificar.

La programación de los controles numéricos ha sufrido una gran evolución en los últimos años.

Si bien se habla todavía de programación manual y programación automática o asistida por computadora, la realidad es que hoy en día, al contar con los controles con una microcomputadora incorporada, la programación manual dispone de muchas facilidades reservadas hasta hace poco a la programación automática.

Sería quizás más adecuado efectuar otra clasificación:

- a) La programación a pie de máquina, apoyada en los lenguajes y facilidades de que disponen los CNC.
- b) La programación en oficina técnica, apoyada en equipos y software propios dentro de las técnicas de CAD-CAM.

2.2.1.1 Sistemas de control

Existen principalmente dos sistemas de control numérico para las máquinas-herramienta, que son: lazo abierto y lazo cerrado.

2.2.1.1.1 Sistemas de lazo abierto

Este es un sistema de control sin retroalimentación como se muestra en la figura N° 1, es decir, sin comparación entre la posición real y la prescrita. En este sistema se alimenta la cinta perforada a una lectora de cinta que decodifica la información y la almacena hasta el momento que se va a poner en marcha la máquina; cuando esto sucede, la convierte en impulso o señales eléctricas, que a su vez se envían a la unidad de control, la cual energiza a las unidades de servocontrol. Estas unidades controlan los servomotores para que ejecuten determinadas funciones de acuerdo con la información perforada en la cinta.

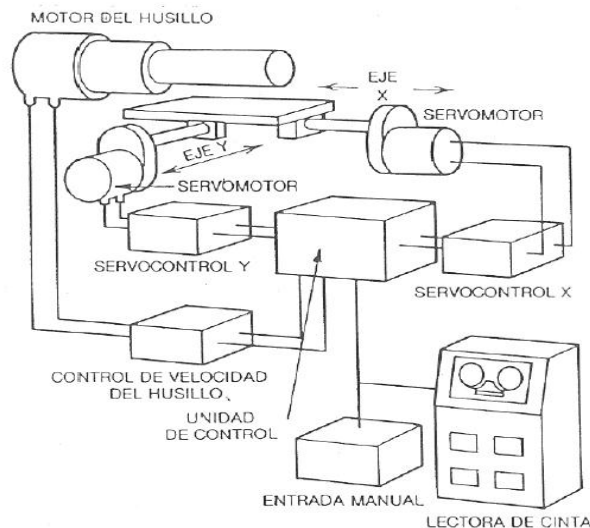


Figura N° 1: Sistema de control de lazo abierto

La distancia que cada *servomotor* mueve el sinfín de avance de la máquina depende del número de impulsos que reciba desde la unidad de control. Los sistemas de

lazo abierto tienen una exactitud de 0.02mm, y las dimensiones de los planos se indican en 3 decimales.

También el desplazamiento de los dispositivos móviles, puede ser accionado por un motor “paso a paso”, que da una cantidad de pasos o desplazamientos elementales definitivamente por la unidad de control.

En este sistema no hay forma de comprobar si el servomotor ha efectuado su función de modo concreto, y no se debe utilizar cuando se requiere una exactitud mayor de 0.02mm. La figura muestra un diagrama de un sistema de control numérico de lazo abierto.

Motor paso a paso. (También denominados motores de velocidad gradual) Es un motor eléctrico utilizado como accionamiento en sistemas de lazo abierto. El rotor de ese motor gira un ángulo dado cada vez que recibe un impulso de corriente, o sea que un tren de impulsos determinado se traduce un número de desplazamientos elementales.

Servomotor. Es un motor eléctrico que puede girar desde fracciones pequeñas de radian hasta el giro continuo en respuesta de una señal proveniente de control. Es un motor utilizado en los servomecanismos para producir movimientos lineales o rotativos.

Servomecanismos. El servomecanismo proporciona la amplificación y la potencia de controlables necesarias para mover rápidamente grandes masas en presencia de fuertes perturbaciones de la carga. La realimentación negativa hace posible que se establezca la comparación entre la señal de entrada de referencia y el valor medio del movimiento controlado por la salida. Se produce así una señal-error de energía reducida.

El servoamplificador y el servomotor responden a la señal-error haciendo que el movimiento de salida se aproxime al valor deseado. A menudo, la señal de referencia a la entrada adopta la forma de corriente o tensión eléctrica. También puede adoptar la de cualquiera de muchos fenómenos físicos, tales como la presión hidráulica o neumática, velocidades de circulación de fluidos, movimientos mecanizados, fuerzas mecánicas, transferencia de calor, variaciones de temperatura o intensidad de radiaciones nucleares; sin embargo, la salida de un servomecanismo es siempre el movimiento de un eje, palanca, articulación o elemento mecánico similar. Para lograr el mejor comportamiento posible del servomecanismo, el dispositivo de medida debe poseer un alto grado de precisión siendo estático, siendo en cambio, despreciable del retardo dinámico en relación a las características de respuesta dinámica del resto del sistema.

2.2.1.1.2 Sistemas de lazo cerrado

Un sistema de control de lazo cerrado es un sistema como se muestra en la figura N° 2 que cuenta con una unidad de retroalimentación (transductor). En este sistema la magnitud de salida es medida de forma continua o por incrementos, y transmitida por medio de un lazo o bucle de retorno a un comparador para que dicha magnitud sea comparada con la magnitud de entrada. La magnitud de salida es la posición real del dispositivo móvil y la magnitud entrada es la posición prescrita por el programa. El comparador establece una comparación entre estas posiciones y suministra una señal de error hasta que coinciden.

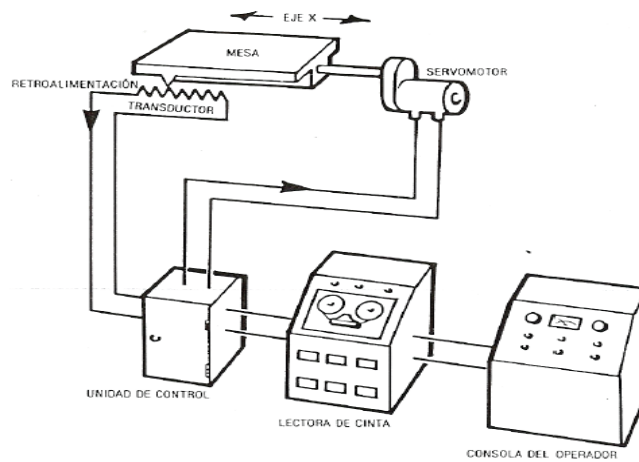


Figura N° 2: Sistema de control de lazo cerrado

Los transductores de los sistemas de lazo cerrado permiten una exactitud de 0.002 mm (0.0001 pulg.) y las dimensiones en los planos se indican con cuatro decimales. La figura muestra un sistema de control numérico de lazo cerrado.

2.2.1.2 Tipos de control de las máquinas-herramienta con CNC

Hay tres tipos disponibles de controles para ubicar y controlar la herramienta de corte y la pieza de trabajo durante el maquinado, y son los siguientes:

2.2.1.2.1 Punto a punto

Este sistema controla el posicionamiento de la herramienta en los sucesivos puntos donde debe efectuarse una o varias operaciones de maquinado. La trayectoria de la mesa es siempre lineal para pasar de un punto al siguiente y puede llegar a un punto en particular siguiendo más de una trayectoria, pues las funciones de posicionamiento y de maquinado son distintas. En la figura N° 3 se muestra un ejemplo.

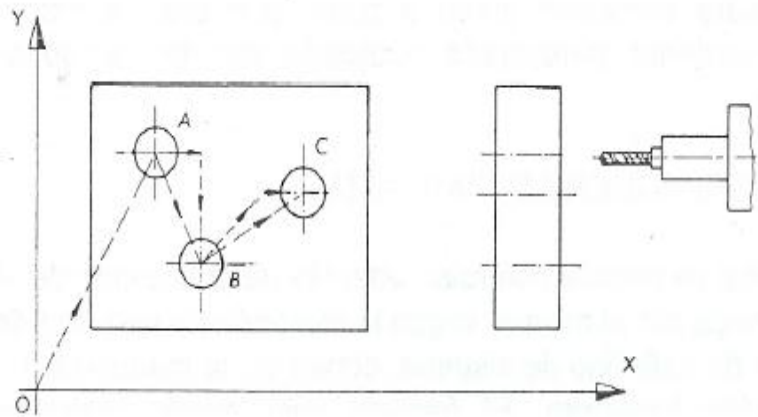


Figura N°3: Sistema de control punto a punto

En el ejemplo de la figura, la mesa para desplazarse a los puntos A, B y C puede desplazarse a lo largo de las siguientes trayectorias:

- Desplazamientos efectuados con direcciones paralelos a los ejes, o sea, primero sobre el eje X y luego sobre el eje Y, o viceversa.
- Desplazamiento dando un orden simultáneo a los dos ejes, pero sin existir coordinación alguna en los sistemas de mando de cada uno de los motores. La trayectoria se aproxima a 45° .

Se desea programar desplazamientos angulares o segmentos de arco, estos deben ser programados como una serie de cortes lineales; como se muestra en la figura N° 4.

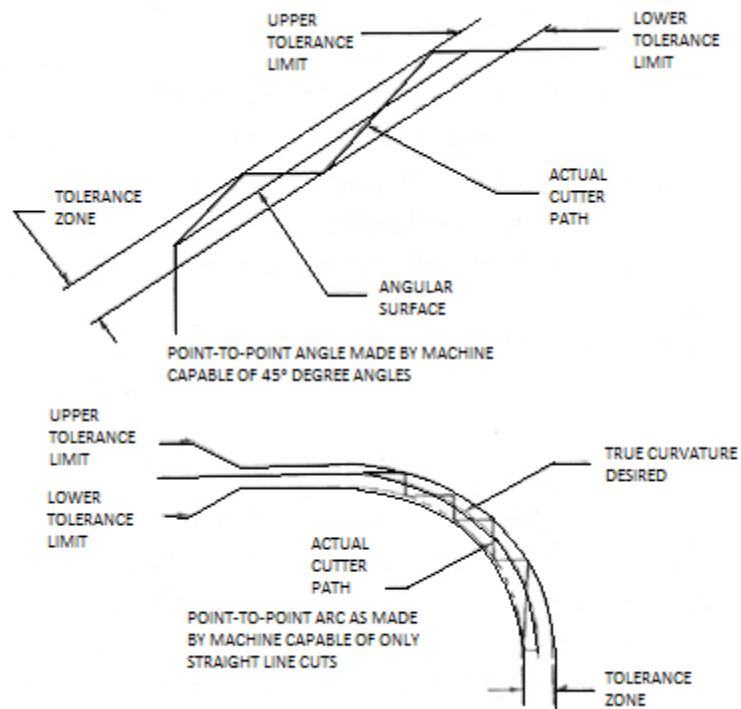


Figura N° 4: Desplazamiento angular

Las órdenes para ubicación punto a punto son ideales para taladrar, rectificar; este tipo suele usar un sistema de lazo abierto.

2.2.1.2.2 Paraaxial

Con este sistema es posible controlar, además de la posición del dispositivo móvil, la trayectoria seguida por el mismo, según la dirección de ejes coordenadas "X" y "Y".

Una aplicación a este tipo de sistema, dentro de la máquina-herramienta, se halla en la taladradora-fresadora. El fresado sólo puede realizarse en trayectorias rectilíneas paralelas a alguno de los ejes coordenados. Ver figura N° 5.

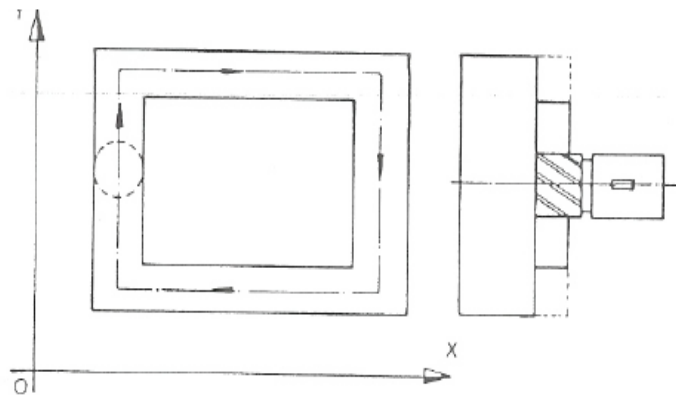


Figura N° 5: Desplazamiento paraaxial

2.2.1.2.3 Trayectoria continua

En los sistemas de programación de trayectoria continua, los desplazamientos del dispositivo móvil son controlados en todo momento, de manera que las posiciones del mismo deben corresponder a la trayectoria preestablecida. Esta programación permite la fabricación de los contornos perfiles irregulares y la obtención de curvas complejas cuyo mecanizado normal presentaría muchas dificultades.

Con ayuda de la figura N° 6 vamos a analizar el método utilizado por la máquina para efectuar este tipo de desplazamiento.

Las ordenes de desplazamiento con comunicadas al dispositivo móvil bajo forma de componentes de un vector que permite alcanzar un punto próximo. La curva elemental descrita por este dispositivo será por lo tanto un segmento de recta. Para ejecutar contorneado cualquiera, ésta se descompondrá en segmentos elementales llamados incrementos. Su número dependerá de la precisión con que se quiera efectuar el trabajo. Por ejemplo muchas piezas requieren segmentos de longitud

mayor de 0.0125mm (0.0005 pulg.) o cuerdas correspondientes a arcos de 1°.

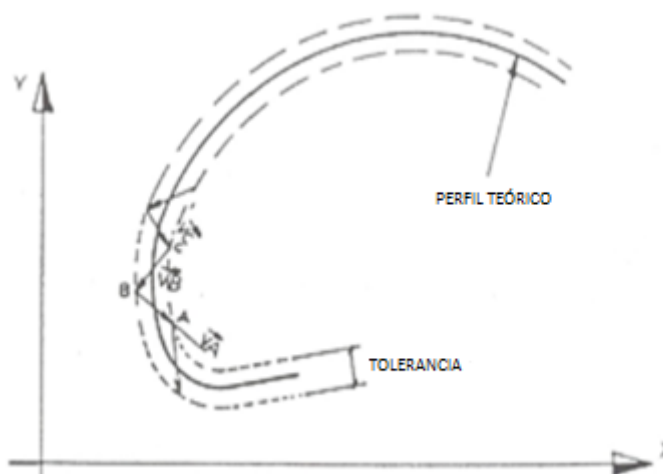


Figura N° 6: Trayectoria continua

A esta descomposición de la curva se le denomina “interpolación”. Es necesario sincronizar velocidades siguiendo las direcciones O-X y O-Y, con el objeto de definir los valores de V_A ; V_B , etc. La curva real así situada en el interior de una zona delimitada por dos curvas paralelas en donde la anchura de estas está definida por la precisión del maquinado.

Para el caso de un desplazamiento continuo con un plano, se sincronizarán dos ejes, llamándose por lo tanto “máquina de dos ejes”. Para un maquinado en el espacio se deberá utilizar una máquina de tres ejes.

2.2.1.3 Principio de funcionamiento

Todas las máquinas CNC tienen dos o más grados de movimiento llamados ejes, cada eje o grado de movimiento puede ser lineal o rotacional, este concepto está ligado a la complejidad de la máquina, esto es, entre más ejes tiene una máquina más compleja es o tiene mayor capacidad de maquinar piezas complejas.

Los ejes de las máquinas están encargados de los movimientos que tienen que hacer la herramienta para el proceso de manufactura que se requiere. Los ejes son llamados con letras, comúnmente los ejes lineales son X, Y, y Z, y los ejes rotacionales son A, B y C.

2.2.1.4 Movimiento de ejes

Si un tornillo le colocamos una tuerca, y giramos el tornillo evitando que la tuerca gire, la tuerca se desplazará proporcionalmente al giro del tornillo.

El planteamiento anterior es el principio de movimiento en las máquinas de CNC, sin embargo el tornillo del que hablamos no es un tornillo común, entrando en detalle, si hablamos de un tornillo común, de cuerda triangular, tendremos el problema de que existe cierto juego entre el tornillo y la tuerca, tendremos el problema de que existe cierto juego entre el tornillo y la tuerca, y si elimináramos éste juego, la fuerza necesaria para mover la tuerca sería muy alta, a la par que se desgaste entre el tornillo y tuerca nos pondría en el caso del juego en poco tiempo. Si usáramos una cuerda cuadrada, el caso sería más exactamente el mismo, salvo que el tornillo resistiría más que con la cuerda triangular.

Para una máquina de control numérico se requiere de un sistema tuerca-tornillo con un juego mínimo, de poco desgaste y que requiera de poca potencia para moverse; la solución es el tornillo embalado. El tornillo embalado tiene un perfil semicircular como se muestra en la figura N° 7.

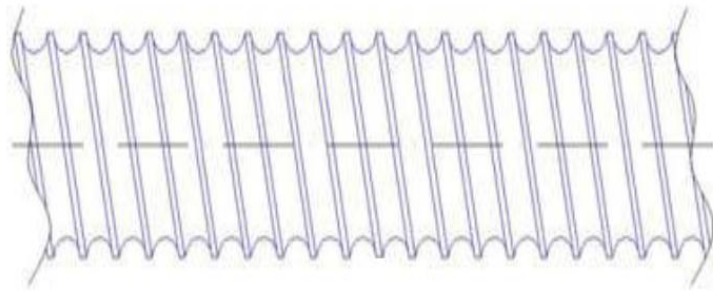


Figura N° 7: Tornillo de cuerda triangular

Y en contraparte, la tuerca lleva la otra mitad de la circunferencia (ver la figura N° 8). Esa circunferencia es con la finalidad de guiar una línea de balines que corre a todo lo largo de la cuerda del tornillo.

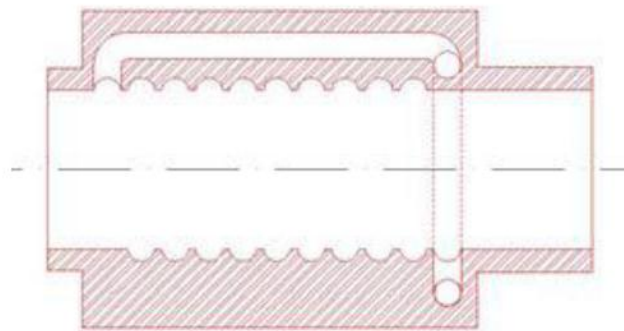


Figura N°8: Tuerca

Dentro de la misma tuerca existe un canal que permite a los balines correr libremente y regresar desde el último hasta el primer filete. Por otro lado, la tuerca se mantendrá sujeta al tornillo con un juego prácticamente inexistente, y por otro, siendo que rueda sobre balines, la potencia necesaria para mover la tuerca es mínima (incluso el peso de la tuerca es suficiente para moverse si el tornillo es puesto en posición vertical). Debe notarse que la tuerca no reposa sobre el tornillo, sino sobre los balines.

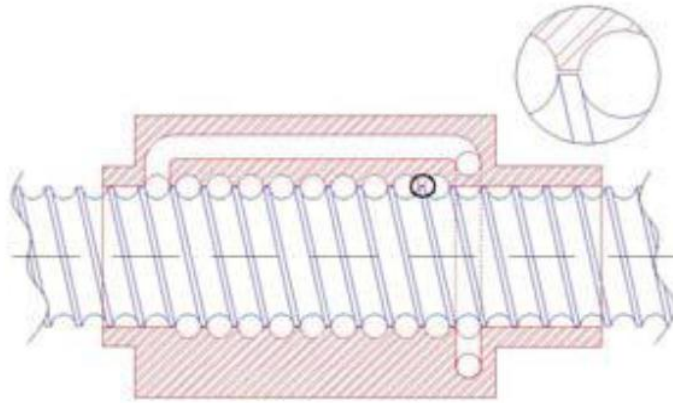


Figura N° 9: Ballscrew o tornillo embalado

A esto se le conoce como un tornillo embalado, o Ballscrew (Ver figura N° 9). Y es la base mecánica de las máquinas de control numérico. El interior de la tuerca está sellado para evitar que la viruta entre hacia los conductos de balines.

2.2.1.5 Características del CNC

Las máquinas de CNC cuentan con más que ejes de movimiento. Estos ejes de movimiento se motorizan para los desplazamientos de los carros según sus ejes principales ver figura N° 10.

Algunas de estas características son:

- Dispone de una posición fija denominado punto de referencia.
- Tienen desplazamientos de la herramienta a lo largo de la línea recta.
 - a. En línea recta.
 - b. Desplazamiento mecánico cónico
 - c. Desplazamiento en forma de arco

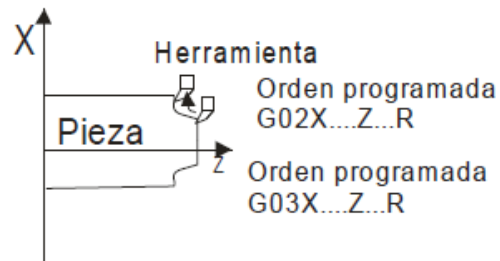
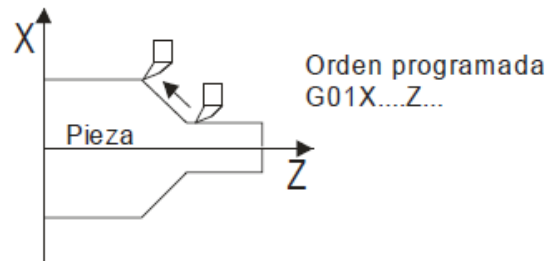
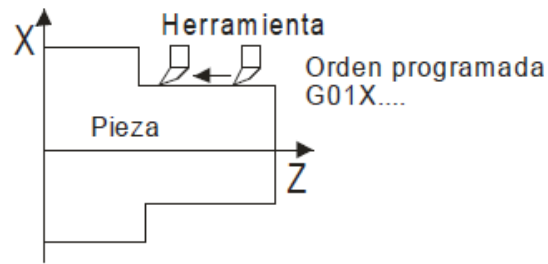


Figura N° 10: Tipos de orden de desplazamiento

- Avance: es el desplazamiento de la herramienta a una velocidad específica para cortar o mecanizar una pieza.
- Plano de la pieza y desplazamiento de la herramienta.
- Velocidad en mecanizado que es la velocidad de la herramienta respecto a la pieza cuando se está mecanizando.
- Selección de la herramienta empleada para distintas operaciones de mecanizado.
- Orden para operaciones.
- Configurar el programa.
- Márgenes de desplazamiento de la herramienta.

Otra característica más de los equipos CNC es el uso de códigos G. Estos se subdividen en dos tipos: el código G simple y el código G modal.

A continuación la tabla N° 1 muestra algunos códigos G y sus funciones:

| SISTEMA DE CÓDIGO G | FUNCIÓN |
|------------------------------------|--|
| G00 | Posicionamiento (avance rápido) |
| G01 | Interpolación lineal |
| G02 | Interpolación circular horaria |
| G03 | Interpolación circular anti horaria |
| G04 | Temporización |
| G10 | Introducción de datos |
| G17 | Selección plano Xp Yp |
| G18 | Selección plano Zp Zp |
| G19 | Selección plano Yp Yp |
| G20 | Entrada de datos en pulgadas |
| G21 | Entrada de datos en valores métricos |
| G22 | Función límite de recorrido activada |
| G23 | Función límite de recorrido desactivada |
| G27 | Comprobación vuelta al punto de referencia |
| G28 | Vuelta al punto de referencia |
| G31 | Mecanizado intermitente |
| G32 | Tallado de rosca |
| G34 | Tallado de rosca paso variable |
| G50 | Fijación de sistemas de coordenadas |
| G65 | Llamado de macro |
| G98 | Avance por minuto |
| G99 | Avance por revolución |

Tabla N° 1: Códigos G y funciones

El código G puede introducirse manualmente a los equipos CNC o se pueden generar archivos que se realizan por un software especializado como por ejemplo: Autocad, Inkscape, Orcad, etc. Estos programas permiten crear diseños no sólo para aplicaciones de ingeniería mecánica, también para diseñar circuitos impresos. Los archivos generados contienen un formato especial que puede ser interpretado por el CNC y que contienen los datos más importantes del diseño como es la inicialización, las coordenadas, la escala, y la finalización, por mencionar algunos parámetros importantes.

2.2.1.6 Movimiento de los ejes en un CNC

Ballscrew es el medio mecánico para desplazar la tuerca, es cuestión de aplicar sus ventajas. Si podemos controlar la velocidad, posición y aceleración del motor, y al motor conectamos el tornillo, entonces podemos controlar la velocidad, posición y aceleración de la tuerca.

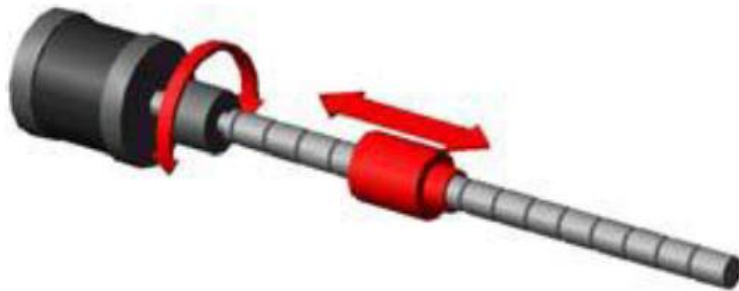


Figura N° 11: Sistema tuerca – tornillo 1 eje

Si a este sistema tuerca-tornillo (ver figura N° 11), le conectamos otro sistema perpendicularmente, entonces tendremos que la segunda será controlada, no solo a lo largo del tornillo, sino en un plano.

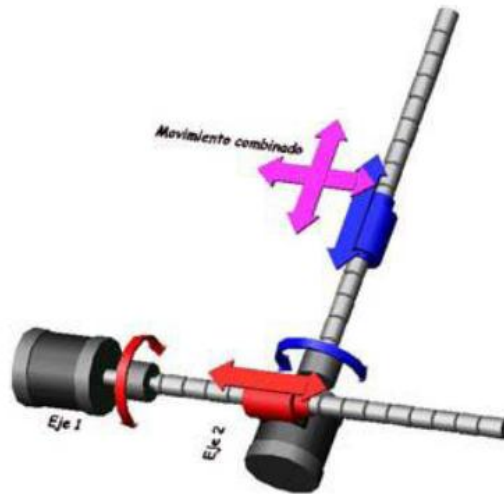


Figura N° 12: Sistema tuerca – tornillo 2 ejes

Podemos citar a este ejemplo el caso de un torno; los dos ejes que conforman el movimiento de la torreta, más el eje C del chuck (ver figura N° 12). En el caso de un centro de maquinado se adjunta un tercer eje.

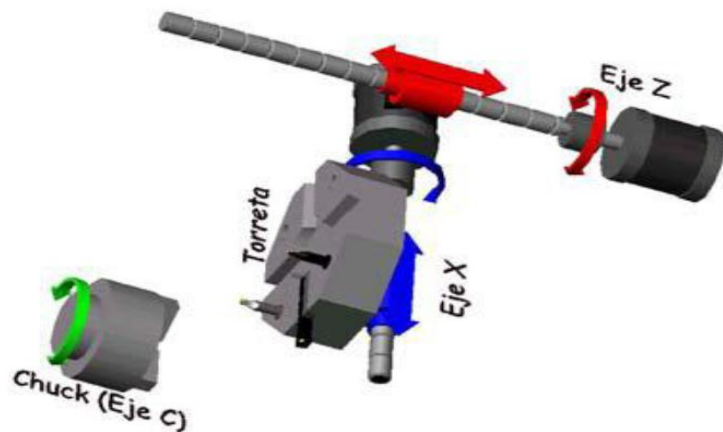


Figura N° 13: Sistema tuerca tornillo de 2 ejes y tercer eje independiente

Como puede verse aquí en la figura N° 13, el primer eje corresponde a eje longitudinal (Z) mientras que el segundo corresponde al eje transversal (X). Puede apreciarse del mismo modo que en el caso de los dos tornillos, que la tercera tuerca es controlada, no solo en el plano sino en el espacio.

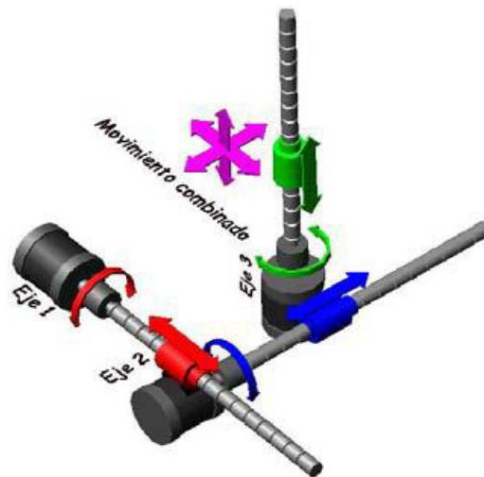


Figura N°14: Sistema tuerca – tornillo 3 ejes

En algunas ocasiones, el tercer eje no se fija a los otros dos, en su lugar, se fija a un cabezal que sostendrá el husillo que hará girar a la herramienta de corte, mientras que la prensa que sujeta a la pieza a cortar se sujeta a los otros dos ejes (ver figura N° 14). Esto, para efectos de la pieza a cortar no importa, pues, en lo que la pieza concierne, la punta de herramienta se mueve en el espacio.

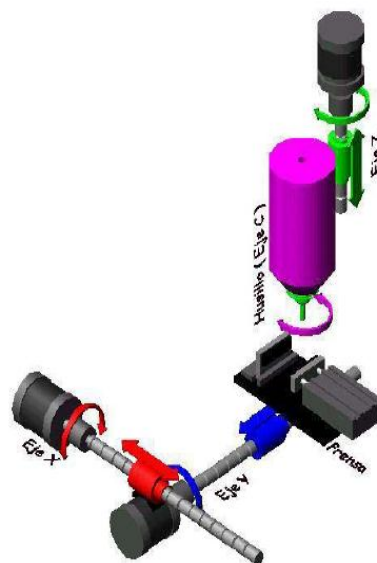


Figura N° 15: Planteo de sistema para fresadora 3 ejes

Aquí como puede apreciarse en la figura N° 15, se cuenta con los tres ejes, X y Y para el movimiento en el plano de la prensa, y Z en el cabezal, para desplazar la punta de la herramienta en el tercer eje. En las figuras anteriores se muestran imágenes en las que los tornillos van conectados directamente a los motores.

Estos, claro, son sólo uno diagramas ilustrativos. Normalmente para mejor control y rendimiento de potencia, el motor está conectado a una caja de transmisión. Sin embargo, de poco sirve contar con un sistema mecánico de precisión y bajo requerimiento de potencia si no podemos controlar esos motores para realizar los movimientos que la pieza a trabajar requiere.

2.2.1.7 Motores de un CNC

Existen varios tipos de motores eléctricos, cada uno con propiedades distintas. Los más comunes, son los motores de CC (corriente continua). En este tipo particular, podemos controlar la velocidad variando el voltaje en un cierto rango; con una curva apropiada de voltaje podemos parcialmente controlar su aceleración, pero no podemos controlar su posición y menos aún su torque.

Otro tipo de motor muy difundido es el motor CA (corriente alterna) que puede variar su torque en base al voltaje suministrado dentro de un pequeño rango y puede variar su velocidad en base al cambio de frecuencia de CA que se le entre las bobinas, la velocidad, aceleración y posición. Sin embargo, dada la naturaleza de este motor, el torque puede controlarse muy poco, y al igual que el motor CC se corre el riesgo, en caso de sobrecarga de quemar el motor. Si el sistema tuerca-tornillo a controlar es de baja potencia, el suministra. Sin embargo no se puede controlar su posición.

Los motores paso a paso son motores alimentados por DC, lleva un arreglo de varias bobinas que permite variar, a través de la velocidad de cambio de pulso motor a pasos es una solución viable y económica.

Los servomotores son motores especiales que cuentan con una etapa de retroalimentación; esta le indica al motor cuánto se ha movido y si requiere moverse más, o incluso, indica al motor aplicar fuerza para mantener la posición actual. Este es el tipo de motores ideales para una máquina CNC, sin embargo su costo reduce su difusión.

2.2.1.8 Sistema de control de ejes de una CNC

Todo movimiento puede descomponerse en tres vectores instantáneos. Estos serían, X, Y y Z. Para efectos de un ejemplo, usaremos solo dos ejes, X y Y. Ahora, si intentamos llevar la punta de la herramienta de un punto A a un punto B como se observa en la figura N°16, requerimos que los motores realicen un movimiento acorde con la distancia a recorrer en cada eje, más o menos del siguiente modo:

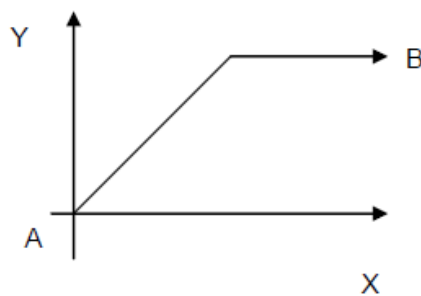


Figura N° 16: Movimiento combinado diagonal y recto

Puede notarse en la figura N° 17, siendo las distancias a recorrer de los ejes son desiguales, tendremos un movimiento a 45° hasta que uno de los motores agote la distancia a recorrer. A partir de ahí, el movimiento será tan solo en el eje que aún no ha llegado a su destino; esto es lo que conoce

como un movimiento no interpolado. Es decir, cada motor va por su cuenta. Pero, si nuestro control puede calcular el tiempo estimado de arribo en base a la velocidad normal de uno de los motores, y con ello calcular una velocidad conveniente para que el segundo motor llegue a su punto final al mismo tiempo, entonces tendremos un movimiento interpolado.

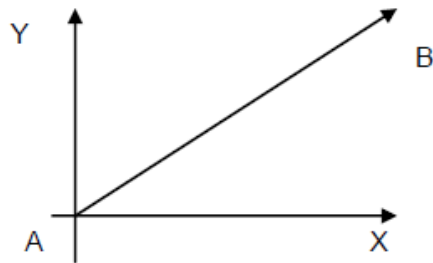


Figura N° 17: Movimiento en diagonal

Este movimiento en particular es mucho más útil que el anterior, pues nos permite trazar líneas rectas entre puntos, en lugar de depender de velocidades fijas de los motores. Más aún, si podemos variar controladamente las velocidades en los motores, por decir algo, de modo senoidal, mientras que el otro lo variamos de modo cosenoidal, podemos trazar arcos regulares con la trayectoria de la herramienta. Si podemos variar esta velocidad para que sea fija, o variable de modo regular, podemos realizar curvas no regulares (ver figura N° 18).

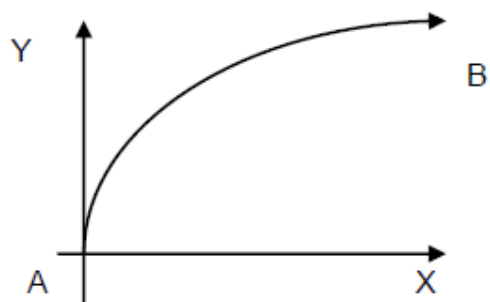


Figura N° 18: Movimiento en curva

Esta habilidad para controlar las velocidades de motores y ejes a través de ecuaciones matemáticas es lo que da este tipo de máquinas su nombre control numérico computarizado. En los ejemplos anteriores se tiene que, de un punto A se parte para llegar a un punto B, en programación CNC existen dos métodos para hacer este movimiento: por medio del sistema absoluto y por sistema relativo.

2.2.2 Circuito impreso

Una placa de circuito impreso (PCB) es una plancha de material rígido aislante, cubierta por unas pistas de cobre en una de sus caras o en ambas, para servir como conductor o de interconexión eléctrica entre los distintos componentes que se montarán sobre ella.

La materia prima consiste en una plancha aislante, típicamente de “fibra de vidrio” o “baquelita”, cubierta completamente por una lámina de cobre. Dependiendo del tipo de placa, el cobre puede ir a su vez protegido por una capa de resina fotosensible.

Cuando existe soldadura sólo en una cara son llamados monocapa como se figura N° 19 y cuando existe soldadura en ambas caras son conocidas como bicapa como se figura N° 20.

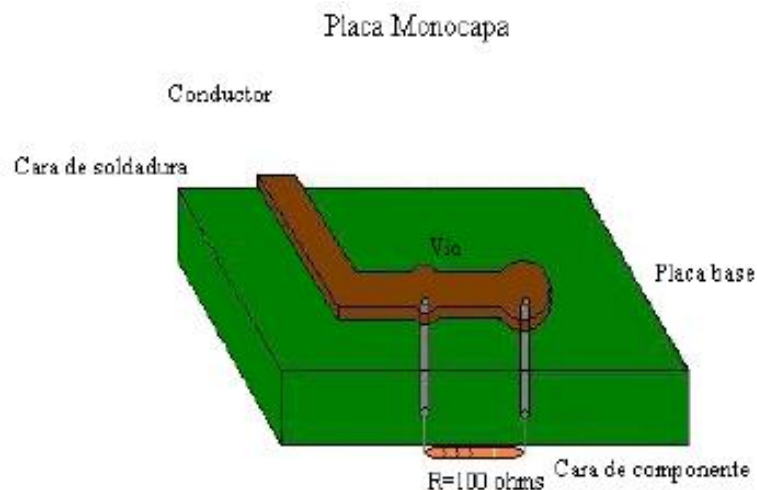


Figura N° 19: Placa monocapa

En algunas situaciones es necesario emplear pistas en la cara el componente como en la soldadura. En este caso los agujeros pasantes deben ser metalizados, es decir recubiertos de un material semiconductor para permitir la unión eléctrica entre las pistas de la cara de la soldadura y la de componentes como se muestran en la figura.

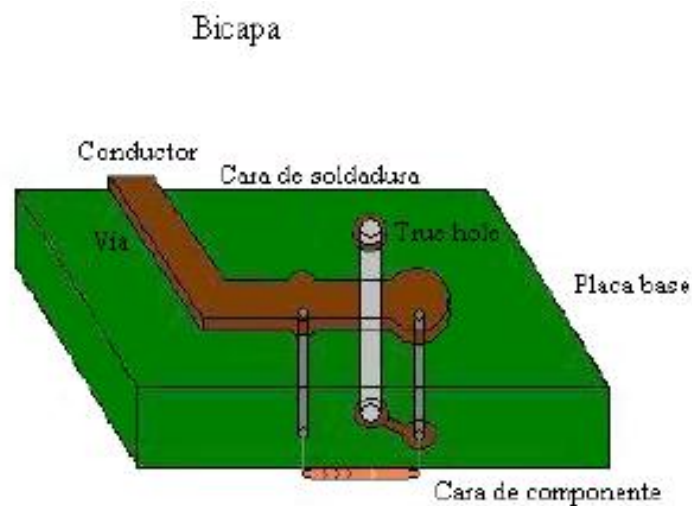


Figura N° 20: Placa doble cara o bicapa

Con la incorporación de la tecnología de montaje superficial (SMT: superficial mount technology) la colocación de componentes no solo se hace por la cara de componentes, también por la cara de la soldadura, aumentando así el número de componentes sobre la placa.

Conforme aumenta el número de componentes a colocar, también aumenta el número de pistas a realizar, siendo insuficientes las dos caras de la placa. Para evitar aumentar el área de la placa se desarrolla la tecnología multicapa en la cual se generan partes de caras de placas que se van superponiendo en forma de “sándwich” y que se conectan en varias vías.

2.2.2.1 Funciones básicas de un circuito impreso

- Soportar sus propios componentes.

- Soportar sus interconexiones eléctricas.

Todo ello siguiendo unas reglas establecidas a la vista de unas tolerancias impuestas por la naturaleza de los equipos o sistemas electrónicos.

En el diseño de los circuitos impresos, encontraremos una serie de factores variables que habrán de ser seleccionados y combinados de una forma óptima en cada caso.

La colocación de los componentes en la propia placa base del circuito, el material dieléctrico de la base, el tipo de los conductores, el número de capas de conductores, la rigidez, la densidad o compactado del equipo en la placa, etc. , combinados de manera adecuada influirán en el rendimiento, calidad y coste del producto .

En el diseño habrá que pensar también en las condiciones de Fabricación, creando una información adecuada para Fabricar, conociendo los medios y los costos que van a intervenir en las distintas operaciones a seguir, para encontrar procedimientos viables y rentables.

2.2.2.2 Elementos básicos del circuito impreso

- a. Soporte aislante.
- b. Agujeros para montaje de componentes y/o interconexión.
- c. Conectores de interconexión.
- d. Terminales de entrada y de salida.

2.2.2.3 Clasificación de las placas impresas

Categorías de las placas impresas según su densidad en componentes y en interconexiones.

Se consideran tres categorías básicas según sus densidades en orden de menor a mayor:

- a. De Simple Cara, con conductores en una sola superficie plana de la base aislante.
- b. De Doble Cara, con conductores en ambas caras de la base aislante, con agujeros metalizados para la interconexión entre caras, u otros medios.
- c. Multicapa con tres o más capas de conductores separados por material aislante y usualmente interconectados a través de agujeros metalizados.

2.2.2.4 Densidades de los circuitos impresos

En toda placa impresa, es necesario conjugar la limitación de su superficie con los elementos (componentes e interconexiones) que es necesario equipar según indica el circuito. Existen o pueden existir una serie de incompatibilidades, dada la diversidad de tamaños y formas de sus componentes, el número de éstos, la complejidad de sus interconexiones, etc.

Es deseable, según esto, conocer una medida que de idea del orden de la densidad de una placa impresa y que permita tipificarlas.

Se toma como unidad de densidad el número de agujeros, para montar componentes, por decímetro cuadrado de superficie útil. Esta unidad no es perfecta, pero puede servir como referencia para conocer, en una primera aproximación la porción de circuito que puede montarse eficazmente en cada caso.

Usualmente, los valores indicados en la tabla N^a 2 corresponden con las distintas clases de placas impresas.

| CIRCUITOS IMPRESOS | NUM. DE AGUJEROS PARA MONTAJE POR DECÍMETRO CUADRADO DE SUPERFICIE ÚTIL |
|---------------------------|--|
| SIMPLE CARA | Entre 50 y 150 |
| DOBLE CARA | Entre 150 y 300 |
| MULTICAPA | Más de 300 |

Tabla N° 2: Número de agujeros por tipo de circuito impreso

2.2.2.5 Sistema de clasificación

Existe un sistema de clasificación de placas impresas por sus densidades, que proporciona el grado de concentración de conductores, nudos y agujeros. Este dato, junto con otros factores tales como el tamaño de la placa, determinan las tolerancias permitidas en las distintas fases del diseño y de los procesos de fabricación.

El sistema de clasificación consiste en dos dígitos. El primer dígito representa el tipo de placa (número de capas y tipo de conexiones a través de ellas), y el segundo dígito está relacionado con el máximo de concentración local de conductores.

Para el primer dígito, podemos formar el cuadro N° 3 de clasificación siguiente:

| 1er Dígito | Tipo de Placa |
|-------------------|---|
| 1 | Simple o doble cara, sin agujeros metalizados |
| 2 | Doble cara con agujeros metalizados |
| 3 | Multicapa, con agujeros metalizados |

Tabla N° 3: Dígito según tipo de placa

El segundo dígito de la clasificación indica la máxima concentración de conductores, de tal manera que cuanto mayor sea la densidad en la placa impresa, más elevado será el valor de este dígito.

Para considerar la cuantía de la densidad de las placas impresas se introducen las tres variables siguientes:

- a. Anchura nominal de los conductores.
- b. Separación nominal entre los conductores.
- c. Diferencia entre el diámetro nominal de los nudos y el diámetro nominal de los agujeros correspondientes.

Según esto, el segundo dígito de la clasificación de una placa impresa en diseño, será el menor número para el cual los valores mínimos correspondientes a las variables arriba indicadas, están satisfechos sobre toda la placa.

2.2.2.5.1 Límites mínimos dimensionales para cada placa impresa

A continuación, en cuadros por separado, se establecen los límites mínimos que definen a cada clase de placa impresa, en cuanto a densidad se refiere, por el sistema de dos dígitos.

- a) Placas sin agujeros metalizados

El primer dígito de la clasificación de este tipo de la tabla N° 4 de placas será 1, y el segundo tomará los valores 1, 2 ó 3 según los tres parámetros a), b) y c) indicados.

| Dimensiones (mm) | Clasificación densidad | | |
|---------------------------------------|-------------------------------|-----------|-----------|
| | 11 | 12 | 13 |
| Anchura nominal mínima del conductor. | 0.8 | 0.6 | 0.4 |

| | | | |
|--|-----|-----|------|
| Separación nominal mínima entre conductores. | 0.7 | 0.5 | 0.35 |
| Diferencia mínima entre diámetro nominal del nudo y del agujero. | 1.6 | 1.2 | 0.8 |

Tabla N° 4: Límites dimensionales para placa sin agujeros metalizados

b) Placas con agujeros metalizados

El primer dígito será 2 y el segundo tomará los valores 1, 2, 3, o 4 según los parámetros a), b) y c) indicados de la tabla N° 5.

| Dimensiones (mm) | Clasificación densidad | | | |
|---|------------------------|-----|------|-----|
| | 21 | 22 | 23 | 24 |
| Anchura nominal mínima del conductor | 0.8 | 0.5 | 0.4 | 0.3 |
| Separación nominal mínima entre conductores | 0.7 | 0.5 | 0.35 | 0.3 |
| Diferencia mínima entre diámetro nominal del nudo y del agujero | 1.3 | 0.8 | 0.64 | 0.6 |

Tabla N° 5: Límites dimensionales para placa con agujeros metalizados

c) Placas multicapa

El primer dígito será 3 para el caso de tres capas y el segundo dígito será 3 el siguiente cuadro N° 6.

| Dimensiones (mm) | Clasificación densidad |
|--|---------------------------|
| | 33 |
| Anchura nominal mínima del conductor | 0.4 |
| Separación nominal mínima entre conductores. | 0.35 |
| Diferencia mínima entre diámetro nominal del nudo y nominal del agujero. | 0.64 |

Tabla N° 6: Límites dimensionales para placa multicapa

2.2.2.6 Clasificación de las técnicas para la elaboración de circuitos impresos

Se ha clasificado los métodos de elaboración de PCB en función de las técnicas para eliminar el excedente de cobre; así como el método de transferencia del arte de circuito a la placa de cobre, tal como se indica a continuación:

2.2.2.6.1 Elaboración de PCB mediante ataque químico

- Elaboración y Transferencia del Arte de Circuito Impreso a la placa de cobre:
- Manual o Artesanal: El circuito se realiza directamente sobre la placa de cobre, se utiliza solo en casos de circuitos electrónicos de baja complejidad como se muestra en la figura N° 21.



Figura N° 21: Creación de placa PCB por ataque químico

- Marcador con Tinta Indeleble, Resistente al Agua, contiene acrílico,
- Cintas plásticas adheribles y/o Transferibles de Pistas y sus variaciones según encapsulados de componentes: PAD, SIP, DIP o DIL, entre otros.
- Con ambos recursos
- Diseño Asistido por Computadora o por sus siglas en inglés (CAD):
- Transferencia o impresión del Arte a la placa de Cobre:
 - Serigrafía: Malla o Pantalla (Marco y Tela), Paleta, Foto emulsión.
 - Pintura Acrílica
 - Papel "Transfer".
 - Tarjeta Sensibilizada
 - Ataque Químico con ácido para la oxidación o eliminación del cobre (Cu) no deseado.

2.2.2.6.2 Elaboración de PCB por máquinas Fresadoras o Sin Químicos:

- Diseño del PCB con la ayuda de un CAD.

- Máquinas Fresadoras o Router (todas requieren de CAD) para eliminar el Cu no deseado.

Conforme avanza la tecnología, los circuitos impresos se convierten en parte fundamental en el desarrollo de circuitos electrónicos, cada vez con mayor densidad y por ende exige exactitud y precisión, esto hace necesario la utilización de diseños asistidos por computadora, es por ello que nuestro trabajo, se centra en definir y especificar este método, así como también la eliminación de excedente de cobre con ataque químico, dejando a un lado, no por menos importante, el método manual o artesanal y la elaboración de PCB por máquinas Fresadoras.

2.2.2.7 Técnicas y consideraciones en el diseño del arte de PCB

2.2.2.7.1 Pasos generales para la elaboración de circuitos impresos utilizando los métodos de ataque químico

- Diseño: Un circuito complejo requiere el uso de herramientas computacionales que permitan diseñar y simular el esquema electrónico y su arte de circuito impreso.
- Cortado: Conocido el tamaño del circuito impreso, se procede a realizar el corte de la tarjeta.
- Impresión en la placa de cobre: una vez que la máscara esta lista, se procede a grabarla en la placa, este procedimiento depende de la técnica a utilizar.
- Atacado del cobre: se inserta la placa de cobre previamente grabada, en soluciones acidas que eliminan el cobre no deseado.
- Limpieza y taladrado: se realiza el lavado y limpieza de la placa para eliminar todas las impurezas, luego

se perforan los orificios en donde se colocaran los componentes.

- Soldadura: etapa donde se realiza el montaje (colocación y soldadura) de los componentes.
- Pruebas de Funcionamiento: Antes de realizar interconexiones se verifica el funcionamiento del circuito (Cortocircuitos, circuitos abiertos, soldaduras frías, entre otros.)

2.2.2.7.2 Herramientas computacionales de diseño

En el diseño, es necesario definir el esquemático del circuito y el diagrama de conexiones que formarán las pistas de cobre sobre la placa. Generalmente se hace en dos fases; en primer lugar, partiendo de las especificaciones sobre la funcionalidad del circuito, se deciden los componentes a utilizar y las interconexiones necesarias entre ellos, y a través del editor de esquemas del software se realiza el diagrama esquemático. Después, con esa información se define la máscara en el editor del arte del PCB, que es una representación virtual de los componentes sobre la placa, y se establece la forma física de las conexiones entre ellos.

Es recomendable la Elaboración y Simulación del Circuito Esquemático previo a la elaboración del PCB.

2.2.2.7.3 Elección adecuada de los componentes

- Tomar en cuenta las limitaciones de inventarios en los proveedores: se deben adquirir todos los componentes y conectores o tener seguridad del despacho de los encapsulados a utilizar, motivado a que posiblemente se fabrique una tarjeta, que no

pueda ser operativa con los componentes vigentes en el momento de su uso.

- Antes de hacer la placa verifique cuidadosamente la interconexión y numeración de cada componente y de cada circuito integrado, así como también la correspondencia del tamaño de cada uno de ellos en el PCB.

2.2.2.7.4 Efecto resistivo en las pistas

- Diseñar las pistas teniendo en cuenta la longitud, grosor y máxima corriente que deban conducir: es recomendable utilizar herramientas de software para determinar y calcular las dimensiones de pistas a usar según parámetros solicitados; ya que estas dimensiones, mal diseñadas, podrán incrementar el valor de una resistencia no deseada y causar problemas por caídas de tensión o servir de fusible al limitar el paso de corriente. Todo material conductor presenta una resistividad propia y según las dimensiones del mismo, tendremos una resistencia eléctrica, expresado por la ecuación 1:

Ecuación 1. Resistencia eléctrica

$$R = \rho \frac{l}{A} \dots \dots \dots (1)$$

En donde:

R: Resistencia eléctrica, expresado en Ohm [Ω]

p: Resistividad del Material en unidades de [Ωm]

l: Longitud que por las unidades del Sistema Internacional es el metro [m]

A: Área transversal dado en [m^2]

- En los circuitos de instrumentación y de medición, ubicar los componentes de tal forma que la longitud de las pistas sea lo más pequeña posible, para evitar efectos de carga en la línea de interconexión.

2.2.2.7.5 Efecto térmico

- Estudiar la colocación de los componentes teniendo en cuenta la interconexión, interferencias térmicas e interferencias electromagnéticas. Las condiciones de temperatura podrán causar ligeras variaciones en el valor de resistencia, motivado a que la resistividad de un material (ecuación 2) dependerá de las condiciones térmicas en las que se encuentre:

Ecuación 2. Resistividad de un material.

$$\rho = \rho_o(1 + \alpha(T - T_o))\dots\dots(2)$$

En donde:

A: Coeficiente de Temperatura

ρ_o : Resistividad a una Temperatura de referencia

T: Temperatura actual

T_o : Temperatura referencial

- Colocar disipadores a los dispositivos de potencia, acompañado de grasa termo conductivo entre ellos. Los espacios de aire se deben evitar motivados a que es un mal conductor térmico, lo que implicaría

una mala disipación, recalentamiento y daño de componente.

- Ubicar los disipadores en lugares ventilados y alejados de componentes susceptibles a la temperatura.
- Efecto Capacitivo e Inductivo
- Para evitar corrientes inducidas, producto de circuitos de alta potencia, se debe distanciar los circuitos de control de los circuitos de potencia.
- Se conoce que la capacitancia para placas paralelas viene expresado de la forma.

Ecuación 3. Capacitancia para placas paralelas

$$C = \epsilon_0 \frac{\epsilon_r A}{d} \dots \dots \dots (3)$$

En donde:

ϵ_0 : constante dieléctrica del vacío

ϵ_r : constante dieléctrica o permitividad relativa del material dieléctrico entre las placas.

A: el área efectiva de las placas

d: distancia entre las placas o espesor del dieléctrico

- Por lo anterior, para evitar el efecto de capacitancias parásitas, se debe evitar el paralelismo entre pistas o planos y se aconseja utilizar rectas horizontales en una cara (Cara de Componentes o lado Superior) con rectas verticales en la otra (Lado inferior o cara de soldadura)
- Para reducir el ruido de conmutación en los circuitos digitales, colocar un condensador de 0,1uF entre

fuente y tierra, lo más cercano posible a cada integrado. Colocar cada 10 integrados un condensador de 10uF y por cada módulo o tarjeta electrónica colocar un condensador de 47uF [2,4].

- Orientar de forma perpendicular al PCB las bobinas y transformadores, para evitar las influencias magnéticas que sobre otros circuitos; ya que en un solenoide, el campo magnético se concentra sobre su eje axial.
- Para los circuitos de alta frecuencia, es recomendable que las curvas de las pistas no superen un ángulo de 45°, ya que podría producirse un auto inducción sobre la misma, deformando su señal.
- Para evitar las EMI (Interferencias Electromagnéticas) y brindar protección eléctrica, se deben separar los planos de tierra analógico y digital o utilizar acoplamientos (transformador, opto acopladores, aisladores de radio frecuencia, etc.). Para circuitos susceptibles, se debe dejar espacio para fijar las jaulas de Faraday.

2.2.2.7.6 Transferencia del arte a la placa

- Colocar etiqueta o texto al PCBs, para identificar el lado adecuado a transferir en el cobre.
- Verifique que la impresora no genere discontinuidades y que imprima a la escala correcta.

2.2.2.7.7 Pruebas de funcionamiento

- Poner puntos de prueba (TP o test point) a la salida de cada etapa, en lugares de la placa donde se pueda conectar instrumental fácilmente.

- Simplificar el ensamblaje.
- Diseñar las placas de forma estandarizada y modular, de tal manera que la fabricación y ensamblaje sea práctico.

2.2.2.7.8 Otras consideraciones

- No poner pistas, ni colocar componentes cerca de los bordes de las placas donde puedan tener contacto con los tornillos de fijación, guías o con la estructura.
- La separación mínima entre 2 pistas adyacentes debe de ser 0.8 mm, lo que garantiza un buen aislamiento eléctrico de hasta 180 voltios, en condiciones normales
- Realizar los Pad (Área Plana conductiva donde se realiza las soldaduras de componentes) en función del tamaño y peso de los componentes. Así como también de acuerdo a las fuerzas y tensiones mecánicas que deba soportar.
- Es recomendable que los componentes puedan cambiarse sin necesidad de extraer otro.
- Llenar con planos de tierra la parte de la placa que no lleve pistas; así se evita la contaminación de los oxidantes (ácidos).

Un ejemplo de aplicación de las técnicas de diseño en el CAD se observa en la figura N° 22, muestra el diagrama esquemático de un circuito electrónico y la figura N° 23, muestra el arte de PCB de dicho diagrama, en este caso el programa utilizado fue el EAGLE.

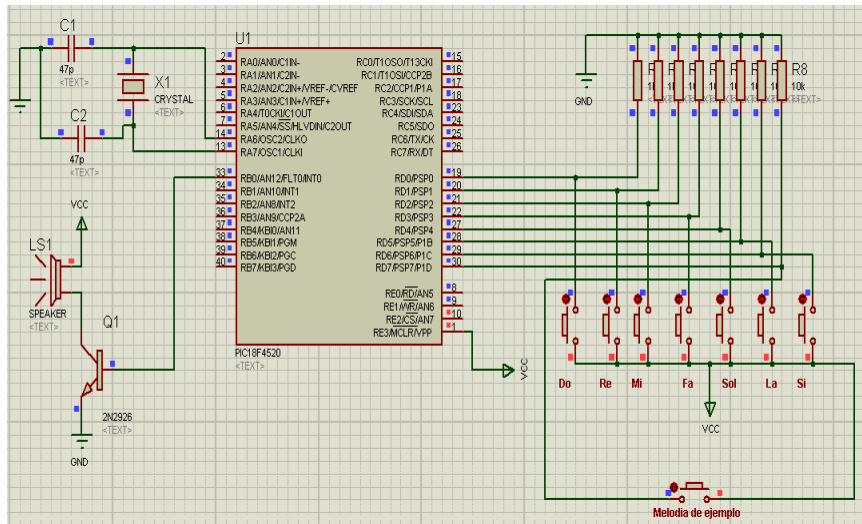


Figura N° 22: Diseño de circuito eléctrico

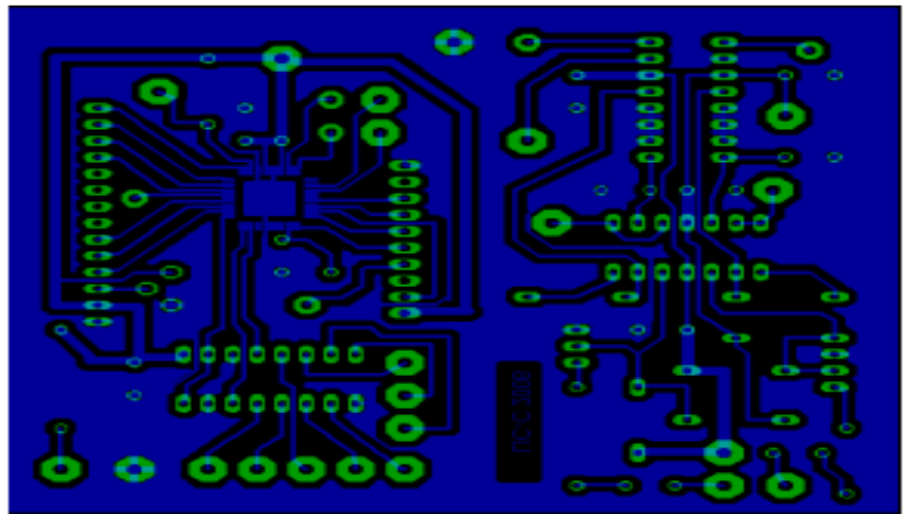


Figura N° 23: Diseño de circuito de placa en Orcad

2.2.3 Arduino

Arduino es una plataforma libre de computación de bajo coste basada en una placa de entrada-salida y en un entorno de desarrollo IDE que implementa el lenguaje Processing/WiringHardware.

Arduino se puede usar para desarrollar objetos interactivos automáticos o conectarse a software en el ordenador (Pure Data, Flash, Processing; MaxMSP).

2.2.3.1 Hardware

2.2.3.1.1 Elementos de la placa

La placa es una placa de circuito impreso donde va instalado el microprocesador, la memoria, las conexiones de entrada y salida y la conexión para el puerto usb, como se muestra en la figura N° 24.

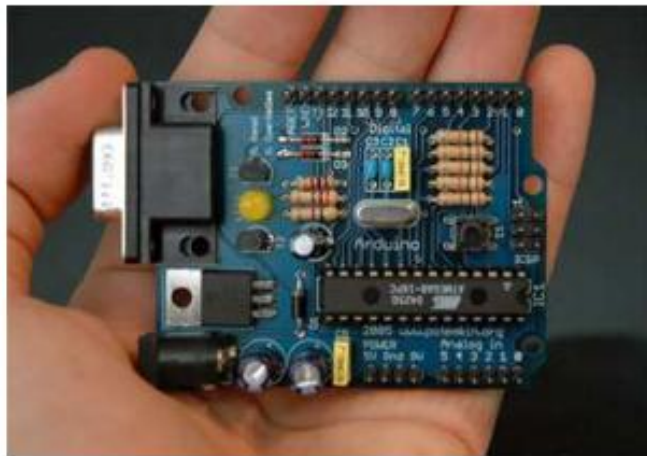


Figura N° 24: Placa Arduino Uno

Puerto usb: a través de él se cargan las instrucciones a ejecutar, el programa que es realizado en el entorno de programación de arduino. Comunicación Arduino-Ordenador.

Microprocesador: realiza las instrucciones almacenadas en el programa de forma cíclica. Es un circuito integrado que contiene muchas de las mismas cualidades que una computadora. Escribe en los pines DS2-13 y lee en los DE2-13 AE0-5.

Pines de entrada y salida: Permiten conectar elemento que dan información y crean actuaciones.

Botón de reset: permite resetear el programa y permite cargar uno nuevo.

2.2.3.1.2 Elementos del microcontrolador

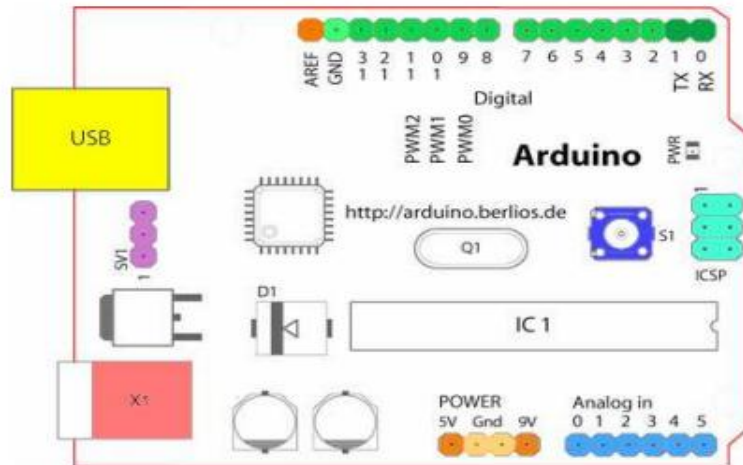


Figura N° 25: Estructura de tarjeta Arduino Uno

- Aref- Pin de referencia analógica (naranja)
- GND- Señal de tierra digital (verde claro)
- Pines digitales 2-13. Entrada y salida (verde)
- Pines digitales 0-1 / entrada y salida del puerto serie: TX/RX (verde oscuro)
- Botón de reset- Pulsador (azul oscuro)
- Pines de entrada analógica 0-5 (azul claro)
- Pines de alimentación y tierra (fuerza: naranja, tierra: naranja claro)
- Entrada de la fuente de alimentación externa (9-12V DC) X1 (rosa)
- Puerto USB (amarillo)

2.2.3.1.3 Entradas y salidas

El microcontrolador recibe información de las entradas (read), la procesa y escribe un 1 o un 0 (5v ó 0v) en las salidas (Write), actuando sobre el dispositivo que tenemos conectado.

Al microcontrolador por lo tanto conectamos unos sensores a la entrada y unos actuadores a la salida como se observa en la figura N° 26, para que en función del programa y de la lectura de los sensores se produzcan una serie de actuaciones.



Figura N° 26: Diagrama de funcionamiento de Arduino

Señales analógicas y digitales

Una señal es analógica cuando puede tomar infinitos valores entre su valor mínimo y máximo.

Elementos analógicos: Potenciómetro, LDR, NTC, Zumbador, motor eléctrico, Led. Una señal es digital cuando solo puede tomar 2 valores, el máximo asociado a 1 o a cierto “on” y el mínimo asociado a cero, falso o “off”.

Elementos digitales: pulsador, detector de presencia, led, timbre.

Funcionamiento del arduino

Función DigitalWrite

High: el microcontrolador escribe un 1 en el pin de salida, es decir 5v. El led se enciende (ver figura N° 27).

Low: el microcontrolador escribe un 0 en el pin de salida, es decir 0v. El led no se enciende.

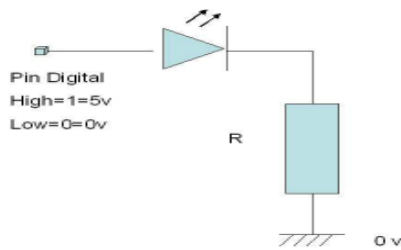


Figura N° 27: Circuito para Salida de Arduino

Función Digital Read

El microcontrolador lee la tensión en el pin de entrada si $V > 3,5v$ asigna un 1, si $V < 3,5v$ asigna un 0 (Ver figura N° 28).

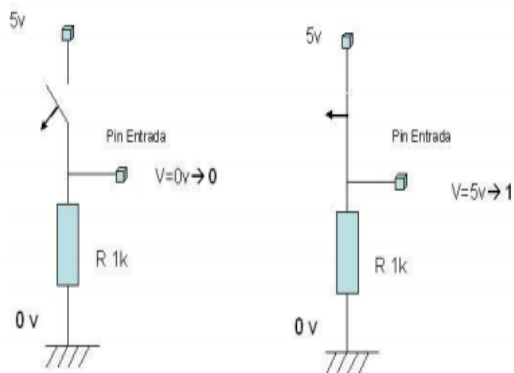


Figura N° 28: Circuito para Entrada de Arduino

2.2.3.2 Software

Para comunicarnos con el microcontrolador y cargarle los programas usamos IDE (Integrated Development Environment), sistema de desarrollo de Arduino, sirve para escribir programas, compilarlos y descargar los programas a la placa de arduino.

Un programa es una secuencia de instrucciones que son ejecutadas por una computadora o un microcontrolador, es una secuencia específica para realizar una tarea. Los programas

son escritos en diferentes tipos de lenguajes, tales como “C”, Fortran o BASIC.

El programa (lenguaje de alto nivel, “humano”) se compila (se transforma en 1 y 0) y se carga en el microprocesador que lo ejecuta de forma cíclica. Para programar el micro se utiliza el IDE de Arduino, en él se escribe un programa que se carga en el microprocesador mediante el puerto usb.

Para ello ejecutar arduino.exe desde la carpeta c:/arduino. Se debe configurar el puerto a usar (previamente hay que instalar el driver de la placa para que reconozca el puerto, buscarlo en c:/arduino/drivers/FTDI USB Drivers). Para mirar el puerto a usar Mi Pc>Propiedades del Sistema>Herramientas>Administrador de dispositivos> Puertos.

El lenguaje de programación de Arduino es muy sencillo. Un programa puede tener una estructura como:

```
//Blink: Programa que hace que un led parpadee  
//declaración de variables //Asocia el 13 al nombre ledrojo  
  
Int ledrojo = 13;  
  
//Declaración de set up //Configuración de puertos  
  
Void setup()  
{  
    pinMode(ledrojo,OUTPUT);  
}  
//Declaración de loop //Bucle infinito  
Void loop()  
{  
digitalWrite(ledrojo, HIGH); //Escribe un 1 (5V) en el Pin rojo  
delay(1000); //Espera 1 segundo = 1000ms  
digitalWrite(ledrojo,LOW); //Escribe un 0 (0V) en el Pin rojo  
delay(1000); //Espera 1 segundo = 1000ms  
}
```

Donde la columna izquierda es lo que lee el microprocesador y la de la derecha son comentarios para facilitar nuestra comprensión.

Es importante el orden y la limpieza en la realización de los programas y el uso de los comentarios //, y la tabulación en los bucles para facilitar su comprensión.

2.2.3.2.1 Estructura general

Declaración de variables: en esta parte declaramos todas las variables que vamos a usar en el programa. Nos permite asociar nombres a números lo que nos será de mucha utilidad a la hora de modificar los programas.

Void setup {} es la función de configuración de los pines de Arduino y sólo se ejecuta una vez, mientras que loop () se ejecuta una y otra vez hasta que apaguemos el sistema, o se gasten las baterías.

Bucle infinito Void loop {} es la parte del programa que se ejecuta de forma cíclica.

Comandos básicos

HIGH - 5V - TRUE (Verdadero) - 1 lógico

LOW - 0V - FALSE (Falso) - 0 lógico

2.2.3.2.2 Variables

Int entero corto

Los vectores son conjuntos de variables, se declaran como las variables normales, pero con unos corchetes y se dan los valores entre llaves, ejemplo: int vector[] = {1, 5, 23}. En inglés se llaman arrays.

2.2.3.2.3 Estructura de control

FOR (CONDICIÓN INICIAL; CONDICIÓN FINAL; INCREMENTO)

Es un comando para repetir la misma operación un cierto número de veces.

```
For(n=0; n<3; n++){  
  (Conjunto de instrucciones a repetir)  
}
```

IF(CONDICIÓN){}

Es un comando que sirve para discriminar si se dio una determinada condición. Las comparaciones son: == igualdad, != desigualdad, > mayor que, < menor que, >= mayor o igual que, <= menor o igual que. Todo lo que figure entre las llaves será ejecutado sólo si se da la condición entre paréntesis.

```
If (p<2) {  
  (Conjunto de instrucciones a repetir)  
}
```

IF ELSE

Verifica si se cumple una condición y ejecuta lo que está entre llaves, sino se cumple ejecuta lo que está debajo del else.

```
If (p<2) {  
  (Conjunto de instrucciones a repetir)  
}  
Else{  
  (Conjunto de instrucciones a repetir)  
}
```

WHILE

Ejecuta el conjunto de instrucciones entre llaves mientras se cumpla la condición.

```
While (p<2) {  
(Conjunto de instrucciones a repetir)  
}
```

2.2.3.2.4 Temporizadores

DELAY (MS TIEMPO)

Sirve para parar los procesos de la placa durante un cierto tiempo en milisegundos.

DELAYMICROSECONDS(US TIEMPO)

Sirve para parar los procesos de la placa durante un cierto tiempo en microsegundos.

MILLIS()

Devuelve el tiempo en milisegundos, desde que la tarjeta Arduino activó el programa que se está ejecutando.

2.2.3.2.5 Pines digitales

PINMODE (PIN, MODO)

Sirve para declarar un pin como entrada (INPUT) o como salida (OUTPUT)

DIGITALWRITE (PIN, VALOR)

Sirve para escribir un valor al pin, el valor podrá ser 1 lógico (HIGH) o 0 lógico (LOW)

DIGITALREAD (PIN)

Lee un valor de 0 ó 1 de la entrada digital.

2.2.3.2.6 Pines analógicos

ANALOGREAD (PIN)

AnalogWrite(pin, valor) escribe una señal al pin 9, 10, u 11 cuyo ancho de pulso se controla con valor. El ancho se codifica en 256 niveles, es decir, podremos introducir números del 0 al 255.

PWM (Pulse Width Modulation), la modulación por anchura de pulso es una técnica que empleamos para producir señales digitales que, filtradas, se comportarán como señales analógicas. El PWM en Arduino funciona a una frecuencia constante por encima de los 60Khz. Mediante el comando.

2.2.3.2.7 Comunicación con el puerto serie

SERIAL.BEGIN (SPEED)

Inicializa o abre el puerto serie asignándole una velocidad de transmisión de datos (bits por segundo). Dicha función debe ser llamada desde dentro de la estructura o sección setup(). Equivaldría a la función beginSerial ().

Int speed: velocidad de datos, en bits por segundo (baudios)

SERIAL.PRINT (DATA, DATA TYPE)

Vuelca o envía un número o una cadena de caracteres, al puerto serie. Dicho comando puede tomar diferentes formas, dependiendo de los parámetros que utilicemos para definir el formato de volcado de los números.

Data: el número o la cadena de caracteres a volcar o enviar.

Data type: determina el formato de salida de los valores numéricos (decimal, octal, binario, etc...)

2.2.4 Linux

Linux es el núcleo del sistema operativo libre más popular y constituye la base del sistema operativo Linux (también llamado GNU/Linux), que ha comenzado a competir con sistemas operativos no libres como Unix y Windows.

El sistema operativo GNU/Linux es un sistema que se basa en Unix, es multiusuario y multitarea, el núcleo de Linux fue desarrollado por un estudiante finlandés de informática, Linus Torvalds de la Universidad de Helsinki en Finlandia, y que publicó su código fuente en 1990.



Figura N° 29: Logo oficial de Linux

2.2.4.1 Componentes de LINUX

Linux se puede dividir generalmente en cuatro componentes principales: el núcleo (kernel), el shell, el sistema de archivos y las utilidades.

- El núcleo: es el programa medular que ejecuta programas y gestiona dispositivos de hardware tales como los discos y las impresoras.
- El Shell: proporciona una interfaz para el usuario. Recibe órdenes del usuario y las envía al núcleo para ser ejecutadas.
- El sistema de archivos: organiza la forma en que se almacenan los archivos en dispositivos de almacenamiento, tales como los discos. Los archivos están organizados en directorios.

Cada directorio puede contener un número cualquiera de subdirectorios, cada uno de los cuales puede a su vez, contener otros archivos.

2.2.4.2 UNIX

Es un núcleo de un Sistema Operativo de tiempo compartido: un programa que controla los recursos de unas computadoras y los asigna entre los usuarios. Permite a los usuarios ejecutar sus programas; controla los dispositivos periféricos (discos, terminales y otros) conectados a la máquina.

Unix no sólo abarca el núcleo, sino que incluye también programas esenciales, entre ellos: compiladores, editores, programas para copiado, etc. Linux se basa sobre Unix

Se basa con una característica: este sistema operativo sería Software Libre. Esto significa que una vez que el usuario compra un CD o bien lo descarga gratuitamente desde Internet, es libre de redistribuirlo y modificarlo a su antojo, siempre que, como lo indica la Licencia Pública General GNU - acrónimo de "GNU is Not Unix", del modo que ha dispuesto la Free Software Foundation, se incluya el código fuente. Esto también incluye derecho a poder instalar Linux en cualquier número de ordenadores o equipos de cómputo que el usuario desee.

2.2.4.3 Diseño de UNIX

Los conceptos centrales en el modelo del sistema UNIX son los archivos y los procesos como se muestra en la figura N° 30. En la siguiente figura muestro un diagrama de bloques del kernel, mostrando varios módulos y las relaciones entre ellos. En particular, se muestra el subsistema de archivos en la parte izquierda y el subsistemas de control de procesos en la parte derecha, los dos mayores componentes del Kernel.

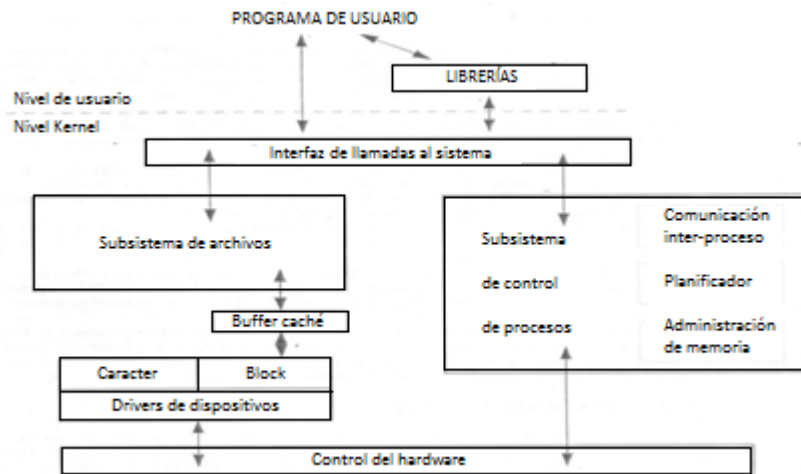


Figura N° 30: Diseño de UNIX

Las llamadas interactúan con el subsistema de archivos y con el subsistema de control de procesos. El subsistema de archivos controla los archivos, asigna espacio a los archivos, administra el espacio libre, controla el acceso a los archivos, etc. Los procesos interactúan con el subsistema de archivos mediante un específico conjunto de llamadas al sistema.

El subsistema de archivos accede a los archivos de datos usando mecanismo de búffer que regula el flujo de datos entre el Kernel y los dispositivos de almacenamiento secundario. El mecanismo de buffer interactúa con los controladores de dispositivos de E/S de tipo de bloque para iniciar la transferencia de datos desde y hacia el Kernel.

El subsistema de procesos es el responsable de la sincronización de los procesos, la comunicación entre procesos, administración de memoria principal y la planificación de procesos. Tanto el subsistema de archivos y el subsistema de procesos interactúan cuando se carga un archivo en memoria para su ejecución.

El módulo de administración de memoria controla la asignación de memoria. Si en algún momento no hay suficiente memoria

física para todos los procesos, el kernel los mueve entre la memoria principal y la secundaria.

El módulo de planificador o scheduler asigna la CPU a los procesos. Planifica los procesos para ser ejecutados por turno hasta que voluntariamente liberen la CPU mientras esperan un recurso o hasta que el Kernel los saca su tiempo de ejecución supera el tiempo quantum.

Finalmente, el control de hardware es el responsable de las interrupciones y de las comunicaciones con la máquina. Los dispositivos como los discos o terminales pueden interrumpir a la CPU mientras un proceso se está ejecutando. Así, el Kernel debe restablecer la ejecución del proceso interrumpido después de servir a la interrupción.

2.2.4.4 Proyecto GNU

El proyecto GNU, fue iniciado por Richard Stallman y fue lanzado en 1984, con el objeto de crear un sistema operativo libre completo tipo UNIX, el sistema GNU.

Las variantes del sistema operativo GNU que utilizan el núcleo llamado Linux, son utilizadas ampliamente en la actualidad, aunque a menudo estos sistemas se refieren como “Linux”, la manera correcta de llamarlos es sistemas GNU/Linux GNU es un acrónimo recursivo para GNU No es Unix y se pronuncia en español.

DISTRIBUCIÓN GNU/LINUX

Una distribución Linux o GNU/Linux, es un conjunto de aplicaciones reunida en torno al núcleo del sistema, que permiten brindar mejoras para instalar fácilmente un sistema Linux o GNU/Linux. Son “Sabores” de Linux que en general, se destacan por las herramientas para configuración y sistemas de paquetes de software a instalar.

También existe micro-distribuciones que pueden almacenar en uno o dos piquetes. Entre las distribuciones más conocidas pueden citarse como se muestra en la figura N° 31:



Figura N° 31: Distribuciones más conocidas Linux

2.2.4.5 Ubuntu

Es un sistema operativo predominante enfocado en la facilidad de uso e instalación, la libertad de los usuarios, y los lanzamientos regulares (cada 6 meses).

El nombre proviene del concepto africano Ubuntu, que significa “humanidad hacia otros” o “yo soy porque nosotros somos”. También es el nombre de un movimiento humanista sudafricano. Ubuntu aspira a impregnar de esa mentalidad al mundo de las computadoras. El eslogan de Ubuntu “Linux para seres humanos” resume una de sus metas principales: hacer de Linux un sistema operativo accesible y fácil de usar.

PRINCIPIOS DE UBUNTU.

La filosofía de Ubuntu se basa en los siguientes principios:

- El usuario debe tener la libertad de descargar, ejecutar, copiar, distribuir, estudiar, compartir, cambiar y mejorar sus

software para cualquier propósito, sin tener que la pagar derechos de licencia.

- Debe ser capaz de utilizar su software en el idioma de su elección.
- Debe ser capaz de utilizar todo el software independientemente de su discapacidad.

Al instalar Ubuntu se puede estar seguro de que todo el software cumple con estos ideales. Además, estamos continuamente trabajando para asegurar que cualquier programa que sea necesario está disponible bajo una licencia que ofrezca esas libertades.

UBUNTU ES SOFTWARE LIBRE

El proyecto Ubuntu está totalmente basado en los principios del Software Libre y anima a que la gente use, mejore y distribuya software libre.

Según la Free Software Foundation, el software es libre cuando sus usuarios gozan de las siguientes libertades:

- La libertad de usar el programa, con cualquier propósito.
- La libertad de estudiar cómo funciona el programa, y modificarlo, adaptándolo a cada necesidad.
- La libertad de distribuir copias del programa, con el cual se puede ayudar al prójimo.
- La libertad de mejorar el programa y compartir esas mejoras con los demás, de modo que toda la comunidad se beneficie.

La filosofía del software libre establece que la gente pueda usar el software de todas las formas que considere “socialmente útiles”: el código está disponible para poder descargarlo, modificarlo y usarlo del modo más conveniente. Por tanto, además del hecho de que el software libre normalmente puede

conseguir sin coste alguno, esta libertad también tiene ventajas técnicas: al desarrollar programas puede utilizarse el trabajo de los demás y construir a partir de este.

UBUNTU ES CÓDIGO ABIERTO:

El código abierto es el poder colectivo en acción. El poder de una comunidad mundial de expertos altamente calificado que construyen, comparten y mejoran el software de última generación, y a continuación, lo ponen a disposición de todo el mundo.

Hay 10 principios básicos de software de código abierto:

- El software debe ser libre de redistribuir
- El programa debe incluir el código de fuente
- La licencia debe permitir a la gente a experimentar con las modificaciones y redistribuir.
- Los usuarios tienen derecho a saber quién es responsable por el software que está utilizando.
- No debe haber ninguna discriminación contra cualquier persona o grupo.
- La licencia no debe restringir a nadie que haga uso del programa en un campo específico.
- Nadie debería tener que adquirir una licencia adicional para usar o redistribuir el programa.
- La licencia no debe ser específica de un producto.
- La licencia no debe restringir otro software.
- La licencia debe ser tecnológicamente neutral

El término código abierto fue acuñado en 1998, surgió del movimiento software libre para eliminar la ambigüedad en la palabra “free” – libre y gratis en inglés – y sigue disfrutando de un éxito creciente y un amplio reconocimiento.

2.3 Marco conceptual

Las siguientes definiciones han sido utilizadas en el desarrollo de la Tesis:

ARDUINO: es una plataforma libre de computación de bajo coste basada en una placa de entrada-salida y en un entorno de desarrollo IDE que implementa el lenguaje Processing / WiringHardware.

CONTROL NUMERICO COMPUTARIZADO: dispositivo de automatización de una máquina que, mediante una serie de instrucciones codificadas (programa) controla su funcionamiento.

FRESADORA: es una máquina herramienta utilizada para realizar mecanizados por arranque de viruta mediante el movimiento de una herramienta rotativa de varios de corte denominada fresa. Mediante el fresado es posible mecanizar los más diversos materiales como madera, acero, fundición de hierro, metales no férricos y materiales sintéticos, superficies planas o curvas, de entalladura, de ranuras, de dentado, etc.

LINUX: es el núcleo del sistema operativo libre más popular y constituye la base del sistema operativo Linux (también llamado GNU/Linux), que ha comenzado a competir con sistemas operativos no libres como Unix y Windows.

PLACA DE CIRCUITO IMPRESO (PCB): es una plancha de material rígido aislante, cubierta por unas pistas de cobre en una de sus caras o en ambas, para servir como conductor o de interconexión eléctrica entre los distintos componentes que se montarán sobre ella.

CAPITULO III

DISEÑO Y CONSTRUCCIÓN DE LA HERRAMIENTA

3.1 Diseño de la herramienta

3.1.1 Etapas del proyecto

El proyecto de Tesis comprende cinco etapas, las mismas que se observan en la figura N° 32.

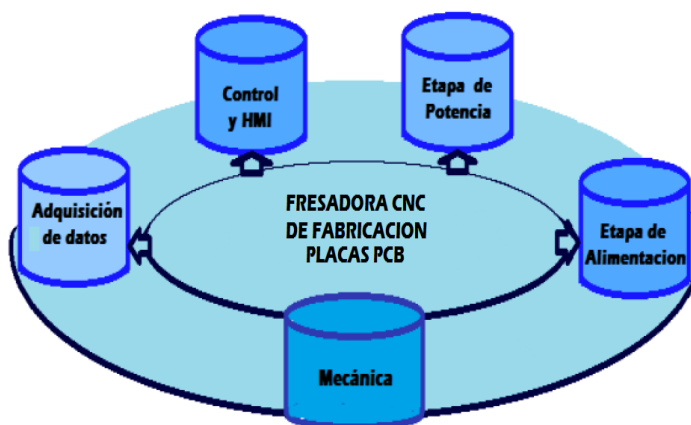


Figura N° 32: Etapas del proyecto

Para poder comenzar con el diseño y el desarrollo de la fresadora CNC de fabricación de placas PCB, se necesita plantear desde la etapa adquisición y envío de datos por parte los finales de carrera por medio del Arduino, el control por medio de los software usados en el sistema operativo Ubuntu Linux, la etapa de potencia para

poder controlar los motores paso de la fresadora, como también mecánica con la elección de los motores, el diseño de la estructura para la fresadora, y finalmente la etapa de alimentación para poder suministrar la energía constante para que la fresadora pueda realizar el trabajo correctamente, en la figura N° 33 se observa la estructura general de la etapa electrónica de la fresadora CNC.

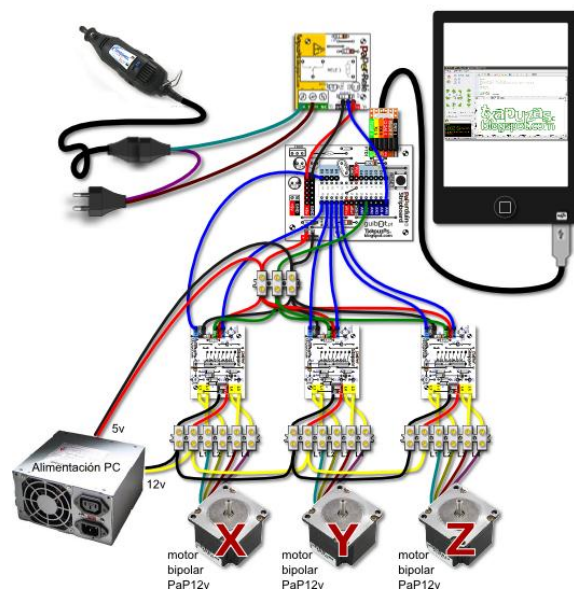


Figura N° 33: Estructura general de la etapa electrónica de la Fresadora CNC

3.1.2 Control y monitoreo HMI

El control y monitoreo (HMI) de la fresadora CNC para fabricación de placas PCB es realizada por una computadora a bordo con plataforma Atmel consistente de un módulo Arduino Uno R3. El control de la fresadora se realiza por medio del programa realizado en Gambas 3.5.3, este programa que es de interfaz y de control es utilizado en el sistema operativo Ubuntu Linux.

3.1.2.1 Arduino Uno R3

Arduino uno R3 trabaja bajo una plataforma de código abierto basado en prototipos de electrónica flexible y fácil de usar tanto en hardware y software (ver Figura N° 34). Contiene un microcontrolador Atmega8 y utiliza el lenguaje de programación

Arduino (basado en cableado) y el entorno de desarrollo Arduino (basado en procesamiento).

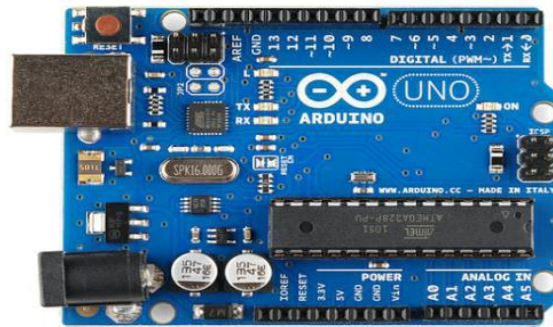


Figura N° 34: Arduino Uno

Como también la recepción de los datos por parte de los finales de carrera que se encuentra en la fresadora CNC. Toda esta información llega y es enviada por medio de un puerto RS232 emulado.

Para la realización del programa HMI se utilizará el sistema operativo Ubuntu Linux ya que si bien tiene algunos parecidos a otros como Microsoft, este nos brinda la ventaja es que es gratuito.

Ubuntu Linux es un sistema operativo predominante enfocado en la facilidad de uso e instalación, la libertad de los usuarios, El proyecto Ubuntu está totalmente basado en los principios del Software Libre y anima a que la gente use, mejore y distribuya software libre.

El programa de control a realizarse en el Sistema Operativo Ubuntu Linux, es el Gambas V3.5.3.

3.1.2.2 Gambas V3

Gambas V3 es un entorno de desarrollo libre basado en una base de intérprete con extensiones de objetos, un poco como Visual Basic TM.

Gambas es un entorno de lenguaje-objeto y de desarrollo completo construido en un intérprete de BASIC. Es liberado bajo la GNU Licencia Pública General.

Su arquitectura está inspirada en gran parte por Java. Así Gambas se compone de:

- Un compilador.
- Un intérprete.
- Un archivador.
- Un guionista.
- Un entorno de desarrollo.
- Muchos de los componentes de extensión.



Figura N° 35: Logo de software Gambas V3

La comunicación entre la computadora y el arduino será por medio de conexión USB, realizando una conexión RS-232 emulado, esta nos permite tener una conexión full-duplex para poder enviar los comandos como también de recibir información si alguno de los finales de carrera se activa.

3.1.2.3 Proceso

UNTECSCNC es una fresadora casera de 2,5D. Es decir permite realizar mecanizados sobre una superficie (X,Y), a distintas alturas (Z). UNTECSCNC se compone de varias partes:

La propia fresadora realizada con madera según las indicaciones de David Carr en su blog: Make your bot. Un programa realizado

en gambas para sistemas Linux denominado UntecsCNC_TX, que consta de un editor de programas Gcode, un simulador para ver visualmente el funcionamiento de los programas que realizamos en un subprograma de envío de dichos comandos Gcode a la fresadora. Entre el programa de mando Untecs CNC_TX y la fresadora necesitamos una interface que decodifique los comandos y en consecuencia envíe las señales de control oportunas al motor paso a paso que gobiernan los ejes de la fresadora.

Esta interface está compuesta por un Arduino, que en su interior tiene cargado un programa (Untecs CNC_RX) que comunica con el programa de mando, decodifica los comandos Gcode y manda señales de control a los motores paso a paso. Estas señales de control son amplificadas por la etapa de potencia que controlan los motores paso a paso y el motor del cabezal de la fresadora. Como se observa en la figura N° 36 se representa el diagrama de bloques de la fresadora CNC.

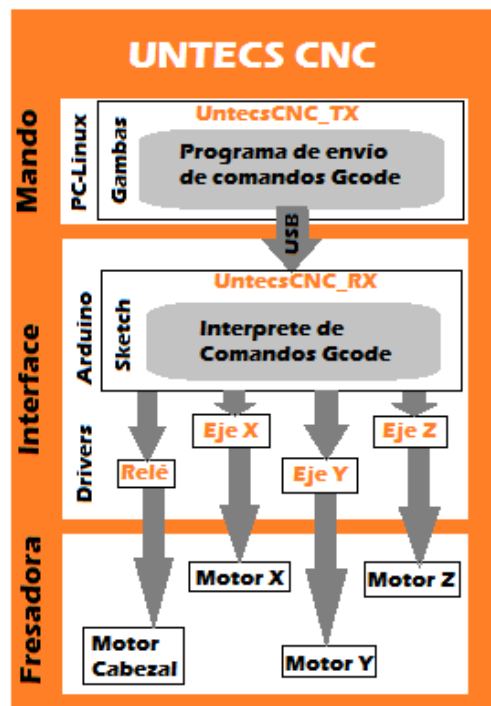


Figura N° 36: Diagrama de bloques de la etapa electrónica de la fresadora CNC

3.1.2.4 Estado del arte

3.1.2.4.1 EMC2

Es un sistema Linux específico para CNC, de tiempo real y preparado para controlar motores paso a paso por el puerto serie. Disponible como CD-Live. Dispone del sistema de visualización y control. Es el ordenador el que realiza todas las tareas de control de los motores, y conectando en el puerto paralelo unos drivers de potencia podremos controlar los motores paso a paso. Es un sistema técnicamente muy bueno, pero está pensado para tener un PC dedicado al control de la fresadora (si actualizas linux puede dejar de funcionar), y además controla directamente los motores paso a paso a través del puerto paralelo, puerto que se va quedando obsoleto hoy en día.

3.1.2.4.2 REP-RAP

Es un sistema de impresión 3D preparado para funcionar en Windows, Linux y Mac, ya que está realizado en Java. El objetivo es el realizar una máquina autoreplicante.

Para poder usarlo en un sistema multitarea, el ordenador tiene cargado un programa denominado Rep-Rap que envía comandos Gcode a un microprocesador Arduino. El Arduino tiene cargado un programa que interpreta los comandos y manda señales de control a unos drivers de potencia que controlan los motores paso a paso.

Este sistema también permite utilizar ReplicatorG para enviar datos al microprocesador. Este sistema con mucho apoyo por parte de la comunidad tiene el objetivo de crear extrusionadoras de plástico para crear formas 3D, y utiliza una versión de Arduino.

3.1.2.4.3 CONTRAPTOR CNC

Es una variante del sistema Reprap para su utilización con mini fresadoras. La fresadora se construye mediante perfiles perforados. Tiene un programa realizado en Processing que permite el envío de programas Gcode al Arduino.

3.1.2.4.4 UNTECS CNC

Es un sistema derivado de Contraptor CNC, con las siguientes modificaciones.

El programa de envío de datos UNTECS CNC TX es una transcripción en gambas del programa original del Contraptor CNC realizado en Processing. (Creo que se podría utilizar el programa de envío de contraptor en UNTECS CNC, que al estar realizado en Processing-Java permite su uso en sistemas windows).

El Sketch de recepción de datos cargado en Arduino, se ha modificado para permitir utilizar controladoras de motor paso a paso “primitivas”, es decir con dos hilos de control (D1 y D2), en vez de utilizar Drivers avanzados con señales de STEP y DIR.

La fresadora se realiza con madera según las indicaciones de David Carr.

Los componentes necesarios son:

- Un ordenador con sistema Linux, gambas2 y el programa Untecs CNC_TX instalado.
- Un Arduino que tenga instalado el programa Untecs CNC_RX.
- Una fresadora casera en el que los ejes se muevan por motores paso a paso con dos señales de control.

- Etapa de potencia, que suministren la potencia necesaria que permitan mover los motores, y poder habilitar la máquina de fresado.

3.1.2.5 Untecs CNC Transmisión TX

Untecs CNC_TX es un programa realizado en Gambas simultáneamente un editor de texto, un simulador y una consola de control CNC.

EDITOR DE PROGRAMAS

Es un sencillo editor de textos con resaltado para comandos Gcode, con una característica *fundamental*: envía por el puerto serie o USB con serie virtual una a una las líneas del editor esperando previamente a recibir señales “OK” del controlador Arduino.

SIMULADOR

Es un visualizador de programas Gcode (2D) que muestra el recorrido en el plano XY. Dispone de zoom, Desplazamiento, simulación rápida y con retardo.

3.1.2.5.1 Control

Es una consola de mando desde donde se puede manejar directamente la fresadora, moviendo los motores y activando el cabezal.

Este programa reconoce archivos con coordenadas absolutas o relativas (G90/G81), en milímetros o pulgadas (G21/G20) y con parámetros (#).

3.1.2.5.2 Instalación

Como Untecs CNC_TX está realizado en gambas2, es necesario que se tenga instalado el programa completo

Gambas2, para instalar el programa a la PC, se descargará e instalará desde el “Centro de software de Ubuntu”.

Una vez instalado, solo se abre el programa Untecs CNC_TX y el gambas se ejecutará automáticamente, para que funcione el programa se debe tener instalado el componente de gambas2 gb.net para evitar generar cualquier error en el programa.

3.1.2.5.3 Componentes

Excluyendo el menú principal, podemos dividir Untecs CNC_TX en cuatro componentes (ver figura N° 37):

- Un control para mover directamente la fresadora.
- Una consola de salida que nos muestra mensajes informativos sobre el estado del sistema.
- Un editor de códigos G
- Un simulador de Códigos G.



Figura N° 37: Partes del Software de control de Gambas V3

3.1.2.5.4 Editor

Es configurable desde el menú/Preferencias/Editor. El editor tiene una barra de herramientas con los comandos

más usuales en este tipo de programas (Ver figura N° 38). Hay que destacar los últimos 5 controles, que son los específicos de CNC.

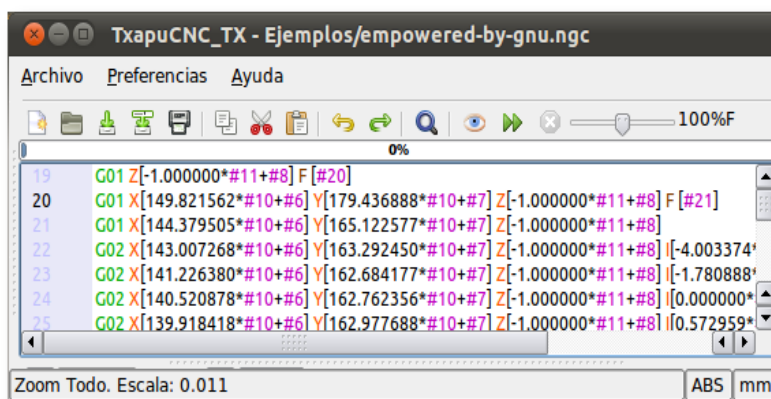


Figura N° 38: Editor



El primero visualiza en el simulador el programa cargado en el editor. Este proceso se ejecuta automáticamente al cargar un programa G Code.



El segundo Ejecuta el programa en la máquina, es decir envía los comandos al Arduino, y simultáneamente lo visualiza en el simulador.



El tercero es el pulsador de parada, que detiene tanto la simulación como el envío de datos al controlador Arduino.

Este control modifica el parámetro de velocidad de avance leído del archivo y que se enviará al controlador. Es decir mediante este control podemos modificar la velocidad especificada en cada línea del programa desde el 10% al 1000%. Se puede utilizar para no tener que cambiar el programa si vemos que el material a mecanizar es demasiado duro o blando.

3.1.2.5.5 Simulador

Esta ventana muestra una visualización del archivo cargado en el editor (Ver figura N° 39).



Para que se visualice el archivo es necesario apretar el botón de simulación situado en la barra de herramientas del editor.

La simulación puede ser instantánea o pausada para ver la ejecución de cada instrucción. El modo pausado se activa al apretar el botón con la imagen de un caracol. Al activarse se puede modificar el tiempo de espera entre la visualización de cada paso expresado en centésimas de segundo. También nos puede interesar que el programa simule el diámetro de la herramienta de mecanizado.

Al activarlo podemos modificar el diámetro de la herramienta expresado en décimas de milímetro. Si se requiere analizar una parte del mecanizado se realiza arrastrando con el mouse y también se pueden hacer zoom. Si queremos que la simulación ocupe toda la ventana se puede apretar el botón situado en la parte inferior derecha de la pantalla.

En la parte inferior izquierda del editor existe otro botón que coloca la simulación en el origen. En la parte derecha de la ventana del simulador vemos una representación de la herramienta de mecanizado y su posición en el eje Z. Los números situados encima y debajo de la imagen, son los valores máximos y mínimos del eje Z, encontrados en el archivo cargado en el editor.

El color de la herramienta de mecanizado varía para indicar si el motor del cabezal está encendido o apagado.

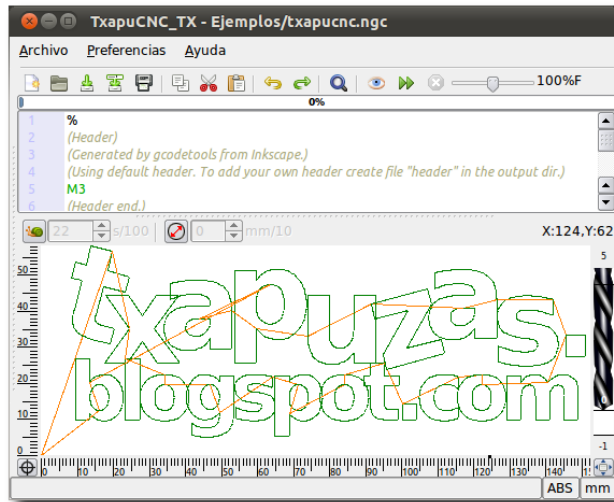


Figura N° 39: Simulador

3.1.2.5.6 Controlador

La última ventana es la del control. Desde aquí se puede mandar comandos independientes al controlador (Arduino). Algunos de estos comandos disponen de un botón propio.

Los indicadores de posición situados en la parte superior, no indican la posición real mientras el CNC se está moviendo. Solamente cuando ha llegado a la posición de destino. Mientras se mueve el CNC, se muestran los puntos de destino, y éstos parpadean indicando que se está moviendo. Al ser el controlador (Arduino) el encargado de mover los motores, Untecs CNC_TX no sabe la posición del CNC, solamente cuando recibe la orden "ok" Untecs CNC_TX conoce que el CNC ha llegado a la posición especificada.

Define el origen de coordenadas del CNC, enviando el comando: "G28 X0 Y0 Z0" Si previamente se ha definido el origen de coordenadas, el programa podrá mandar la orden de ir al origen, "G0 X0 Y0 Z0".

Con estos botones hacemos que el CNC vaya a los límites del mecanizado (los valores representados en negro) a la velocidad indicada por el desplazable “Avance”
Por ejemplo si apretamos el primer botón teniendo el archivo de ejemplo cargado de la derecha, enviaría el siguiente comando al CNC:

“G1 X0 Y58.17 F500”

Con estos pulsadores movemos un paso el eje en el sentido indicado y a la velocidad de avance especificada por el desplazable. El tamaño del paso será el indicado en la lista desplegable “Tamaño de paso”.

Si en este control hemos seleccionado la opción “Continuo”, el control enviará repetidamente al CNC la orden de moverse 0,1mm (ó 0,01' pulgadas si las unidades son pulgadas) en el sentido especificado mientras mantengamos pulsado un botón.

Por ejemplo, si apretamos “Y-” enviaremos al CNC el siguiente comando: “G1 Y X (actual-5) F500”.

Estas teclas (X-, X+, Y-, Y+) pulsadas con el ratón se pueden pulsar directamente en el teclado (Ver figura N° 40). Si además pulsamos la tecla “Control”, el paso será más pequeño, y si pulsamos simultáneamente la tecla “Mayúsculas”, el paso será mayor. Con este pulsador activaremos o pararemos el motor del cabezal. Enviando bien “M3” ó “M5”.

Por último, el cuadro de texto y el botón situado a su derecha permiten enviar al CNC cualquier comando que escribamos en el cuadro de texto.

Hasta que no se haya establecido una conexión con el Arduino, no estarán accesibles ninguno de los botones y

comandos que permiten enviar comandos Gcode como se observa en la Tabla N° 7 al Arduino. El software tiene un listado de acceso rápido el cual se presenta en la Tabla N° 8. Solamente podemos utilizar el programa como un sencillo editor de texto para modificar programas Gcode.

En realidad el sistema completo es una adaptación de diversos proyectos existentes, modificados con el objetivo de simplificar la fabricación. Untecs CNC_TX reconoce archivos en milímetros y pulgadas, con coordenadas absolutas o relativas.

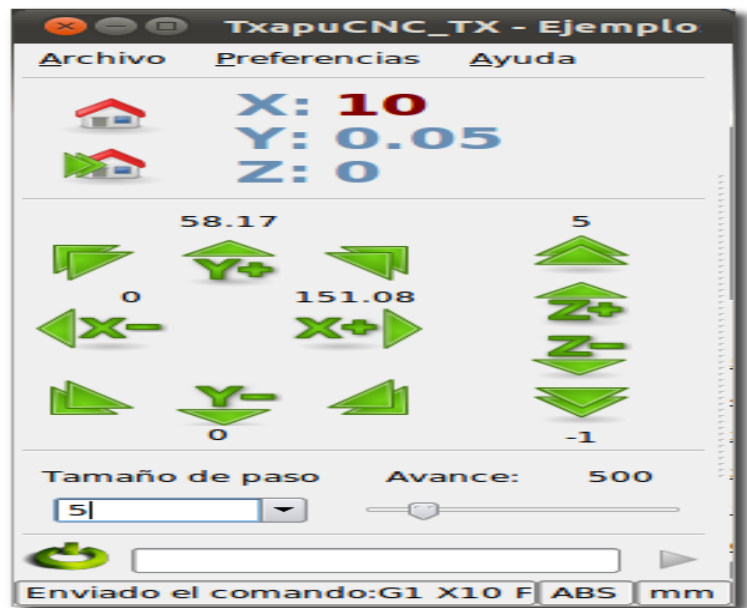


Figura N° 40: Control

| Comando | Ejemplo | Descripción | TX | RX |
|---------|--------------------------|---|----|----|
| G0 | G0 X10 | Movimiento lineal Rápido | Si | Si |
| G1,G01 | G1 X10 Y15 Z0 [F100] | Movimiento lineal Controlado (Avance: 100) | Si | Si |
| G2,G02 | G02 X80 Y30 I30 J-10 F02 | Movimiento curvo (sentido horario) Controlado | Si | Si |
| G3,G03 | G03 X80 Y30 I10 J20 | Movimiento curvo (antihorario) Controlado | Si | Si |
| G4,G04 | G4 P200 | Pausa con retardo (Retardo: 200ms) | Si | Si |
| G20 | G20 | Definir Unidades en Pulgadas | Si | Si |
| G21 | G21 | Definir Unidades en milímetros | Si | Si |
| G28 | G28 | Ir a Origen | Si | Si |
| G30 | G30 X10 Y20 Z30 | Ir a Origen a través de un punto | Si | Si |
| G90 | G90 | Definir Coordenadas absolutas | Si | Si |
| G91 | G91 | Definir Coordenadas relativas | Si | Si |
| G92 | G92 | Definir punto actual como origen | Si | Si |
| M0 | M0 | Paro (Pausa programada) | Si | No |
| M3,M03 | M3 | Marcha del cabezal | Si | Si |
| M5,M05 | M5 | Paro del cabezal | Si | Si |

Tabla N° 7: Comandos G reconocidos por UNTECS CNC

| Tecla | Función | Descripción |
|-------------------|----------------|---|
| Escape | Paro | Detiene la visualización/ejecución del programa |
| Barra de Espacio | Pausa | Realiza una pausa en la visualización/ejecución del programa |
| F1 | Ayuda | Muestra el archivo de ayuda |
| F2 | Maximizar | Aumenta al máximo la ventana de TxapuCNC |
| F3 | Buscar | Muestra el cuadro de diálogo "buscar" |
| + | % Velocidad + | Aumenta un 25% especificada la velocidad del programa |
| - | % Velocidad - | Disminuye un 25% especificada la velocidad del programa |
| Flecha arriba* | Y+ | Mueve el CNC en el sentido positivo del eje Y |
| Flecha abajo* | Y- | Mueve el CNC en el sentido negativo del eje Y |
| Flecha izquierda* | X- | Mueve el CNC en el sentido negativo del eje X |
| Flecha derecha* | X+ | Mueve el CNC en el sentido positivo del eje Y |
| Control | Disminuir paso | Al pulsarla junto a alguna flecha mueve el eje un "micropaso" |
| Mayúscula | Aumentar paso | Al pulsarla junto a alguna flecha mueve el eje un "macropaso" |
| Inicio | Define Inicio | Define la posición actual del CNC como origen de coordenadas. |
| | | |

Tabla N° 8: Listado de funciones de acceso rápido.

3.1.2.6 Untecs CNC Recepción RX

Para que el programa Untecs CNC_TX funcione correctamente, es decir para que mueva los motores del CNC, es necesario instalar un programa en el Arduino.

Este programa se encarga de leer los comandos que le envía Untecs CNC_TX, los interpreta (los comandos que conoce) y en consecuencia manda las órdenes oportunas a los drivers del CNC o al motor. Cuando la máquina llega a la posición de destino, Untecs CNC_RX envía el mensaje de “ok” al PC para que Untecs CNC_TX lo sepa, y pueda enviar el siguiente comando.

3.1.2.6.1 Instalación

Simplemente carga el programa Untecs CNC_RX_xx.pde en el entorno Arduino, y desde ahí, con el Arduino conectado, cárgalo en la placa pulsando el botón señalado en la imagen.

Al Abrir el programa Untecs CNC_RX_xx.pde, se abrirán automáticamente 4 archivos en el Editor de Arduino.

Deberemos modificar los parámetros especificados en el Archivo _init en función de las características de nuestros motores paso a paso:

```
#define X_STEPS_PER_MM 160
```

Para calcular este valor:

Si utilizamos varilla roscada de M8 para mover los ejes, sabemos que el paso es de 1,25, es decir por cada vuelta del eje una tuerca insertada en él se movería 1,25mm. Calcularemos lo que hay que girar el eje para que se desplace 1mm:

si $360^{\circ} \Rightarrow 1,25$

con $X^{\circ} \Rightarrow 1$

$X=360/1,25=288^{\circ}$

Luego son necesarios girar el eje 288° para que se mueva la tuerca 1mm.

Si en cada paso de nuestro motor el eje gira $1,8^{\circ}$

$288^{\circ}/1,8= 160$ Pasos.

Luego serán necesarios 160 pasos para mover la tuerca (la plataforma) 1 mm.

```
#define X_STEPS_PER_INCH 4064
```

Para el caso de pulgadas, simplemente multiplicamos por 24,5

$160 \times 24,5=4064$

Repetiremos este proceso para los otros dos ejes.

```
//our maximum feedrates
```

```
#define FAST_XY_FEEDRATE 2400
```

```
#define FAST_Z_FEEDRATE 2400
```

En esta instrucción se ingresan los valores de velocidad (Avance) máxima para los ejes X, Y y Z. Este valor es empírico. Simplemente desde el control (Untecs CNC_TX) se pueden mover los ejes y aumenta el avance hasta que se observa que se pierde sincronismo (el motor hace más ruido y no se mueve), se coloca un valor ligeramente más bajo. Este valor será la velocidad con el que el Arduino moverá los motores al recibir una instrucción G0 (Movimiento a velocidad máxima).


```
// Units in curve section
```

```
#define CURVE_SECTION_INCHES 0.019685
```

```
#define CURVE_SECTION_MM 0.5
```

Estos son los segmentos rectos mínimos en los que el CNC descompone una curva (El cnc siempre mueve los motores en líneas rectas). Normalmente no es necesario modificar estos valores.

```
// Set to one if sensor outputs inverting (ie: 1 means open, 0 means closed)
```

```
// RepRap opto endstops are *not* inverting.
```

```
#define SENSORS_INVERTING 0
```

Para habilitar sensores finales de carrera para cada eje, se tiene la posibilidad de utilizar normalmente abiertos o normalmente cerrados.

```
/******
```

```
* digital i/o pin assignment
```

```
* this uses the undocumented feature of Arduino:
```

```
* pins 14-19 correspond to analog 0-5
```

```
*****/
```

```
//cartesian bot pins
```

```
#define X_STEP_PIN 8
```

```
#define X_DIR_PIN 9
```

```
#define X_MIN_PIN 4
```

```
#define X_MAX_PIN 2
```

```
#define X_ENABLE_PIN 15 //Pin Analog 1
```

```
#define Y_STEP_PIN 10

#define Y_DIR_PIN 11

#define Y_MIN_PIN 3

#define Y_MAX_PIN 5

#define Y_ENABLE_PIN 15 //Pin Analog 1

#define Z_STEP_PIN 12

#define Z_DIR_PIN 13

#define Z_MIN_PIN 7

#define Z_MAX_PIN 6

#define Z_ENABLE_PIN 15 //Pin Analog 1

#define MOTOR_PIN 19 //Encendido/apagado cabezal (Pin Analog 5)
```

Como ya se ha indicado, el programa está preparado para finales de carrera que impidan que la máquina supere esos topes máximos. X_MAX, X_MIN, Y_MAX, Y_MIN, Z_MAX, Z_MIN.

Todos los pines _ENABLE se han unido para ahorrar pines, y se unen a los respectivos enables de la etapa de potencia de los motores paso a paso.

Los pines EJE_STEP y EJE_DIR son en realidad las entradas de control D1 y D2 de la etapa de potencia de los motores paso a paso de cada eje.

Por último el relé que activa el cabezal de la fresadora está conectado al Pin 19 (AN5) que controla el motor del cabezal.

3.1.3 Etapa de alimentación

La fuente que se va a utilizar es una ATX (una fuente de ordenador) con la que podremos alimentar desde los chips hasta los motores y la fresadora. Se elige una ATX ya que es la más barata y además se puede conseguir de una computadora de escritorio.

La fuente tiene salidas de diferentes voltajes para poder conectar toda la electrónica a la misma ATX y así solo usamos una toma de corriente a la pared.

Los voltajes de las diferentes salidas de la fuente se muestran en la siguiente figura N° 41:

| Pin | Señal | Color |
|-----|--------|--------|
| 1 | 3.3V | Orange |
| 2 | 3.3V | Orange |
| 3 | Ground | Black |
| 4 | 5V | Red |
| 5 | Ground | Black |
| 6 | 5V | Red |
| 7 | Ground | Black |
| 8 | PW-OK | Grey |
| 9 | 5VSB | Purple |
| 10 | 12V | Yellow |
| 11 | 3.3V | Brown |
| 12 | -12V | Blue |
| 13 | Ground | Black |
| 14 | PS-ON | Green |
| 15 | Ground | Black |
| 16 | Ground | Black |
| 17 | Ground | Black |
| 18 | -5V | |
| 19 | 5V | Red |
| 20 | 5V | Red |

Figura N° 41: Valores de cableado por color de Fuente ATX

Se deja una lista de los voltajes que se utilizan en todo el proyecto.

- 5V: Se utiliza para energizar los finales de carrera, arduino y L298.
- 12V: Se utiliza para energizar los motores paso a paso.
- Ground: Para todo el circuito.

Se requiere para que funcione correctamente la fuente de alimentación ATX que se puentee el pin 14 PS-ON con una fuente de 5v, ya que en el caso de que no se realice ello la fuente nunca funcionará.

Para evitar el mal funcionamiento de la máquina todas las tierras deben estar conectadas entre sí, es decir, que la tierra de la etapa de potencia debe estar conectada con la de los motores paso a paso y con la tierra de Arduino. No es problema que los voltajes sean diferentes, aunque el chip tenga 5 V y el motor 12 V las tierras se pueden interconectar entre sí. Eso sí, esto funciona si la fuente es una ATX como se muestra en la figura N°42.



Figura N° 42: Fuente ATX

3.1.4 Etapa de potencia

3.1.4.1 Control de motores paso a paso bipolar

L298N es un driver de potencia para motores paso a paso bipolares utilizando el chip L298N. Un chip que permite controlar motores paso a paso de más potencia que el L293D, pero con la desventaja de que no tiene integrados diodos de protección en la salida.

- Entradas de control: 2 TTL (D1 y D2)

- Salidas de potencia 4 (L1, L2, L3 y L4)
- Tensión alimentación: 5v
- Tensión máxima salida: 36v
- Corriente máxima salida: 2A por salida (picos de 3A)

3.1.4.1.1 Identificación del tipo de motor

Para comprobar el tipo de motor paso a paso que tienes, para el proyecto se tiene un motor paso a paso de tipo unipolar, por lo que se requiere volver el motor unipolar a bipolar lo cual se realiza de la siguiente manera en la figura N° 43:

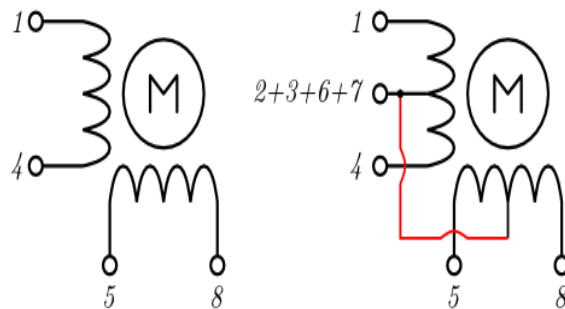


Figura N° 43: Transformación motor unipolar a bipolar

Los motores unipolares vienen con 5 cables, un común que se tiene en ambas bobinas, para realizar la conversión a motor bipolar es solo de eliminar los pines comunes de las bobinas para poder obtener solo 4 cables, como se observa en la figura de la izquierda, de esta manera se logra obtener un motor bipolar el cual funciona con la etapa de potencia basada en el integrado L298.

3.1.4.1.2 Descripción del controlador

Este proyecto utiliza un chip de mayor potencia (L298N) para motores bipolares más grandes.

Para la realización de este proyecto se basa en el circuito siguiente en la figura N° 44:

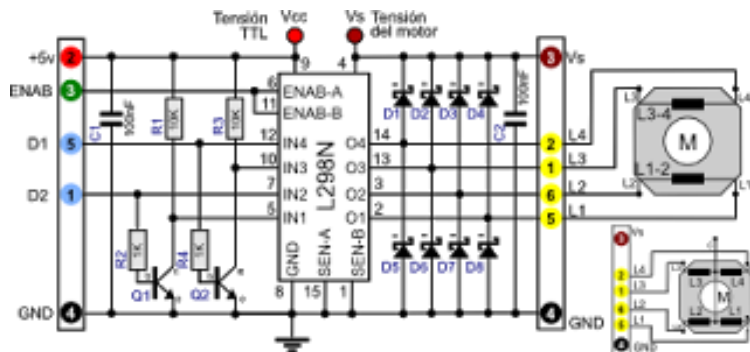


Figura N° 44: Circuito de control de motor paso a paso bipolar

En este circuito se observa que con sólo 2 pines de control (D1 y D2) se controla las 4 bobinas del motor paso a paso (L1, L2, L3 y L4). Esto se puede hacer ya que para mover un motor paso a paso se debe alimentar las bobinas en el orden siguiente como se muestra en la figura N° 45:

| N° DE PASOS | 1 | 2 | 3 | 4 |
|-------------|---|---|---|---|
| devanado1 | 1 | 1 | 0 | 0 |
| devanado2 | 0 | 0 | 1 | 1 |
| devanado 3 | 1 | 0 | 0 | 1 |
| devanado 4 | 0 | 1 | 1 | 0 |

Figura N° 45: Control de funcionamiento de motor paso a paso bipolar

En esta tabla se observa que los valores del devanado 2 es el invertido del 1 y que los del devanado 4 son los invertidos del 3, por lo que si se invierte la señal del L1 se obtiene la señal de control necesaria para L2 y de la misma manera si se invierte la señal de L3 se obtendrá la señal de control de L4, con lo que la tabla para poder mover el motor quedaría de la forma siguiente figura N° 46:

| N° DE PASOS | 1 | 2 | 3 | 4 |
|--------------------|----------|----------|----------|----------|
| dev1=-dev2 | 1 | 1 | 0 | 0 |
| dev3=-dev4 | 1 | 0 | 0 | 1 |

Figura N° 46: Control de funcionamiento de motor paso a paso con dos pines

Esta inversión de señal se realiza en el circuito utilizando 2 transistores Q1 y Q2, que cuando están activos (señal IN1 o IN4) derivan a masa las entradas IN2 o IN3 que están a positivo con la resistencia Pull-Up R1 y R3.

Normalmente, un motor paso a paso unipolar se puede utilizar como bipolar (si no se utilizan los terminales centrales de las bobinas), de forma que puedes utilizar este circuito para controlarlos.

3.1.4.2 Activación de cabezal

3.1.4.2.1 Descripción

A veces es necesario controlar una carga de potencia (bien en alterna o en continua) a partir de una señal digital de 5 v. Una forma de conseguirlo y además de aislar la zona de 5v y la de alta tensión puede ser mediante la utilización de un relé.

Normalmente los micro controladores suministran unos 25mA (Arduino hasta 40) pero para poder gobernar con seguridad la bobina de un relé de 5v (80mA) es mejor hacerlo mediante un transistor como se muestra en la figura N° 47.

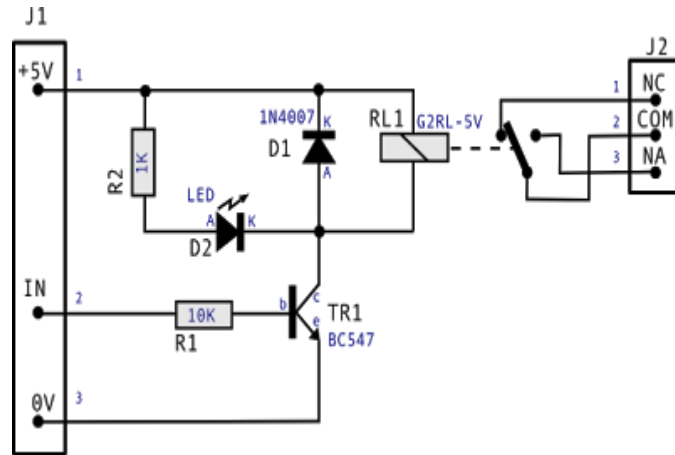


Figura N° 47: Circuito de control con Relé

El diodo D1 es necesario para eliminar los picos de corriente producidos por la bobina del relé.

El diodo LED y su resistencia R2 se utilizan como indicadores del estado de la señal de control, en el caso de no necesitarse se pueden eliminar tranquilamente (como por ejemplo al utilizar la salida digital 13 de Arduino).

Parece ser que la resistencia R1 de 10K es un poco elevada y puede darse el caso de que no lleve al transistor a saturación, si ese es tu caso, puedes sustituir la resistencia R1 de 10K a 4,7K.

3.1.4.2.2 Características técnicas

- Consumo: ~400mW
- Tensión de entrada: 0-5v TTL standard
- Intensidad señal de entrada: despreciable (IN)
- Tensión máxima salida: 220v (depende del relé)
- Intensidad máxima salida: 8A (depende del relé)

3.1.5 Etapa de adquisición y envío de datos

Se tuvo en cuenta para la adquisición de datos: los finales de recorrido de cada eje tanto para el eje X, Y, Z.

Para que la máquina fresadora pueda detectar que se encuentra llegando a los finales de sus recorridos, se utiliza un total de 6 finales de carreras denominados:

- X_MIN
- X_MAX
- Y_MIN
- Y_MAX
- Z_MIN
- Z_MAX

Por tal motivo se utilizó el siguiente circuito eléctrico para que el arduino pueda reconocer tanto 5V como 1, y 0V como 0 (ver figura N° 48).

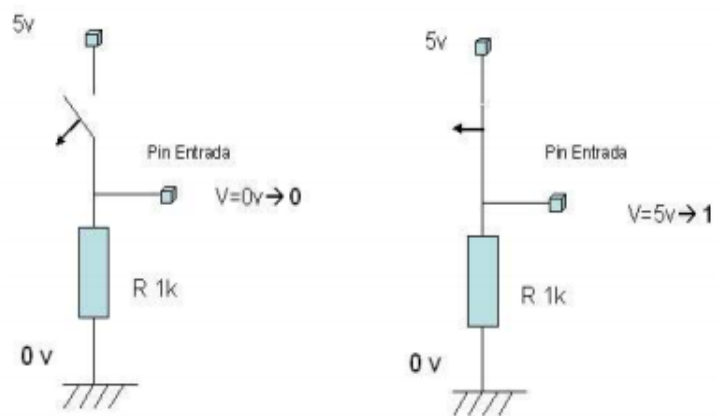


Figura N° 48: Circuito de control para finales de carrera

Por medio de la parte del programa se muestra la configuración de los pines para la configuración de los finales de carrera en el programa del Arduino.

```
//cartesian bot pins  
#define X_STEP_PIN 8  
#define X_DIR_PIN 9  
#define X_MIN_PIN 4  
#define X_MAX_PIN 2  
#define X_ENABLE_PIN 15 //Pin Analog 1
```

```
#define Y_STEP_PIN 10  
#define Y_DIR_PIN 11  
#define Y_MIN_PIN 3  
#define Y_MAX_PIN 5  
#define Y_ENABLE_PIN 15 //Pin Analog 1  
  
#define Z_STEP_PIN 12  
#define Z_DIR_PIN 13  
#define Z_MIN_PIN 7  
#define Z_MAX_PIN 6  
#define Z_ENABLE_PIN 15 //Pin Analog 1
```

3.1.6 Etapa mecánica

Para la etapa mecánica se consideró la fresadora del diseño Mantis-CNC, que es una fresadora realizada con madera según las indicaciones de David Carr.

3.1.6.1 Componentes

Los materiales que se requiere para realizar la fresadora son los siguientes:

- Plan de madera MDF 13mm de 800 x 400 mm, un total de 12 piezas para la fabricación de la fresadora, se tiene en cuenta que varias piezas fueron modificadas con respecto a los componentes encontrados localmente.
- Varilla roscada M8 1000 mm, las cuales se dividen en 3 varillas para los 3 ejes.
- Varilla 3/8 1000 mm, las cuales se dividen en un total de 6 guías para los 3 ejes.
- Casquillos o guías M8, un total de 12.
- Tuercas M8, un total de 6

- Acoples para de motor a varilla roscada.
- Tornillería y pernería diversa.

Los planos se pueden descargar de la misma páginas de del Blog de David Carr, las que se pueden imprimir sobre una hoja DIN-A3 (ver figura N° 49 y 50) y pegarlas sobre la madera para que puedan ser cortadas a la medida que se requiere.

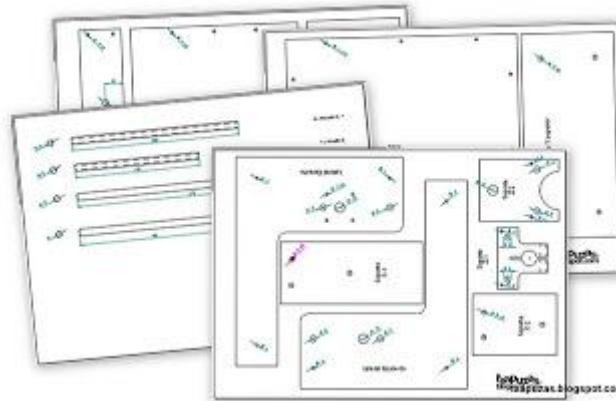


Figura N° 49: Plantillas de fabricación de fresadora CNC

Los planos están hechos para madera de 13mm de espesor, para que las varillas queden paralelas, se debe unir las piezas por parejas y las taladros simultáneamente (algunas piezas tienen unos pequeños agujeros de posicionado).

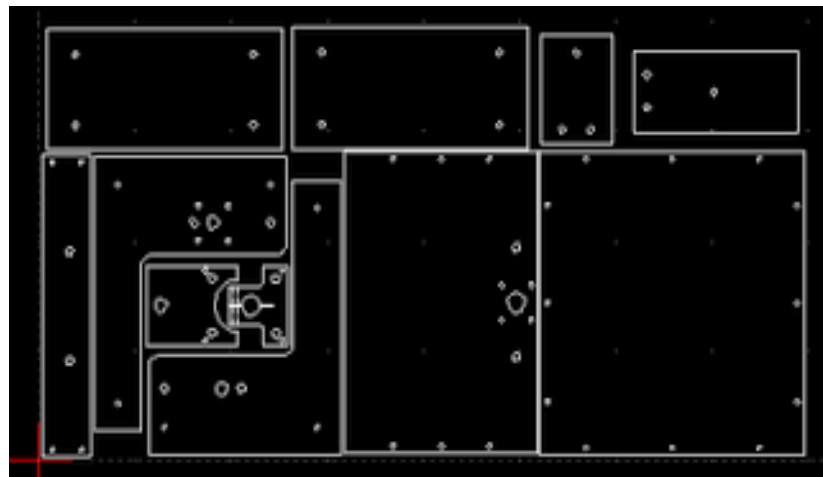


Figura N° 50: Plantilla de diseño de fresadora en Autocad

3.1.6.2 Diseño de eje x

Para la elaboración del eje X de la máquina fresadora CNC, se utiliza las piezas basadas en el modelo de David Carr, las cuales son mostradas a continuación:

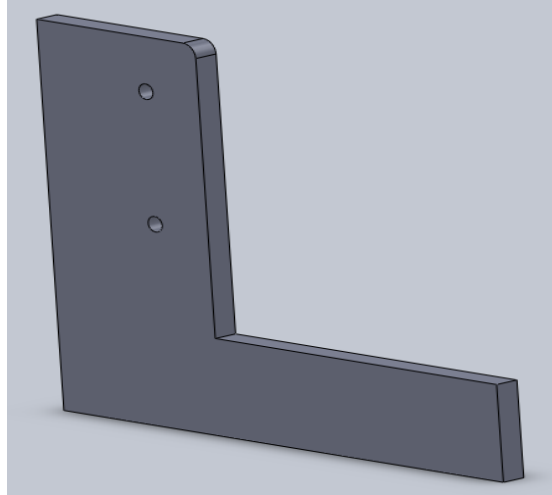


Figura N° 51: Pieza lateral izquierda de eje X

Como se muestra en la figura N° 51 para formar el eje X se han realizado dos piezas en forma de L, los cuales en total tienen dos agujeros de tipo 3/8" para poder realizar la instalación de las bocinas que sirven como guidores.

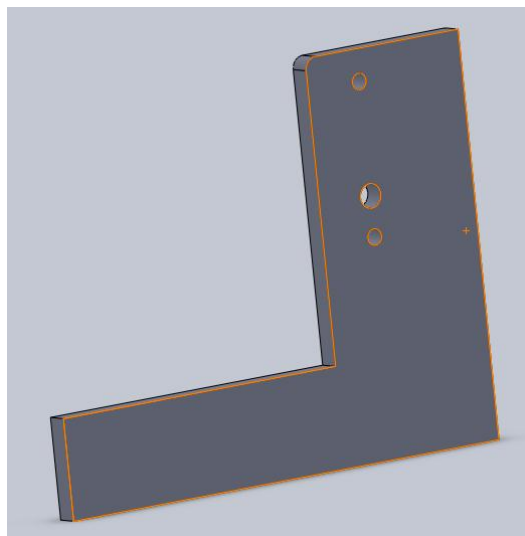


Figura N° 52: Pieza lateral derecha de eje X

Se observa en la figura N° 52 que la pieza en L número 2 tiene un agujero extra, el cual será la posición por la cual se fijará el motor paso a paso cuando se realice la instalación de la misma. Por medio de este agujero pasa la varilla roscada con el acople del motor y la varilla roscada.

Para poder realizar el eje X de la fresadora se utilizaron 2 barras redonda de 3/8" las cuales servirán como guías para el traslado del cabezal en todo el eje, cada una de las barras tendrá instaladas 2 bocinas para que se puedan desplazar por toda la barra se muestra en la figura N° 53.

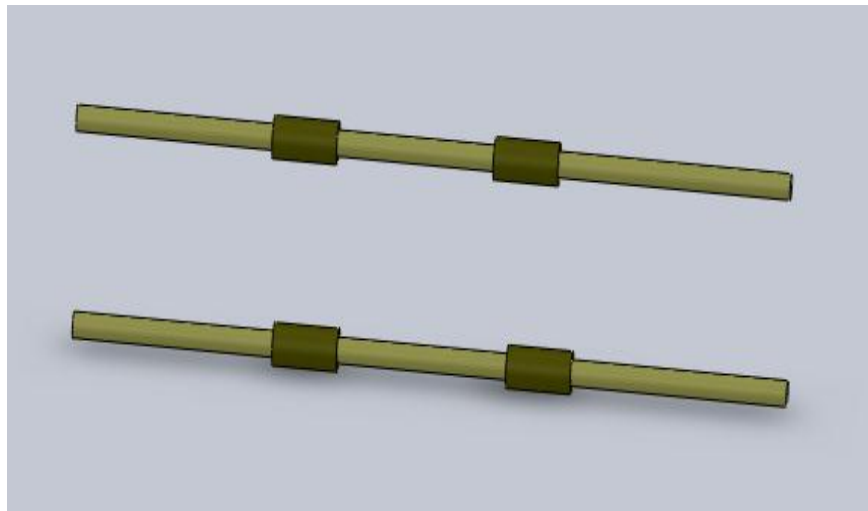


Figura N° 53: Varillas de acero guía y bocinas de eje X

Los guidores se realizan de manera milimétrica para evitar cualquier error de medida al realizar el trabajo en las pruebas futuras.

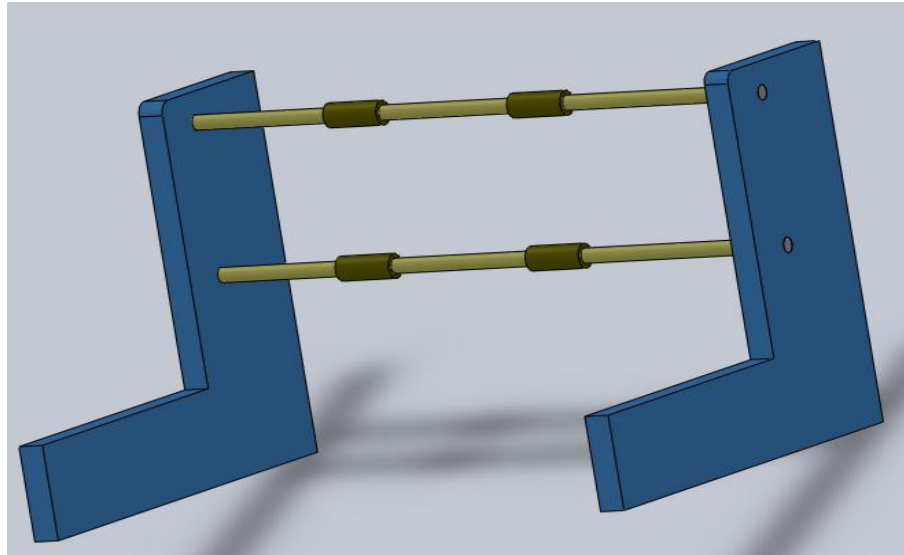


Figura N° 54: Ensamble de eje X

Al finalizar se tiene que instalar las varas redondas 5/16, en los orificios de ambos piezas de madera formando el eje X, se considera un total de 2 guidores por varilla para que luego sean fijadas el eje Z como se observa en la figura N° 54.

3.1.6.3 Diseño de eje y

Para poder realizar el eje Y de la fresadora se utilizaron 2 barras redonda de 5/16" las cuales servirán como guías para el traslado del cabezal en todo el eje, cada una de las barras tendrá instaladas 2 bocinas para que se puedan desplazar por toda la barra.

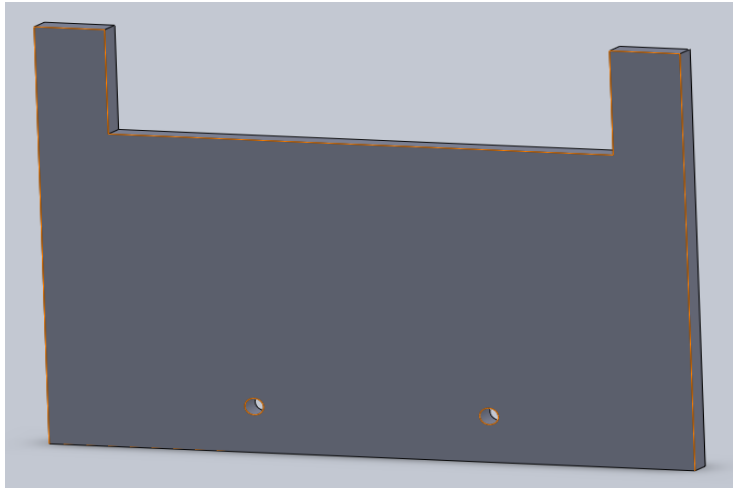


Figura N° 55: Pieza posterior de eje Y

La figura N° 55 es la parte posterior del eje Y, por esta pieza se realiza la fijación del motor para el eje Y, al igual que en el eje X esta piezas presentan 2 agujeros de para introducir las varas de guías al unir las dos piezas.

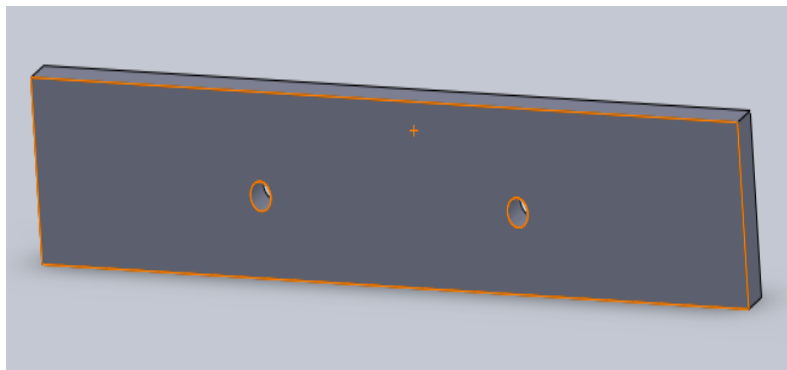


Figura N° 56: Pieza frontal de eje Y

La figura N° 56 la parte frontal es de menor tamaño debido que permite la visión del trabajo de la fresadora, de igual manera que la que la pieza posterior esta presenta dos agujeros para introducir las varillas guías.

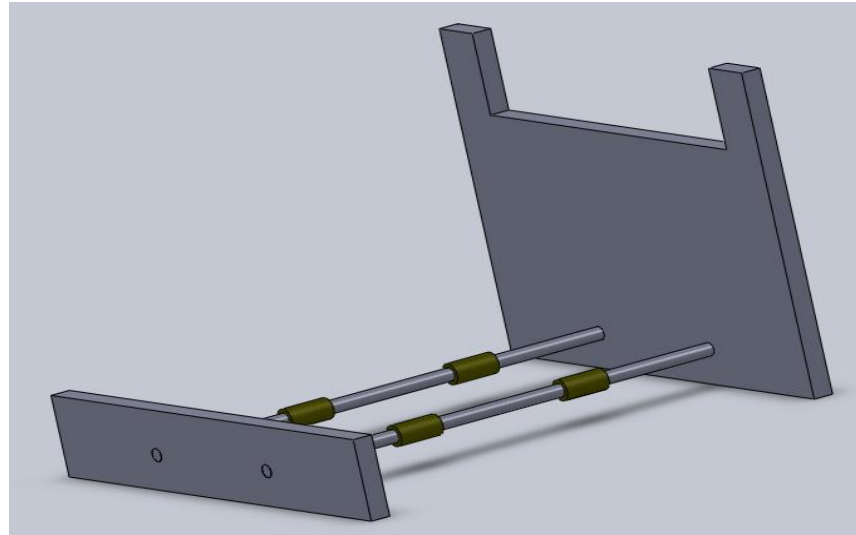


Figura N° 57: Ensamble de eje Y

Para realizar la instalación se consideran las varillas guías con un total de 4 guizadores los cuales se instalan 2 en cada guía con ello se puede posteriormente fijar la base de la mesa de fresado que ira reposada sobre los acoples, se debe tener en cuenta que los agujeros de las dos piezas deben ser exacto debido a que el error en ello, puede provocar una desnivelación en la mesa de fresado lo cual puede verse en la figura N° 57.

3.1.6.4 Diseño de eje z

Para poder realizar el eje Z de la fresadora se utilizaron 2 barras redonda de 5/16" las cuales servirán como guías para el traslado del cabezal en todo el eje, cada una de las barras tendrá instaladas 2 bocinas para que se puedan desplazar por toda la barra.

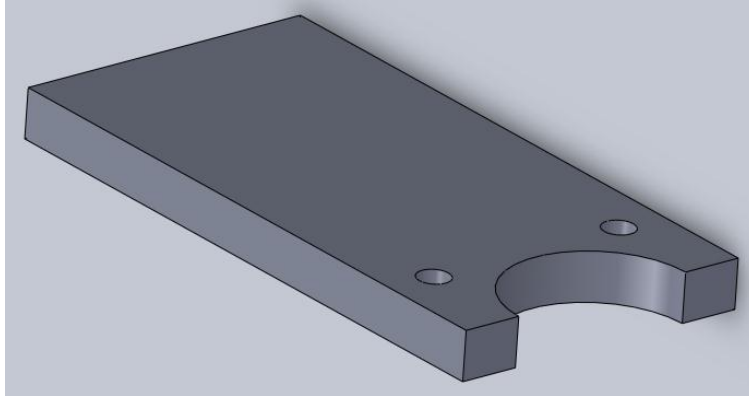


Figura N° 58: Pieza superior eje Z

Como se muestra en la figura N° 58 la primera pieza viene a ser la parte superior del eje Z, en la cual va a ser ubicada tanto las varillas guías para el eje Z, además como base del cabezal de la fresadora.

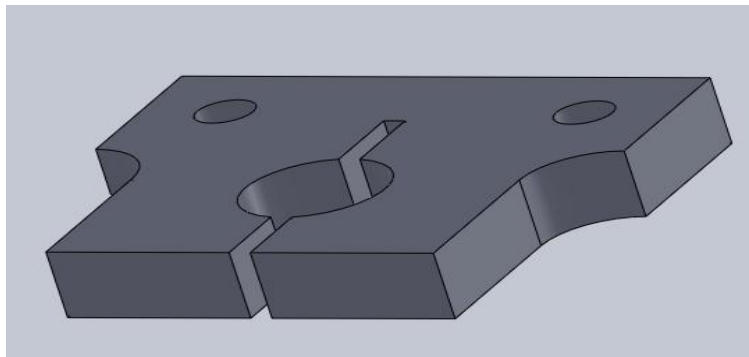


Figura N° 59: Pieza inferior de eje Z

En la figura N° 59 viene a ser la pieza inferior del eje Z, la cual se conforma de los dos agujeros para las varillas guías como también de un agujero regulable el cual permite sujetar el cabezal de la fresadora para que tenga una buena fijación.

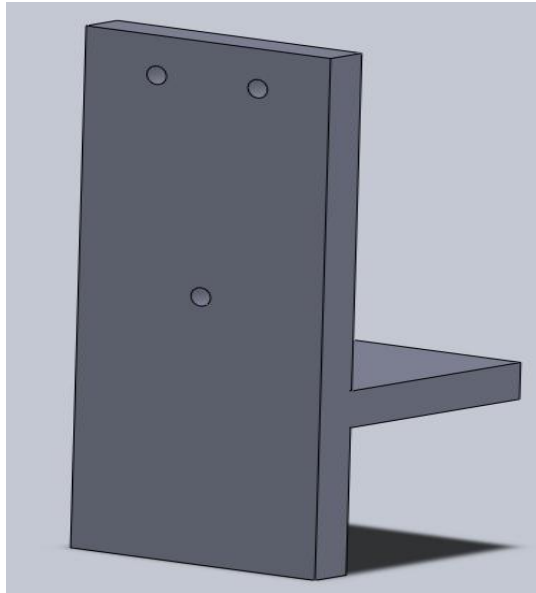


Figura N° 60: Soporte de Fresadora y Eje Z parte A

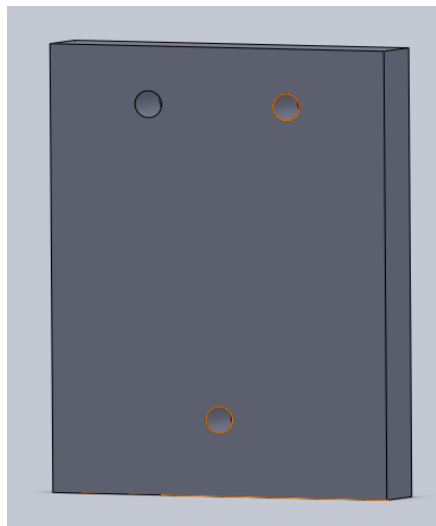


Figura N° 61: Soporte de Fresadora y Eje Z parte B

Las siguientes dos piezas (Ver figura N° 60 y 61) conforma en total la base de soporte para las guías, cabezales y el motor del eje Z, con esto como se observa en la siguiente figura al unirnos permite instalar la fresadora.

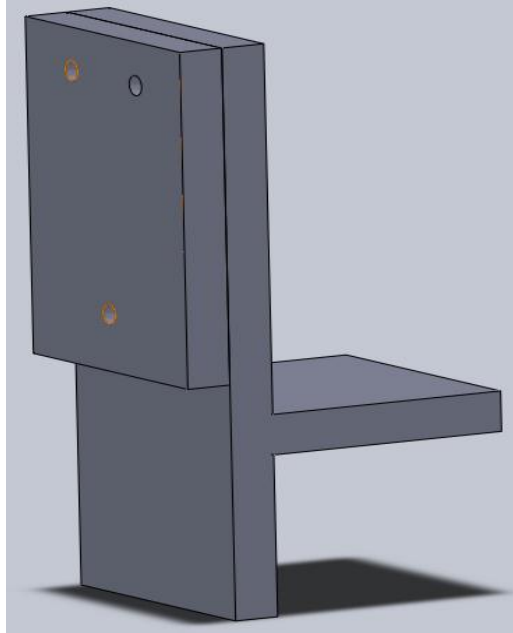


Figura N° 62: Ensamble de soportes de fresadora parte A y B

Se debe considerar al preparar las bases para el eje Z, la correcta instalación de los soportes para que el cabeza tenga trabajo perpendicularmente con la pieza a trabajar, se observa en la figura N° 62.

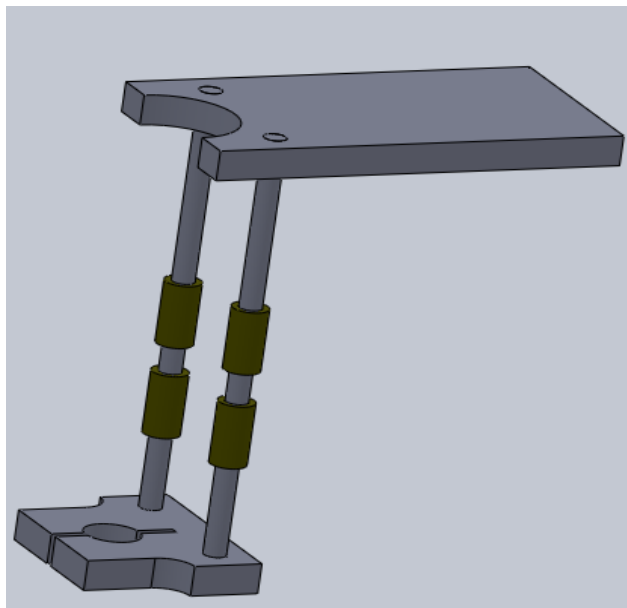


Figura N° 63: Ensamble de eje Z

En la siguiente figura N°63 se procede a instalar las varilla guías con los guidores para ello se considera tal como se

muestra en la figura N° 64, tener en cuenta de que se hayan instalados correctamente.

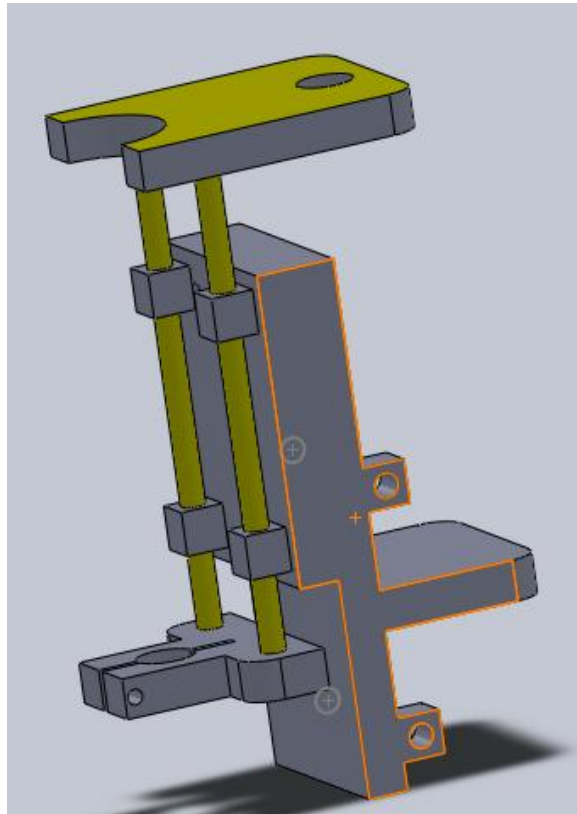


Figura N° 64: Ensamble de eje Z con soporte

Al obtener en ensamble de todas las piezas se observa el trabajo definitivo del eje Z, el cual se estará uniendo a los guidores del eje X.

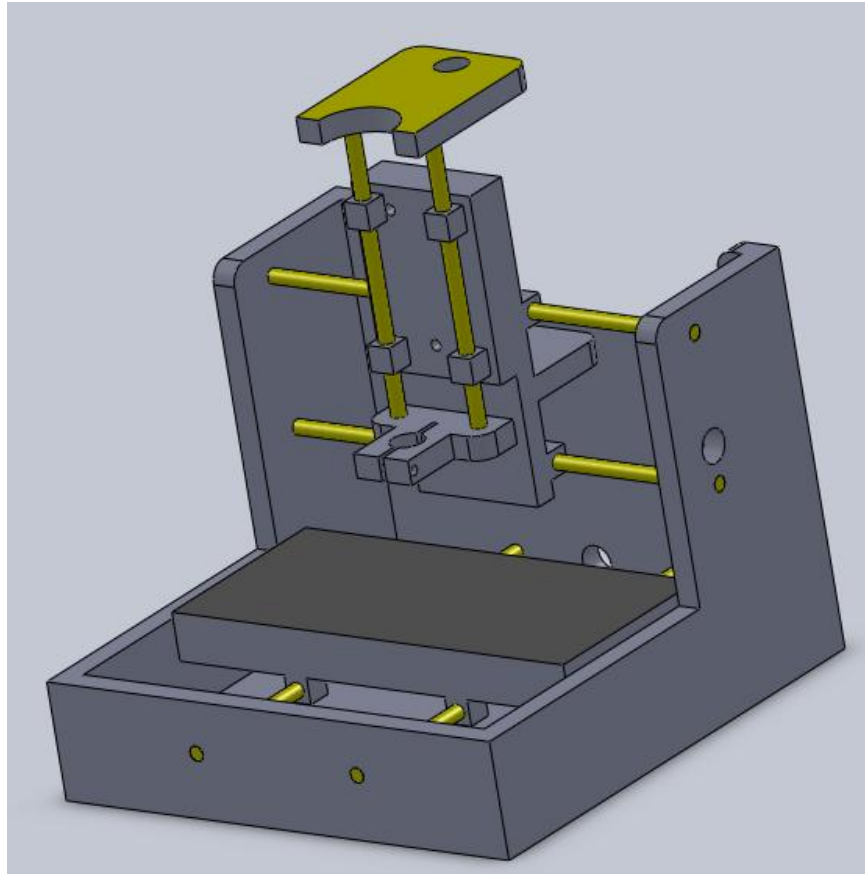


Figura N° 65: Ensamble general de la fresadora

Finalmente en la figura N° 65 al acoplar todos los ejes, se observa el resultado donde se tiene una fresadora de 276 mm x 276 mm con la capacidad de realizar tarjetas, los 3 ejes deben estar bien nivelados para evitar variaciones con el trabajo a realizar, la fresadora que se muestra tiene un rango de trabajo para placas de aproximadamente 10cms x 15 cms.

3.1.6.5 Diseño de los acoples motor – varilla roscada

Para el diseño de los acoples del motor con la varilla roscada se creará realiza mediante un varilla de fierro redondo de 3/4 “ los cuales por uno de los lados viene a ser soldado con la varilla roscada en el centro de su eje, se debe considerar que si la misma no fue soldada justo en el centro esto puede provocar que al generar el movimiento esto provoque un mal movimiento.

Para poder fijar el acople al motor se realiza el procedimiento siguiente. Se realiza un hueco con la medida del eje del motor mediante el uso de un torno, es posible realizarlo manualmente pero no muy recomendable, para fijar el eje con el motor se utiliza un perno allen de tipo prisionero el cual se fija y sujeta el eje el motor por el lado lateral.

De esta manera en la figura N° 66 se logra fijar tanto la varilla roscada como el eje del motor paso a paso.



Figura N° 66: Ensamble de motor con varilla roscada

3.2 Construcción de la fresadora CNC

3.2.1 Consideraciones para el armado de los ejes de la fresadora

Para la construcción de la fresadora se debieron tener las siguientes consideraciones con la finalidad de mantener la precisión en la construcción del material.

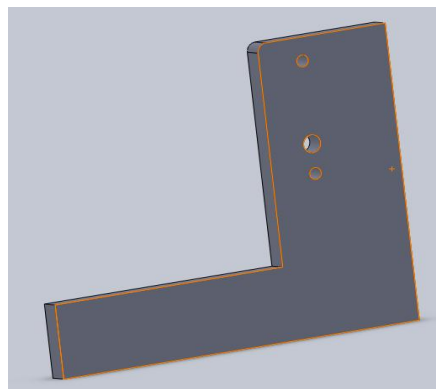


Figura N° 67: Pieza de la fresadora

El material para la realización de la fresadora fue madera MDF 12 mm, debido a que con respecto a los planos utilizados este es el espesor requerido para la instalación.

Para poder mantener nivelados todas las piezas de la fresadora se tuvo que realizar lo siguiente:

- Lijado de todos lados de las piezas para evitar que el corte realizado pueda generar un desnivel, lo que provocaría que la máquina no encaje en el momento del armado.
- Taladrado debe ser realizado con para los ejes X e Y, se debe considerar tener las dos piezas unidas y haber sido lijadas juntas, además de utilizar un taladro de mesa para tener la mejor precisión en el agujero, se requiere considerar estos detalles debido a que por estos se colocarán las varillas lisas que sirven como guías.
- Instalación de los soporte de tuerca (Tee nuts o Araña) estos componentes deben ser instalados antes de la instalación y armado de la máquina ya que estos servirán como soporte de las piezas a mecanizar posteriormente.
- Realizar los trabajos en una superficie lisa y nivelada ya que esto puede incurrir en problemas de nivelación cuando se esté armando toda la fresadora en conjunto.

3.2.2 Instalación de varillas lisas y armado de ejes

3.2.2.1 Instalación de las varillas lisas del eje Y

Para la instalación del eje Y se utilizó las piezas 1, 2, a las cuales se tuvieron en cuenta algunas modificaciones debido a que no se encontró el material indicado por David Carr, se utilizó varillas lisas de 3/8", además de casquillos aluminio fabricados con tubo de aluminio de 1/2".

Para realizar la instalación se colocaron las varillas lisas en los agujeros previamente realizados con 2 casquillos de aluminio en cada varilla lisa, se debe considerar que la medida entre las dos piezas debe ser de 276 mm, en el caso quede muy largo o muy corto se deben o bien abrir o cerrar para mantener la distancia requerida se muestra los resultados en la figura N° 68.



Pieza N° 68: Ensamblaje de Eje Y

3.2.2.2 Instalación de las varillas lisas del eje X

Para la instalación del eje X se utilizó las piezas 3, 4 a las cuales se realizaron los agujeros para varillas lisas de 3/8", además de los casquillos de aluminio de 1/2".

Para realizar la instalación se colocaron las varillas lisas en los agujeros previamente realizados con 2 casquillos de aluminio en cada varilla lisa, se debe considerar que la nivelación de las varillas sea el correcto debido a que este eje soportara al eje Z y al cabezal que realizará el trabajo, su desnivelación provocará fallos en el resultado esperado.

La distancia entre las piezas 3 y 4 debe ser de 276 mm por lo que se debe considerar esta distancia tanto en la parte superior e inferior para evitar que esta mantenga la medida pero se encuentre descuadrada, se muestra los resultados en la figura N° 69.

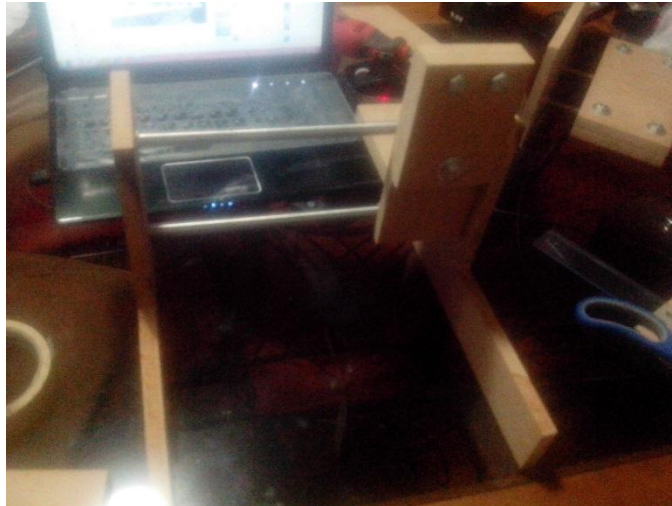


Figura N° 69: Ensamble de eje X y soporte de fresadora

3.2.2.3 Cuadrado de ejes X e Y

Paso seguido a realizar los armados de los ejes X e Y se requiere cuadrar las piezas, por lo que para el eje Y consiste en nivelar las bases de las piezas 1 y 2 con el nivel de la mesa donde se realiza el trabajo, esto se logra realizando presión hacia la mesa. De la misma manera se consigue para el eje X donde haciendo la misma presión hacia la mesa para lograr la nivelación.

Para cuadrar los ejes con se deben ensamblar el eje X sobre el eje Y, donde las mismas deben formar un cuadrado de 276mm x 276mm que es justamente la medida considerada en ambos ejes, en el caso se encuentre una variación en la medida se puede ajustar y acercar las piezas para poder cuadrar los ejes de la fresadora.

3.2.2.4 Ensamble de la mesa de fresado en el eje Y

Al realizar la instalación de la mesa de fresado en el eje Y, se debe haber realizado previamente la unión de las piezas 5 y 6 por medio del soporte de tuerca (Tee nut) e instalado los pernos para tener un solo bloque.

Para realizar la instalación de la mesa se debe invertir el eje Y y soportarlo en el borde de la mesa de trabajo, la mesa de fresado va a ser fijada mediante los 4 casquillos de aluminio que se encuentra ya instalados en la varilla lisa, para fijar los casquillos se utiliza masilla plástica, se consideró frente a otros materiales como el epoxy, un mejor componente para fijar la mesa.

La mesa de fresado debe reposar sobre los 4 casquillos de igual manera en todos los lados para una primera etapa solo es necesario aplicar la masilla plástica en un solo lado de los casquillos ya que en el caso de error este debe ser retirado.

La mesa de fresado deberá queda como se muestra en la figura N° 70 siguiente:



Figura N° 70: Instalación de mesa de fresado

3.2.2.5 Ensamblaje del soporte del eje Z y cabezal de la fresadora en el eje X

Al realizar la instalación del soporte del eje Z y cabezal de la fresadora en el eje X, se debe haber realizado previamente la unión de las piezas 7 y 8 por medio del soporte de tuerca (Tee nut) e instalado los pernos para tener un solo bloque.

Al soporte debe haberse instalado previamente el soporte del motor del eje Z, debido a que después de instalado es complicado y puede dañar la fijación de los casquillos del eje.

Para realizar la instalación del soporte el eje X debe estar invertido y ubicado en el borde de la mesa de trabajo, para la instalación se debe considerar que el soporte se encuentre nivelado y que los casquillos reposen de igual manera en todo el soporte. Para fijarlos se utiliza de igual manera que en el eje Y masilla plástica y debe ser instalada a un solo lado del casquillo para que este pueda verificarse posteriormente, los resultados se observan en la figura N° 71.



Figura N° 71: Fijación de soporte de fresadora en eje X

3.2.2.6 Instalación de las varillas en el eje Z

Para realizar la instalación del eje Z se debe tener principal consideración debido a que de este depende la correcta instalación del cabezal y que este trabaje de manera nivelada evitando cualquier inconveniente con el trabajo a querer mecanizar.

Para comenzar se debe tener las piezas 9 y 10, las cuales no es necesario que se lijén con la finalidad de tenerlas niveladas ya que no depende ello, lo que debe realizarse es los agujeros con la mayor precisión posible.

Para las piezas 9 y 10 se utilizó una madera MDF de 15 mm debido a que se requiere que se tenga una mayor fijación de las varillas lisas de 3/8", para poder realizar la fijación de las varillas lisas se debe agregar un poco de pegamento epoxi en la pieza 10, posteriormente agregarle un poco de cinta para instalar los casquillos y que los mismos no pueda juntarse con el pegamento, para finalizar se instala la pieza 9 en los agujeros respectivos, se observa en la figura N° 72.



Figura N° 72: Fijación de fresadora y eje Z en soporte

3.2.2.7 Ensamble del eje Z al eje X

Para realizar la instalación del EJE Z al EJE X, se debe reposar el eje X para poder mantener lo más nivelado el soporte del Eje Z, para su instalación los casquillos del eje Z deben reposar sobre el soporte de manera equitativa, para fijarlos se utiliza la masilla plástica en uno de los lados de los casquillos para verificar que el eje funciona correctamente.

3.2.2.8 Pruebas de movimiento de los ejes

Después de haber instalado los ejes estos deberán moverse en todo el rango en el caso del eje Y la mesa de fresado deberá moverse en todo su rango, si se observa alguna fricción debe ser revisada, para el caso del eje X de igual manera en todo el rango.

Para el eje Z se debe verificar que baje y suba lo más nivelado posible debido a que este eje es el que soporta el cabezal de la fresadora. En el caso que funcione correctamente todos los ejes de la fresadora se debe de utilizar la masilla plástica para cubrir todos los casquillos y estos puedan ya quedarse de manera fija y definitiva en la máquina.

3.2.3 Instalación de varillas roscadas y motores

3.2.3.1 Instalación del motor y varilla roscada en eje Y

Para la instalación del motor y la varilla roscada M8 del eje Y se utilizó un acople el cual se sujeta mediante un perno prisionero al eje del motor, tener en cuenta que si la varilla roscada no se instaló de manera correcta con el acople, el motor al funcionar provocará que la misma no gire correctamente provocando que la mesa de fresado también se observe el movimiento.

El motor del eje Y se fija en la pieza posterior por medio de 2 stove bolts, la varilla roscada pasa por el agujero que se

encuentra en la pieza posterior del eje Y, para realizar la fijación se utilizan 2 tuercas M8 unidas para tener un mayor espacio en la aplicación de la masilla plástica, el resultado se observa en la figura N° 73.



Figura N° 73: Instalación de varilla roscada en eje Y

Para realizar la fijación las tuercas deberán estar en el centro y equidistante a las varillas lisas del eje Y, las tuercas son fijadas con masilla plástica en un solo lado hasta realizar la verificación.

El movimiento de la mesa de la fresadora se genera cuando el motor gire provocara que la varilla roscada M8 también gire provocando que las tuercas generen un desplazamiento en todo el eje según el giro del motor (Ver figura N° 74).



Figura N° 74: Instalación de motor en eje Y

3.2.3.2 Instalación del motor y varilla roscada en eje X

El motor del eje X se fija a la pieza lateral derecha por medio de 2 stove bolts, la varilla roscada pasa por el agujero que se encuentra en la pieza lateral derecha del eje X, para realizar la fijación se utilizan 2 tuercas M8 unidas para tener un mayor espacio en la aplicación de la masilla plástica.

Para la fijación las tuercas se encontrarán más cerca a la varilla lisa inferior, debido a que en ese punto no va a interferir con la posición del motor del eje Z, debido a que este eje justamente sujeta el eje Z, se observa en la figura N° 75.



Figura N° 75: Instalación de varilla roscada en eje X

El movimiento del soporte del eje Z se genera cuando el motor gira, provoca que la varilla roscada M8 también gire provocando que las tuercas generen un desplazamiento en todo el eje X, tener en consideración que si se gira el motor más de sus rangos estos pueden dañar el trabajo, tal como se observa en la figura N° 76.



Figura N° 76: Instalación de motor en eje X

3.2.3.3 Instalación del motor y varilla roscada en eje Z

El motor del eje Z se fija en la base que se encuentra en la parte posterior del soporte del cabezal y del eje Z, este se fija por medio de 2 stove bolts, este motor se instala de manera vertical a diferencia de los otros ejes, debido a que este se encarga de la altura en el trabajo de la fresadora.

Para la fijación las tuercas se encuentran una en la parte inferior del soporte superior del eje Z y uno en la parte superior, para este eje se utiliza arandelas en ambos lados para mejor fijación.

El movimiento del cabezal de la fresadora se genera cuando el motor gira, provoca que la varilla roscada M8 también gire provocando que las tuercas generen un desplazamiento en todo el eje Z como se observa en la figura N° 77, tener en consideración que si se gira el motor más de sus rangos estos pueden los topes.

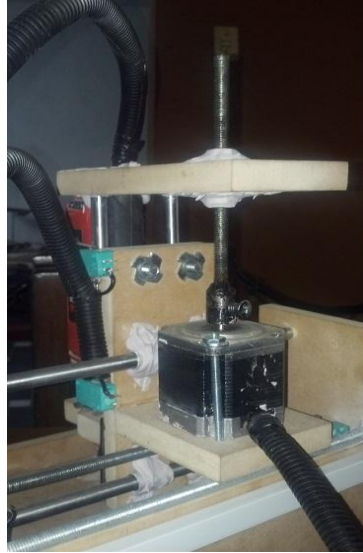


Figura N° 77: Instalación de motor y varilla roscada en eje Z

3.2.4 Instalación de finales de carrera

Para evitar que la fresadora sobrepase los rangos establecidos se utilizan un total de 6 finales de carreras ubicados en el eje X, eje Y y eje Z como se muestra en la figura N° 78, los cuales en el momento que lleguen al tope se activaran indicando al programa que se encuentra en el tope.

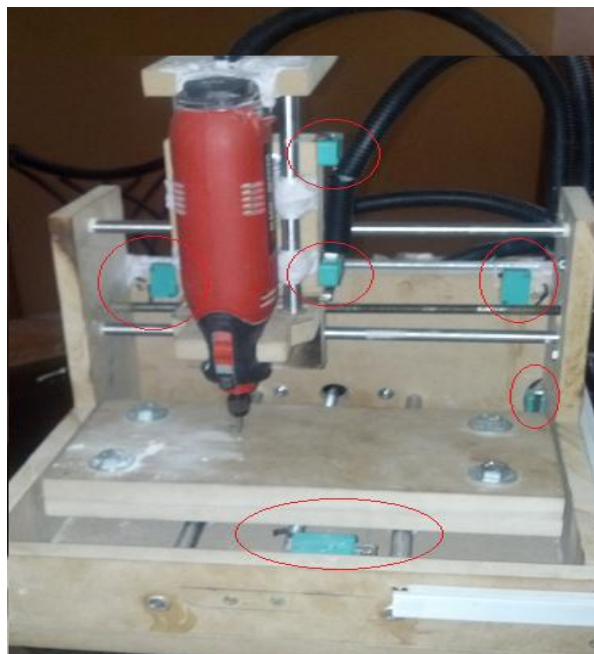


Figura N° 78: Instalación de finales de carrera

3.2.5 Fabricación del tablero eléctrico para la fresadora CNC

Para la instalación del tablero eléctrico de la fresadora se consideraron las etapas de control y adquisición en la parte superior, en la parte intermedia la etapa de alimentación y control de la fresadora, y en la parte inferior la etapa de potencia de los motores paso a paso.

3.2.5.1 Parte superior etapa de control y adquisición

Para la parte superior (ver figura N° 79), se consideró en la etapa de control el arduino, además del controlador previo a las señales para los motores paso a paso fabricado con un circuito integrado cd40106 que consiste en un schmit trigger, el cual enviará la señal positiva y negativa a los controladores de los motores paso a paso, para terminar con la etapa de control y adquisición estos se encuentra el circuito de para los finales de carrera, los cuales se conectan con la máquina por medio de un conector DB9 adaptado.



Figura N° 79: Etapa de control y adquisición de datos

3.2.5.2 Parte intermedia etapa de alimentación y activación del cabezal

Para la parte intermedia (Ver figura N° 80), se considera la etapa de alimentación por medio de la fuente ATX, la cual se utilizó las señales de 5v, GND, 12V los cuales brindaran las señales correspondientes a todos los circuitos del tablero eléctrico, además se considera que la fuente ya fue puenteada por lo que al conectarse y activar el switch se encenderá sin problemas.

Para la activación del cabezal se utiliza el circuito por medio de un relé por lo que al activarse por medio del arduino este envía 220v al conector habilitando el cabezal.



Figura N° 80: Etapa de alimentación y activación de cabezal

3.2.5.3 Parte inferior etapa de potencia

En la parte inferior del tablero (Ver figura N° 81) se encuentran las controladoras L298 de los motores paso a paso distribuidos de la siguiente manera, para el eje X una controladora, para el

eje Y dos controladoras en paralelo y para el eje Z una sola controladora.

Para poder conectar la etapa de potencia con los motores se ha dispuesto de conectores DB9 adaptados para que envíen la señal a los motores de los 3 ejes, para mantener las controladoras lo más frías posibles se han instalado dos coolers de 12v, con lo que uno funciona insertando el aire, el otro cooler funciona retirando aire, generando un flujo de aire constante por todas las controladoras.

Cabe considerar que se han instalado foco led de indicacion y un switch de encendido de la máquina, además de conector para la fuente de alimentación y conexión USB para conectar el arduino.

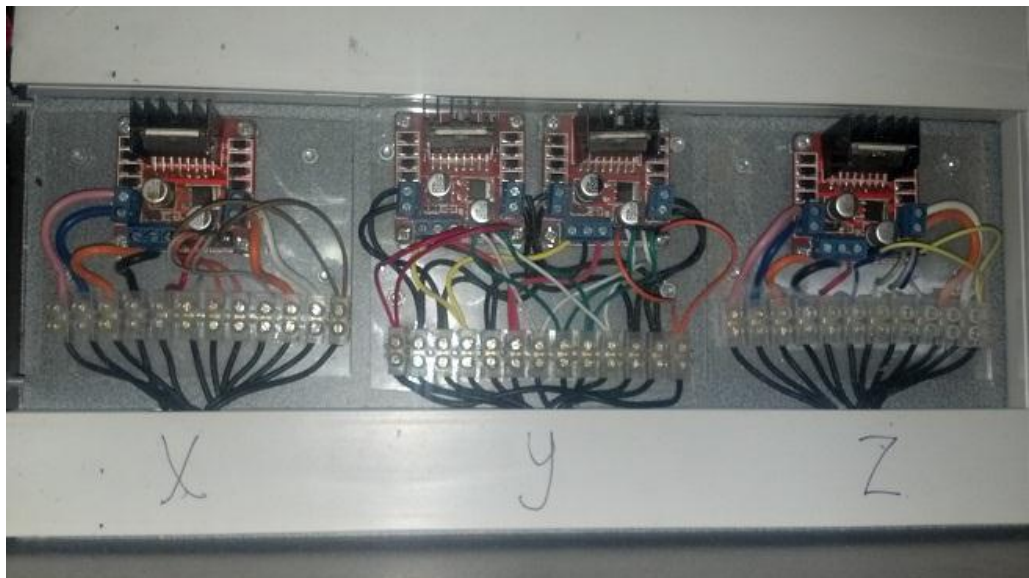


Figura N° 81: Etapa de potencia

3.3 Pruebas y resultados

3.3.1 Pruebas

Para realizar la prueba de la fresadora CNC, se inicia con la identificación del circuito eléctrico, por tanto se observa en la siguiente

figura N°82 el circuito eléctrico realizado en el software ISIS de Proteus 8 Professional.

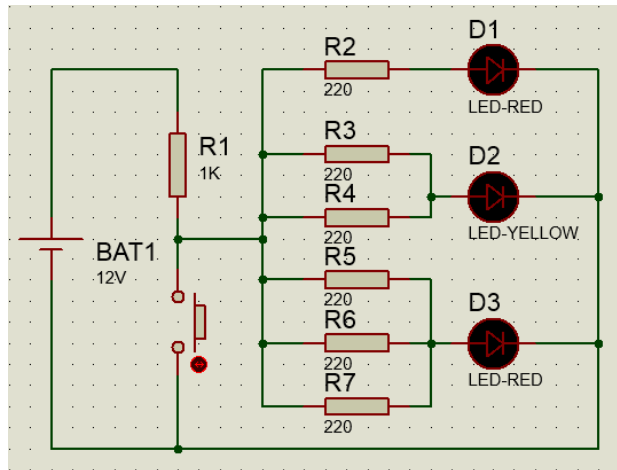


Figura N° 82: Circuito eléctrico para pruebas en software ISIS PROTEUS

El circuito eléctrico consta de resistencias, leds, botón y una batería que activa el circuito eléctrico, luego de ello lo que se debe realizar es trasladar a un software de diseño para placas en este caso se utiliza el ARES Proteus 8 profesional (Ver figura N° 83). Se debe tener en cuenta la mejor distribución de los componentes para que se pueda realizar el trabajo de mecanizado de la mejor manera.

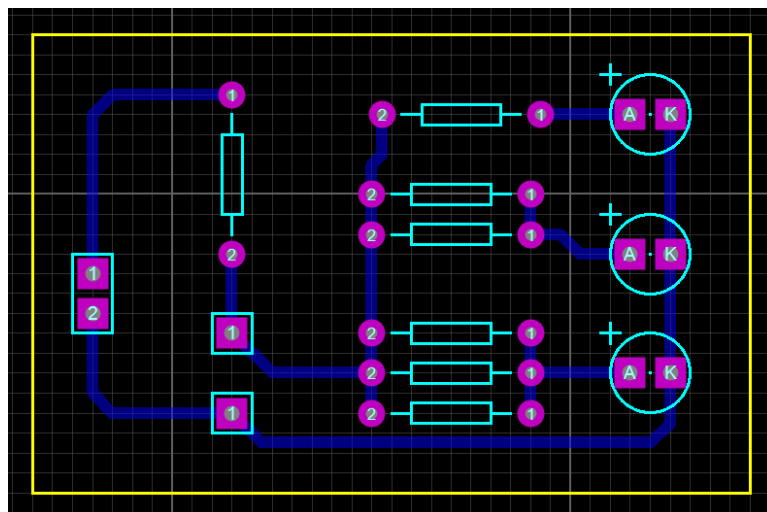


Figura N° 83: Diseño de pistas en software ARES PROTEUS

Luego para asegurar el estado de las pistas se observa que por medio de la opción 3d visualization el acabado final de la tarjeta electrónica como se observa en la figura N° 84.

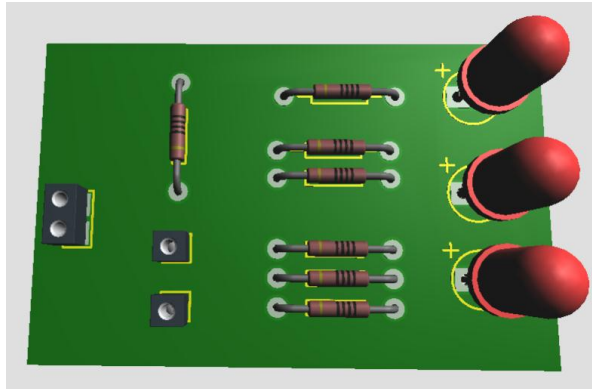


Figura N° 84: Vista en 3D de la placa

Para finalizar la etapa del diseño de la tarjeta electrónica por lo que se va a exportar mediante un formato PDF teniendo la consideración que sé debe imprimir en modo espejo y sólo la capa posterior en el cual se observa las pistas del circuito electrónico.

Ya con la tarjeta electrónica diseñada y en formato PDF se tiene que importar al programa Inkscape para poder generar el archivo de mecanizado, el software inkscape sólo soporta archivos .SVG por lo que se realiza la conversión de PDF a .SVG para hacerlo compatible con el software Inkscape, se muestra el resultado en la figura N° 85.

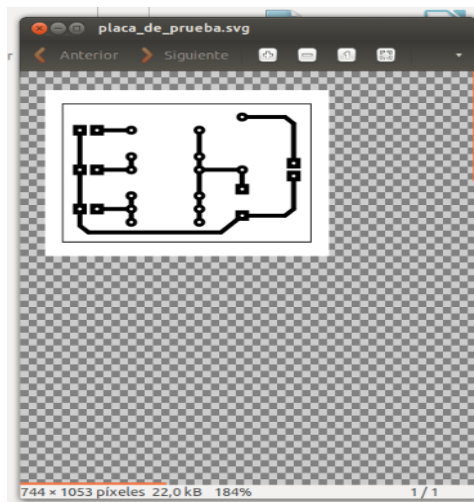


Figura N° 85: Pistas en formato .SVG

Luego de haber transformado el archivo .SVG se puede importar al software Inkscape con el que se va a realizar el mecanizado para generar el archivo para la máquina fresadora CNC.

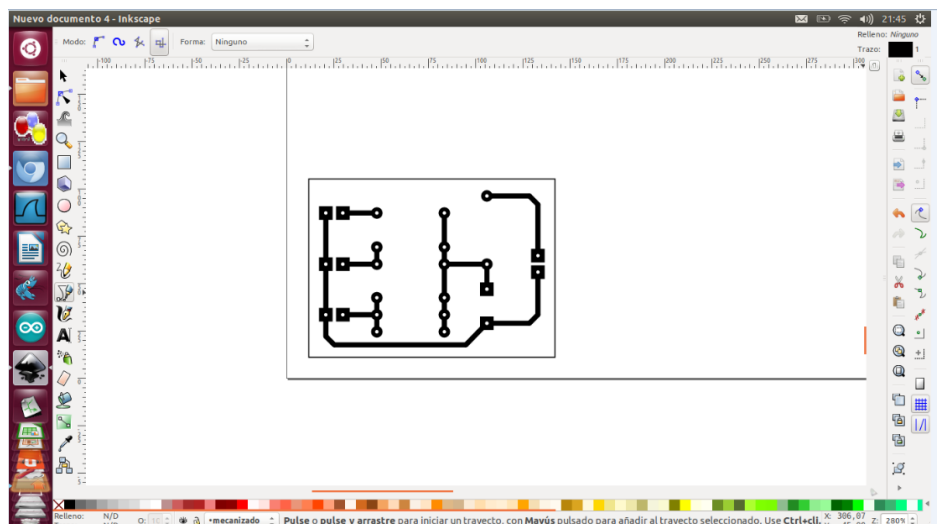


Figura N° 86: Pistas importadas en software INKSCAPE

Luego de haber realizado la importación del archivo (Ver figura N° 86) lo que continúa es trazado o dibujo por encima de la placa que ya se tiene, por motivos de pruebas se utilizó la herramienta bezier para y se han líneas rectas tanto horizontales como verticales delimitando la posición de los componentes. La siguiente figura N° 87 muestra cómo queda después de haber delimitado todas las pistas.

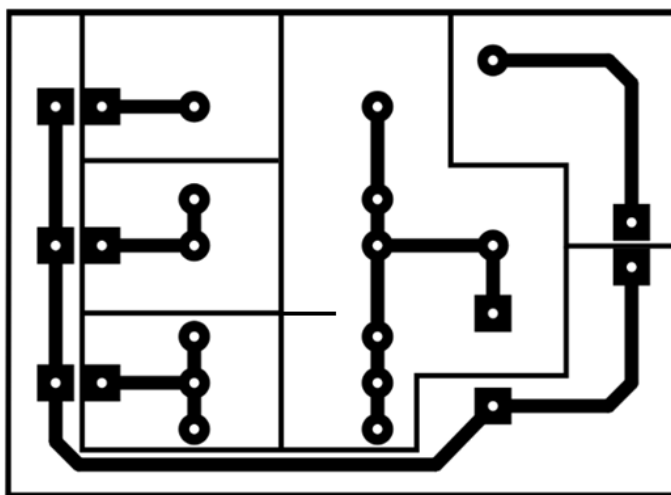


Figura N° 87: Diseño de pistas a trazar

Los procedimientos siguientes que se realizan para las pruebas es el de configurar el diseño previamente mostrado lo primero que se debe realizar es asignar el punto de orientación para el diseño, por medio

de la herramienta orientation points se asigna el punto de inicio del trazado, el resultado se observa en la figura N° 88.

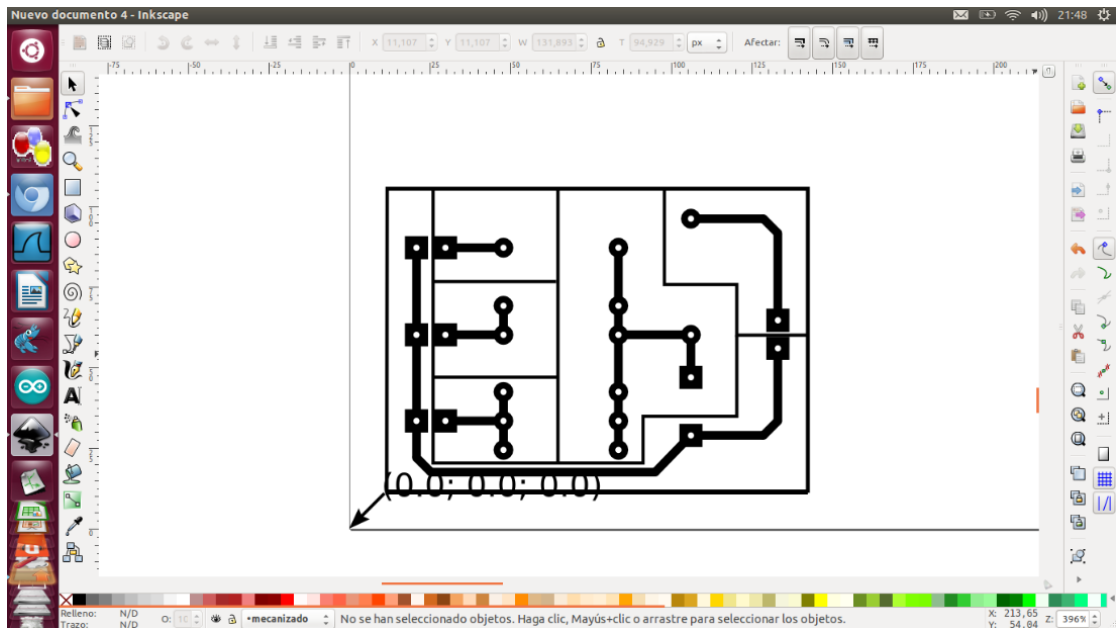


Figura N° 88: Fijación de punto de inicio con orientation points

Para la elección de la herramienta se debe escoger uno de tipo cilíndrico, la velocidad para el trazado se define según el tipo de motor que se tenga, la máquina se puede configurar la velocidad de los motores desde 1200 a 2500, observando que para una mejor precisión en el momento del trazado se utiliza una velocidad de 2200 con la cual no presenta ningún inconveniente. La velocidad para levantar la fresadora cuando no va a ser el trazo va a ser de 500 ya que no se necesita un movimiento rápido al momento de realizar los cambios de posición, en la siguiente imagen N° 89 se adjunta los parámetros ya asignados a la imagen que va a ser trazada.

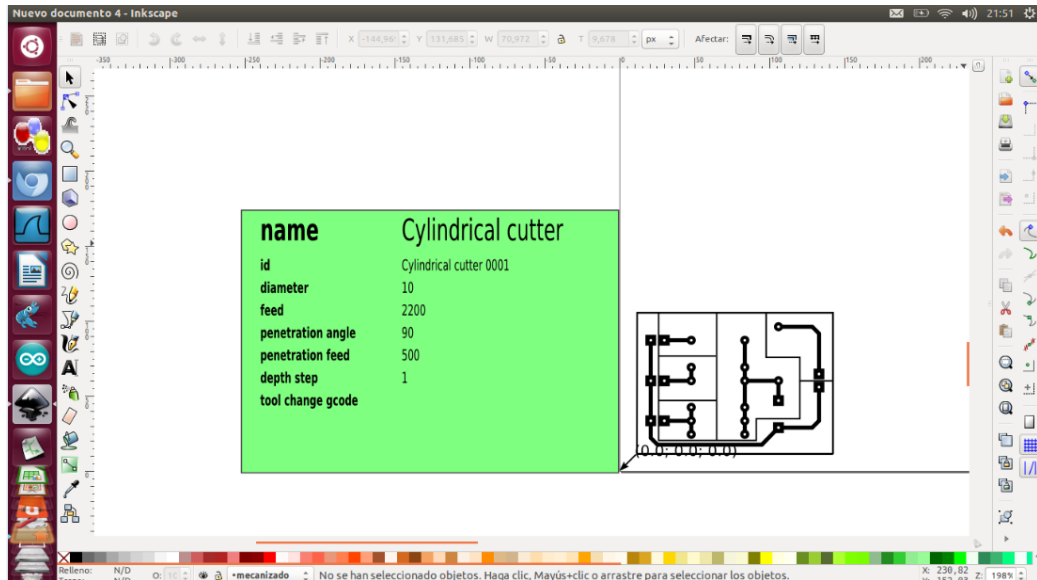


Figura N° 89: Selección de herramienta para fresado

Luego de proceder en escoger la herramienta y las velocidades de trabajos se debe generar el trazado y el archivo de código G, para esto se utiliza la herramienta Path to Gcode la cual va a realizar el mejor trazado para cubrir todo el diseño como el menor número de instrucciones, como se observa en la figura siguiente N° 90 se genera el archivo mecanizado.ngc y se muestran los trazos que se van a seguir cuando se haga la placa de circuito eléctrico.

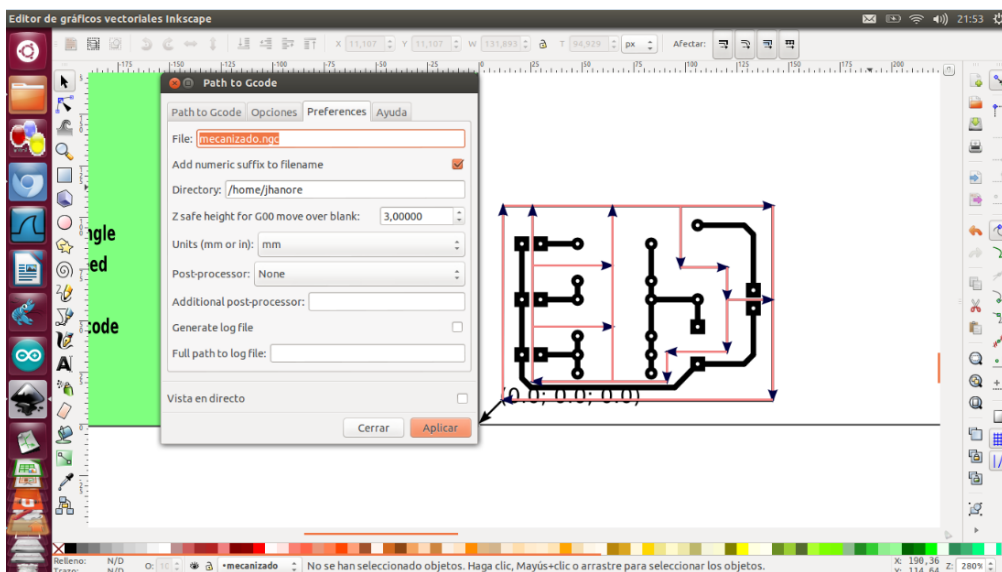


Figura N° 90: Trazos a seguir por fresadora CNC

Posteriormente se observa en la figura N° 91 que el archivo fue generado con el nombre mecanizado 001.ngc con el cual se observa que el código G para la fresadora se ha creado sin inconvenientes.

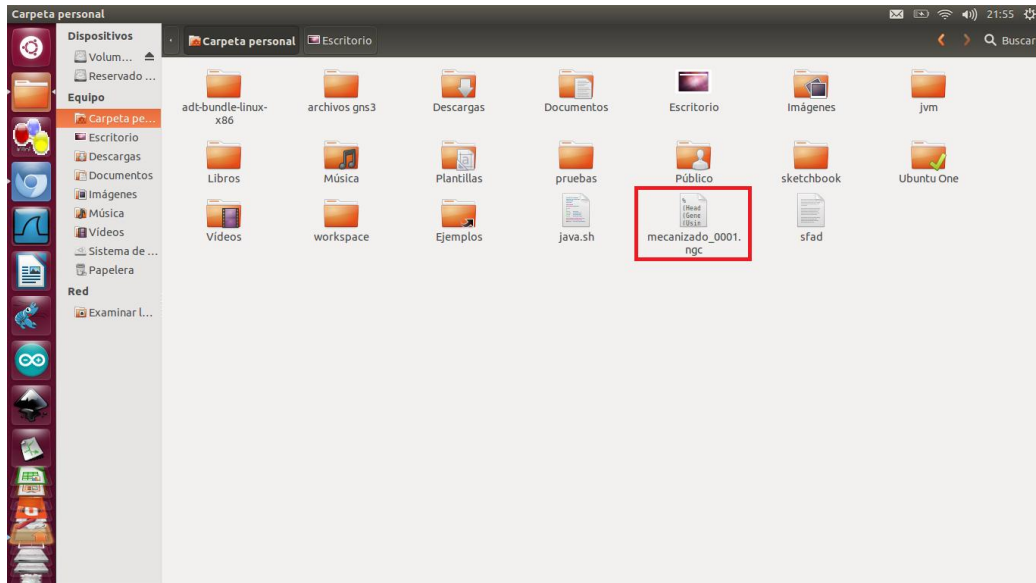


Figura N° 91: Creación del archivo de mecanizado

Se realizaron algunas modificaciones al archivo mecanizado_0001.ngc con la finalidad de tener las instrucciones con valores absolutos para debido a que el software solo lee e interpreta este tipo de valores por lo que, después de la modificación se observa cómo queda el archivo, se observa la modificación en la figura N° 92.

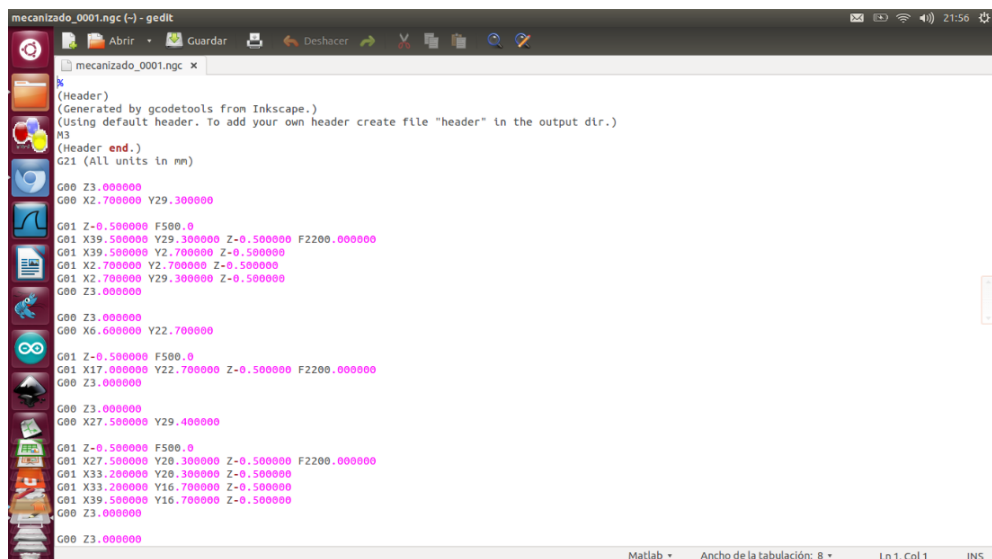


Figura N° 92: Modificación al archivo de mecanizado

Ya para finalizar sólo queda abrir el archivo en el programa de la fresadora para poder realizar la placa del circuito eléctrico, sólo queda posteriormente agregar una placa PCB en para que realice el grabado correspondiente se observa el archivo en la figura N° 93.

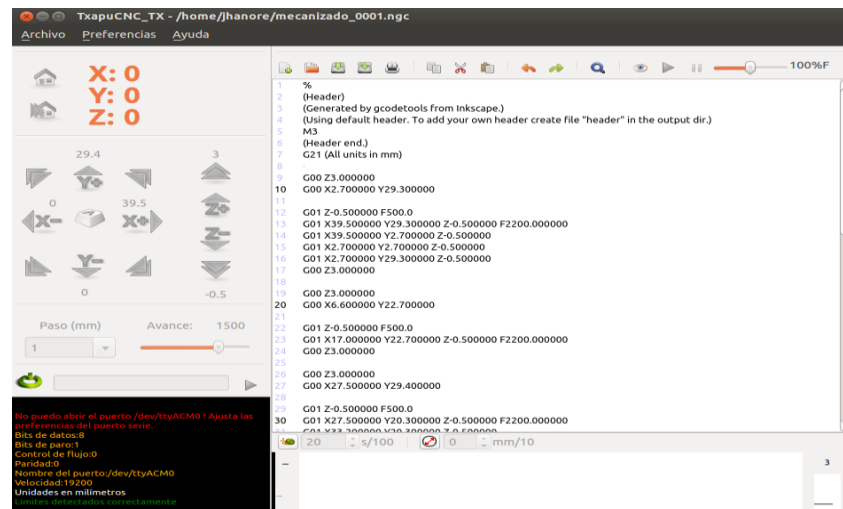


Figura N° 93: Archivo de mecanizado en software de fresadora

Para comenzar a realizar el trazado de las pistas se tiene que ubicar la herramienta de fresado en el punto de inicio y se verifica que funcione tanto la herramienta de fresado como los finales de carrera para que no salga del espacio de trabajo evitando cualquier daño, luego de haber comprobado lo mencionado se procede a abrir el programa para comience a realizar el trazado, como muestra la figura N° 94.



Figura N° 94: Ubicación de herramienta de fresadora en punto de origen

Luego de iniciar el programa para mejorar la precisión de la máquina se utiliza cinta doble contacto para sujetar la placa mientras el trazado se va realizando, tras el momento en el que se va realizando el trazado se debe observar que la placa no se desprenda como que los límites de carrera no sean activados lo que puede provocar un fallo en el trazado, después de finalizar el trazado la herramienta de fresado se elevará para poder retirar la placa, como se observa al realizar el trazado se genera residuos de la placa PCB por lo que se recomienda evitar que estas caigan a las guías de la fresadora, el resultado se observa en la figura N° 95.



Figura N° 95: Fabricación de las pistas

Al terminar de realizar el trabajo se observan los resultados en la figura N° 96 para verificar el estado de la placa después del trazado, en la cual se observa que ha delimitado bien las áreas como se había diseñado en el software Inkscape.

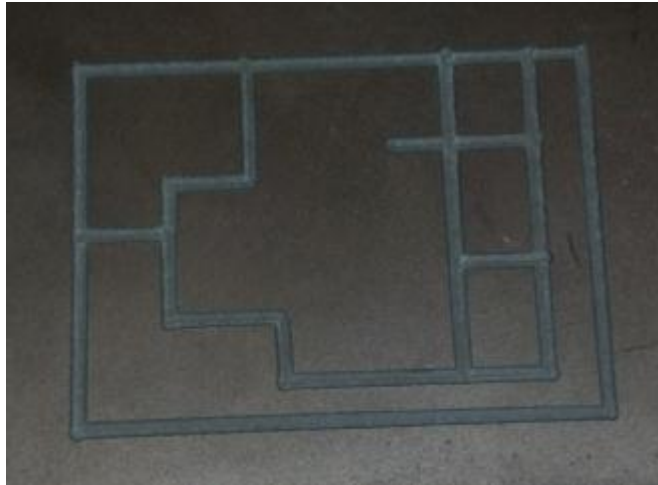


Figura N° 96: Resultados de la placa PCB

Se pasa a realizar los agujeros de la placa basándose en los puntos que se tienen en el diseño de la placa diseñada en ARES (Ver figura N° 97) con la finalidad de realizarlos, se puede realizar con la fresadora con un software de habilitación de agujeros, se considera que si se desea realizar el taladrado se tiene que cambiar la herramienta de la fresadora para poder realizar los agujeros.

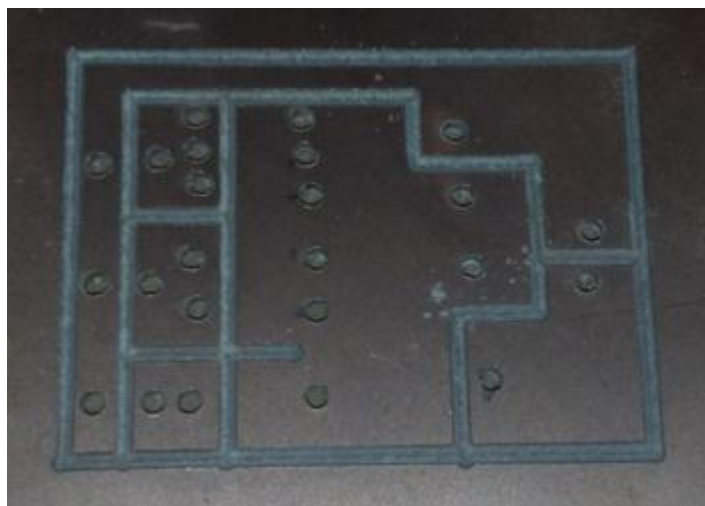


Figura N° 97: Fabricación de agujeros en placa PCB

Para finalizar lo que queda es soldar los componentes indicados en el circuito electrónico los led, resistencias y botones para poder verificar el funcionamiento del circuito eléctrico en la figura N° 98.

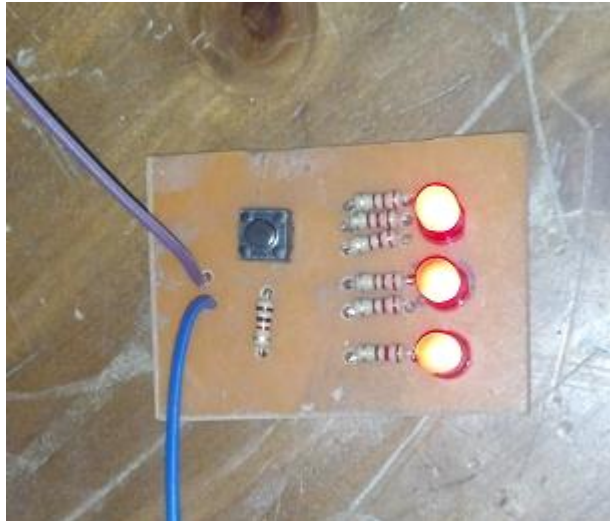


Figura N° 98: Funcionamiento de la placa PCB

3.3.2 Resultados

Después de haber realizado todo el proceso de fabricación de las placas según el circuito indicado anteriormente se observan los siguientes resultados en la siguiente figura N° 99:

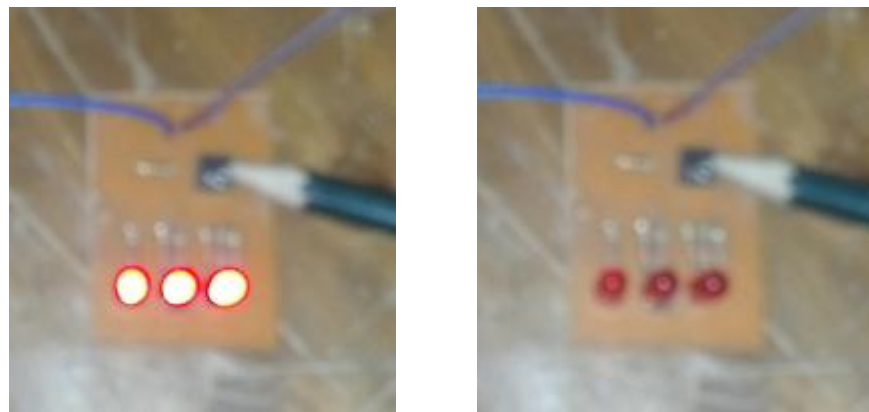


Figura N° 99: Pruebas de funcionamiento

Se observa que el circuito funciona correctamente por lo que se puede dar como resultados que el diseño del mecanizado y la fabricación del mecanizado funcionan correctamente teniendo en cuenta las consideraciones indicadas en el proceso de pruebas.

Se puede mejorar la precisión de la máquina fresadora en la fabricación del mecanizado teniendo en cuenta la velocidad de los motores y la velocidad de la herramienta de fresado, además de tener

en cuenta el tamaño de las herramientas de fresado ya que mientras mayor sea el espesor utilizará más espacio para realizar las pistas lo que puede provocar que los componentes no entren correctamente en el circuito.

Se pueden realizar circuitos desde 1x1 cm hasta de 10x10cm teniendo los rangos máximos de la fresadora, se debe tener en cuenta que el mismo tablero de control puede ser utilizado en máquinas de mayor tamaño.

Como se observa el proceso de la fabricación de las placas puede tomar muy poco tiempo en realizar por lo que se acorta el tiempo de fabricación y se obtiene una mejora en la calidad de frente a otros métodos.

CONCLUSIONES

- Se logró diseñar e implementar una fresadora CNC de 3 ejes de libertad para la fabricación de tarjetas electrónicas usando Arduino y Ubuntu Linux, con la finalidad de brindar una herramienta para los estudiantes de la Universidad Nacional Tecnológica del Cono Sur.
- Se determinó los mejores dispositivos electrónicos para las etapas de HMI, adquisición de datos y etapa de potencia de la fresadora CNC, según los alcances que se tienen localmente.
- Se configuró la interface de un software libre en Ubuntu Linux con la máquina fresadora CNC por medio del tablero de interface, y se verificó el correcto funcionamiento de la máquina fresadora CNC con el programa HMI en gambas V3.
- Se determinó con los resultados que la fresadora CNC puede fabricar tarjetas electrónicas PCB de buena calidad, reduciendo el tiempo y disminuyendo los costos.
- Se determinó los rangos de trabajo, velocidades de mecanizado, y herramientas para la máquina fresadora CNC, considerando que puede utilizarse para otros fines aparte de la creación de placas PCB.
- La fresadora CNC de 3 ejes de libertad para fabricación de tarjetas electrónicas brinda una solución y puede ser utilizada por los estudiantes de la Universidad Nacional Tecnológica del Lima Sur.

RECOMENDACIONES

- Diseñar correctamente el circuito eléctrico que se desee fabricar tanto en un software de simulación y de fabricación de placas de circuito impreso.
- Calibrar la máquina fresadora CNC conforme a las necesidades del usuario y a los motores y componentes que se hayan seleccionado para la fabricación de la misma.
- Seguir el procedimiento de fabricación de las placas electrónicas de acuerdo a lo redactado en la etapa de pruebas de esta Tesis.
- Considerar el rango máximo de trabajo de las placas de circuito impreso que se puedan realizar en el modelo de fresadora presentado por un total máximo de 10cm x 10cm.
- Al momento de fabricar las placas de circuito impreso se sugiere que se hagan los movimientos en líneas horizontales y verticales con el fin de mejorar la precisión al momento de realizar las placas PCB.
- Considerar el tipo de herramienta de fresado, la velocidad de trabajo de la fresadora y la velocidad de traslado de la herramienta de la máquina para poder generar el circuito impreso tal cual como se requiere.
- Tener sistemas de seguridad para el tablero de control tanto como fusibles de protección como interruptores para apagar el tablero en cualquier caso de emergencia.
- Mantener lubricadas las guías de los ejes de la fresadora además de las varillas roscadas para disminuir la fricción y mantener la precisión de la máquina fresadora.

BIBLIOGRAFÍA

- ARDUINO, (2014). *El Microcontrolador Arduino*, recuperado de <http://arduino.cc/en/Guide/HomePage>.
- BARRIENTOS, A., (2007). *Fundamentos de Robótica*, España, McGraw-Hill, 2da Edición.
- BURGANO, G., (2011). *Mesa robótica XY para la perforación de placas PCB con conexión USB*, Tesis de Ingeniería, Universidad de San Francisco de Quito, Quito, Ecuador.
- CORONA, A., (2010). *Manual de prácticas para el maquinado de piezas metálicas del ajedrez en el laboratorio LPAIC en CNC*, Tesis de ingeniería publicada, Instituto Politécnico Nacional, México D.F., México.
- HERRERA, D., (2012). *Diseño y construcción de una máquina para la elaboración de circuitos impresos basado en control numérico computarizado*, Tesis de ingeniería publicada, Escuela Politécnica Nacional, Quito, Ecuador.
- LEY CUEN, C., (2007). *Teoría de Linux*, recuperado de <http://leycuen.blogspot.com/2007/02/como-iniciar-en-linux.html>.
- MUÑOZ, M., (2010). *Prototipo electromecánico para dibujar circuitos impresos sobre una placa de cobre*, Tesis de ingeniería publicada, Universidad Tecnológica de la Mixteca, Oaxaca, México.
- SALAS, R., (2010). *Técnicas de diseño, desarrollo y montaje de circuitos impresos*, Universidad de los Andes, Venezuela.
- UBUNTU, (2014). *Sobre Ubuntu*, recuperado de http://www.ubuntu-es.org/sobre_ubuntu#.U1hmmKh5O8F.
- VALERIO, J., (2007). *Diseño e implementación de una plataforma de desarrollo, aplicada a un prototipo de máquina fresadora CNC*, Tesis de ingeniería publicada, Universidad de Chile, Santiago, Chile.

ANEXOS

ANEXO 1: Imagen del proyecto completo



ANEXO 2: Datasheet de microcontrolador de Arduino Uno

Features

- High Performance, Low Power AVR[®] 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4K/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48PA/88PA/168PA/328P)
 - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
 - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
 - Active Mode: 0.2 mA
 - Power-down Mode: 0.1 µA
 - Power-save Mode: 0.75 µA (Including 32 kHz RTC)



8-bit AVR[®]
Microcontroller
with 4/8/16/32K
Bytes In-System
Programmable
Flash

ATmega48PA
ATmega88PA
ATmega168PA
ATmega328P

Rev. 8161D-AVR-10/09

ANEXO 3: Datasheet de chip L298 de etapa de potencia



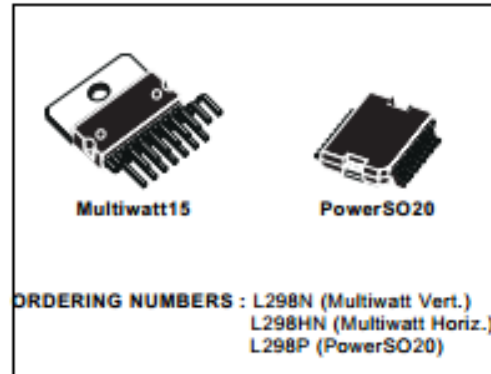
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

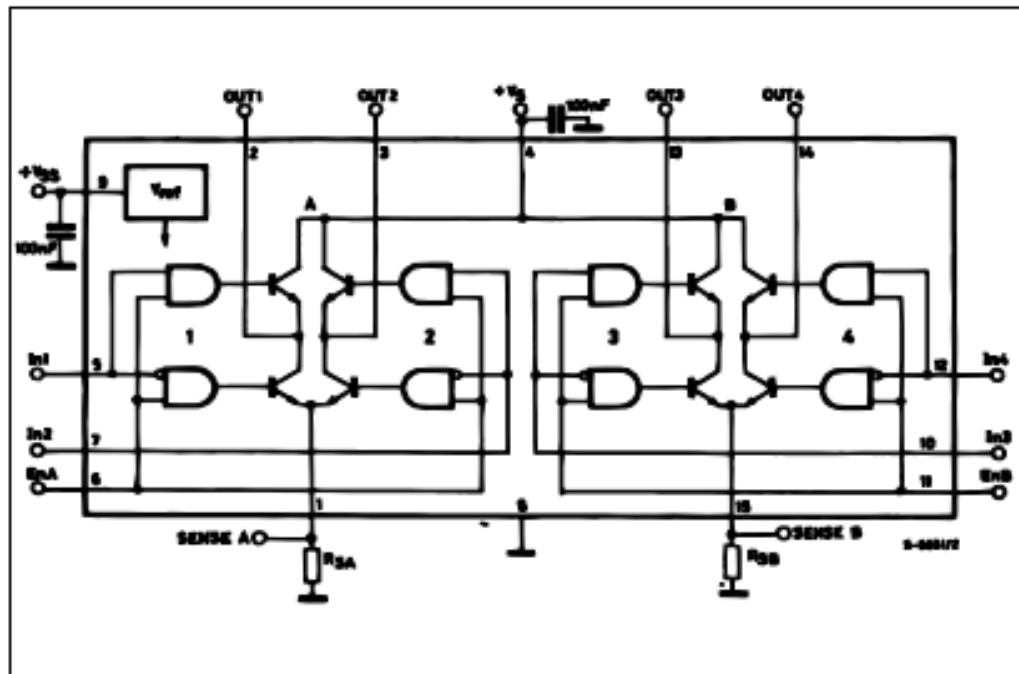
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM



January 2000

1/13

ANEXO 4: Descripción de la herramienta de mecanizado para placa electrónica



106 Engraving Cutter

★★★★★
[Write a review](#)

Engraving cutter. Quality cutters for detail engraving, carving and routing in wood, fiberglass, plastic and soft metals. Two cutters included.

Share this: [f](#) [t](#) [p](#) [+](#)

[Email a Friend »](#)
[Printer Friendly »](#)



| Features | Specifications | Reviews | Product Support |
|----------|----------------|---------|-----------------|
|----------|----------------|---------|-----------------|

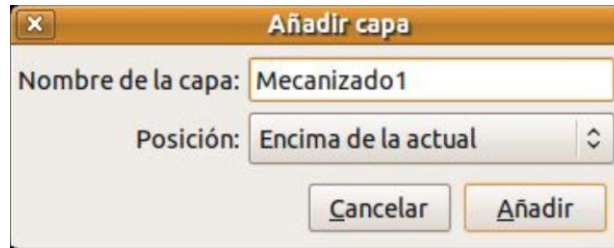
106 Engraving Cutter

- For detail engraving, carving and routing
- For wood, fiberglass, plastic and soft materials
- Not for use with hardened materials
- Use higher speeds for hard woods, slower speeds for metals
- 1/8" shank

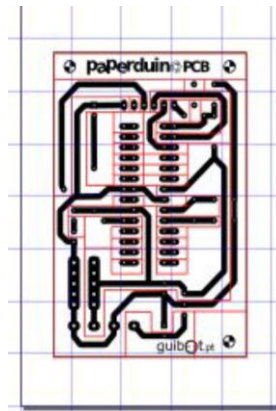
Accessories

- 106 Engraving Cutter (2)

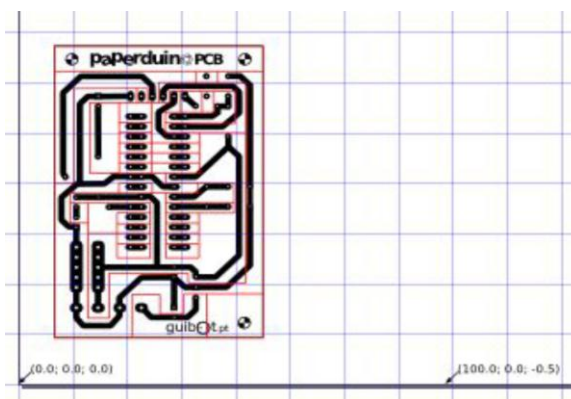
- Se crea una capa nueva con Capa/Añadir Capa ó (Mayús+Ctrl+N) y se le asigna un nombre.



- En esta nueva capa, con la herramienta de Bezier (curva) se dibuja las líneas de separación de pistas (se dibuja con líneas rectas por sencillez)



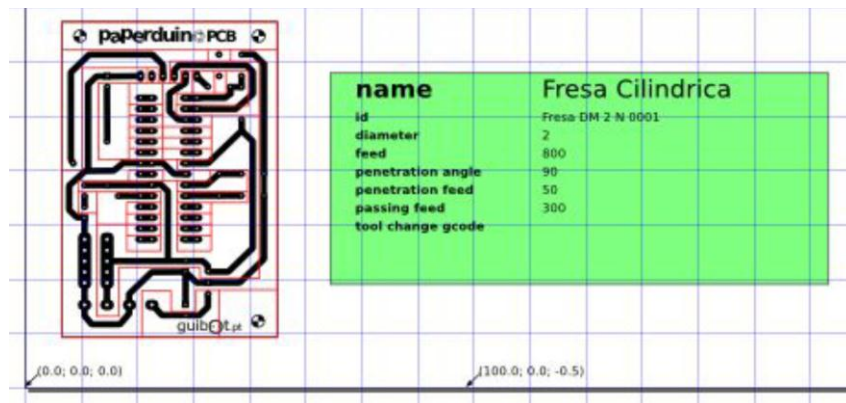
- Se selecciona Extensiones/Gcodetools/Puntos de Orientación.



En el cuadro de diálogo se define a que altura está la cara superior de la pieza (normalmente a 0mm), y cuanto se quiere que profundice el

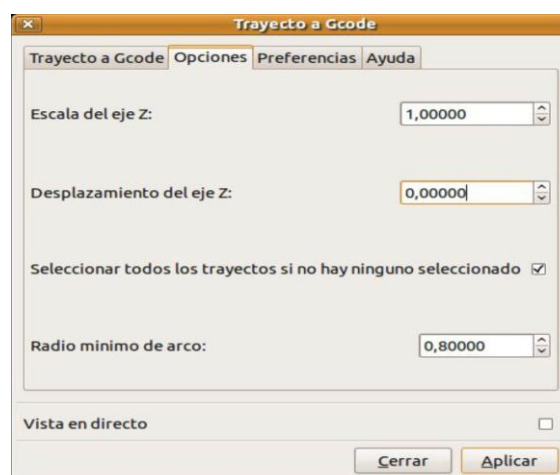
mecanizado (entre -0,5 y -1 mm), y Gcodetools definirá los puntos de control.

7. Se selecciona Extensiones /Gcodetools/Biblioteca de Herramientas. Se selecciona la forma de nuestra fresa, y al apretar el botón de aplicar, creará un cuadro con los valores por defecto de la herramienta.

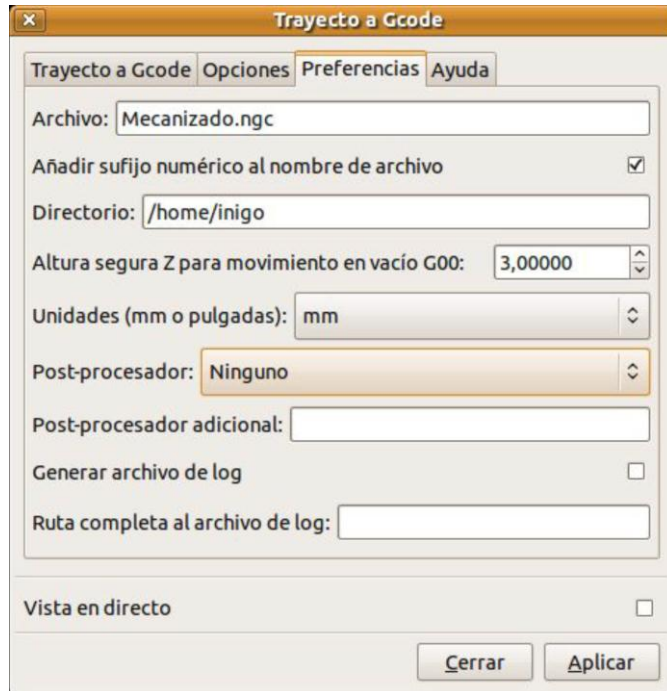


Con la herramienta de texto se modifican los campos de Diámetro de fresa, Velocidad en vacío, Ángulo de penetración, Avance de acuerdo a la fresa que utilizemos, el material a mecanizar, es decir, se adecua a las necesidades requeridas.

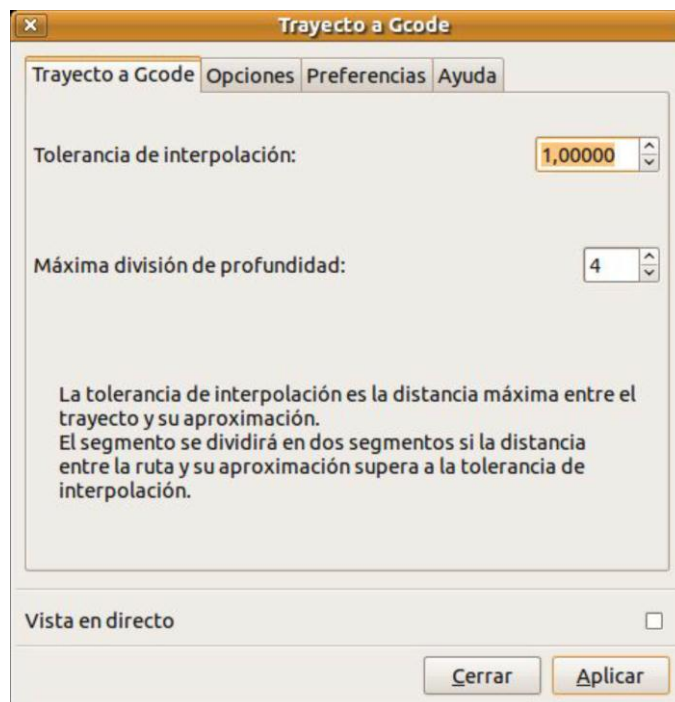
8. Se selecciona los trayectos que se quiere mecanizar y se apreta en el menú Extensiones/Gcodetools/Trayecto a Gcode



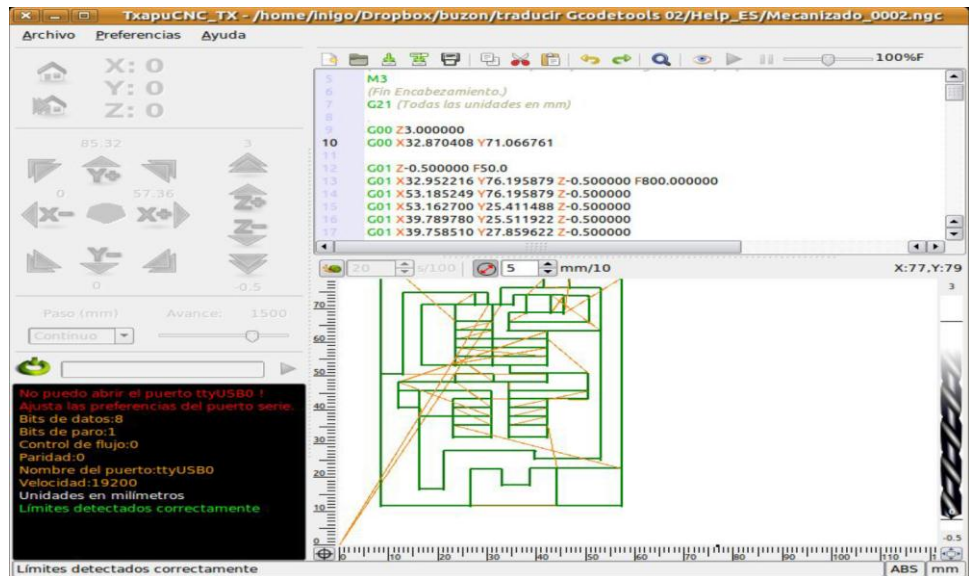
9. Se comprueba que los valores de la lengüeta Opciones son correctos.



10. Se comprueba también que los valores de la lengüeta Preferencias son correctos.



11. Se va a la lengüeta Trayecto a Gcode. Se comprueba también los valores. (En la imagen, El valor de la tolerancia de interpolación es “un poco grande”).



Al apretar el botón Aplicar, se observa que representa en el archivo inkscape la ruta de mecanizado indicando el sentido de movimiento en cada trayecto.

Pero lo más importante es que ha generado en el directorio que se ha indicado el archivo con el Gcode listo para mecanizar.

ANEXO 6: Programa UNTECS CNC Recepcion RX en Gambas V3

UNTECS CNC RX

```
// Arduino G-code Interpreter
#include <HardwareSerial.h>
//our command string
#define COMMAND_SIZE 128
char palabra[COMMAND_SIZE];
byte serial_count;
int no_data = 0;
void setup()
{
    //Do startup stuff here
    Serial.begin(19200);
    Serial.println("start");
    //other initialization.
    init_process_string();
    init_steppers();
}
void loop()
{
    char c;
    //keep it hot!
    //read in characters if we got them.
    if (Serial.available() > 0)
    {
        c = Serial.read();
        no_data = 0;

        //newlines are ends of commands.
        if (c != '\n')
        {
            palabra[serial_count] = c;
            serial_count++;
        }
    }
    //mark no data.
    else
    {
        no_data++;
        delayMicroseconds(100);
    }
}
```

```

//if theres a pause or we got a real command, do it
if (serial_count && (c == '\n' || no_data > 100))
{
    //process our command!
    process_string(palabra, serial_count);
    //clear command.
    init_process_string();
}
//no data? turn off steppers
if (no_data > 1000)
    disable_steppers();
}

```

INIT

```

// define the parameters of our machine.
#define X_STEPS_PER_INCH 4064 //(160x24,5)
#define X_STEPS_PER_MM 160 //(M8=>Paso:1,25mm;
360/1,25=>1mm=288°; 288°por mm/1,8°por paso=160 Pasos por mm

#define Y_STEPS_PER_INCH 4064
#define Y_STEPS_PER_MM 160

#define Z_STEPS_PER_INCH 4064
#define Z_STEPS_PER_MM 160

//our maximum feedrates
#define FAST_XY_FEEDRATE 2400
#define FAST_Z_FEEDRATE 2400

// Units in curve section
#define CURVE_SECTION_INCHES 0.019685
#define CURVE_SECTION_MM 0.5

// Set to one if sensor outputs inverting (ie: 1 means open, 0 means closed)
// RepRap opto endstops are *not* inverting.
#define SENSORS_INVERTING 1

// How many temperature samples to take. each sample takes about 100
usecs.

/*****

```

```

* digital i/o pin assignment
*
* this uses the undocumented feature of Arduino - pins 14-19 correspond to
analog 0-5
*****/

//cartesian bot pins
#define X_STEP_PIN 12
#define X_DIR_PIN 13
#define X_MIN_PIN 4
#define X_MAX_PIN 5
#define X_ENABLE_PIN 18

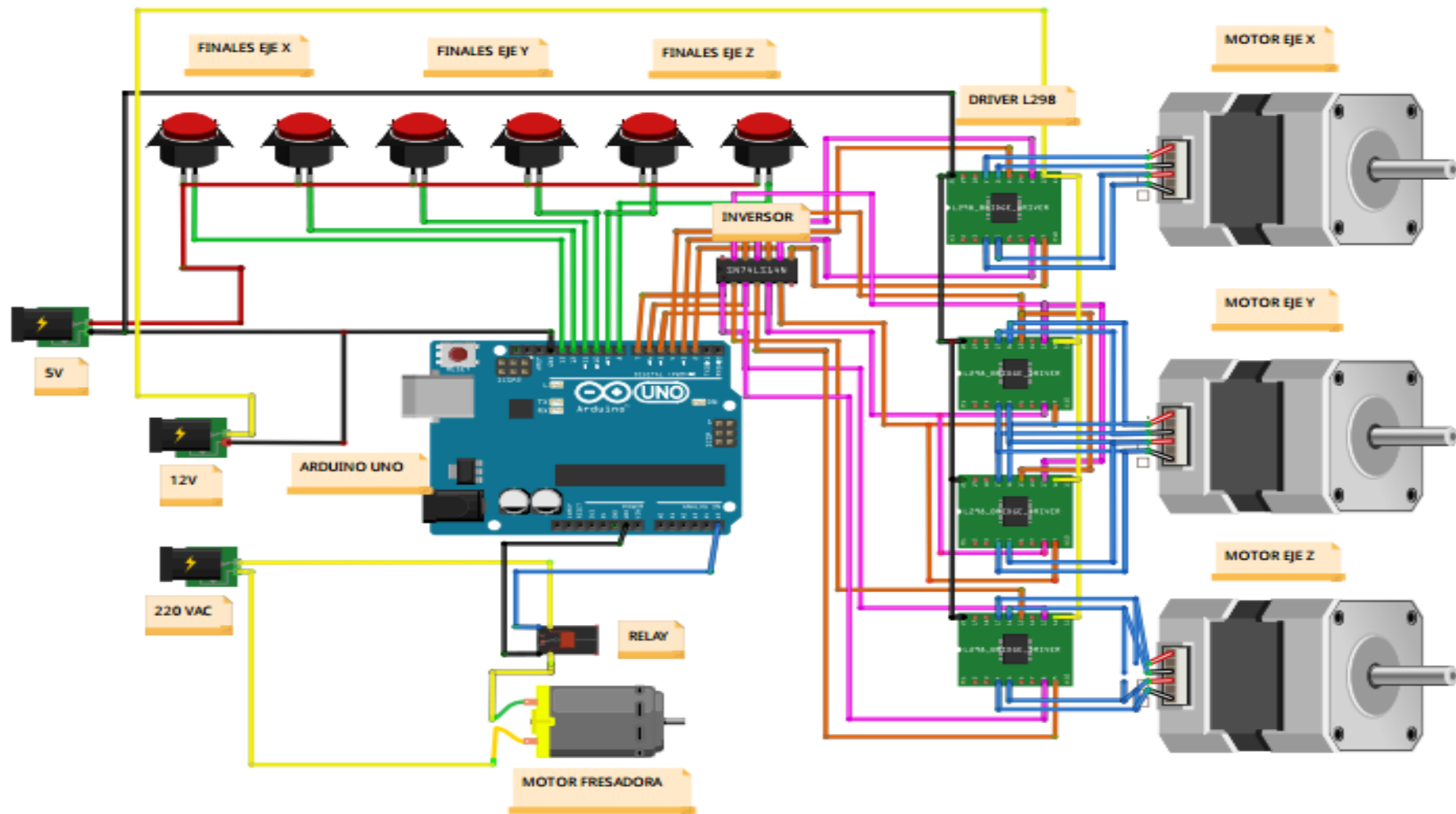
#define Y_STEP_PIN 10
#define Y_DIR_PIN 11
#define Y_MIN_PIN 7
#define Y_MAX_PIN 6
#define Y_ENABLE_PIN 17

#define Z_STEP_PIN 8
#define Z_DIR_PIN 9
#define Z_MIN_PIN 3
#define Z_MAX_PIN 2
#define Z_ENABLE_PIN 16

#define MOTOR_PIN 19 //Encendido/apagado cabezal (Analog 5)

```

ANEXO 7: CIRCUITO ELÉCTRICO DE LA FRESADORA



fritzing

ANEXO 8: PLANOS DE LA FRESADORA CNC

ANEXO 9: PRESUPUESTO

| | Descripción | Cantidad | Costo Parcial (S/.) | Costo Total (S/.) |
|--------------|---------------------|----------------|---------------------|-------------------|
| Materiales | Motores Paso a Paso | 3 | 8 | 24 |
| | Fresadora | 1 | 100 | 100 |
| | Espárragos M8 | 1 | 1.5 | 1.5 |
| | Varilla DM8 | 3 | 10 | 30 |
| | Molde Impresión A0 | 1 | 5 | 5 |
| | Pernería en General | 1 | 10 | 10 |
| | Tarjeta Arduino Uno | 1 | 40 | 40 |
| | Driver L298 | 4 | 15 | 60 |
| | Cooler | 3 | 5 | 15 |
| | CD40106 | 1 | 2 | 2 |
| | Relé | 1 | 2 | 2 |
| | Finales de Carrera | 6 | 1.5 | 9 |
| | Fuente ATX | 1 | 20 | 20 |
| | Borneras | 50 | 0.1 | 5 |
| | Cableado | 1 | 10 | 10 |
| | Servicios | Servicio torno | 1 | 50 |
| Carpintería | | 1 | 50 | 50 |
| Mano de Obra | | 1 | 300 | 300 |
| Otros | | 1 | 200 | 200 |
| | | | <i>Total (S/.)</i> | 933.50 |