



UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:

INGENIERO/A EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

**TEMA: DISEÑO DE UNA FRESADORA CNC PARA PCBs MEDIANTE
ARCHIVOS GERBER Y FUENTE EXCELLON**

AUTOR: CARLOS ROBERTO PASQUEL CASTILLO

TUTOR: Ing. RENE ERNESTO CORTIJO LEYVA, Mg

QUITO- ECUADOR

AÑO: 2019

UNIVERSIDAD TECNOLÓGICA ISRAEL

DECLARACIÓN

Yo, Carlos Roberto Pasquel Castillo, con cedula de identidad 1717432734, en calidad de estudiante de la Carrera de Electrónica y Telecomunicaciones, declaro que los contenidos de este trabajo de titulación, requisito previo a la obtención del Grado de Ingeniera en Electrónica y Telecomunicaciones, son absolutamente originales, auténticos y de exclusiva responsabilidad legal y académica del autor.

Quito, enero del 2019

Carlos Roberto Pasquel Castillo
CC: 171743273-4

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de tutor del componente practico certifico:

Que el trabajo de titulación “**DISEÑO DE UNA FRESADORA CNC PARA PCBs MEDIANTE ARCHIVOS GERBER Y FUENTE EXCELLON**”, presentado por el Sr. Carlos Roberto Pasquel Castillo, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M. Enero del 2019

TUTOR

.....

Ing. Rene Ernesto Cortijo Leyva, Mg

DEDICATORIA

Luego de varios años de trabajo, sacrificio y esfuerzo han dado como resultado alcanzar una meta profesional y se lo dedico completamente a Dios, a mi amada esposa María Antonieta Cano y mi madre María Piedad Pasquel, por su apoyo incondicional dentro del transcurso de la carrera.

Gracias a todos los que han formado parte en esta etapa de mi vida.

AGRADECIMIENTO

Agradezco a Dios por darme la vida y la salud y permitirme culminar mis estudios universitarios, a mi amada esposa María Antonieta Cano por estar a mi lado en los momentos buenos y malos y darme su apoyo incondicional en todo momento. A mi madre María Piedad Pasquel por su sacrificio de haberme apoyado siempre y ser un ejemplo de fortaleza ante las adversidades de la vida, a mi hermana Brenda y su padre don Luis Tipan por sus consejos, a mi amigo el Ing. Jorge López por darme la oportunidad de lograr esta meta, a mi amigo Cristian Masabanda por su apoyo técnico y consejos dentro del desarrollo del proyecto, a mis compañeros de aula y profesores con los que compartí momentos únicos. Gracias por estar presentes desinteresadamente en el trayecto de este logro.

RESUMEN

El proyecto trata del diseño y construcción de una fresadora CNC para PCBs, la cual a través de un microcontrolador Arduino UNO controla la velocidad y posición de los motores a pasos que accionan los ejes de la fresadora, y con el arranque de viruta después del proceso de fresado y taladrado se obtiene los circuitos PCBs utilizados hoy en día en diferentes campos como la industria, educación, investigación entre otros.

Se implementó un programa para el control PWM con el *software App Inventor 2* que controla vía Bluetooth la velocidad del *Spindle* con la interacción de un microcontrolador arduino NANO el cual a su vez gestiona alarmas de los finales de carrera de seguridad de límites de trabajo de la fresadora CNC que bloquean el movimiento, esto es con el fin de evitar daños en las piezas de la máquina y materiales.

La implementación es gestionada a través de un Smartphone con sistema operativo Android con la que se activa y desactiva las etapas de control principal de la fresadora teniendo un control remoto de la máquina.

Dentro de las ventajas significativas esta la precisión y el corto tiempo de elaboración en una producción en serie, y una desventaja es el alto costo al adquirir un equipo de similares características con lo cual se obtuvo un ahorro del 40 % al construir la fresadora localmente.

La implementación tiene cuatro puntos principales los cuales se determinaron dentro del diseño previo a la construcción: mecánica, eléctrica, electrónica y *software* para el control.

Palabras clave: *Spindle*, Bluetooth, fresadora, PCBs, CNC, *App Inventor 2*.

ABSTRACT

The project deals with the design and construction of a CNC milling machine for PCBs, the quality through an Arduino UNO microcontroller controls the speed and position of the motors, the steps of the milling machine shafts and the chip removal after the Milling and drilling process is obtained. Circuits PCBs are used today in different fields such as industry, education, research among others.

A program for the PWM control was implemented with App Inventor 2 *software* that controls the speed of the *spindle* via Bluetooth with the interaction of an NANO Arduino microcontroller, which in turn manages alarms for the safety limits of the work limits. of the CNC milling machine that blocks the movement, this is in order to avoid damage to the parts of the machine and the materials.

The implementation is managed through a smartphone with Android operating system that activates and deactivates the main control stages of the milling machine by having a remote control of the machine.

Among the advantages, you get the precision and short time of the production of a series production, and a disadvantage is the cost of acquiring a similar equipment with which you get a saving of 40% to build the router locally.

The implementation has four main points which were determined within the design prior to construction: mechanical, electrical, electronic and *software* for control.

Keywords: Spindle, Bluetooth, Milling machine, PCBs, CNC, App Inventor 2.

TABLA DE CONTENIDO

DECLARACIÓN.....	ii
APROBACIÓN DEL TUTOR	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
RESUMEN	vi
ABSTRACT	vii
TABLA DE CONTENIDO	viii
INTRODUCCIÓN.....	1
ANTECEDENTES DE LA SITUACIÓN Y OBJETO DE ESTUDIO	1
PLANTEAMIENTO DEL PROBLEMA	4
JUSTIFICACIÓN	5
OBJETIVOS	6
Objetivo general.....	6
Objetivos específicos	6
ALCANCE.....	6
DESCRIPCIÓN DE CAPITULOS	6
CAPÍTULO I.....	8
FUNDAMENTACIÓN TEÓRICA	8
1.1 CONTROL NUMÉRICO POR COMPUTADORA	8
1.1.1 Archivo Gerber	8
1.1.2 Archivo Excellon	9
1.1.3 Código G.....	9
1.1.4 Código M	10
1.1.5 CAD	11
1.1.6 CAM	11

1.1.7	CAI.....	11
1.1.8	CAT.....	11
1.2	SISTEMA DE COORDENADAS.....	12
1.2.1	Sistema de Coordenadas Absolutas.	12
1.2.2	Sistema de Coordenadas Incrementales.....	13
1.3	ARDUINO UNO	13
1.4	ARDUINO CNC <i>SHIELD</i>	14
1.5	CONTROLADORES A4988	14
1.6	MOTOR PASO A PASO.	15
1.6.1	Descripción de los motores de paso.....	15
1.6.2	Motores paso a paso Bipolares.	16
1.7	MOTOR DE HUSILLO (<i>SPINDLE</i>).....	16
1.8	BLUETOOTH ELECTRONICS	17
1.8.2	Puerto serie.....	18
1.9	SOFTWARE.....	19
1.9.2	Arduino Versión 1.8.4.....	19
1.9.3	App Inventor 2	19
1.9.4	Universal Gcode Sender Versión 1.0.8.....	20
1.9.5	FlatCAM Versión 8.5.....	20
1.9.6	Proteus Versión 8.6.....	21
CAPÍTULO II.....		22
2.1	MARCO METODOLOGICO	22
2.1.1	Métodos aplicados	22
2.1.2	Fases del proyecto.....	23
CAPÍTULO III		24
3.1	PROPUESTA	24

3.2	ARDUINO UNO.....	26
3.2.1	Programa Arduino UNO.....	27
3.3	ARDUINO NANO.....	28
3.4	MOTORES PASO A PASO.....	29
3.5	MOTOR DC <i>SPINDLE</i>	30
3.6	TERJETA CONTROL MOTOR DC <i>SPINDLE</i>	31
3.7	CONTROLADOR A4988.....	32
3.8	Comunicación Bluetooth.....	33
3.9	FUENTES DE PODER.....	33
3.10	DESARROLLO DE SOFTWARE.....	34
3.10.1	MIT app inventor 2.....	34
3.11	DESBASTE Y PERFORACIÓN.....	35
3.12	FRESAS Y BROCAS.....	36
3.13	ANÁLISIS DE FACTIBILIDAD.....	38
3.13.1	Factibilidad Técnica.....	38
3.13.2	Factibilidad Económica.....	39
3.13.3	Factibilidad Operativa.....	39
3.14	ANÁLISIS DE COSTOS DEL PROYECTO.....	39
3.15	VENTAJAS DE LA FRESADORA CNC PARA PCB.....	42
CAPÍTULO IV.....		43
4.1	IMPLEMENTACION.....	43
4.2	DESARROLLO.....	43
4.2.1	Diseño y construcción de la fresadora CNC.....	43
4.2.2	Diseño y construcción de la tarjeta de control.....	45
4.2.3	Desarrollo del Software.....	47
	MIT App inventor 2.....	47

Programa Arduino NANO	54
4.2.4 Desarrollo de cálculos en la implementación	60
Controlador A4988	60
Motores paso a paso.....	60
Relación de velocidad del <i>Spindle</i> y el avance de la mesa	62
Calibración de parámetros iniciales de la fresadora CNC a través de Universal Gcode Sender.....	64
Parámetros principales a calibrar en FlatCAM.....	68
4.2.5 PRUEBAS DE FUNCIONAMIENTO.....	69
Operación local.....	69
Operación remota.....	70
4.2.6 ANÁLISIS DE RESULTADOS.....	71
CONCLUSIONES.....	74
RECOMENDACIONES	75
BIBLIOGRAFÍA.....	76
ANEXOS	78

LISTA DE FIGURAS

Figura 1: Evolución de los archivos de trazados.....	8
Figura 2: Representación de un sistema de coordenadas absolutas.	12
Figura 3: Representación de un sistema de coordenadas incrementales:.....	13
Figura 4: Arduino UNO.	14
Figura 5: Placa de control CNC <i>Shield</i> compatible con Arduino.	14
Figura 6: Diagrama eléctrico del controlador A4988.	15
Figura 7: Partes de un motor bipolar a pasos.	16
Figura 8: Partes internas del motor DC.....	17
Figura 9: Bluetooth diagrama y representación eléctrica.....	18
Figura 10: Diagrama de un puerto de comunicación serie.....	19
Figura 11: Conversión Serial a USB.	19
Figura 12: Diagrama que ilustra la relación entre archivos y objetos.....	20
Figura 13: Diagrama de bloques de la fresadora CNC.	24
Figura 14: Proceso de diseño, control y elaboración de placas PCBs	25
Figura 15: Partes principales de un Arduino UNO.	26
Figura 16: Aceleración suave del firmware GRBL.....	27
Figura 17: Partes principales de un Arduino Nano.	29
Figura 18: Motor a pasos Nema 17.	30
Figura 19: Dimensiones motor DC <i>Spindle</i>	31
Figura 20: Tarjeta de control motor DC <i>Spindle</i>	31
Figura 21: Diagrama de temporización de la interfaz lógica.	32
Figura 22: Fuentes de poder.....	34
Figura 23: Dimensiones fresa cónica de 30°.....	37
Figura 24: Dimensiones broca helicoidal.....	37
Figura 25: Partes de la fresadora CNC.....	44
Figura 26: Diseño estructura fresadora en AutoCAD.	45
Figura 27: Acople de motores a paso con las bridas de sujeción y el tornillo.	45
Figura 28: Diseño tarjeta electrónica control.	46
Figura 29: Tarjeta control seguridades, control PWM, activación remota.	47
Figura 30: Tarjeta enlace señales micros	47
Figura 31: Distribución de la ventana principal de la App.	48

Figura 32: Distribución de la ventana de control y alarmas.....	49
Figura 33: Distribución de la ventana de control principal.....	49
Figura 34: Distribución de la ventana de información.....	50
Figura 35: Programación de la navegación entre las ventanas.	51
Figura 36: Programación en bloques del regreso entre las ventanas.	52
Figura 37: Programación en bloques de la conexión del Bluetooth con la App.	52
Figura 38: Programación en bloques del envío de la velocidad del <i>Spindle</i> desde la App.....	53
Figura 39: Programación en bloques de la lectura de los avisos de alarmas	53
Figura 40: Programación en bloques de los estados de los relés de control.	54
Figura 41: Código para la creación de un Serial entre el arduino y el Bluetooth.	55
Figura 42: Código para la lectura de datos del Serial entre arduino y Bluetooth.	55
Figura 43: Diagrama de flujo lectura de datos.	55
Figura 44: Diagrama de flujo velocidad del <i>Spindle</i>	56
Figura 45: Lectura del valor enviado por el serial y asignación de velocidad o estado.	56
Figura 46: Control del estado de los relés dependiendo del valor enviado desde la App.	57
Figura 47: Diagrama de flujo de estado de los relés.	58
Figura 48: Diagrama de flujo de escritura de datos.	59
Figura 49: Código para la escritura de datos en el serial del Arduino y el Bluetooth.	59
Figura 50: Diseño pistas de prueba en Proteus.	69
Figura 51: PCB obtenida de la fresadora CNC.	70
Figura 52: Dimensiones de pistas obtenidas.	71
Figura 53: Dimensiones entre placa e integrado.	71
Figura 54: Visualización de alarmas en Smartphone	71

LISTA DE TABLAS

Tabla 1:Lista de Códigos G más utilizados en los programas CNC.	10
Tabla 2: Principales códigos M usados en el Código CNC.	10
Tabla 3: Librerías del firmware GRLB utilizado para la CNC.	28
Tabla 4: Secuencia control motores paso a paso bipolares.	30
Tabla 5: Datos diagrama temporización de la interfaz lógica.	33
Tabla 6: Separación mínima entre pistas para diferentes voltajes.	35
Tabla 7: Ancho de pista para PCB.	36
Tabla 8: Especificaciones técnicas del proyecto.	38
Tabla 9: Tabla de costos de los componentes mecánicos CNC.	40
Tabla 10: Tabla de costos de componentes electrónicos CNC.	40
Tabla 11: Costos de componentes de importación de CNC.	41
Tabla 12: Costos de implementación y desarrollo CNC.	41
Tabla 13: Tabla de costo total del proyecto.	41
Tabla 14: Análisis Costo Construcción local Vs. Importación.	42
Tabla 15: Función de cada uno de los estados de los relés.	57
Tabla 16: Ecuaciones para el cálculo del torque requerido por los motores a pasos.	60
Tabla 17: Datos motores a paso.	61
Tabla 18: Resultados del torque del motor a pasos.	61
Tabla 19: Datos de los motores a pasos.	63
Tabla 20: Valores referenciales de velocidad.	64
Tabla 21: Parámetros para calibración de los movimientos de ejes.	67
Tabla 22: Mascara para configuración finales de carrera e inversión de giro de motores.	68

LISTA DE ECUACIONES

Ecuación 1: Calculo de intensidad de corriente del controlador A4988.....	60
Ecuación 2: Calculo de voltaje de referencia del controlador A4988.....	60
Ecuación 3: Calculo de torque motores a pasos.....	61
Ecuación 4: Potencia de corte.....	62
Ecuación 5: Avance de mesa.....	62
Ecuación 6: Velocidad del Spindle.....	63
Ecuación 7: Cálculo número de pasos Universal Gcode Sender.....	65

INTRODUCCIÓN

Las máquinas de control numérico computarizado CNC nacen a partir de los años cincuenta, como solución a la industrialización y producción en serie, su uso era reducido y con el avance tecnológico y computacional cada vez es más común encontrar máquinas CNC tanto en pequeñas como grandes industrias. Así mismo el campo de aplicación se ha incrementado, por lo que actualmente se fabrican fresadoras, rectificadoras, tornos, etc. Estas máquinas son controladas a través de una computadora la cual mantiene un control estricto de velocidad y posición de los motores que accionan los ejes de la máquina.

El control numérico por computadora es un método que se enfatiza en controlar movimientos de las partes de una máquina en determinada función, el cual a través de instrucciones codificadas transmite información a un controlador que gobierna componentes electro-mecánicos los cuales responden a la orden dada por el controlador.

El presente trabajo de investigación se enfoca en el diseño e implementación de una fresadora CNC para Placas de Circuito (PCB), mediante archivo Gerber y fuente Excellon para lo cual se estudian métodos de implementación, para obtener rigidez en la máquina y precisión en el acabado de la (PCB) optimizando tiempo y recursos, logrando obtener una máquina a un bajo costo, pero en las mismas o mejores características que una importada.

ANTECEDENTES DE LA SITUACIÓN Y OBJETO DE ESTUDIO

Según Bavaresco (2001), “Los antecedentes de la investigación hacen referencia al conocimiento previamente elaborado de la variable o el sector donde se desarrolla la investigación, que pueda proporcionar algún soporte a la investigación”.

Los procesos utilizados para la elaboración de PCBs son muy diversos entre los más comunes se tiene un método conocido como transferencia térmica, y reducción del cobre con ácido clorhídrico, el cual consiste después de haber obtenido el circuito se debe imprimirlo en una impresora láser en papel fotográfico o papel Press-n-Peel, se corta la baquelita de la medida necesaria y con la ayuda de una plancha para la transferencia, el proceso de reducción

del cobre es realizado con ácido y agua caliente, por lo que conlleva tiempo en el proceso de la preparación del archivo tanto en *software* como en *hardware*, y algo más crítico y riesgoso a la salud de la o las personas que ejecutan este proceso al trabajar con ácido, pues existe el riesgo de generarse afecciones nasales, daños en la piel, complicaciones o pérdida de la vista por contacto con este químico y en varias ocasiones pérdida de recursos por existir fallos en este proceso. Cabe destacar que otros métodos solo varían la forma como transferir el circuito impreso a la placa de cobre los cuales siguen usando ácido.

Existen soluciones temporales dentro del ámbito de la salud hacia las personas que realizan este proceso las cuales se manejan con equipos de protección, guantes de látex, mascarillas, gafas, etc., pero el riesgo es latente, muchas veces presenta fallos en la elaboración se da como pérdida total de la PCB que se esté realizando y a esto no hay otra solución más que volver a empezar teniendo pérdida de material, tiempo y dinero.

Por otro lado, tenemos un factor muy importante, que en el país no se puede obtener de manera común una fresadora CNC de PCBs y los equipos se los consigue en pocos lugares y a costos elevados razones que han motivado la elaboración de la fresadora CNC con equipos, dispositivos y elementos comunes en nuestro medio, tratando de abaratar costos ya que una fresadora CNC para PCBs según lo investigado hasta la fecha bordea los 1500 dólares importándola.

Cabe destacar que la automatización de métodos ha venido evolucionando desde hace mucho tiempo con lo cual se ha implementado mejoras en varios procesos, la elaboración de placas electrónicas ha evolucionado en países industrializados de tal manera que ya no se realizan manualmente, sino que lo hacen utilizando máquinas CNC para evitar errores humanos, afecciones por uso de ácido entre otros. Una máquina CNC debe operar bajo estricto control de calidad por lo que es indispensable que el personal técnico mantenga a punto todos los ajustes para evitar lotes de producción con fallas.

El control numérico (CN) surgió a partir de la revolución industrial de 1760-1840, con la incorporación de maquinaria cuyo funcionamiento se hace de manera autónoma y continua.

Fue desarrollado por (Parsons, 1948), (un contratista de maquinado para la Fuerza Aérea de los Estados Unidos) (USAF) y el Instituto Tecnológico de Massachusetts (MIT). Así mismo la primera máquina fresadora de corte numérico, construida con bulbos, usando un código binario y cinta perforada.

No obstante, a partir de los sesentas se añadieron computadoras para aumentar la flexibilidad del proceso, y se les denominó Control Numérico Computarizado (CNC), como es conocido hoy en día. Alrededor de 1972 vino el verdadero apogeo del control numérico computarizado (CNC) una década después de la introducción de microcomputadoras asequibles lo cual ha generado un cambio total en la obtención de productos realizados en serie como son máquinas de metales, plásticos, madera entre otros las cuales se obtienen con un alto grado de precisión y optimización de recursos.

Al momento casi todos los productos que son utilizados a diario sea en el área, de la informática, la electrónica, la manufactura, la construcción, el transporte, las comunicaciones, entre otras dependen de las máquinas CNC para su fabricación, obteniendo automóviles, aviones, televisores, artículos para el hogar, aparatos y muchos otros productos en los cuales basamos nuestra vida diaria.

En el Ecuador la universidad de las fuerzas armadas ESPE Latacunga en el año 2005 desarrolló un tema por (Acuña y Sánchez, 2005) denominado diseño y construcción de un prototipo de una máquina taladradora de control numérico computarizado.

De igual forma en el año 2018 la universidad de las fuerzas armadas ESPE Quito ha desarrollado un proyecto por (Cadena y Zamora; 2018) con el tema de investigación diseño y construcción de un equipo router CNC para fabricación y perforación de circuitos PCB, en los cuales se centran en diseño mecánico dejando de lado seguridades y conexiones remotas por fuera.

Tomando en cuenta que el control numérico computarizado continúa en vías de desarrollo, en el presente proyecto, se implementó seguridades de alarmas remotas las cuales se puede visualizar dónde está el fallo por límites excedidos y poder solucionar de manera rápida, adicional se gestiona el encendido y apagado remoto de las tarjetas CNC *Shield* y la

tarjeta del *Spindle* y se desarrolla un control PWM remoto del *Spindle*, todo este control vía Bluetooth a través de un Smartphone.

PLANTEAMIENTO DEL PROBLEMA

La elaboración de PCBs ha tenido grandes avances tecnológicos con los cuales se ha logrado optimizar recursos y estandarizar el proceso, con la globalización se ha generado un avance en la ciencia y tecnología logrando obtener máquinas totalmente automáticas y autónomas eliminando la intervención humana por dispositivos mecánicos y electrónicos. El país ha venido sufriendo ciertos cambios en la matriz productiva en lo cual se ha enfatizado a la optimización de recursos destinados a generar transferencia tecnológica con esto se incentiva a la competitividad y productividad y contribuir así al desarrollo de la ciencia y tecnología. Pero aún existe limitaciones dentro del desarrollo tecnológico con lo cual nos vemos condicionados a competir de manera óptima.

Uno de los métodos utilizados para la elaboración de PCBs es a través de transferencia térmica, y reducción del cobre con ácido clorhídrico que conlleva tiempo, riesgos a la salud y en varias ocasiones pérdida de recursos por existir fallos humanos en este proceso, son considerados ineficientes y contaminantes. Dentro del avance tecnológico, específicamente en el área electrónica está la automatización de procesos, dentro del cual se presenta la implementación de una fresadora CNC para PCBs con la que se va a reducir tiempos en la elaboración de tarjetas electrónicas la optimización de los recursos, facilitando la producción a escalas mayores.

En el Ecuador ha venido incrementándose la tecnología CNC, dando cabida a importar máquinas CNC a elevados costos desde los 1500 dólares en adelante sin contar los trámites aduaneros y de la misma importación, generando así la necesidad de investigación para poder implementar y diseñar con lo que tenemos en nuestro medio, palpando la complejidad de adquirir varios de los repuestos a nivel local e importar los que no existen en el país.

El problema se presenta cuando dentro del proceso de elaboración de PCBs se realiza manualmente y no se logra estandarizar y controlar variables en la elaboración de placas electrónicas implicando pérdidas materiales, económicas y de tiempo, y algo más crítico la

salud de las personas que elaboran el trabajo al tener que manipular agentes químicos que desencadena en enfermedades a corto y largo plazo, contaminación al medio ambiente, teniendo en cuenta un factor primordial que es el querer adquirir una fresadora CNC importándola resulta mucho más costoso que ensamblar una CNC con dispositivos de compra local y diseñando la o las tarjetas necesarias que controlaran los motores X, Y, Z, limitándonos a importar solo lo necesario con esto se abarata costos y se obtiene un equipo funcional.

JUSTIFICACIÓN

Al adquirir un método de control automático, el cual gobierne las máquinas de un determinado proceso se obtiene mayor calidad en los productos, y se consigue la eliminación de los errores propios del ser humano, ocasionando grandes ahorros de tiempo y material al eliminarse la producción de piezas defectuosas.

Por otra parte, el trabajo incluye el diseño y construcción de una fresadora CNC para PCBs, con el cual se optimizará tiempo, recursos y se aplicará en la elaboración de placas electrónicas de un determinado tamaño en una producción en serie.

Por lo antes expuesto, una de las problemáticas con estos equipos es que son muy costosos, es por tal motivo que se plantea el presente proyecto, con el fin de obtener una máquina reduciendo costos de fabricación de al menos un 35 %; con materiales altamente resistentes.

Con el aporte de métodos, como el experimental, inductivo, deductivo y un profundo análisis bibliográfico, se implementará el *software* y *hardware* necesario para la comunicación entre la computadora y los drivers para el control de la fresadora CNC.

Así también, este proyecto se justifica al contribuir en el cuidado del medio ambiente, erradicando el uso de productos químicos en la elaboración de PCBs artesanalmente, los cuales son letales para nuestro planeta.

OBJETIVOS

Objetivo general

- Desarrollar una fresadora CNC para PCBs a partir de archivos Gerber y fuente Excellon

Objetivos específicos

- Diseñar el circuito electrónico y acoplar al controlador, con el mecanismo para el control automático de la fresadora CNC.
- Construir la parte estructural y mecánica de la fresadora CNC.
- Implementar los controladores necesarios para el manejo de los motores de la fresadora CNC.
- Definir la interfaz y *software* necesario para el control de la fresadora CNC.
- Crear los algoritmos necesarios para los archivos Gerber y fuente Excellon.
- Realizar pruebas en el equipo y corrección de fallos generados.

ALCANCE

La fresadora CNC para PCBs trabajara con un diseño previo de los circuitos en un *software* CAD (Proteus), las dimensiones máximas de la baquelita serán de 170 mm x 140 mm, la interfaz de usuario deberá siempre estar activa (Universal Gcode Sender) así como la interfaz Bluetooth con la cual se activará el sistema, receptará alarmas de fallo por límites de seguridad y controlara a través de la entrada PWM de la tarjeta del *Spindle* su velocidad, se contará con los planos del diseño de la fresadora y se entregara el manual de usuario.

DESCRIPCIÓN DE CAPITULOS

La investigación se estructura de Introducción, está un preámbulo del tema principal en donde se plantea el alcance, justificación y antecedentes del proyecto.

Capítulo I Fundamentación Teórica, donde se expone los componentes utilizados en la implementación del proyecto con sus características principales.

Capítulo II Marco metodológico, se explica los diferentes métodos de investigación para el desarrollo de la fresadora CNC.

Capítulo III Propuesta, se explica los diferentes equipos y materiales utilizados para la elaboración de la fresadora los programas implementados y los diferentes costos y tiempo utilizados para la ejecución.

Capítulo IV Implementación, aplicando los métodos y técnicas existentes de diseño y construcción se arma, programa, y se realiza pruebas manuales y automáticas de la máquina terminada.

Conclusiones y Recomendaciones, Referencias bibliográficas, Anexos.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

1.1 CONTROL NUMÉRICO POR COMPUTADORA

El control numérico por computadora o CNC, es un método de controlar en todo momento la posición de un elemento electromecánico móvil. Esto se obtiene con *software* específico y ordenes elaboradas de forma definida previamente, las órdenes dadas a la máquina son implantadas mediante códigos numéricos las cuales siguen una secuencia lógica, que se denomina programa para mecanizado, este ejecuta un trazado de las pistas y perforación en la placa de baquelita a través del desbaste de material.

1.1.1 ARCHIVO GERBER

Es utilizado en la producción de circuitos a través del maquinado, creado por (Gerber Systems Corporation) hoy en día conocida como (Ucamco) (Tabernier, 2015, p. 58), desde el año 1970 creó diversas impresoras de fotograbado de control numérico, se formalizó en 1980 por (Electronic Industries Association) y recibió el archivo de trazado EIA RS-274-D, dentro de diversos cambios hoy manejamos el archivo de trazado RS-274 X, los cuales son archivos de código ASCII que manejan coordenadas e instrucciones que interpreta el circuito a fabricar.

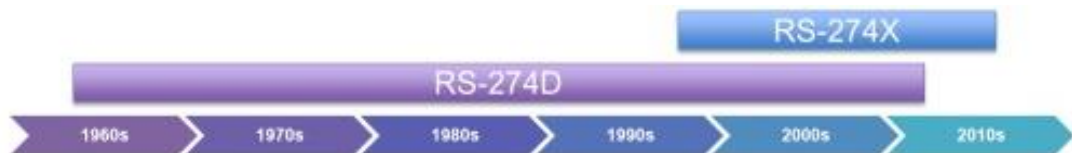


Figura 1: Evolución de los archivos de trazados.

Fuente: (“Alternatives to Gerber RS-274-X”, s/f)

Al elaborar un circuito se maneja diversas formas como círculos, rectángulos, etc., y con el archivo de trazado RS-274-D se manejaba diversos archivos para cada forma, con la

implementación del archivo de trazado RS-274X se unifica toda la definición de diámetros, formas y coordenadas necesarias para la elaboración de los circuitos.

Los archivos Gerber en si se refieren a la trayectoria de una herramienta de corte que es controlada por diversas directivas y códigos los cuales responden a coordenadas en los ejes X, Y, Z, que son controlados según el circuito a realizarse.

1.1.2 ARCHIVO EXCELLON

La estructura de un fichero Excellon según, (Tabernier, 2015, p. 58), es similar a un Gerber, en esta se encuentra información necesaria para la perforación de agujeros con diámetros y posiciones específicas del circuito en la placa de baquelita. La función principal es controlar el sistema de taladro de la fresadora CNC. El nombre Excellon proviene de la compañía “Excellon Automation Company” (www.excellon.com). En la década de los 80’s, esta empresa copaba el 90 % del mercado en la venta de máquinas de taladrado de PCBs.

El archivo de trazado que define el formato Excellon para PCB es (IPC-NC-349), el IPC Association Connecting Electronics Industries, es una asociación de carácter comercial cuyo objetivo principal es el de estandarizar los requisitos para el ensamblaje y producción de equipos electrónicos la cual es acreditada por el American National Standards Institute (ANSI) como una organización de desarrollo de estándares, la cual es conocida a nivel mundial por la Implementación de estándares.

1.1.3 CÓDIGO G

De acuerdo a lo expuesto por (Hood y Floyd, 2009, p. 206), se ha generado una lista de funciones de los movimientos de la máquina entre ellos movimientos rápidos, avances, avances radiales, pausas y ciclos. Estos contienen variables (direcciones) definidas por el programador para cada función específica, maneja un lenguaje de programación vectorial básicamente se escriben segmentos de recta y arcos de circunferencia la programación se efectúa mediante un lenguaje de bajo nivel, considerando que la norma (ISO 6983-1: 2009) es el estándar con el cual se rigen las máquinas CNC y por ende el código G.

Tabla 1: Lista de Códigos G más utilizados en los programas CNC.**Fuente: Elaborado por el autor.**

Código	Función
G00	Avance lineal a alta velocidad para posicionamiento.
G01	Avance lineal a velocidad programada para realizar el corte.
G02	Avance circular en sentido de las manecillas del reloj, a velocidad programada
G03	Avance circular en sentido opuesto a las manecillas del reloj, a velocidad programa.
G28	HOME: Regreso al cero de máquina.
G90	Uso de coordenada absolutas.
G91	Uso de coordenadas relativas.
G92	Programación del cero de pieza.

1.1.4 CÓDIGO M

De acuerdo a Williams (2009. P. 221), son funciones misceláneas las cuales son requeridas para ayudar a maquinar una placa de baquelita, estos códigos no interfieren con el movimiento de la máquina, paro del programa, cambio de herramienta, refrigerante, sistema de aspiración, etc.

Tabla 2: Principales códigos M usados en el Código CNC.**Fuente: Elaborado por el autor.**

Código	Función
M00	Paro del programa
M02	Indica el fin del programa
M03	Indica el giro del <i>Spindle</i> en sentido de las manecillas del reloj.
M04	Indica el giro del <i>Spindle</i> en sentido contrario de las manecillas del reloj.

1.1.5 CAD

CAD (*Computer Aided Design*), el diseño asistido por computadora utiliza *software* informático específico como herramienta de soporte para el diseño y documentación del mismo y generar un proceso de fabricación de un producto específico, según (Hood y Floyd, 2009), “This is specialized software that allows you to design two- and three-dimensional objects for the CNC machine to cut, drill, and perform other actions on”.

1.1.6 CAM

CAM (*Computer Aided Manufacturing*), Hood y Floyd (2009) manifiestan que “*CAM software takes the design you created with the CAD software and converts it into a “language” called G-Code. This G-Code is then used by the final type of software, Control. Control software is the actual application that talks to your CNC machine; it takes the G-Code from the CAM software and uses it to send the proper electrical signals*”. Partiendo de un modelo CAD se genera un programa de control numérico con el cual podemos manejar simulaciones de trayectorias para el mecanizado, producciones en serie, con lo cual se convierte los diseños en piezas físicas.

1.1.7 CAI

CAI (*Computer Aided Inspection*), es la inspección asistida por computadora la cual a través de un *software* específico se observa en tiempo real la ejecución del proceso.

1.1.8 CAT

CAT (*Computer Aided Testing*), ensayo asistido por computadora con el cual se realiza pruebas en *software* del proceso de maquinado sin correr riesgos de daños en maquinaria y material o peligros al operador y gestionar de manera eficaz los cambios necesarios antes de realizar el trabajo en campo.

1.2 SISTEMA DE COORDENADAS

Los sistemas de coordenadas son en pulgadas o métrico, los cuales pueden ser absolutos o incrementales y son los planteados en el plano cartesiano. El cual contiene ejes de coordenadas o ejes de trabajo, en este caso formado por tres rectas perpendiculares entre sí (X, Y, Z) que se intersecan en el origen (0,0,0). Los comandos en X e Y se usan para el posicionamiento de la bancada o mesa de trabajo y el comando Z para el posicionamiento de la herramienta o *Spindle*.

1.2.1 Sistema de Coordenadas Absolutas.

El punto cero es el punto inicial de trabajo o punto inicial de referencia en cada diseño, lo que quiere decir que cada vez que se quiera mover el *Spindle* o la bancada partiremos desde al punto cero absolutos y los colocaremos en el punto deseado, pero siempre partiendo del punto cero iniciales, dicho esto el código para que la CNC nos entienda para las coordenadas absolutas es el G90.

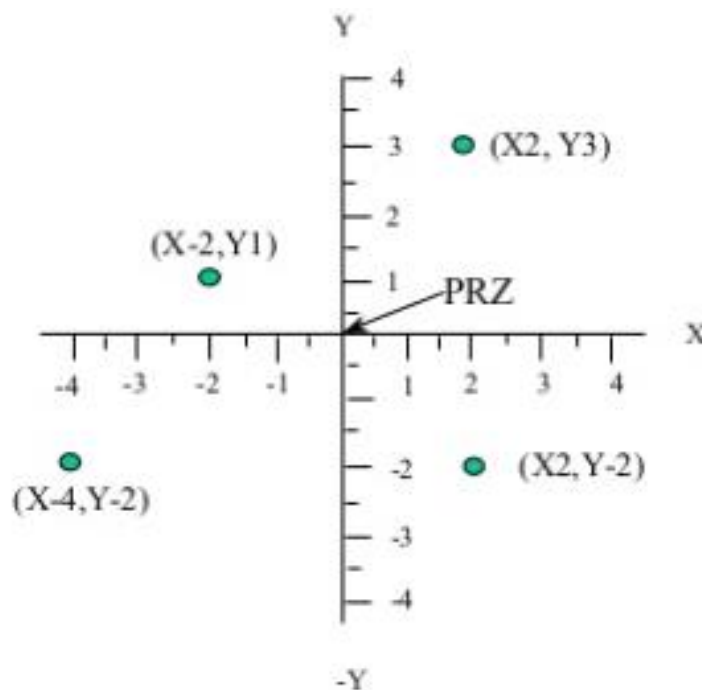


Figura 2: Representación de un sistema de coordenadas absolutas.

Fuente: (José Mecánico, 09:52:25 UTC)

1.2.2 Sistema de Coordenadas Incrementales.

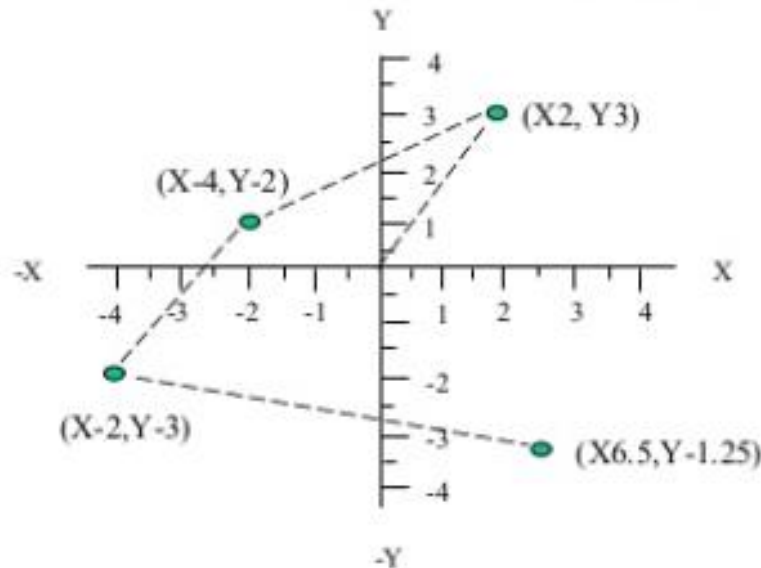


Figura 3: Representación de un sistema de coordenadas incrementales:

Fuente: (Jose Mecanico, 09:52:25 UTC)

Se programa la distancia que se debe desplazar el *Spindle* o la bancada respecto al último punto conseguido, es decir el punto cero se va modificando de acuerdo al avance del *Spindle* denominándose una trayectoria en cadena la cual depende de cada punto anterior al movimiento realizado. El código en la CNC para las coordenadas incrementales es el G91.

1.3 ARDUINO UNO

Es una tarjeta de desarrollo para la elaboración de proyectos de electrónica, robótica, automatización, etc., esta contiene un microcontrolador ATmega 328 y trabaja con *software* y *hardware* libre, por su bajo costo y su entorno de programación facilita la elaboración de proyectos. Es capaz de ejecutar las órdenes almacenadas en su memoria la cual contiene bloques funcionales que cumplen una tarea específica, en su interior maneja tres unidades principales que son unidad central de procesamiento, memoria y periféricos de entradas y salidas (Arduino.cc, 2019).

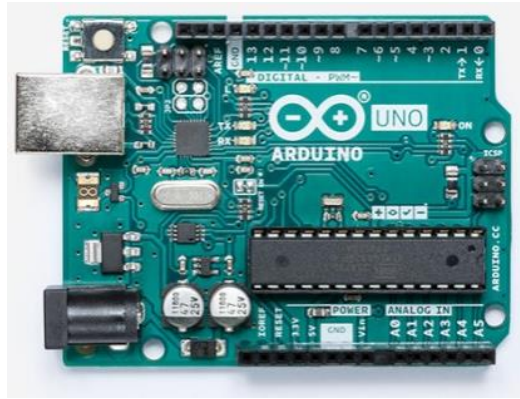


Figura 4: Arduino UNO.

Fuente: (“Arduino UNO Rev3”, s/f)

1.4 ARDUINO CNC SHIELD

Es una placa que permite conectar hasta 4 controladores (drivers) con sus respectivos motores paso a paso, y la interconexión entre el Arduino UNO y varios componentes externos como interruptores de final de carrera, salidas a relé, y varios tipos de sensores, reduciendo así una gran cantidad de cables y conexiones.

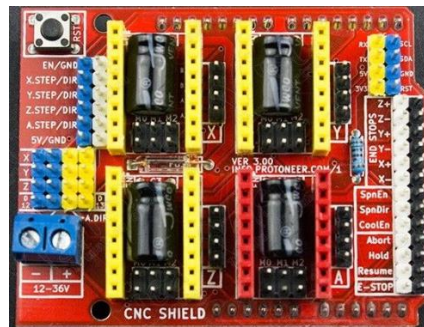


Figura 5: Placa de control CNC Shield compatible con Arduino.

Fuente: (“Arduino CNC Shield V3 - VISTRONICA SAS”, s/f)

1.5 CONTROLADORES A4988

Es un controlador (driver) que minimiza el manejo del motor paso a paso desde el Arduino UNO. Permite controlar altos voltajes e intensidades que requieren los motores a pasos y limitar la corriente que circula por el motor. La intensidad máxima de trabajo es de 2 A y el rango de voltaje de trabajo, 8 a 35 V, el diseño de este controlador es para motores de paso bipolares de paso completo, medio, cuarto, octavo y modos de dieciséis pasos.

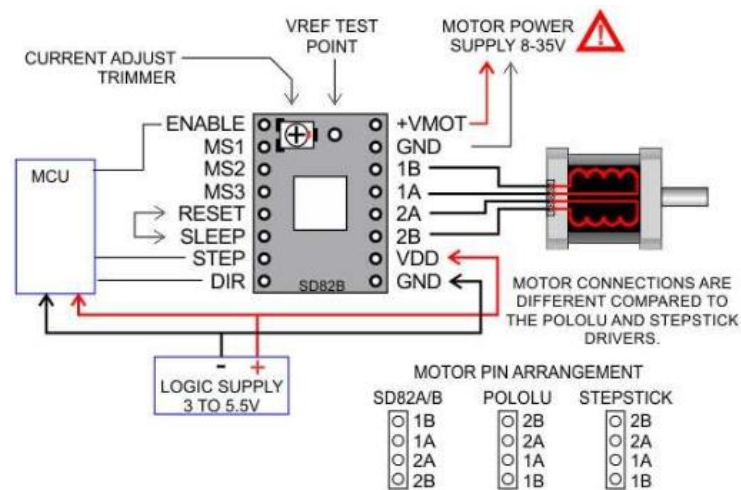


Figura 6: Diagrama eléctrico del controlador A4988.

Fuente: (“SureStepr SD82B Stepper driver”, s/f)

1.6 MOTOR PASO A PASO.

Es un dispositivo electromecánico que convierte una serie de pulsos eléctricos en desplazamientos angulares, dando como resultado un movimiento de un paso por cada pulso digital aplicado. Las bobinas del motor están en el estator y pueden ser de imán permanente o reluctancia variable. A medida que se activa los bobinados del motor en un orden predefinido permite que fluya una corriente a través de ellos con lo cual el estator se magnetiza y se generan polos electromagnéticos los cuales dan como resultado la propulsión del motor.

1.6.1 Descripción de los motores de paso

Los motores paso a paso pueden posicionarse con precisión sin ningún sensor de retroalimentación y se lo suele presentar como un controlador de circuito abierto con lo cual no necesita un control de lazo cerrado, una de sus características principales es el poder rotar en ambas direcciones con lo cual se obtiene un mayor desempeño en sus diferentes aplicaciones, tener movimientos angulares precisos y mantener un torque de retención a velocidad cero.

1.6.2 Motores paso a paso Bipolares.

Estos motores tienen cuatro cables de salida de sus bobinas el cual necesita un drive para poder cambiar el sentido de giro y así la dirección del flujo de corriente, con lo que se induce un campo magnético en las bobinas y por ende un polo magnético norte y sur, y generar el movimiento del rotor, en la Figura 7 se observa las partes principales de un motor a pasos.

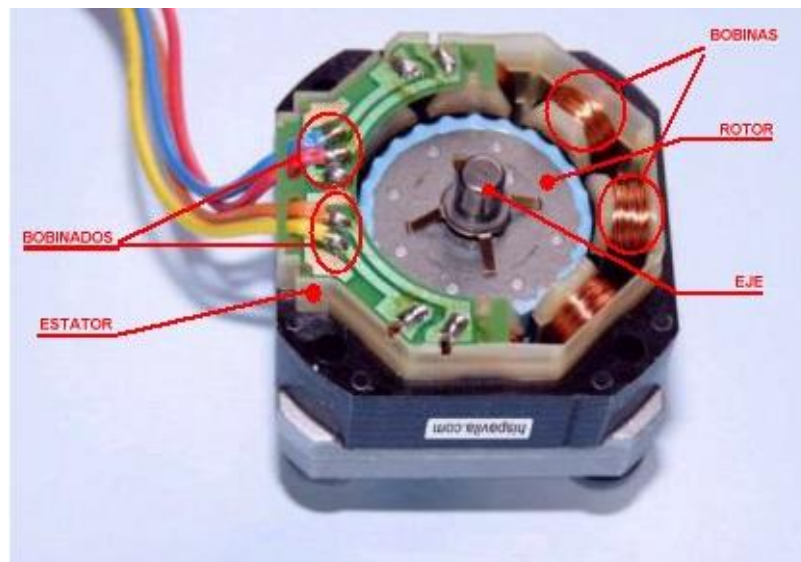


Figura 7: Partes de un motor bipolar a pasos.

Fuente: (García, s/f)

1.7 MOTOR DE HUSILLO (SPINDLE)

Es un motor DC sin escobillas (*Brushless*), de acuerdo con el criterio del *datasheet de microchip* (AN885. *Brushless DC Motor Fundamentals*, Microchip Technology Inc, 2013), manifiesta que al no tener un elemento que provoque rozamiento entre el rotor y el estator del motor, evita los chispazos entre los carbones y el colector de los motores DC convencionales, con lo cual se logra mantener una comunicación entre dispositivos electrónicos de manera limpia y sin interferencias, una característica de este motor en especial es que incluye el sistema de ventilación del motor con el mismo sistema del eje central del rotor, la conmutación es realizada electrónicamente a través de un drive el cual controla el motor, el rango de velocidades es muy amplio de 0 a 12000 rpm, y el voltaje de

trabajo es de 12 a 48 VDC, con sus respectivas normalización (12V-3000 rpm, 24V-6000 rpm, 36V-9000 rpm, 48V-12000 rpm).

Al no tener retención magnética su giro es suave con lo cual se facilita el control de velocidad, tiene una relación lineal entre voltaje/velocidad, carga/velocidad y carga corriente, en la Figura 8 se observa las partes principales de un motor DC *Brushless*.

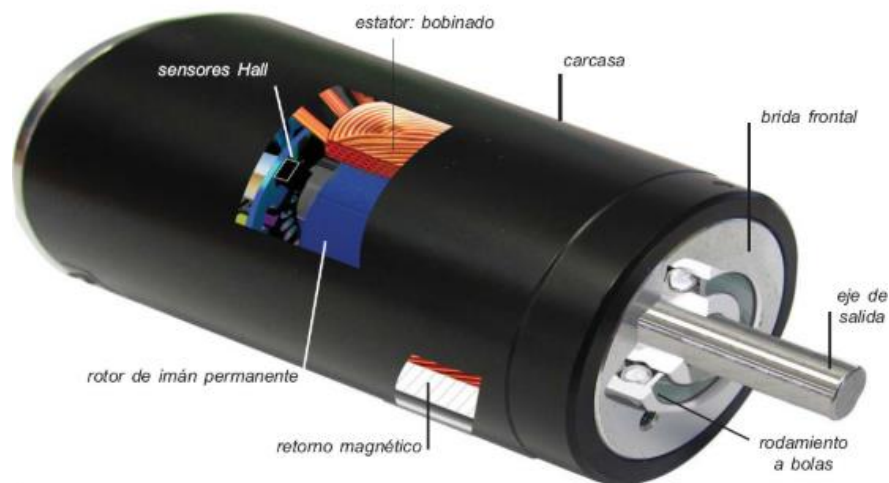


Figura 8: Partes internas del motor DC.

Fuente: <http://ecoelektric.com.ar/>

1.8 BLUETOOTH ELECTRONICS

El HC-05 B Bluetooth es un módulo con el cual se genera una comunicación entre el Arduino NANO y una aplicación de Android en este caso generada con *App Inventor*, este maneja puertos de TX y RX y un rango de distancia de hasta 10 m, la programación se la realiza por comandos AT la cual es enviada por un puerto serial, el voltaje de trabajo es de 3,3 voltios, puede ser usado como maestro/esclavo.

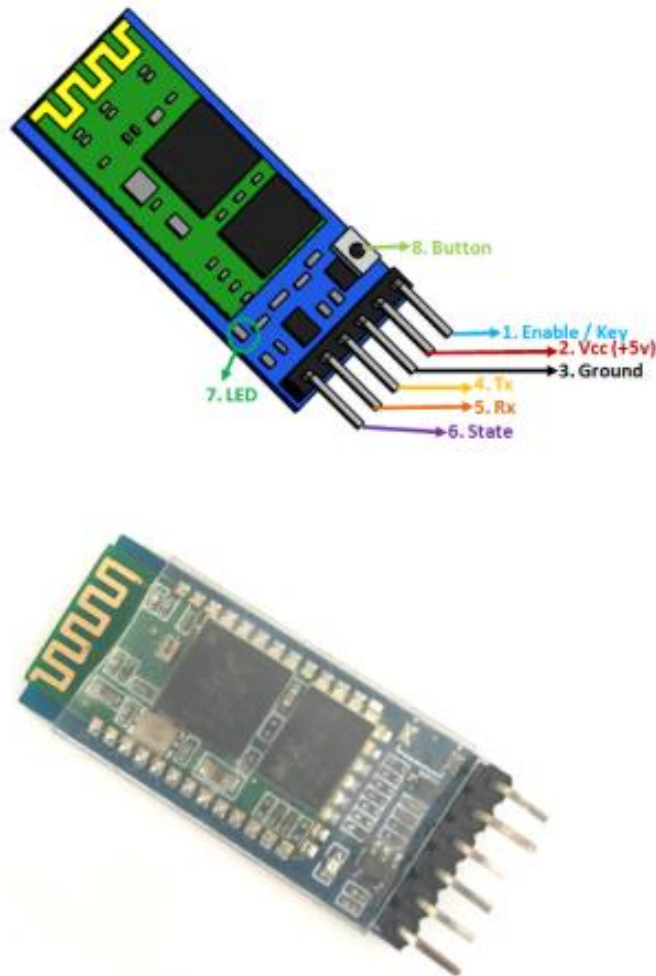


Figura 9: Bluetooth diagrama y representación eléctrica.

Fuente: (“HC-05 Bluetooth Module Pinout, Specifications, Default Settings, Replacements & Datasheet”, s/f)

1.8.2 Puerto serie

Es usado para una interfaz física o virtual en la comunicación de dos dispositivos u ordenadores, la información es enviada en secuencia de bits a través de un par de cables uno para TX y otro para RX, también se los denomina como UART (*Universal Asynchronous Receiver/Transmitter*), Transmisor-Receptor Asíncrono Universal, con lo cual se controla puertos y dispositivos serie logrando conectar, comunicar y proveer de alimentación eléctrica entre periféricos y dispositivos electrónicos, el estándar (EIA 232, 1969) *Electronics Industry Association*, es quien especifica las conexiones entre un sistema y un modem.

COMUNICACIÓN SERIE

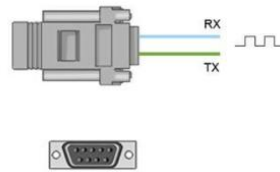


Figura 10: Diagrama de un puerto de comunicación serie.

Fuente: (“Comunicación de Arduino con puerto serie”, s/f)

Arduino UNO utiliza un convertidor serial hacia el USB del ordenador con lo cual se establece la comunicación Mediante el USB del ordenador y el Arduino UNO.

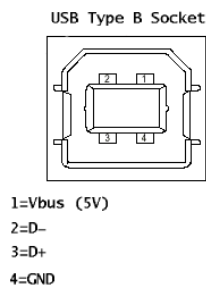


Figura 11: Conversión Serial a USB.

Fuente: ("aprendiendoarduino.wordpress.com/2016/11/09/uart-y-usb-en-arduino/")

1.9 SOFTWARE

1.9.2 Arduino Versión 1.8.4

Es un *software* de código abierto IDE (entrono de desarrollo integrado), de fácil acceso y escritura del programa, amigable al usuario, el entorno de programación está escrito en Java y está basado en *Processing*,

1.9.3 App Inventor 2

Es un medio para desarrollar *software* el cual fue creado por (Google Labs, 2010), en el cual se genera aplicaciones para el sistema operativo Android, el método de programación

es visual a partir de una serie de bloques se crea la aplicación, es un *software* libre y está basado en JavaScript.

1.9.4 Universal Gcode Sender Versión 1.0.8

Esta desarrollado en Java (*Sun Microsystems*, 1995), “Escribir una vez, ejecutar en cualquier lugar”, es un lenguaje de programación de computadora orientado a objetos y al ser un programa multiplataforma trabaja en cualquier entorno Windows, Mac, Linux, está basado en el *firmware* GRBL el cual es compatible con código G, este controla hasta 3 motores, es dedicado para aplicaciones de control de máquinas CNC, la comunicación entre el Arduino y Universal Gcode Sender es mediante la norma RS-232 y la transmisión se realiza mediante un cable USB,

1.9.5 FlatCAM Versión 8.5

Este *software* fue creado por (Caram, 2014-2016), en el cual se prepara los trabajos de la CNC para la fabricación de las placas electrónicas, toma los archivos Gerber o Excellon y se crea el Gcode, es de código abierto y su lenguaje de programación es Python.

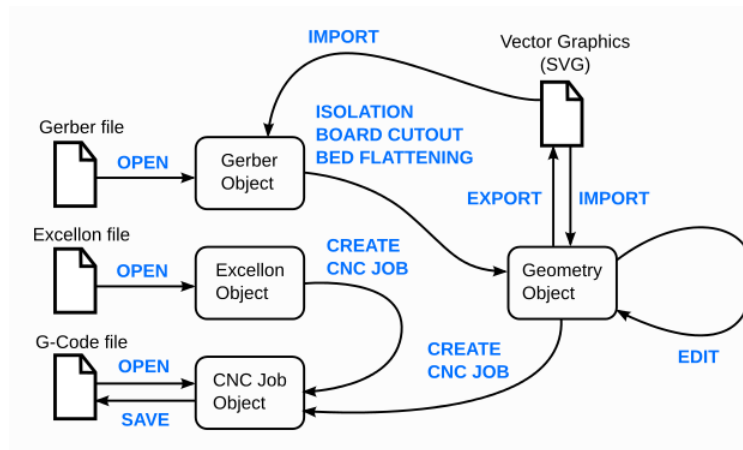


Figura 12: Diagrama que ilustra la relación entre archivos y objetos.

Fuente: <http://flatcam.org/manual/basics.html>

Los datos en FlatCAM se encuentran de 4 diferentes tipos de objetos: Gerber, Excellon, *Geometry* y CNC Job. Los objetos Gerber, Excellon y CNC Job se crean directamente leyendo archivos en formatos Gerber, Excellon y G-Code. Los objetos de geometría son un paso intermedio disponible para manipular datos. Se pueden crear

mediante diferentes operaciones, así como directamente importados desde archivos SVG. En la Figura 12 se ilustra la relación entre archivos y objetos. Las flechas que conectan los objetos representan un subconjunto de tareas que se pueden realizar en FlatCAM.

1.9.6 Proteus Versión 8.6

Es un programa en el cual se diseña un esquema electrónico, se programa el *software* del programa, se diseña la placa de un circuito impreso, se realiza simulaciones de todo lo implementado, compilación y detección de errores generación de archivos Gerber y Excellon, todo esto con el fin de mantener un control de documentación y su posterior construcción.

CAPÍTULO II

2.1 MARCO METODOLOGICO

En este capítulo se plantea los aspectos metodológicos y prácticos con los cuales se realizó el diseño, implementación y puesta en marcha de la fresadora CNC para PCBs, de igual forma se presenta el tipo de investigación previo al desarrollo del proyecto, por otra parte Sabino (2007, p.87), manifiesta que “tiene como objeto proporcionar un modelo de verificación adecuado para contrarrestar hechos con teorías y su forma es la de una estrategia o plan general tendente a determinar las operaciones necesarias para realizarlo”.

2.1.1 Métodos aplicados

De acuerdo a (Arias, 1997, p. 47), señala que la investigación documental “es aquella que se basa en la obtención y análisis de datos provenientes de materiales impresos u otros tipos de documentos”, para el desarrollo del proyecto se debe realizar un análisis documental con lo cual se determinó la viabilidad del proyecto, se investigó los diversos modelos de máquinas CNC existentes, tomando como referencia estructural para el diseño a (Williams, 2003), planteando así de forma clara el desarrollo de la fresadora.

Por otro lado, (Palella y Martins, 2006, p. 88), manifiestan que el método inductivo “Aluden a aquellos procedimientos que van de lo simple a lo compuesto, es decir, de las partes al todo.”, debido a que se deberá acoplar módulos de potencia, electrónica y comunicaciones para la interfaz entre la computadora y el módulo de los motores a controlar con su respectivo diseño de seguridades y control.

El método deductivo según (Rodríguez y Pérez, 2017), señalan que “Mediante este procedimiento, se organizan hechos conocidos y se extraen conclusiones mediante una serie de enunciados”, puesto que se aplicará principios ya descubiertos dentro del ámbito de diseño y construcción en el funcionamiento de dispositivos electrónicos.

2.1.2 Fases del proyecto

Las actividades planteadas para el avance de la investigación se exponen en diferentes fases las cuales se basan en procedimientos que se llevaran a cabo para la implementación y desarrollo del proyecto.

Fase I. Determinar las especificaciones del *hardware*

Se desarrolla en forma general las etapas en las que se implementará el sistema estructural de la fresadora CNC para PCBs, mecánica, eléctrica y electrónica, rigiéndose bajo normas preestablecidas con el fin de garantizar precisión y optimización de recursos.

Fase II. Adaptación entre *hardware* y *software*

Al tener ensamblada la maquina e instalado los programas, se realizará pruebas de funcionamiento de sincronismo y trabajo en conjunto con todos los equipos, posteriormente se realizará un análisis de resultados del enlace vía Bluetooth con el fin de mantener comunicación permanente entre todos los dispositivos y así garantizar la funcionalidad del sistema.

Fase III. Implementación de *software*

El desarrollo del *software* se gestionará con los diferentes programas, de manera conjunta con la vinculación de las librerías necesarias para el control de sus dispositivos de dependencia.

Fase IV. Depuración de *hardware* y *software*

Se filtrará cada etapa corrigiendo desperfectos del proyecto, realizando pruebas rigurosas de funcionamiento hasta obtener el producto planteado.

CAPÍTULO III

3.1 PROPUESTA

La propuesta planteada es el diseño y construcción de una fresadora CNC para PCBs, con materiales de adquisición local como los motores, controladores, Arduino UNO y demás elementos, con la excepción del motor *Spindle* (husillo) el cual no hay en el país, y se lo importo, con esto se redujo costos realizando la implementación y desarrollo de forma eficiente. Los elementos instalados están regidos por normas y estándares internacionales tales como ISO e EIA.

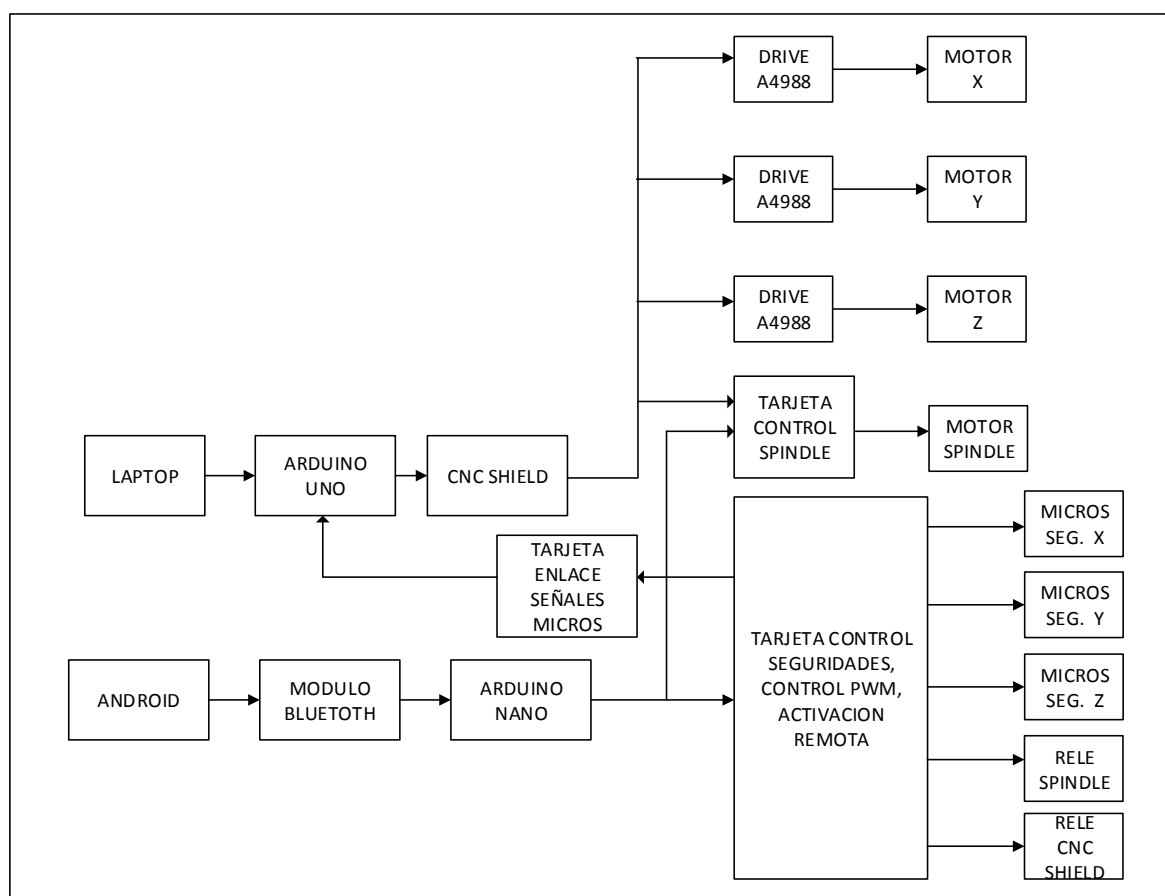


Figura 13: Diagrama de bloques de la fresadora CNC.

Fuente: Elaborado por el autor.

Se realizará una interfaz gráfica a través de un Smartphone y el programa *App Inventor* 2 con el cual se gestionará alarmas de los finales de carrera de los ejes X, Y, Z, vía Bluetooth.

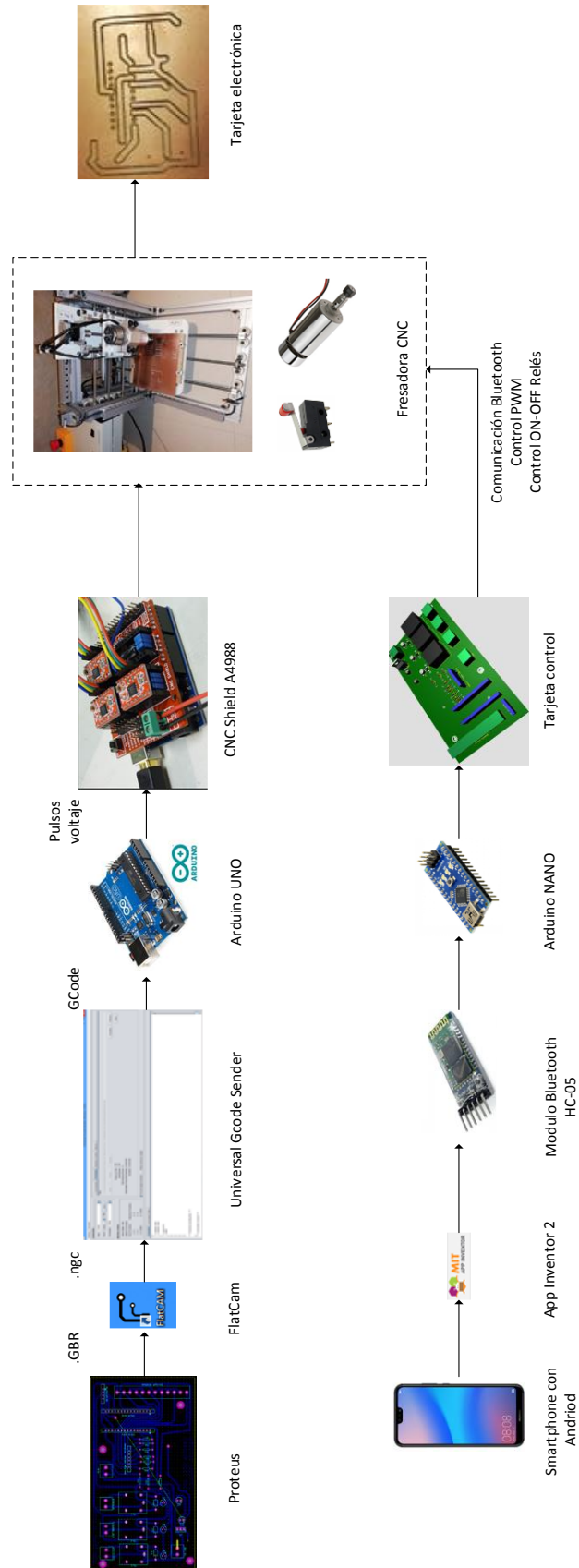


Figura 14: Proceso de diseño, control y elaboración de placas PCBs

Fuente: Elaborado por el autor.

Se controlará la velocidad del *Spindle* a través de la entrada PWM de la tarjeta, se habilitará vía remota las tarjetas de control de los diferentes elementos de la fresadora CNC.

En la Figura 13 se observa la distribución física de los elementos que interactúan entre sí, para poder controlar la fresadora CNC.

3.2 ARDUINO UNO.

Es una tarjeta de microcontrolador basada en un microprocesador ATmega 328, tiene 14 pines digitales entre entradas y salidas digitales dentro de los cuales hay seis pines que se pueden utilizar como salidas PWM, seis entradas analógicas, un cristal de cuarzo de 16 MHz, un conector USB para comunicación y programación, un conector de alimentación de 7 - 12 Vdc, un botón de *reset*, una entrada ICSP (*In Chip Serial Programmer*) esta da acceso a la memoria de programa AVR (*Flash*) con la cual se puede grabar cualquier programa al microcontrolador desde la PC sin necesidad del puerto USB, con la misma entrada ICSP en caso de cambiar el microprocesador se podrá grabar el *Bootloader* (gestor de arranque) con el cual se carga y se hace funcionar los programas realizados con el *software* de Arduino, un microprocesador ATmega 16U2 actúa como un convertidor de USB a Serie.

En el análisis para el desarrollo del proyecto el factor económico es fundamental y se implementó este por tener *hardware* “*Open-Source*”,

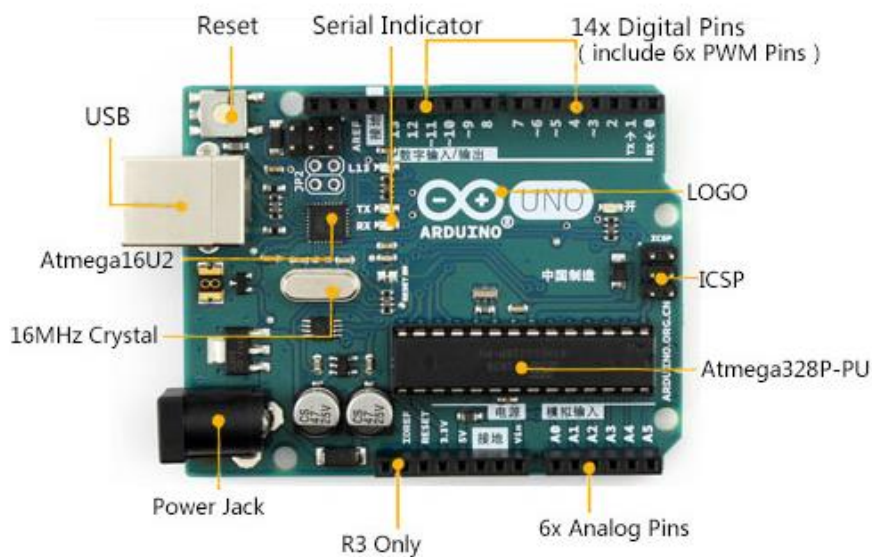


Figura 15: Partes principales de un Arduino UNO.

Fuente: (“Fichier:Arduino uno Pinout (paysage).png — MCHobby - Wiki”, s/f).

3.2.1 Programa Arduino UNO

Firmware GRBL Controller

Grbl es un *software* gratuito, de código abierto y alto rendimiento permite controlar el movimiento de máquinas CNC, la forma estándar de comunicarse es mediante Código G, el *software GRBL controller* es un sistema de control computarizado encargado de enviar línea a línea dicho código a la tarjeta de control, y así permitir la activación de los actuadores.

La ventaja de uso de este *firmware* es que permite la aceleración suave, permitiendo una vez los motores inmóviles, acelerar progresivamente ayudando a vencer la inercia que se presenta al arrancar de los motores, tiene otra aplicación en los casos de movimientos en esquinas o cambios de giros permitiendo desacelerar de manera progresiva para así poder realizar el movimiento en la posición correcta.

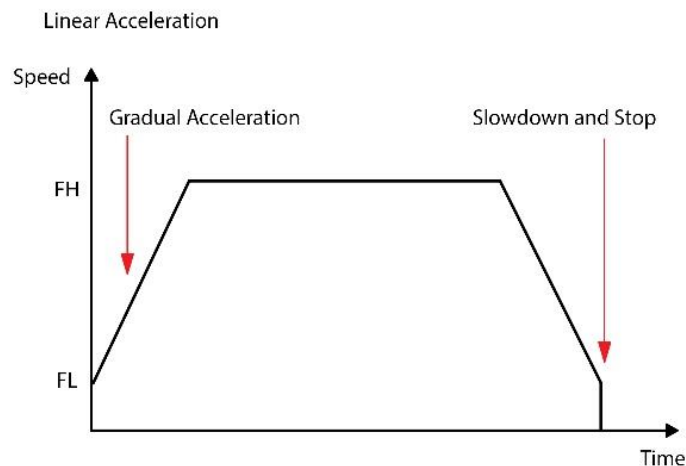


Figura 16: Aceleración suave del firmware GRBL.

Fuente: ("Using a Dedicated Pulse Control LSI vs. a CPU for Motion Control | Nippon Pulse - Your Partner in Motion Control - Radford, VA", s/f)

Librerías CNC Arduino UNO

El *firmware GRBL Controller* consta de 17 librerías, cada una de ellas tiene una tarea en específico a fin de poder interpretar de manera correcta el código CNC, las librerías más relevantes son:

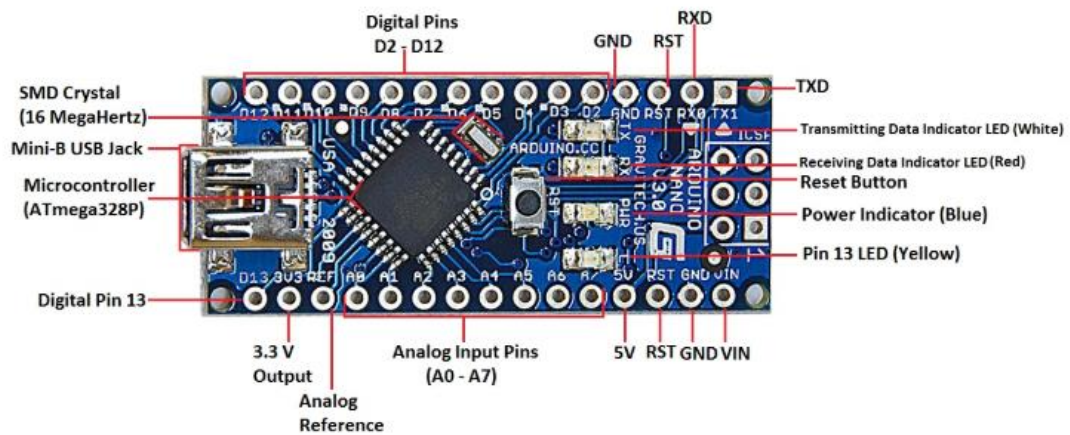
Tabla 3: Librerías del firmware GRLB utilizado para la CNC.**Fuente: Elaborado por el autor.**

Librería	Función
Config	Realiza una configuración completa del Arduino, la tarea principal de configuración es definir cada uno de los pines, como son los de control de motores, las señales de los fines de carrera.
Defaults	Tiene la configuración por defecto de algunos parámetros de maneja de máquina como son los la velocidad de los motores, la aceleración, el punto cero de la máquina, la precisión del movimiento, estos parámetros los tiene definidos para 4 tipos diferentes de movimientos.
Gcode	Librería encargada de interpretar cada una de las líneas de comandos enviadas por el Universal Gcode Sender, sus funciones principal es leer la instrucción y diferencias cada una de ellas, las más importantes se pueden observar en la Tabla 1 y Tabla 2.
Grblmain	Librería principal, es la encargada de controlar cada una de las otras permitiendo que cada una se active y desactive en el instante en el que sean necesarias.
Limits	Liberia encargada de controlar, las señales que proporcionan los finales de carrera de los límites de cada uno de los ejes sea X, Y o Z.
Motion_control	Librería encargada del movimiento de los motores, dependiendo de número de hilos del tornillo, los grados de giro del motor, para posicionar en el punto solicitado por el código CNC.
Serial	Librería encargada de la comunicación entre la computadora y la tarjeta de control(Arduino), permitiendo así él envío de código G hacia la misma.

3.3 ARDUINO NANO.

Es una tarjeta de microcontrolador de dimensiones reducidas con un microprocesador ATmega 328, posee una entrada mini USB para poder programarlo, tiene 14 puertos digitales que pueden ser usados como entradas o salidas dentro de los cuales tenemos 6 entradas

PWM, 8 puertos análogos, una memoria de 16 KB, 1 KB de SRAM, 512 de EPROM, el rango de voltaje de alimentación es de 6 a 20 VDC.



Arduino Nano V3.0 Pinout

Figura 17: Partes principales de un Arduino Nano.

Fuente: store.arduino.cc

Este elemento contiene el programa de seguridades de los micros y la salida PWM para controlar el *Spindle*, la comunicación se realiza a través del módulo Bluetooth HC05 con un Smartphone que contiene la interfaz gráfica en la plataforma Android.

3.4 MOTORES PASO A PASO

El motor seleccionado es un motor a pasos de Nema 17 como se muestra en la Figura 18, serán utilizados tres motores para generar los movimientos en los ejes X, Y, Z.

Son motores bipolares los cuales tienen 2 bobinas y cuatro terminales con esto para una secuencia determinada necesita la inversión de la corriente que circula en sus bobinas, a cada inversión de la polaridad acciona el movimiento del eje en un paso cuyo movimiento está determinado por la siguiente secuencia como se ve en la Tabla 4.

Especificaciones:

- Nema: 17
- Modelo: KS42STH40-1204A
- Torque: 55.5 oz/in

- Inercia: 54 g/cm²
- Corriente: 1.2 A
- Voltaje: 12 VDC



Figura 18: Motor a pasos Nema 17.

Fuente: <https://tienda.bricogeek.com/motores-paso-a-paso/546-motor-paso-a-paso-nema-17-32kg-cm.html>

Tabla 4: Secuencia control motores paso a paso bipolares.

Fuente: Elaborado por el autor.

PASO	TERMINALES			
	A	B	C	D
1	+V	-V	+V	-V
2	+V	-V	-V	+V
3	-V	+V	-V	+V
4	-V	+V	+V	-V

3.5 MOTOR DC SPINDLE

El motor de husillo (*Spindle*) es un motor DC *Brushless* sin escobillas modelo HQUDJ GS52 de 400 W de 12000 rpm, Rev: 3000-12000r/min (12v-3000 r/min, 24V-6000 r/min, 36V-9000 r/min, 48V-12000 r/min) y 0.50 Nm, viene adaptado con su propia fuente de 12 a 48 VDC con la cual se puede controlar la velocidad a través de un potenciómetro o con una entrada PWM (modulación por ancho de pulso) en este caso se utiliza la PWM a través del Arduino NANO y un teléfono Android para en control de la velocidad, el eje viene con refrigeración por aire forzado.

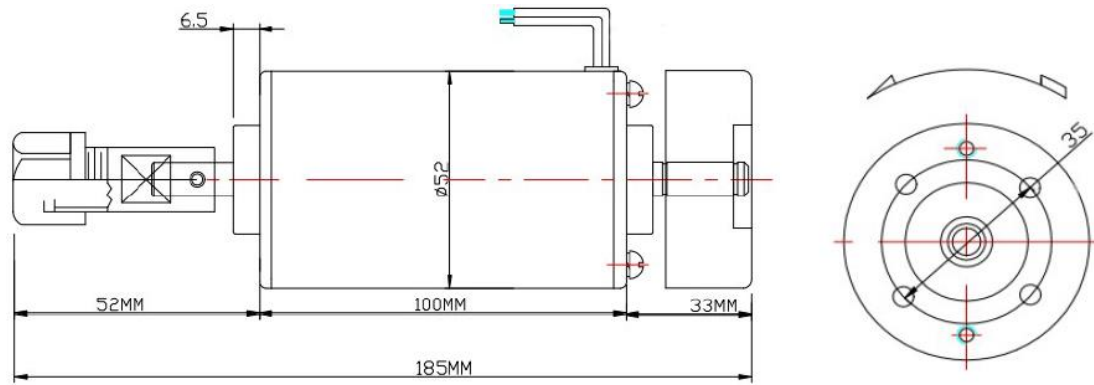


Figura 19: Dimensiones motor DC Spindle

Fuente: www.ekt2.com

3.6 TERJETA CONTROL MOTOR DC SPINDLE

Trabaja con voltajes de 12 a 60 VDC para el accionamiento del motor DC Spindle, tiene dos modos de control a través de un potenciómetro de 10 K y la que se usa en este proyecto a través de la entrada PWM la cual como requisito exige un nivel de 3,5 a 12 VPP, frecuencia 1K a 10 KHZ, la corriente nominal de la tarjeta es de 10 amperios, la potencia nominal de trabajo es 600 W estando con una ventana de 200 W ya que el motor es de 400 W, el rango de modulación del ancho de pulso (PWM) es de 0% a 100%.



Figura 20: Tarjeta de control motor DC Spindle.

Fuente: www.ebay.es

3.7 CONTROLADOR A4988

Se usan tres drivers controladores para los ejes X, Y, Z, respectivamente, el control de un motor paso a paso se ve simplificado con este driver ya que genera las señales para funcionamiento y protege al motor por alta temperatura o corriente. Se alimenta con tensiones para el driver 3 a 5 VDC y para el motor a pasos de 8 a 35 VDC con una corriente nominal de 1 a 2 Amp.

El Arduino UNO envía pulsos de corriente dentro de los cuales se diferencian ya que tenemos pulsos para el avance del motor, dirección del motor, y un GND común con diferentes pines de salida del Arduino, de esta forma cuando el driver recibe un pulso de avance el circuito comprueba el voltaje del pin de dirección y energiza las bobinas en el orden adecuado.

Los pulsos deben ir sincronizados para evitar que el motor tenga movimientos no adecuados o ruidos al girar que provocarían su daño prematuro. Para realizar la conexión entre el Arduino UNO y los drivers se utiliza la tarjeta CNC *SHIELD*.

En la Figura 21 se observa el diagrama de temporización de la interfaz lógica dada a un motor a pasos a través del driver A4988.

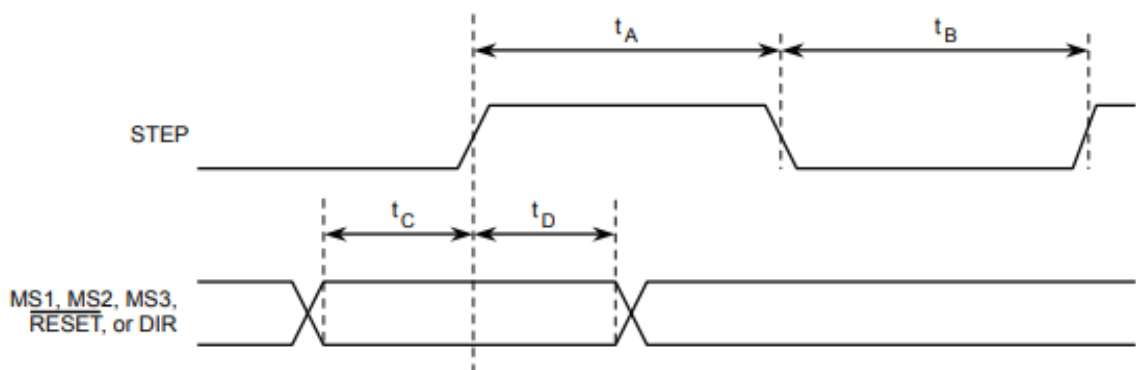


Figura 21: Diagrama de temporización de la interfaz lógica.

Fuente: A4988-Datasheet.pdf

Tabla 5: Datos diagrama temporización de la interfaz lógica

Fuente: A4988-Datasheet.pdf

Time Duration	Symbol	Typ.	Unit
STEP minimum, HIGH pulse width	tA	1	μs
STEP minimum, LOW pulse width	tB	1	μs
Setup time, input change to STEP	tC	200	ns
Hold time, input change to STEP	tD	200	ns

3.8 COMUNICACIÓN BLUETOOTH

El módulo HC-05 Bluetooth trabaja bajo una especificación industrial WPAN (*Wireless Personal Área Network*), con la cual se puede transmitir voz y datos entre dispositivos a través de enlaces de radiofrecuencia electromagnética en la banda ISM (*Industrial, Científic and Medical*) la cual es de uso no comercial, dentro de los 2,4 GHz.

El enlace se lo hace a través de un Smartphone Huawei P 20 LITE el cual contiene el programa realizado en MIT *App Inventor 2* y el Arduino NANO que contiene el programa de alarmas de los micros de seguridad, el encendido y apagado de la tarjeta de control de la CNC, la tarjeta de control del *Spindle* y el control PWM para el control de velocidad del *Spindle*.

3.9 FUENTES DE PODER

Se utilizará 2 fuentes de poder la primera viene con el motor DC *Spindle* la cual trabaja con 110 V AC a la entrada y 48 VDC a la salida, la corriente nominal es de 8,3 amperios, el voltaje de salida puede ser ajustable, esta fuente es exclusiva para el motor del *Spindle* y no se alimentará ningún otro dispositivo.

La segunda fuente se utiliza para todo el control de los dispositivos de seguridad, alimentación de los Arduinos, alimentación de la tarjeta de control y comunicación, control y mando de los motores a pasos, las características principales, trabaja con un voltaje de entrada de 110 VAC y 12 VDC a la salida y la corriente nominal es de 5 amperios, las dos fuentes de alimentación son conmutadas.



Figura 22: Fuentes de poder

Fuente: www.ebay.es

3.10 DESARROLLO DE SOFTWARE

El entorno para el desarrollo del *software* se implementará en Windows 8.1, todos los programas son sin licencias por lo que se adquieren de manera fácil desde el internet.

3.10.1 MIT app inventor 2

Para la aplicación desarrollada en el sistema operativo Android se trabajó con *App Inventor 2* por ser un *software* el cual se va enlazando una serie de bloques para obtener la aplicación necesaria, con una cuenta de correo electrónico se crea un usuario y se procede a programar.

Se desarrolló los bloques para la identificación de los dispositivos Bluetooth dentro del rango de alcance, para la recepción de las alarmas por topes de los finales de carrera de los ejes de trabajo X, Y, Z, la comunicación PWM para controlar la velocidad del motor

Spindle, adicional se generó botones para el encendido y apagado de las tarjetas de control principales todo esto a través de un Smartphone.

3.11 DESBASTE Y PERFORACIÓN

El proceso de desbaste y perforación es la extracción de viruta de la placa de baquelita con la cual se crea un surco que permite separar una pieza de otra o vaciar una parte sobrante de un vaciado. En el proceso tanto de corte como perforación se genera fricción y por ende un cambio térmico en este caso calor, el cual según el avance no debe exceder ya que al contacto con material frío debe mantener la temperatura ambiente de no ser así existe algún fallo en el afilado de la herramienta de corte.

Se debe tener en cuenta la separación de las pistas, la longitud y el ancho de las mismas ya que influye directamente en el producto final de trabajo, tomando en cuenta que al trabajar con voltajes de diversas escalas se puede generar interferencias y conflictos en los dispositivos electrónicos, se establece por norma que el tamaño de las pistas deben ser más ancho que el diámetro de la broca, la norma IPC para el diseño de circuitos electrónicos recomienda valores según la corriente a circular por las pistas como se muestra en la Tabla 6.

Tabla 6: Separación mínima entre pistas para diferentes voltajes.

Fuente: IPC 2221^a

Voltage Between Conductors (DC or AC Peaks)	Minimum Spacing						
	Bare Board				Assembly		
	B1	B2	B3	B4	A5	A6	A7
0-15	0.05	0.1	0.1	0.05	0.13	0.13	0.13
16-30	0.05	0.1	0.1	0.05	0.13	0.25	0.13
31-50	0.1	0.6	0.6	0.13	0.13	0.4	0.13
51-100	0.1	0.6	1.5	0.13	0.4	0.5	0.13
101-150	0.2	0.6	3.2	0.4	0.4	0.8	0.4
151-170	0.2	1.25	3.2	0.4	0.4	0.8	0.4
171-250	0.2	1.25	6.4	0.4	0.4	0.8	0.8
251-300	0.2	1.25	12.5	0.4	0.4	0.8	0.8
301-500	0.25	2.5	12.5	0.8	0.8	1.5	0.8
>500	0.0025 mm/V	0.005 mm/V	0.025 mm/V	0.00305 mm/V	0.00305 mm/V	0.00305 mm/V	0.00305 mm/V

- *B1: Internal Conductors*
- *B2: External Conductors, uncoated, sea level to 3050 m*
- *B3: External Conductors, uncoated, over 3050 m*
- *B4: External Conductors, with permanent polymer coating (any elevation)*
- *A5: External Conductors, with conformal coating over assembly (any elevation)*
- *A6: External Component lead/termination, uncoated, sea level to 3050 m*
- *A7: External Component lead termination, with conformal coating (any elevation)*

Tabla 7: Ancho de pista para PCB.

Fuente: <https://www.electronics-project-design.com/PCB-Design.html>

Courent/ A	Track Width (mm)
1	0.25
2	0.76
3	1.27
4	2.03
5	2.79
6	3.81
7	4.57
8	5.59
9	6.60
10	7.62

Las placas de baquelita están compuestas de dos capas una conductora de cobre y una de base aislante, los estándares del espesor del cobre van desde 17.5, 35, 70 y 105 micras, las baquelitas comunes en el mercado tienen una base de 1.6 mm y un recubrimiento de cobre de 35 micras o 0.035mm. (IPC 2221, 2013)

3.12 FRESAS Y BROCAS

Para el tipo de desbaste a realizar es necesario utilizar fresas cónicas de Carburo de Tungsteno con recubrimiento de Titanio, ya que tienen alta resistencia a la flexión, alta dureza, alta eficiencia, y resistencia al desgaste.

Material: Carburo de Tungsteno endurecido con recubrimiento de Titanio.

Angulo de corte: 10° y 30°.

Diámetro de la punta de corte: 0.1 mm y 0.2 mm.

Diámetro del vástago: 3.175 mm.

Longitud: 34 mm.



Figura 23: Dimensiones fresa cónica de 30°.

Fuente: mercadolibre.com.ec

Las brocas planteadas para la perforación de las placas son de Carburo de Tungsteno, teniendo en cuenta que se necesite que corte y evacue los residuos, las más adecuadas son de forma helicoidal.

Material: Carburo de Tungsteno.

Diámetro de la punta de corte: 0.8 mm y 1 mm.

Diámetro del vástago: 3.175 mm.

Longitud: 38 mm.

Pueden trabajar hasta con 30.000 rpm.

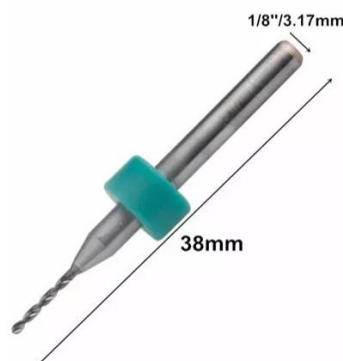


Figura 24: Dimensiones broca helicoidal.

Fuente: mercadolibre.com.ec

3.13 ANÁLISIS DE FACTIBILIDAD

La factibilidad y viabilidad permitió evaluar y conocer los factores a gestionar para el desarrollo integral del proyecto, los costos y beneficios para la investigación, el diseño y construcción que intervino en los siguientes puntos:

3.13.1 Factibilidad Técnica

Se evaluó dos puntos importantes para el diseño y la construcción de la CNC los cuales son el *hardware* y el *software*.

Dentro del *hardware* se investigaron los elementos necesarios para el montaje y la viabilidad de poder adquirirlos localmente generándose trabas en algunos, y logrando gestionar de manera efectiva y solucionar los percances en la adquisición y construcción de la fresadora CNC.

Para el *software* se gestionaron los programas a utilizarse los cuales se buscó en su mayoría *software* libre por cuestión de licencias y se desarrolló los programas necesarios para el control total de la fresadora CNC, se desarrolló una aplicación en Android para un control de velocidad PWM del *Spindle* vía Bluetooth, observar advertencias de finales de carrera si la máquina sale de los límites de guías establecidos, y habilitar o deshabilitar toda la máquina a través de un Smartphone.

Tabla 8: Especificaciones técnicas del proyecto.

Fuente: Elaborado por el autor.

Especificaciones técnicas del proyecto	
Especificaciones	Descripción
Elementos de control	Drivers, tarjetas de control, interfaz de usuario / laptop, celular.
Actuadores	Motores a paso Nema 17, motor DC <i>Spindle</i> .
Elementos mecánicos	Estructura de la fresadora (X,Y,Z), bancada de soporte (mesa), <i>Spindle</i> .
Herramientas	Brocas para taladrado, brocas punta de lanza para fresado.
Zona de trabajo (X,Y,Z)	170mm x 140mm x 25mm

Continuación Tabla 8: Especificaciones técnicas del proyecto.

Velocidad de rotación máxima <i>Spindle</i>	12000 rpm
Precisión de posicionamiento	±0.1 mm
PCBs	Una capa de baquelita
Velocidad X,Y (avance)	
Dimensión total de la fresadora	580mm x 570mm x 410mm

3.13.2 Factibilidad Económica

Se solicitó cotizaciones dentro y fuera del país, con todo este análisis y dentro del presupuesto establecido se realizó la compra localmente, con la excepción de un elemento que se importó.

3.13.3 Factibilidad Operativa

Investigando la operación y gestión de las fresadoras CNC se obtuvo una idea clara de lo que se desea obtener, y con el fin de garantizar a las personas que interactúen con la máquina de una forma adecuada de adaptación y manipulación del equipo y su *software*, se elaboraron manuales de operación, planos de instalación eléctrica y conexión de la fresadora CNC, planos del conjunto total de la fresadora, y se implementaron finales de carrera de seguridad que advierten si las guías salen de los límites establecidos, de igual forma la máquina para por seguridad con el fin de precautelar daños al operador y a los materiales y herramientas en uso, tomando en cuenta para este punto tener una bancada sólida para evitar vibraciones y una activación errónea de las alarmas.

3.14 ANÁLISIS DE COSTOS DEL PROYECTO

Un análisis detallado de los costos es de suma importancia, ya que en función a este se determina la viabilidad del proyecto y con una minuciosa investigación la posible manufactura de la fresadora CNC, referente a las características planteadas, es necesario para proceder a la etapa de fabricación, una vez gestionado lo citado se encontró que los costos para importar una fresadora CNC son muy elevados, estando la más económica en 1310 dólares americanos y de dimensiones de trabajo más pequeñas, con esto se realizó

cotizaciones a nivel local, en otras provincias e internacionales en los países vecinos con el fin de bajar costos en materiales sin descuidar la calidad y así obtener una máquina óptima para el trabajo requerido.

De las proformas obtenidas se analizó los costos más viables con los diferentes proveedores y se clasificó para la manufactura de la máquina en:

- Costos componentes mecánicos CNC
- Costos componentes electrónicos CNC
- Costos componentes importación CNC
- Costo implementación y desarrollo CNC
- Costo total del proyecto
- Análisis costo construcción local Vs. Importación

Tabla 9: Tabla de costos de los componentes mecánicos CNC

Fuente: Elaborado por el autor.

Costos componentes mecánicos CNC			
Descripción	Cantidad	Costo unit.	Costo total
Perfil de aluminio 40*40*6000	1	45	45
Acople Flexible para tubos de 8mm Entrada de 5mm	4	4,07	16,28
Tornillos acerados 8mm 4hilos	1	84	84
Tuerca de bronce 8mm 4hilos	3	4	12
Ejes acerados 8mm	1	10	10
Rodamiento Lineal 8mm Con Agujeros para tornillo	12	7,45	89,4
Soporte para eje 8mm	12	4,5	54
Caja conexiones	1	10	10
	Total		320,68

Tabla 10: Tabla de costos de componentes electrónicos CNC.

Fuente: Elaborado por el autor.

Costos componentes electrónicos CNC			
Descripción	Cantidad	Costo unit.	Costo total
Motores NEMA 17 1.2A	3	22,25	66,75
Arduino UNO + cable USB	1	15	15
Arduino NANO + cable USB	1	9	9

Continuación Tabla 10: Tabla de costos de componentes electrónicos CNC.

A4988 DRIVER	3	4	12
Shield CNC V4	1	12	12
Fuente adaptador 12V a 5VDC 4A	1	15	15
Módulo Bluetooth	1	5	5
	Total		134,75

Tabla 11: Costos de componentes de importación de CNC

Fuente: Elaborado por el autor.

Costos componentes importación CNC			
Descripción	Cantidad	Costo unit.	Costo total
Motor <i>Spindle</i> (importación, desaduanización, nacionalización)	1	210	210
	Total		210

Tabla 12: Costos de implementación y desarrollo CNC.

Fuente: Elaborado por el autor.

Costo implementación y desarrollo CNC			
Descripción	Cantidad	Costo unit.	Costo total
Materiales montaje	1	30	30
Transporte	1	25	25
Alimentación	1	25	25
Alquiler taller y herramientas para montaje	1	40	40
	Total		120

Tabla 13: Tabla de costo total del proyecto.

Fuente: Elaborado por el autor.

Costo total del proyecto		
Descripción	Cantidad	Costo
Costos componentes mecánicos CNC	1	320,68
Costos componentes electrónicos CNC	1	134,75
Costos componentes importación CNC	1	210
Costo implementación y desarrollo CNC	1	120
	Total	785,43

Tabla 14: Análisis Costo Construcción local Vs. Importación**Fuente: Elaborado por el autor.**

Análisis Costo Construcción local Vs. Importación	
Construcción local	\$ 785,43
Importación	\$ 1310
% Ahorro Construcción Vs. Importación	40 %

3.15 VENTAJAS DE LA FRESADORA CNC PARA PCB

- Al realizar una producción en serie el tiempo de repuesta es mínimo, con la misma calidad y precisión de todas las piezas a elaborar.
- Se elimina la contaminación del medio ambiente al realizar la producción bajo este método y no con agentes químicos.
- El proceso de manejo y control de la fresadora CNC es de fácil aprendizaje basta con leer los manuales de este proyecto.
- El mantenimiento y limpieza de la máquina es de fácil acceso y no genera rubros adicionales a más de un agente lubricante para las partes móviles.
- El voltaje de trabajo es de 110V que se lo encuentra en cualquier punto dentro del Ecuador.
- Al tener un peso de 20 Kg aproximadamente es de fácil transporte.

CAPÍTULO IV

4.1 IMPLEMENTACION.

La implementación de la CNC se basó previamente en el diseño y posterior construcción, realizada a través de investigaciones tanto en libros bibliográficos como en la internet descartando elementos de alto costo y difícil adquisición, dentro de los elementos más relevantes en el proceso tenemos los motores paso a paso, el Arduino UNO y NANO, los drivers, el motor del *Spindle* con sus respectivos controladores. Una vez planteado el cronograma se realizó la construcción de la estructura, se armó con todos los componentes necesarios y se realizó pruebas sin carga elemento por elemento.

Se gestionaron las librerías necesarias, y se descargó los programas a ser utilizados, al poseer tanto *hardware* como *software* se realizaron pruebas con energía y se corrigió fallos generados, se implementó las seguridades necesarias con su respectivo programa y diseño en el Smartphone para control y adquisición de alarmas. La implementación se divide en cuatro etapas, mecánica, electrónica, eléctrica y desarrollo del *software*.

4.2 DESARROLLO.

4.2.1 Diseño y construcción de la fresadora CNC

Una vez adquirido todos los elementos y materiales se realizó un plano de la estructura mecánica en AutoCAD como se observa en la Figura 25 y Figura 26, con las dimensiones reales de la máquina, con lo cual se procedió al corte de las medidas adecuadas y su posterior ensamblé.

En la Figura 27 se muestra el ensamble de motores con la unión de las bridas de sujeción, los tornillos para el movimiento de cada uno de los ejes X, Y, Z.

Una vez ensamblado todas las partes, el eje Y, tiene acoplado una base para la sujeción de las baquelitas, en el eje z se acoplo el motor del *Spindle*, que es el encargado de rutear la placa de circuito impreso.

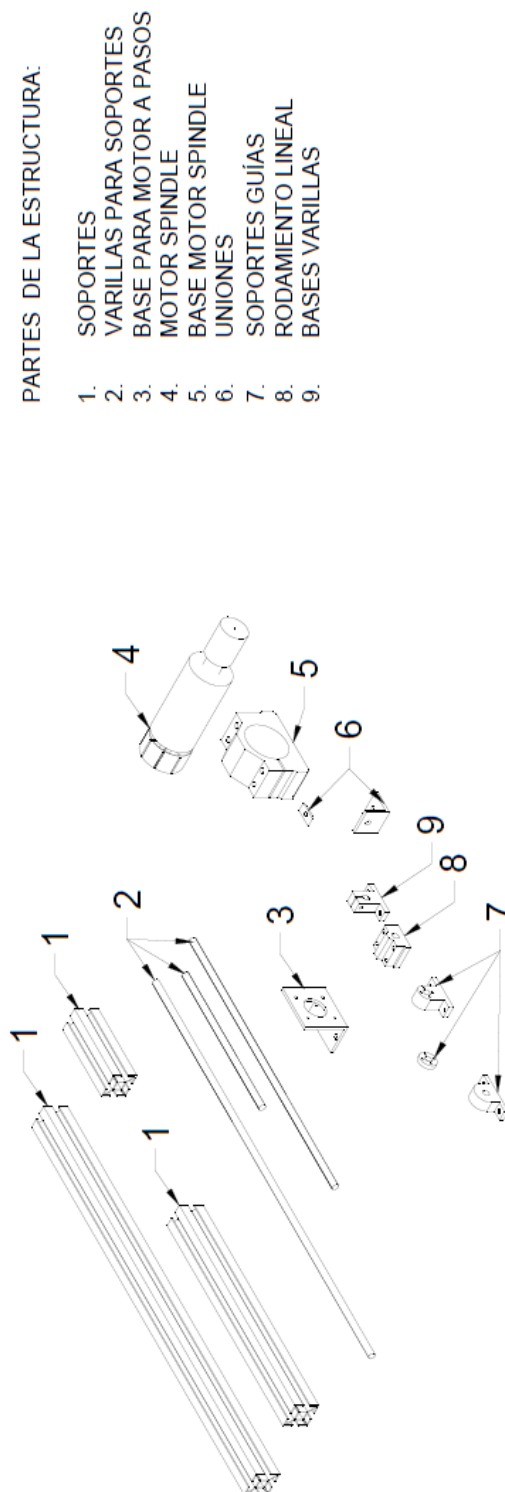


Figura 25: Partes de la fresadora CNC

Fuente: Elaborado por el autor.

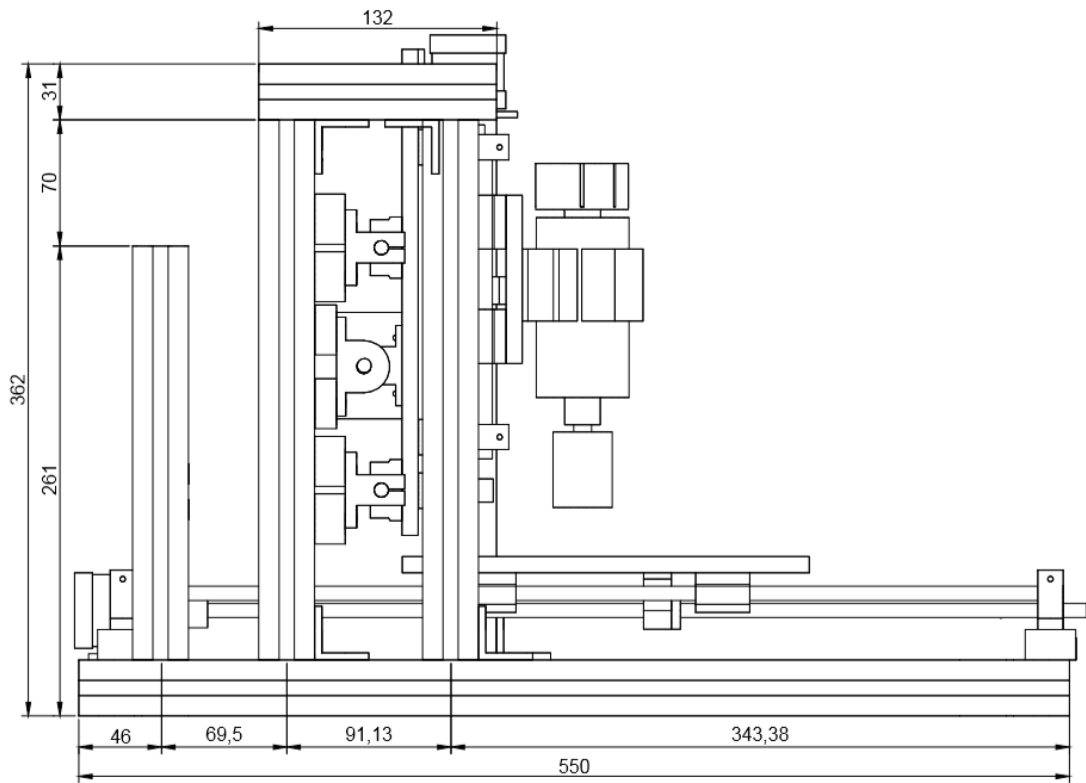


Figura 26: Diseño estructura fresadora en AutoCAD.

Fuente: Elaborado por el autor.



Figura 27: Acople de motores a paso con las bridas de sujeción y el tornillo.

Fuente: Elaborado por el autor.

4.2.2 Diseño y construcción de la tarjeta de control

Se diseñó una tarjeta para el control vía Bluetooth, adquisición de datos de seguridades, la señal PWM para controlar la velocidad del motor *Spindle*, y envía señales a los relés de activación para *CNC Shield* y el *Spindle* quedando un relé auxiliar para alguna

actualización futura, en la tarjeta está el módulo Bluetooth y el Arduino Nano los cuales se conectan directamente con el Smartphone Android el cual tiene el programa para recibir las señales de fallos y activar la máquina remotamente.

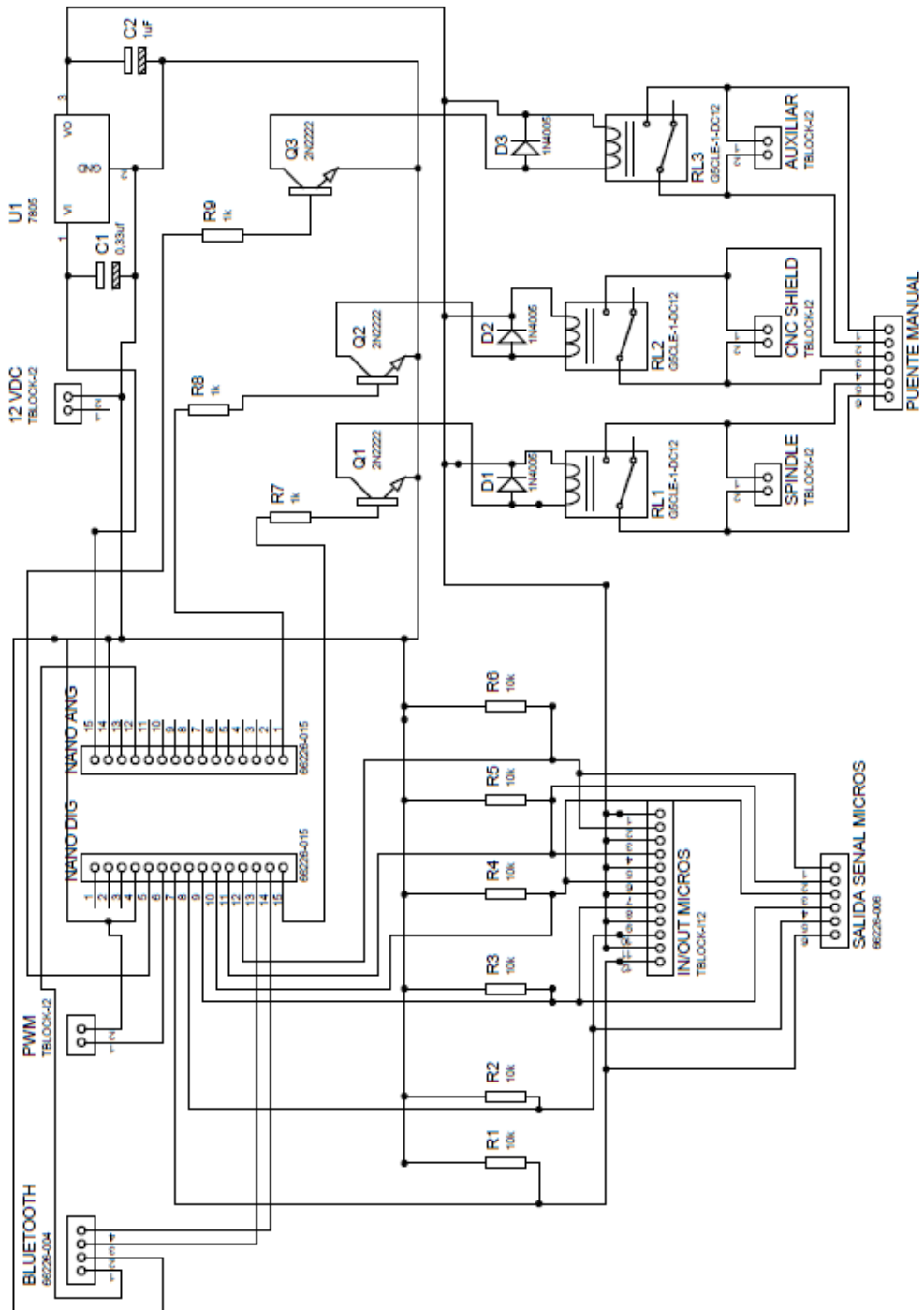


Figura 28: Diseño tarjeta electrónica control.

Fuente: Elaborado por el autor.

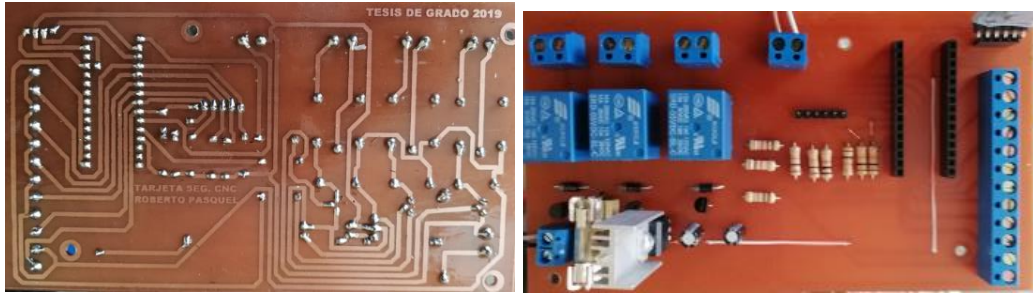


Figura 29: Tarjeta control seguridades, control PWM, activación remota.

Fuente: Elaborado por el autor

Adicional se construyó una tarjeta para enviar la señal de los micros de seguridad desde el Arduino NANO hacia el Arduino UNO.

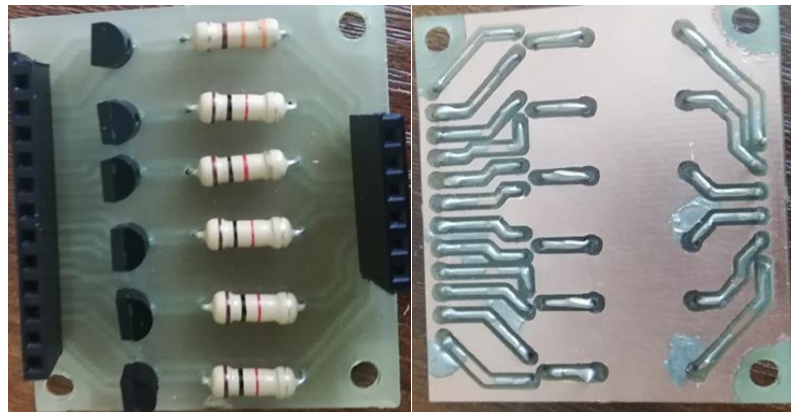


Figura 30: Tarjeta enlace señales micros

Fuente: Elaborado por el autor

4.2.3 Desarrollo del Software

MIT App inventor 2

El desarrollo de la aplicación consta de dos etapas la primera en el entorno de diseñador y la segunda en el entorno de bloques. En la etapa de diseñador se crea la parte visual que tendrá el usuario para conectarse con la CNC, las diferentes ventanas y controles, y la etapa de bloques es el programa que se comunica a través del Smartphone y el Arduino NANO en la TX transmisión y RX recepción de datos.

El entorno de diseño se encuentra dividido en 4 pantallas principales como se explica a continuación:

- Pantalla principal
- Control y alarmas
- Control principal
- Información

Pantalla principal

En la Figura 31, se muestra como se compone la pantalla principal:



Figura 31: Distribución de la ventana principal de la App.

Fuente: Elaborado por el autor.

La pantalla consta de 5 botones que cumplen con las siguientes funciones:

- **Conexión:** botón que permite la conexión de la App con el dispositivo Bluetooth.
- **Desconexión:** termina la conexión entre la App con el dispositivo Bluetooth.
- **Control y alarmas:** dirige a la ventana donde se encuentra un slider que controla la velocidad para el motor *Spindle* y muestra las fallas que se puede presentar en cualquier eje de la fresadora.
- **Control principal:** dirige a la ventana que tiene botones de on, off, que controlan los relés de control principales de accionamiento del *Spindle* y de la *CNC Shield*.
- **Información:** dirige a la ventana que contiene la información acerca del autor y el título del proyecto.

Control y alarmas

En la Figura 32, se muestra como se compone la pantalla de control y alarmas:

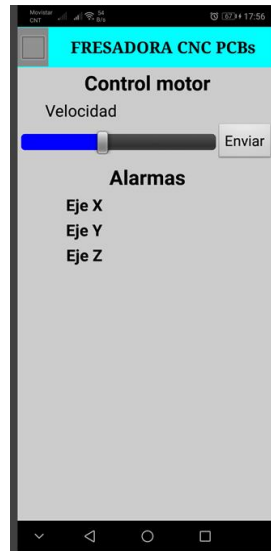


Figura 32: Distribución de la ventana de control y alarmas.

Fuente: Elaborado por el autor.

La parte principal de esta ventana, es el slider y el botón de enviar la velocidad del *Spindle*, además de ello cuenta con un visor de alarmas para cada uno de los ejes, en el caso de que alguno de los finales de carrera de uno se activa se enviara un señal de error a esta pantalla.

Control principal

A continuación, en la Figura 33 se muestra la composición de la ventana control principal:



Figura 33: Distribución de la ventana de control principal.

Fuente: Elaborado por el autor.

Esta ventana está dividida en cuatro botones, los botones tienen la función de controlar la activación y desactivación de los relés que permiten el accionamiento de *Spindle* y *CNC Shield*.

Información

En la Figura 34, se muestra la composición de la ventana de información:

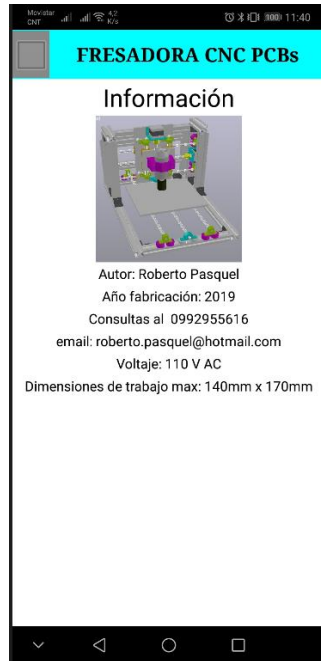


Figura 34: Distribución de la ventana de información.

Fuente: Elaborado por el autor.

En esta ventana tiene la utilidad principal de mostrar el autor de la App, el año en el que fue realizada, los contactos del autor y datos específicos de la fresadora.

Programación en bloques

El diseño de la App cuenta con dos partes principales, la una fue la parte gráfica que es lo que se muestra, pero atrás de ello existe la programación en bloques en la cual está la parte de navegación, como se interactúa entre unas ventanas y otras, los bloques de conexión para Bluetooth, y la escritura y lectura de variables.

Navegación

La navegación entre pantallas, se lo realiza mediante los botones de la ventana principal, una vez se acceda a la ventana, se utilizará el botón de atrás para regresar a la ventana principal y de ahí poder acceder a la ventana necesaria de nuevo.

En la Figura 35, se muestra la programación en bloques para el ingreso a cada una de las ventanas, la lógica utilizada es mientras se dé clic en un botón de la ventana principal, se desactivará esta misma y se activará la que se encuentra asignada al botón.

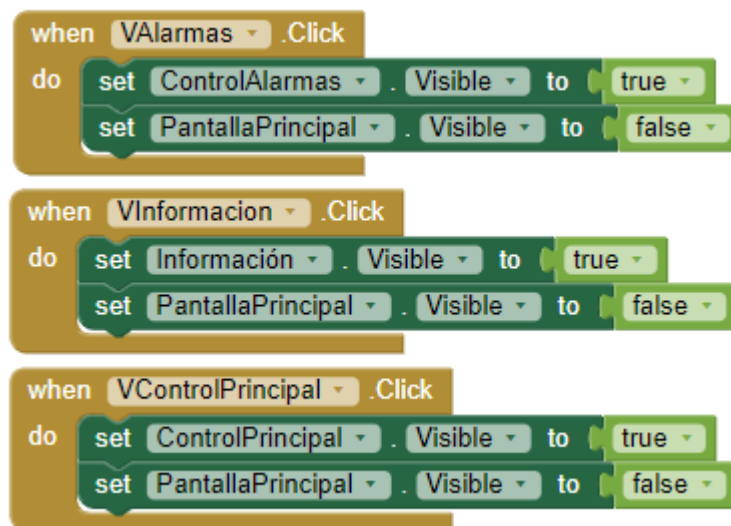


Figura 35: Programación de la navegación entre las ventanas.

Fuente: Elaborado por el autor.

Una vez ingresado a una ventana se utiliza el botón de retroceso, que tiene cualquier versión de Android, y se utiliza la siguiente lógica, en esta lógica una vez se encuentre activado una ventana, se comprobará cual ventana está abierta y se desactivará la visibilidad y se activará la visibilidad de la ventana principal, haciendo esto cada vez que se necesite navegar por la App.

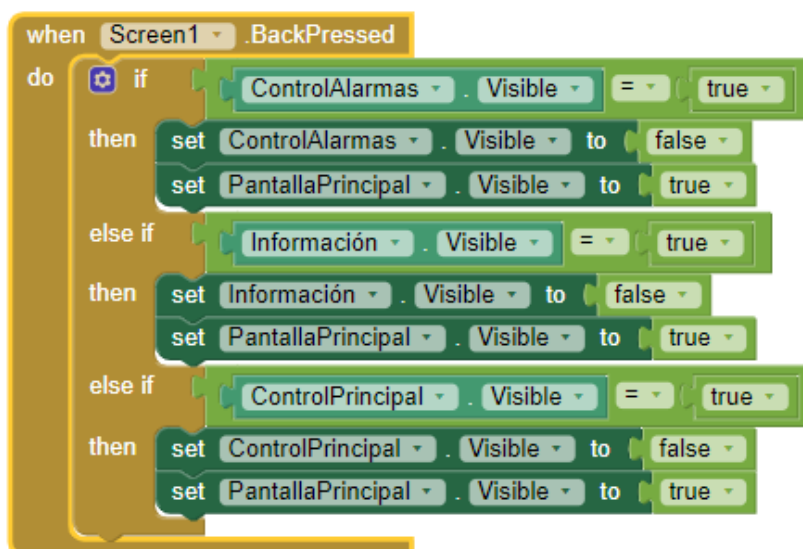


Figura 36: Programación en bloques del regreso entre las ventanas.

Fuente: Elaborado por el autor.

Conexión con el dispositivo Bluetooth

La conexión del Bluetooth se lo realiza a través de dos bloques de control como se muestra en la imagen a continuación:

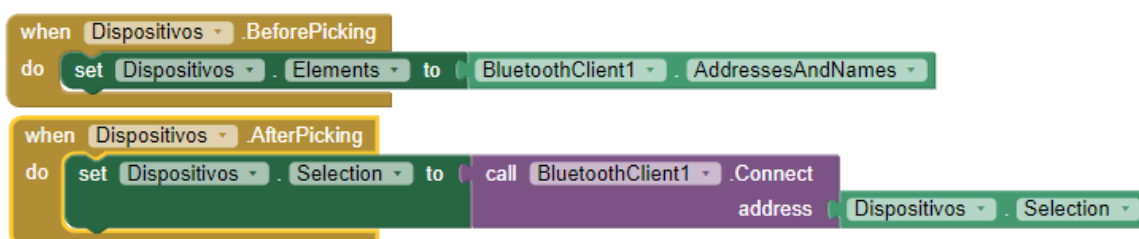


Figura 37: Programación en bloques de la conexión del Bluetooth con la App.

Fuente: Elaborado por el autor.

Envío de velocidad

Para el envío de velocidad es utilizado un *slider* en la ventana de control y alarmas, el valor del mismo será mostrado en un indicador en la pantalla, y mediante la activación del botón enviar como se muestra en los bloques.

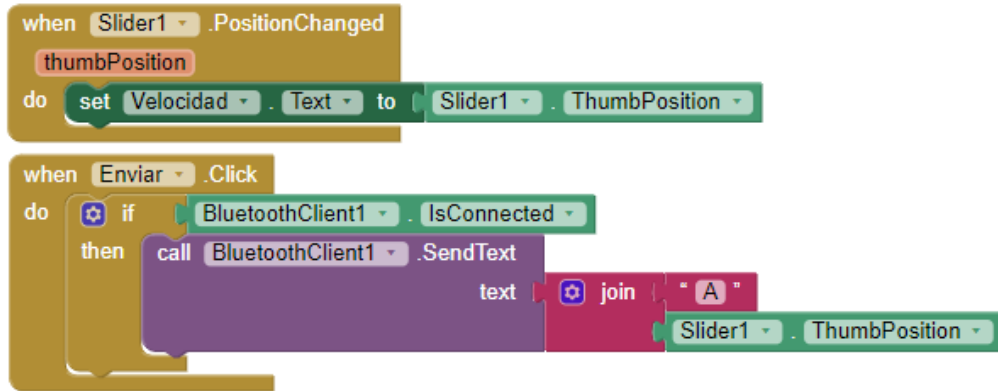


Figura 38: Programación en bloques del envío de la velocidad del *Spindle* desde la App.

Fuente: Elaborado por el autor.

Lectura de los avisos de alarma

Para la lectura de los avisos de alarma fue utilizada una variable global, para guardar el valor escrito en el serial buffer del Bluetooth, luego este valor es comparado con cualquiera de los valores, en este caso el valor puede ir desde a hasta d, en el caso de a el error se produce en el eje x, cuando el valor sea b es un error en el eje y, en caso del valor de c es un error en el eje z, y por último en el caso del valor d, no se produce error en ninguno de los ejes. En la Figura 39 se observa el ejemplo de la programación en bloques en el eje x, los ejes restantes son un símil de este.

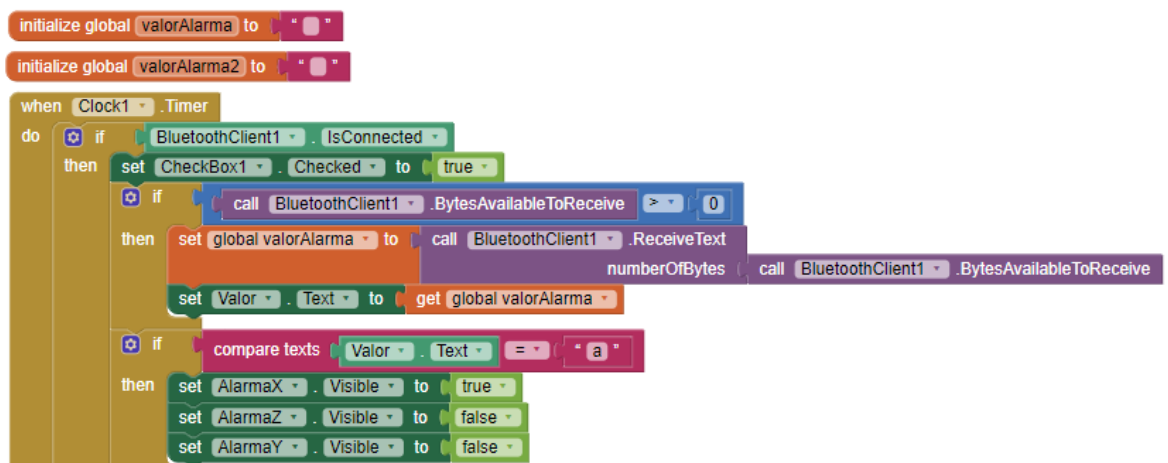


Figura 39: Programación en bloques de la lectura de los avisos de alarmas .

Fuente: Elaborado por el autor.

Escritura para la activación de los relés de energización principal

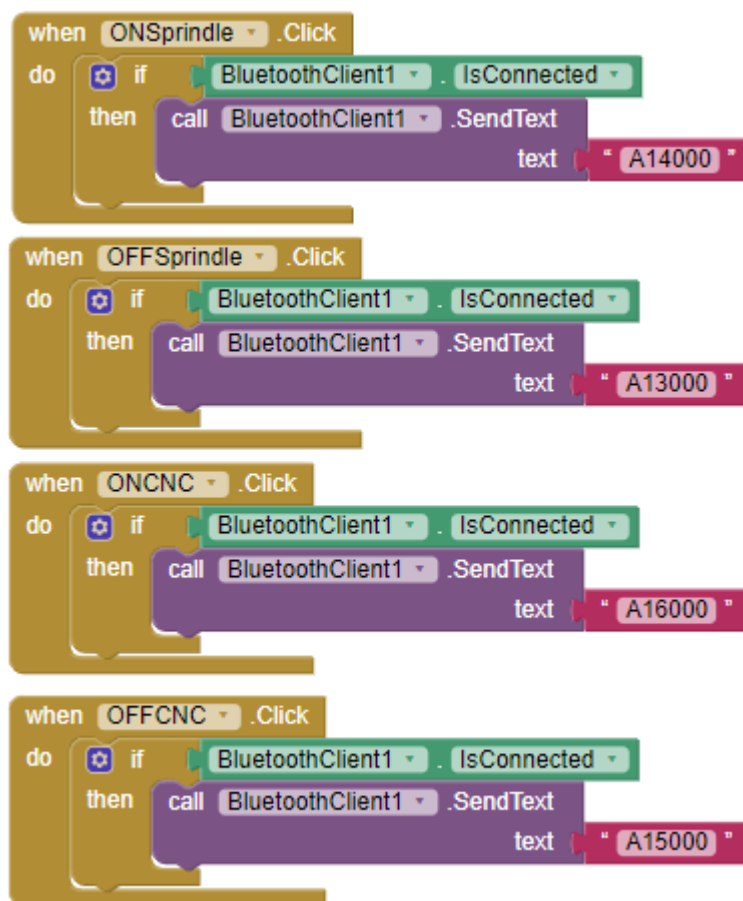


Figura 40: Programación en bloques de los estados de los relés de control.

Fuente: Elaborado por el autor.

Para la activación de los relés del *Spindle* y la *CNC Shield*, se utilizó valores superiores a 12000, como se muestra en la Figura 40 con el fin de discernir si los valores enviados son de velocidad de *Spindle* o de activación de los relés, para ello se utilizó los valores de 13000, 14000, 15000 y 16000, cada valor realiza la activación del relé de *Spindle*, desactivación del relé de *Spindle*, activación del relé de *CNC Shield* y desactivación del relé de *CNC Shield* respectivamente, la programación en bloque se muestra en la Figura 40.

Programa Arduino NANO

El programa de Arduino NANO consta de tres partes principales:

Comunicación Bluetooth

La comunicación con el Arduino NANO y el Bluetooth se la realizó mediante la librería `SoftwareSerial.h`, esta librería permite la creación de un puerto serial virtual entre el

Arduino y el Bluetooth, permitiendo así la escritura y lectura del dispositivo y que pueda ser interpretado por la tarjeta Arduino.

Los pines 10 y 11 fueron utilizados para la comunicación serial entre la placa Arduino y el Bluetooth, nombrada como BT, esta comunicación trabaja a 9600 baudios.

```
#include <SoftwareSerial.h>
SoftwareSerial BT(10,11);
```

Figura 41: Código para la creación de un Serial entre el arduino y el Bluetooth.

Fuente: Elaborado por el autor.

Lectura de datos

```
if (BT.available())
{
  velocidad = BT.parseInt();
  Serial.println(velocidad);
}
```

Figura 42: Código para la lectura de datos del Serial entre arduino y Bluetooth.

Fuente: Elaborado por el autor.

La lectura de datos fue realizada mediante el uso de dos funciones, la primera es la función `BT.available()`, esta función permite leer el número de bytes que se almacenan en el buffer, una vez este valor sea mayor que 0, la siguiente función `BT.parseInt()`, regresa el primer entero válido del serial buffer, leyendo de esta manera el dato enviado desde la App para la velocidad del *Spindle* o la activación o desactivación de los micros de control y guardándole en la variable `velocidad`.

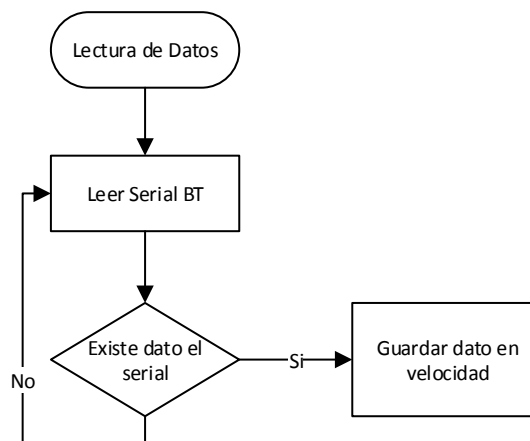


Figura 43: Diagrama de flujo lectura de datos.

Fuente: Elaborado por el autor.

Velocidad del *Spindle*

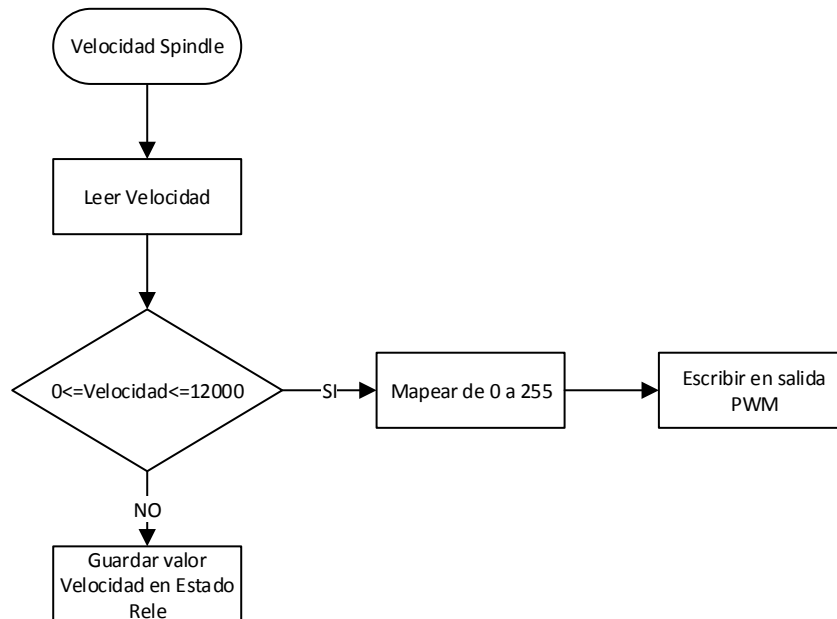


Figura 44: Diagrama de flujo velocidad del *Spindle*.

Fuente: Elaborado por el autor.

```

//Escritura de la velocidad PWM
if(velocidad >= 0 && velocidad <=12000)
{
  valorpwm = map(velocidad,0,12000,0,255);
  analogWrite(pwmPIN,valorpwm);
}
else
{
  //Estado de los redes de activación
  EstadoRele = velocidad;
}
  
```

Figura 45: Lectura del valor enviado por el serial y asignación de velocidad o estado.

Fuente: Elaborado por el autor.

La variable velocidad es comparada con el fin de determinar la señal que fue enviada, para el caso de la velocidad del *Spindle*, el valor esta entre 0 a 12000, que es la velocidad máxima del motor, para controlar la velocidad es necesario una salida PWM, esta salida tiene un valor entre 0 a 255, para ello se va a utilizar la función `map()`, esta función permite realizar equivalencias entre diferentes rangos, permitiendo trasladar el valor de velocidad a valor de salida PWM, y así modulando la velocidad a través del pin digital 3. En el caso de no estar

el valor de velocidad entre el rango de 0 a 12000, este valor será guardado en una variable llamada EstadoRele para posterior comparación.

Encendido de los relés de control principal

```
//Encendido de los relés de activación del control principal
switch (EstadoRele)
{
  case 13000:
    digitalWrite (ReleSpindle,LOW);
    Serial.println("Rele Sprindle Apagado");
    break;
  case 14000:
    digitalWrite (ReleSpindle,HIGH);
    Serial.println("Rele Sprindle Encendido");
    break;
  case 15000:
    digitalWrite (ReleShield,LOW);
    Serial.println("Rele Shield Apagado");
    break;
  case 16000:
    digitalWrite (ReleShield,HIGH);
    Serial.println("Rele Shield Encendido");
    break;
  default:
    break;
}
```

Figura 46: Control del estado de los relés dependiendo del valor enviado desde la App.

Fuente: Elaborado por el autor.

Una vez asignada el valor a la variable EstadoRele esta será comparada a través de la sentencia de control Switch, esta tendrá cuatro casos como son: 13000, 14000, 15000, y 16000, cada uno de estos casos tiene las sentencias de la Tabla 15:

Tabla 15: Función de cada uno de los estados de los relés.

Fuente: Elaborado por el autor.

13000	Apaga el relé del <i>Spindle</i> .
14000	Enciende el relé del <i>Spindle</i>
15000	Apaga el relé del <i>Shield</i>
16000	Enciende el relé del <i>Shield</i>

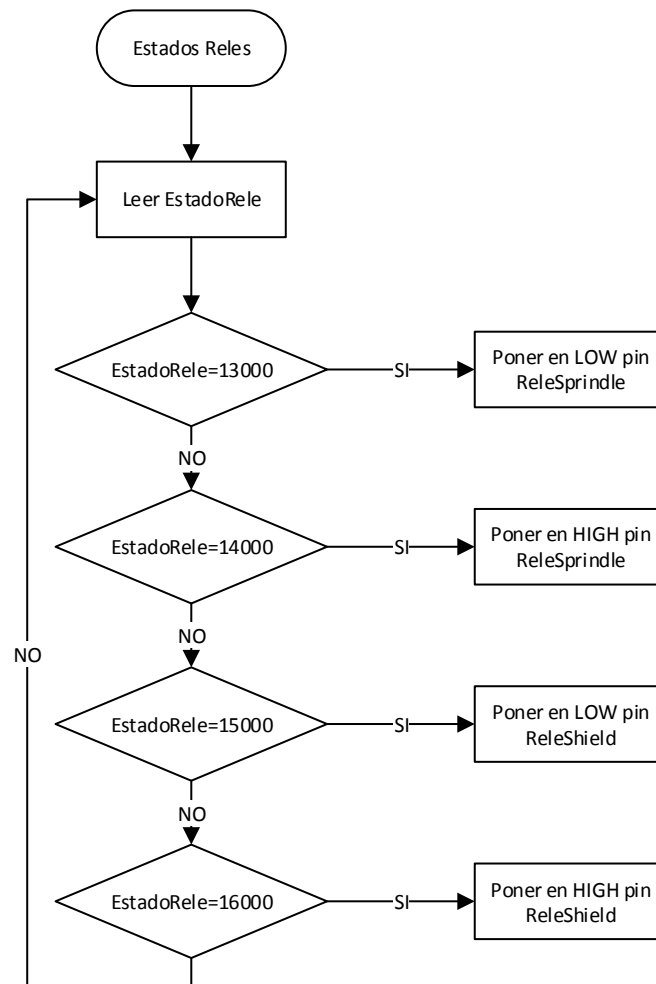


Figura 47: Diagrama de flujo de estado de los relés.

Fuente: Elaborado por el autor.

Escritura de datos

La escritura de datos permite la activación de las alarmas que se encuentra en la App del celular, para ello los ejes cuentan con fines de carrera en los dos límites, la explicación tomara como ejemplo el eje x, el límite máximo y mínimo del eje son conectados en el pin digital 4 y 5 respectivamente. Se utiliza un bucle de control if y se compara los estados de los pines antes nombrados, en el caso de que se encuentre uno de los dos en alto, enviara una señal al Serial del Bluetooth que será recibido por la App y mostrara una señal de alarma en el eje en el que se encuentra. Para el eje y e z son un símil del eje x.

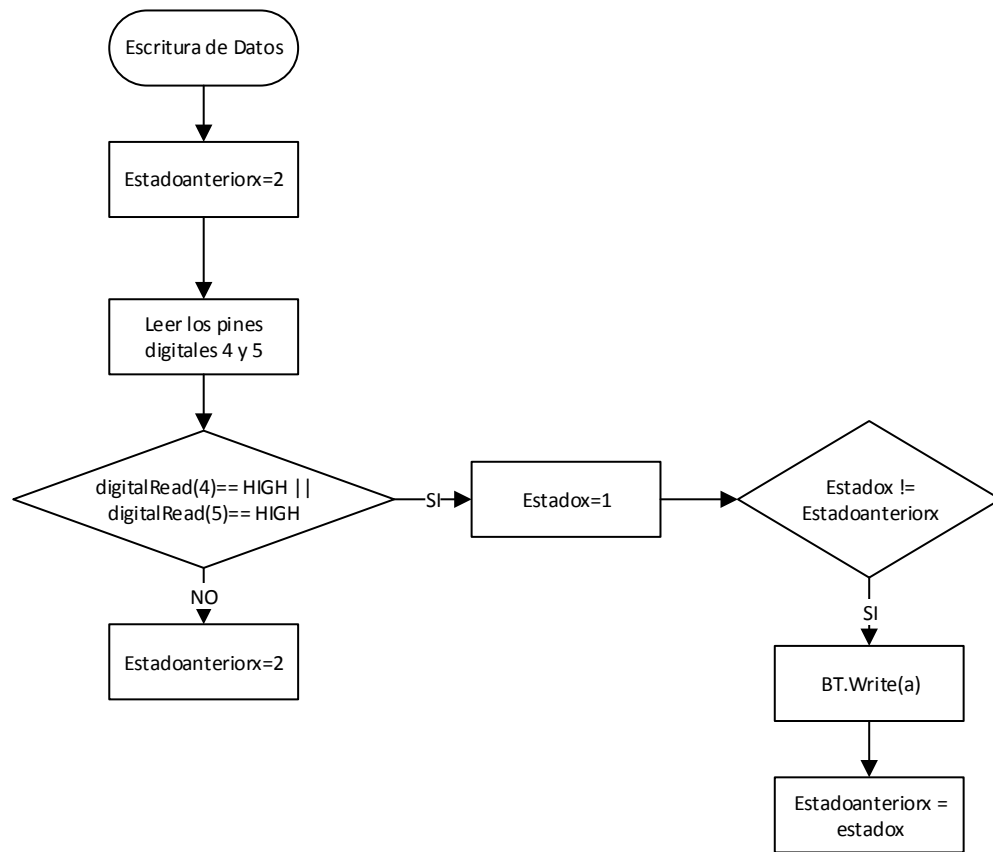


Figura 48: Diagrama de flujo de escritura de datos.

Fuente: Elaborado por el autor.

```

if(digitalRead(4)== HIGH || digitalRead(5)== HIGH)
{
  estadox=1;
  if(estadox != estadoanteriorx)
  {
    BT.write("a");
    Serial.println("a");
    estadoanteriorx=estadox;
  }
}
else
{
  estadoanteriorx=2;
}
  
```

Figura 49: Código para la escritura de datos en el serial del Arduino y el Bluetooth.

Fuente: Elaborado por el autor.

4.2.4 Desarrollo de cálculos en la implementación

Controlador A4988

Es necesario calibrar la intensidad de corriente a través del potenciómetro del driver, esto de acuerdo a la I_n del motor.

$$I = V_{ref} * 2.5$$

$$I = \text{Corriente (Amp)}$$

$$V_{ref} = \text{Voltaje de referencia (V)}$$

$$2.5 = \text{Constante}$$

Ecuación 1: Cálculo de intensidad de corriente del controlador A4988

Fuente: A4988-Datasheet.pdf

El voltaje de referencia se lo toma con un multímetro entre GND y el tornillo del potenciómetro, el motor nema utilizado tiene un I_n de 1,2 Amp.

$$V_{ref} = I / 2.5$$

$$V_{ref} = 1.2 \text{ Amp} / 2.5$$

$$V_{ref} = 0.48 \text{ V}$$

Ecuación 2: Cálculo de voltaje de referencia del controlador A4988

Fuente: A4988-Datasheet.pdf

Motores paso a paso.

Para el cálculo del torque necesario para los motores a paso se lo realizara por medio de las siguientes ecuaciones:

Tabla 16: Ecuaciones para el cálculo del torque requerido por los motores a pasos.

Fuente: Elaborado por el autor.

Función	Ecuación	Descripción
Torque	$T = 2 * I_o * \frac{w}{t} * \frac{\pi\theta}{180}$	<ul style="list-style-type: none"> • T: Torque Requerido • I_o = Carga Inercial

Continuación Tabla 16: Ecuaciones para el cálculo del torque requerido por los motores a pasos.

		<ul style="list-style-type: none"> • θ = Ángulo de paso • w = Velocidad • t = Tiempo para alcanzar la velocidad
Inercia Total	$I_o = I_{eq} + I_{tornillo} + I_{rotor}$	<ul style="list-style-type: none"> • I_{eq} = Inercia Equivalente • $I_{tornillo}$ = Inercia del Tornillo • I_{rotor} = Inercia del rotor
Inercia Equivalente	$I_{eq} = m * \frac{1}{p^2} * 0.025$	<ul style="list-style-type: none"> • m = Peso • p = Paso
Inercia del tornillo	$I_{tornillo} = D^4 * Longitud * 0.028$	<ul style="list-style-type: none"> • D = Diámetro del tornillo

Ecuación 3: Calculo de torque motores a pasos
Fuente: (Sandvik, 2013)

A continuación, en la Tabla 17 se observan los datos utilizados para el cálculo del torque de los motores de los ejes utilizados en la máquina:

Tabla 17: Datos motores a paso.

Fuente: Elaborado por el autor.

Unidades Equivalentes		
Denominación	Unidades	
	Sistema Internacional	Sistema Ingles
Diámetro	8 mm	0.32 in
Longitud	500 mm	19.69 in
Paso	2 mm	5 hilos/in

Tabla 18: Resultados del torque del motor a pasos.

Fuente: Elaborado por el autor.

Torque del Motor		
$I_{tornillo}$	0.005	$lb * in^2$
I_{eq}	0.001	$lb * in^2$

Continuación Tabla 18: Resultados del torque del motor a pasos.

I_{rotor}	0.1023	$lb * in^2$
w'	200	$pasos/s$
θ	1.8	$grados$
t	0.15	s
<i>Torque</i>	68,31	$oz * in$

Este valor de torque equivale en unidades del sistema internacional a un valor de 0.4 Nm.

Una vez conocido el valor del torque se puede realizar la selección de los motores para cada uno de los ejes.

Relación de velocidad del *Spindle* y el avance de la mesa

Con el fin de calcular una relación entre las revoluciones del motor del *Spindle* y el avance de la mesa se ha hecho uso de las siguientes ecuaciones:

- Potencia de Corte

$$pc = \frac{pa * pr * vf * kc}{60 * 1000}$$

Ecuación 4: Potencia de corte
Fuente: (Sandvik, 2013)

Donde:

pc = potencia de corte de fresado [W]

vf = velocidad de avance $\left[\frac{mm}{min}\right]$

kc = fuerza específica de corte $\left[\frac{N}{mm^2}\right]$

pa = profundidad axial de fresado [mm]

pr = profundidad radial de fresado [mm]

- Revoluciones del *Spindle*

$$Vf = f * n$$

Ecuación 5: Avance de mesa
Fuente: (Sandvik, 2013)

$$N = \frac{1000 * Vc}{\pi * d}$$

Ecuación 6: Velocidad del Spindle
Fuente: (Sandvik, 2013)

$N =$ velocidad de rotación del Spindle [rpm]

$Vc =$ velocidad corte superficial [m/min]

$d =$ diámetro de la herramienta [mm]

$f =$ Avance [mm/rev]

$Vf =$ Velocidad $\left[\frac{mm}{min}\right]$

Se utilizaron los siguientes datos como constante

Tabla 19: Datos de los motores a pasos

Fuente: elaborado por el autor.

Dato	Valor
Potencia Motor	400 [W]
Eficiencia	80%
Potencia de Corte	320 [W]
Profundidad axial de fresado [pa]	0.75 [mm]
Profundidad radial de fresado [pr]	0.15 [mm]
Kc	1400 [N/mm ²]

Se realizó la Tabla 20 de valores, se excluyó valores inferiores a 3000 rpm, puesto que en estos valores el avance supera los 40 mm/rev.

Tabla 20: Valores referenciales de velocidad**Fuente: Elaborado por el autor.**

N[rpm]	Vc[m/min]	Potencia de Corte [W]	f[mm/rev]
3000	9.42	320	40.63
3500	11.00	320	34.83
4000	12.57	320	30.48
4500	14.14	320	27.09
5000	15.71	320	24.38
5500	17.28	320	22.16
6000	18.85	320	20.32
6500	20.42	320	18.75
7000	21.99	320	17.41
7500	23.56	320	16.25
8000	25.13	320	15.24
8500	26.70	320	14.34
9000	28.27	320	13.54
9500	29.85	320	12.83
10000	31.42	320	12.19
10500	32.99	320	11.61
11000	34.56	320	11.08
11500	36.13	320	10.60
12000	37.70	320	10.16

Calibración de parámetros iniciales de la fresadora CNC a través de Universal Gcode Sender.

Una vez abierto el programa Universal Gcode Sender se ingresa en la ventana *Command*, los símbolos \$\$, posterior a este paso se despliega el listado de parámetros configurables de la Tabla 21, los cuales son modificables y necesarios calibrar de acuerdo a las características de cada elemento a usar.

La forma de variar los parámetros, por ejemplo, se tomará el valor $\$0=200.000$ (x, step/mm) el cual viene de fábrica, se coloca en la ventana *Command* el dato $\$0=25.000$ y se presiona enter, se vuelve a colocar en la ventana *Command* y se escribe \$\$ con lo cual se despliega nuevamente la lista de parámetros, pero ya con el valor modificado a continuación se detalla la identificación de cada uno de los parámetros y su función.

En los parámetros $\$0$, $\$1$, $\$2$, se calibra el número de pasos de los ejes X, Y, Z, necesarios para mover un milímetro. Los datos necesarios son:

- Motor: 1.8° x paso
- Tornillo
 - Desplazamiento por vuelta: 8mm
 - Numero de hilos: 4
- Un giro completo es 360° se calcula:
 - No. Pasos = $360^\circ/1.8^\circ = 200$ pasos
- Desplazamiento de bancada:
 - $200 \text{ pasos} / 8\text{mm} = 25 \text{ pasos/mm}$
 - Entonces:

$$25 \text{ pasos/mm} = 1\text{mm}$$

Ecuación 7: Cálculo número de pasos Universal Gcode Sender.
Fuente: Universal Gcode Sender

Esta prueba se realizó en los ejes X, Y, Z, quedando medido y calibrada la máquina tanto en teoría como en la práctica.

$\$0=25.000$ (x, step/mm)
 $\$1=25.000$ (y, step/mm)
 $\$2=25.000$ (z, step/mm)

El punto $\$4$ y $\$5$ es para sincronismo de velocidad y evitar que se pierda pasos de los motores.

$\$4=400.000$ (default feed, mm/min)
 $\$5=2000.000$ (default seek, mm/min)

En el punto $\$6$ se puede invertir los giros de los motores a través de los últimos dígitos.

$\$6=192$ (step port invert mask, int:11000000)

En el punto \$7 es el tiempo de espera de la máquina en la transición después de realizar una orden si es muy poco tiempo se pierde pasos y si es mucho tiempo la fresadora trabaja lento.

\$7=15 (step idle delay, msec)

El punto \$8 es para la aceleración de los motores, a baja velocidad existe una rampa muy grande de aceleración con lo cual se pierde torque al arranque, con las pruebas realizadas trabaja de 500 a 700 mm/sec² de manera adecuada.

\$8=700.000 (acceleration, mm/sec²)

Los puntos \$9, \$10, \$11 son para el control de tallado en las curvas o arcos de las pistas.

\$9=0.050 (junction deviation, mm)

\$10=0.100 (arc, mm/segment)

\$11=25 (n-arc correction, int)

El punto \$12 es para el numero de decimales que maneja la máquina.

\$12=3 (n-decimals, int)

El punto \$13 se refiere al reporte que da en mm o pulgadas.

\$13=1 (report inches, bool)

El punto \$14 se refiere a pulso de inicio puede ser interno (1) a través del Universal Gcode sender o un pulsador externo (0).

\$14=1 (auto start, bool)

El punto \$15 es para invertir el pin de enable del driver depende el driver que se utilice.

\$15=0 (invert step enable, bool)

El punto \$16 habilita o deshabilita el ingreso de finales de carrera de seguridad para que la máquina pare por límites de topes.

\$16=1 (hard limits, bool)

El punto \$17 se utiliza para hacer un home se calibra dependiendo los finales de carrera para el punto cero.

\$17=0 (homing cycle, bool)

Tabla 21: Parámetros para calibración de los movimientos de ejes.**Fuente: Elaborado por el autor.**

```

Grbl 0.8c ['$ for help]
>>> $$
$0=25.000 (x, step/mm)
$1=25.000 (y, step/mm)
$2=25.000 (z, step/mm)
$3=10 (step pulse, usec)
$4=400.000 (default feed, mm/min)
$5=2000.000 (default seek, mm/min)
$6=192 (step port invert mask, int:11000000)
$7=15 (step idle delay, msec)
$8=700.000 (acceleration, mm/sec^2)
$9=0.050 (junction deviation, mm)
$10=0.100 (arc, mm/segment)
$11=25 (n-arc correction, int)
$12=3 (n-decimals, int)
$13=1 (report inches, bool)
$14=1 (auto start, bool)
$15=0 (invert step enable, bool)
$16=1 (hard limits, bool)
$17=0 (homing cycle, bool)
$18=0 (homing dir invert mask, int:00000000)
$19=25.000 (homing feed, mm/min)
$20=250.000 (homing seek, mm/min)
$21=100 (homing debounce, msec)
$22=1.000 (homing pull-off, mm)
ok

```

En el punto \$18 se basa en una máscara la cual se puede invertir los finales de carrear a según la Tabla 22 para ser variados por el usuario.

\$18=0 (homing dir invert mask, int:00000000)

El punto \$19 y \$20 es la velocidad de retorno al punto cero cuando se presiona el botón Return to Zero y cuando termina de realizar el fresado.

\$19=25.000 (homing feed, mm/min)

\$20=250.000 (homing seek, mm/min)

El punto \$21 es la velocidad de home.

\$21=100 (homing debounce, msec)

El punto \$22 realiza un desplazamiento inverso después de regresar al punto de origen.

\$22=1.000 (homing pull-off, mm).

Tabla 22: Mascara para configuración finales de carrera e inversión de giro de motores.**Fuente: Elaborado por el autor.**

Configuración del valor	Mascara	Invertir X	Invertir Y	Invertir Z
0	00000000	N	N	N
1	00000001	Y	N	N
2	00000010	N	Y	N
3	00000011	Y	Y	N
4	00000100	N	N	Y
5	00000101	Y	N	Y
6	00000110	N	Y	Y
7	00000111	Y	Y	Y

Parámetros principales a calibrar en FlatCAM

Corte Z: La profundidad de la herramienta mientras se corta. -2 mils o -0.05 mm son un valor típico para enrutamiento de aislamiento.

Viaje Z: la altura sobre la tabla a la que se desplazará la herramienta de corte cuando no esté cortando cobre.

Velocidad de avance: la velocidad de la herramienta de corte mientras se corta en pulgadas / minuto o mm / minuto, según la configuración del proyecto.

Diámetro de la herramienta: El diámetro de la herramienta de corte. Utilice el mismo valor que cuando creó la geometría de enrutamiento de aislamiento en el paso 2.

Diámetro de la herramienta: el diámetro de la herramienta que se utilizará para cortar el área.

Superposición: Fracción del diámetro de la herramienta por la que se solapa cada corte que pasa. El valor predeterminado de 0.15 es el mínimo para garantizar que no quede cobre en los giros de 90 grados de la trayectoria de la herramienta.

Margen: Distancia para que la herramienta se mantenga alejada del límite del polígono. Esto se puede usar para asegurar que una herramienta grande no toque los bordes de cobre que tienen o serán cortados por una herramienta más pequeña y precisa.

Velocidad del Spindle: (Opcional) Velocidad del *Spindle* en revoluciones por minuto (RPM).

4.2.5 PRUEBAS DE FUNCIONAMIENTO.

Se realizó pruebas de la fresadora CNC de forma local y remota con resultados óptimos entre los cuales están:

Operación local

La operación local se realizó con pulsos desde el programa Univesal GCode Sender previo debe estar instalada la librería Grbl en el Arduino UNO, se realizó pruebas de movimiento en milímetros por las dimensiones de la fresadora se controla los ejes X, Y, Z, para poder colocar en el punto cero de trabajo o generar un nuevo punto de inicio.

Se realiza el enlace Bluetooth con una conexión óptima y un tiempo de respuesta instantáneo tanto de la regulación de velocidad del PWM para el *Spindle*, el control de los relés de activación de los módulos CNC *Shield* y accionamiento de la tarjeta del *Spindle* y el control de las alarmas de la misma manera se observa las alertas instantáneas por fallos forzados, refiriéndonos a esta información al manual de usuario generado.

Es necesario tomar en cuenta que en la CNC nos toma un tiempo en la preparación de la máquina y todo el programa y demás para poder ejecutar un proyecto vendría siendo la preproducción, pero al hablar de realizar unas 10000 placas se ve la respuesta al momento de ejecutar un trabajo con exactitud, que no tendríamos al realizarlas manualmente.

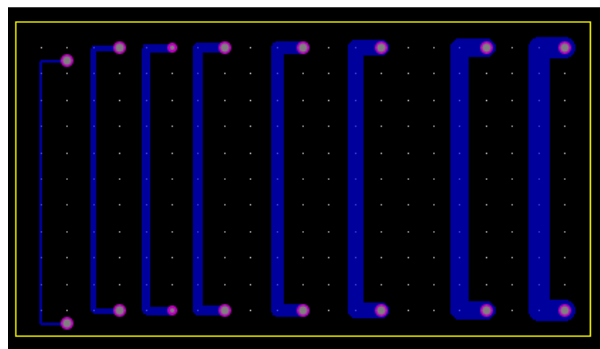


Figura 50: Diseño pistas de prueba en Proteus.

Fuente: Elaborado por el autor.

Como se puede observar en la Figura 50 y Figura 51 se realizó una prueba de dimensiones de pistas realizadas en Proteus y el acabado final con la fresadora comparando dimensiones, precisión y acabado, obteniendo resultados satisfactorios.



Figura 51: PCB obtenida de la fresadora CNC.

Fuente: Elaborado por el autor.

La velocidad de trabajo de la máquina en remoto para los ejes X, Y, Z, se calibra en el programa FlatCAM en mm/minuto, cabe recalcar que este valor se lo asume teniendo en cuenta las herramientas de corte a usarse como la fresa y la broca, es necesario realizar pruebas para ver el acabado ya que según tablas da un valor, pero el resultado final no es el óptimo y es necesario variar la velocidad según las necesidades del usuario.

Los tiempos de aceleración y la velocidad máxima asignada a cada motor a pasos se lo calibra en el programa Universal Gcode Sender, lo cual implica realizar pruebas de acabado y exactitud en perforación, se varia en los parámetros \$4=XX, \$5=XX, \$8=XX.

Operación remota

Para el control y operación remota de las seguridades, activación tarjetas y control PWM en el manual de usuario se puede guiar de manera adecuada el usuario en detalle, partiendo con el circuito ya diseñado en CAD hasta obtener el producto final que es la placa de baquelita cortada para el proceso de suelda que no aplica en este proyecto.

4.2.6 ANÁLISIS DE RESULTADOS.

El análisis realizado contempla como primicia la precisión del acabado con lo cual se realizó varias pruebas de fresado, teniendo incluso que cambiar de *software* ya que existía variación en medias finales y los elementos no se acoplaban de manera correcta.

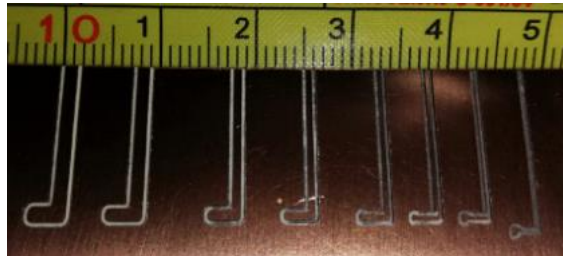


Figura 52: Dimensiones de pistas obtenidas.

Fuente: Elaborado por el autor.

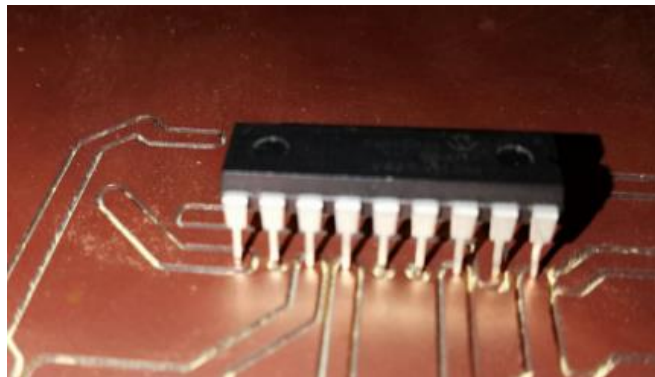


Figura 53: Dimensiones entre placa e integrado.

Fuente: Elaborado por el autor.

Las placas obtenidas muestran que la precisión de las medidas de los integrados y demás elementos calzan perfectamente como se observa en la Figura 53.

Las alertas de las alarmas para los ejes X, Y, Z, funcionan de forma adecuada tanto en el Universal Gcode Sender, que detiene la máquina por la activación de cualquiera de los micros y la visualización de las alertas en el Smartphone son de manera inmediata.

Alarmas		Alarmas		Alarmas	
Eje X	Error X	Eje X		Eje X	
Eje Y		Eje Y	Error Y	Eje Y	
Eje Z		Eje Z		Eje Z	Error Z

Figura 54: Visualización de alarmas en Smartphone

Fuente: Elaborado por el autor

CONCLUSIONES

Una vez realizado el análisis de funcionamiento y especificaciones de una fresadora CNC para PCBs, se logró identificar los elementos necesarios para el diseño y construcción de la misma, obteniendo la adaptación mecánica y electrónica de cada dispositivo, permitiendo alcanzar resultados óptimos en el proceso de funcionamiento tanto en precisión como en control.

- En cuanto al circuito de la tarjeta electrónica se realizó en el programa Proteus 8, conjuntamente con las librerías de Arduino efectuando simulaciones satisfactorias, y obteniendo un diseño compacto y funcional acorde a las especificaciones técnicas, seleccionando componentes electrónicos que interactúen de forma adecuada, facilitando así la implementación de los elementos periféricos hacia la tarjeta electrónica, tanto el control PWM como las seguridades de la fresadora.
- El diseño de la fresadora CNC en AutoCAD ayudo para la construcción de las partes mecánicas, identificando los elementos estructurales principales con lo cual se garantizó rigidez, obteniendo exactitud y precisión en las placas de PCB.
- Ahora bien, para el enlace de control mediante el Arduino UNO y los motores a pasos se utilizó el *driver* A4988 el cual es compatible con la tecnología de Arduino ya que el tiempo de respuesta es inmediato, además es de bajo costo y excelente calidad.
- Para el desarrollo de una aplicación Android para la interfaz de comunicación vía Bluetooth con la cual se gestiona seguridades, velocidad del *Spindle*, control de encendido, la cual facilita la comunicación con el usuario, y facilita la interacción con la fresadora, reduciendo tiempo y costos al momento de trabajar con el equipo.
- Para concluir en relación a validar el sistema mediante pruebas de funcionamiento, se corrigió los fallos tanto en movimientos mecánicos, programas que no coordinaban, y se realizó pruebas con diferentes circuitos obteniendo una alta eficiencia al momento de trabajar, en los diseños en PCBs.

RECOMENDACIONES

- La fresadora se debe utilizar exclusivamente para la fabricación de circuitos PCBs con esto se garantiza su correcto funcionamiento evitando daños prematuros en la fresa y broca.
- La limpieza de los residuos de cobre desbastado se lo debe recoger con una brocha o ser aspirados para su almacenamiento y posterior desecho.
- La fresadora debe ser operada por personal capacitado y bajo ninguna circunstancia personal ajeno al técnico debe manipular los controles y sistema eléctrico o mecánico de la máquina.
- Si la máquina no responde a los comandos vía celular se debe reiniciar el mismo ya que existen equipos con falencias de actualizaciones que no podrían trabajar de manera adecuada.
- Los motores a pasos al ser dispositivos mecánicos deben tener una coordinación precisa de los pulsos aplicados entre frecuencia y tiempo, a través del controlador A4988 y el Arduino que no exista una descoordinación entre la secuencia de pulsos, teniendo una serie de inconvenientes tales como:
 - Realiza giros erráticamente
 - Comienza a vibrar sin girar
 - Se queda estático
 - Tiene avances y retrocesos sin control
- Cuando se utilice el modo de control PWM en la tarjeta de control del motor DC *Spindle* se debe siempre mantener desconectado el potenciómetro o no se habilitará el modo PWM.

BIBLIOGRAFÍA

- Alonso, D., Gil, J., & Martinez, F. (3 de Noviembre de 2014). *revistas.udistrital*. Obtenido de <https://revistas.udistrital.edu.co/ojs/index.php/tekhne/article/viewFile/10439/11456>
- Arduino, H. (s.f.). *Arduino.cc*. Obtenido de <https://www.arduino.cc/>
- Berger, L. (s.f.). *Formulas+Calculations for Optimum Selection of a S tep Motor*. Obtenido de http://www.esteca55.com.ar/downloads/Selection_PaP1-1.pdf
- Garcia, E. (2018). *Compilador C CCS y Simulador Proteus para Microcontroladores PIC*. Mexico: Alfaomega.
- Geoff, W. (2003). *CNC Robotics*. New York: McGraw-Hill.
- Hussam, A. A., Ghassan, A.-K., & Hussien , Z. (8 de Septiembre de 2011). *Towards computer vision feedback for enhanced CNC machining*. Obtenido de <http://ieeexplore.ieee.org/document/6015000/>
- Industria &, E. (16 de Julio de 2014). *Historia y evolución del CNC*. Obtenido de <http://www.industriasyempresas.com.ar/node/2088>
- Modular, S. (s.f.). *3DMS Modular System*. Obtenido de <http://3dmodularsystems.com/en/motors/47-5-nema-17-stepper-motor-for-scalar-family-3d-printers.html>
- Smid, P. (2003). *CNC programming handbook*. New York: EUA: Industrial Press Inc.
- 14:00-17:00. (s/f). ISO 6983-1:2009. Recuperado el 7 de enero de 2019, de <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/03/46/34608.html>
- Alternatives to Gerber RS-274-X. (s/f). Recuperado el 7 de enero de 2019, de <http://www.epdtonthenet.net/article/124342/Alternatives-to-Gerber-RS-274-X.aspx>
- Arduino CNC Shield V3 - VISTRONICA SAS. (s/f). Recuperado el 7 de enero de 2019, de <https://www.vistronica.com/board-de-desarrollo/arduino/arduino-cnc-shield-v3-detail.html>
- Arduino Uno Rev3. (s/f). Recuperado el 7 de enero de 2019, de <https://store.arduino.cc/usa/arduino-uno-rev3>
- Comunicación de Arduino con puerto serie. (s/f). Recuperado el 7 de enero de 2019, de <https://www.luisllamas.es/arduino-puerto-serie/>

Fichier:Arduino uno Pinout (paysage).png — MCHobby - Wiki. (s/f). Recuperado el 7 de enero de 2019, de [https://wiki.mchobby.be/index.php?title=Fichier:Arduino_uno_Pinout_\(paysage\).png](https://wiki.mchobby.be/index.php?title=Fichier:Arduino_uno_Pinout_(paysage).png)

Garcia, V. (s/f). motor bipolar con Arduino, control motor bipolar. Recuperado el 7 de enero de 2019, de <http://hispavila.com/total/3ds/atmega/motorbipolar.html>

HC-05 Bluetooth Module Pinout, Specifications, Default Settings, Replacements & Datasheet. (s/f). Recuperado el 7 de enero de 2019, de </wireless/hc-05-bluetooth-module>

jecrespom. (2016, noviembre 9). UART y USB en Arduino. Recuperado el 7 de enero de 2019, de <https://aprendiendoarduino.wordpress.com/2016/11/09/uart-y-usb-en-arduino/>

Jose Mecanico. (09:52:25 UTC). *Programacion control numerico por computadora*. Educación. Recuperado de <https://es.slideshare.net/josemecanico/programacion-control-numericopor-computadora>

SAE - EIA RS-274 - INTERCHANGEABLE VARIABLE BLOCK DATA FORMAT FOR POSITIONING, CONTOURING, AND CONTOURING/POSITIONING, NUMERICALLY CONTROLLED MACHINES | Engineering360. (s/f). Recuperado el 7 de enero de 2019, de <https://standards.globalspec.com/std/10021063/sae-eia-rs-274>

SureStepr SD82B Stepper driver. (s/f). Recuperado el 7 de enero de 2019, de http://www.panucatt.com/product_p/sd82b.htm

Using a Dedicated Pulse Control LSI vs. a CPU for Motion Control | Nippon Pulse - Your Partner in Motion Control - Radford, VA. (s/f). Recuperado el 7 de enero de 2019, de <https://www.nipponpulse.com/articles/view/using-a-dedicated-pulse-control>

Flat CAM org- manual procedures, FlatCAM's documentation, <http://flatcam.org/manual/procedures.html>

Manual mecanizado Sandvik fresadora, Formulas y definiciones de fresado, <https://www.sandvik.coromant.com/es-es/knowledge/machining-formulas-definitions/pages/milling.aspx>

ANEXOS

ANEXO 1: Programación App Inventor 2

```

cuando Screen1 . BotónAtrás
ejecutar
  si
    poner ControlAlarmas . Visible como cierto
  entonces
    poner ControlAlarmas . Visible como falso
    poner PantallaPrincipal . Visible como cierto
  si no, si
    poner Información . Visible = cierto
  entonces
    poner Información . Visible como falso
    poner PantallaPrincipal . Visible como cierto
  si no, si
    poner ControlPrincipal . Visible = cierto
  entonces
    poner ControlPrincipal . Visible como falso
    poner PantallaPrincipal . Visible como cierto

```

```

cuando Dispositivos . AntesDeSelección
ejecutar poner Dispositivos . Elementos como BluetoothClient1 . DireccionesYNombres

```

```

cuando Dispositivos . DespuésDeSelección
ejecutar poner Dispositivos . Selección como llamar BluetoothClient1 . Conectar
                                     dirección Dispositivos . Selección

```

```

cuando VAlarmas . Clic
ejecutar poner ControlAlarmas . Visible como cierto
        poner PantallaPrincipal . Visible como falso

```

```

cuando VInformación . Clic
ejecutar poner Información . Visible como cierto
        poner PantallaPrincipal . Visible como falso

```

```

cuando VControlPrincipal . Clic
ejecutar poner ControlPrincipal . Visible como cierto
        poner PantallaPrincipal . Visible como falso

```

```

cuando Slider1 . PosiciónCambiada
  posiciónDelPulgar
ejecutar poner Velocidad . Texto como Slider1 . PosiciónDelPulgar

```

```

cuando Enviar . Clic
ejecutar
  si
    BluetoothClient1 . Conectado
  entonces
    llamar BluetoothClient1 . EnviarTexto
    texto unir " A "
    Slider1 . PosiciónDelPulgar

```

```

inicializar global valorAlarma como " 0 "

```

```

inicializar global valorAlarma2 como " 0 "

```



```
cuando Clock1 . Temporizador
ejecutar
  si BluetoothClient1 . Conectado
  entonces
    poner CheckBox1 . Verificado como cierto
    si llamar BluetoothClient1 . BytesDisponiblesParaRecibir > 0
    entonces
      poner global valorAlarma a llamar BluetoothClient1 . RecibirTexto
      poner Valor . Texto como tomar global valorAlarma
      si comparar textos Valor . Texto = " a "
      entonces
        poner AlarmaX . Visible como cierto
        poner AlarmaZ . Visible como falso
        poner AlarmaY . Visible como falso
      si no, si comparar textos Valor . Texto = " b "
      entonces
        poner AlarmaZ . Visible como falso
        poner AlarmaX . Visible como falso
        poner AlarmaY . Visible como cierto
      si no, si comparar textos Valor . Texto = " c "
      entonces
        poner AlarmaY . Visible como falso
        poner AlarmaZ . Visible como cierto
        poner AlarmaX . Visible como falso
      si no, si comparar textos Valor . Texto = " d "
      entonces
        poner AlarmaX . Visible como falso
        poner AlarmaZ . Visible como falso
        poner AlarmaY . Visible como falso
    sino
      poner CheckBox1 . Verificado como falso
```

```
cuando ONSprindle . Clic
ejecutar
  si BluetoothClient1 . Conectado
  entonces
    llamar BluetoothClient1 . EnviarTexto
    texto " A14000 "
```

```
cuando OFFSprindle . Clic
ejecutar
  si BluetoothClient1 . Conectado
  entonces
    llamar BluetoothClient1 . EnviarTexto
    texto " A13000 "
```

```
cuando ONCNC . Clic
ejecutar
  si BluetoothClient1 . Conectado
  entonces
    llamar BluetoothClient1 . EnviarTexto
    texto " A16000 "
```

```
cuando OFFCNC . Clic
ejecutar
  si BluetoothClient1 . Conectado
  entonces
    llamar BluetoothClient1 . EnviarTexto
    texto " A15000 "
```

```
cuando Desconectar . Clic
ejecutar
  llamar BluetoothClient1 . Desconectar
```

ANEXO 2: Programa Arduino NANO.

```

#include <SoftwareSerial.h> // Incluimos la librería SoftwareSerial
SoftwareSerial BT(10,11);

int velocidad;
int valorpwm;
int pwmPIN = 3;

int estadoanteriorx=2;
int estadox=0;
int estadoanteriory=2;
int estadoy=0;
int estadoanteriorz=2;
int estadoz=0;
int estadoanteriorapagado=2;
int estadoapagado=0;

int EstadoRele;
int ReleSpindle = 12; //Activa rele para tarjeta Spindle
int ReleShield = 13; //Activa rele para tarjeta CNC Shield

void setup() {
  // put your setup code here, to run once:
  //BLUETOOTH
  BT.begin(9600); // Inicializamos el puerto serie BT (Para Modo AT 2)
  Serial.begin(9600);

  //MICROS
  for (int i=4; i<=9 ; i++)
  {
    pinMode(i,INPUT);
  }

  //RELES
  pinMode(ReleSpindle, OUTPUT);
  pinMode(ReleShield, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.flush();
  //Lectura del dato recibido por el Bluetooth
  if (BT.available())
  {
    velocidad = BT.parseInt();
    Serial.println(velocidad);
  }
  //Escritura de la velocidad PWM
  if(velocidad >= 0 && velocidad <=12000)
  {
    valorpwm = map(velocidad,0,12000,0,255);
  }
}

```



```

    analogWrite(pwmPIN,valorpwm);
}
else
{
    //Estado de los redes de activación
    EstadoRele = velocidad;
}

//Encendido de los reles de activación del control principal
switch (EstadoRele)
{
    case 13000:
        digitalWrite(ReleSpindle,LOW);
        Serial.println("Rele Sprindle Apagado");
        break;
    case 14000:
        digitalWrite(ReleSpindle,HIGH);
        Serial.println("Rele Sprindle Encendido");
        break;
    case 15000:
        digitalWrite(ReleShield,LOW);
        Serial.println("Rele Shield Apagado");
        break;
    case 16000:
        digitalWrite(ReleShield,HIGH);
        Serial.println("Rele Shield Encendido");
        break;
    default:
        break;
}

//MICROS

if(digitalRead(4)== HIGH || digitalRead(5)== HIGH)
{
    estadox=1;
    if(estadox != estadoanteriorx)
    {
        BT.write("a");
        Serial.println("a");
        estadoanteriorx=estadox;
    }
}
else
{
    estadoanteriorx=2;
}
if(digitalRead(6)== HIGH || digitalRead(7)== HIGH)
{
    estadoy=1;

```

```

    if (estadoy != estadoanteriory)
    {
        BT.write("b");
        Serial.println("b");
        estadoanteriory=estadoy;
    }
}
else
{
    estadoanteriory=2;
}

if(digitalRead(8)== HIGH || digitalRead(9)== HIGH)
{
    estadoz=1;
    if (estadoz != estadoanteriorz)
    {
        BT.write("c");
        Serial.println("c");
        estadoanteriorz=estadoz;
    }
}
else
{
    estadoanteriorz=2;
}


if(digitalRead(4)== LOW && digitalRead(5)== LOW && digitalRead(6)== LOW
&& digitalRead(7)== LOW && digitalRead(8)== LOW && digitalRead(9)== LOW)
{
    estadoapagado=1;
    if (estadoapagado != estadoanteriorapagado)
    {
        BT.write("d");
        Serial.println("d");
        estadoanteriorapagado=estadoapagado;
    }
}
else
{
    estadoanteriorapagado=2;
}
}

```







ANEXO 3: Proformas

Proforma 2

Commercial invoice

		Consignment No 9261299992399326430823		
Buyer name <u>maria Cano</u> 6117 NW 72nd Ave Miami FL 33195-3705 United States		Exporters Reference During two years warranty time, if it has the problem, we will send you the replacement to replace it. Mostly it will be resolve. Finally, if the problem still can not be resolved, then we will send you a new one instead.		
Buyer phone 3058852077		seller phone		
Destination City Name United States	Origin of Goods Germany	Net Mass 5.07	Gross Mass 5.07	
Item	Description of Good	Qty, KG	Unit Value	Total
1	<u>CNC 400W Brushless Spindle Motor ER11&driver speed controller&Mount engraving</u>	1	EUR144.91	USD115.9
		TOTAL	USD115.9	
 Chop Company		Name of Authorised Signatory: <u>Sanven Industrial Co., Ltd.</u>		
Date of Issue: 2017\12\10				

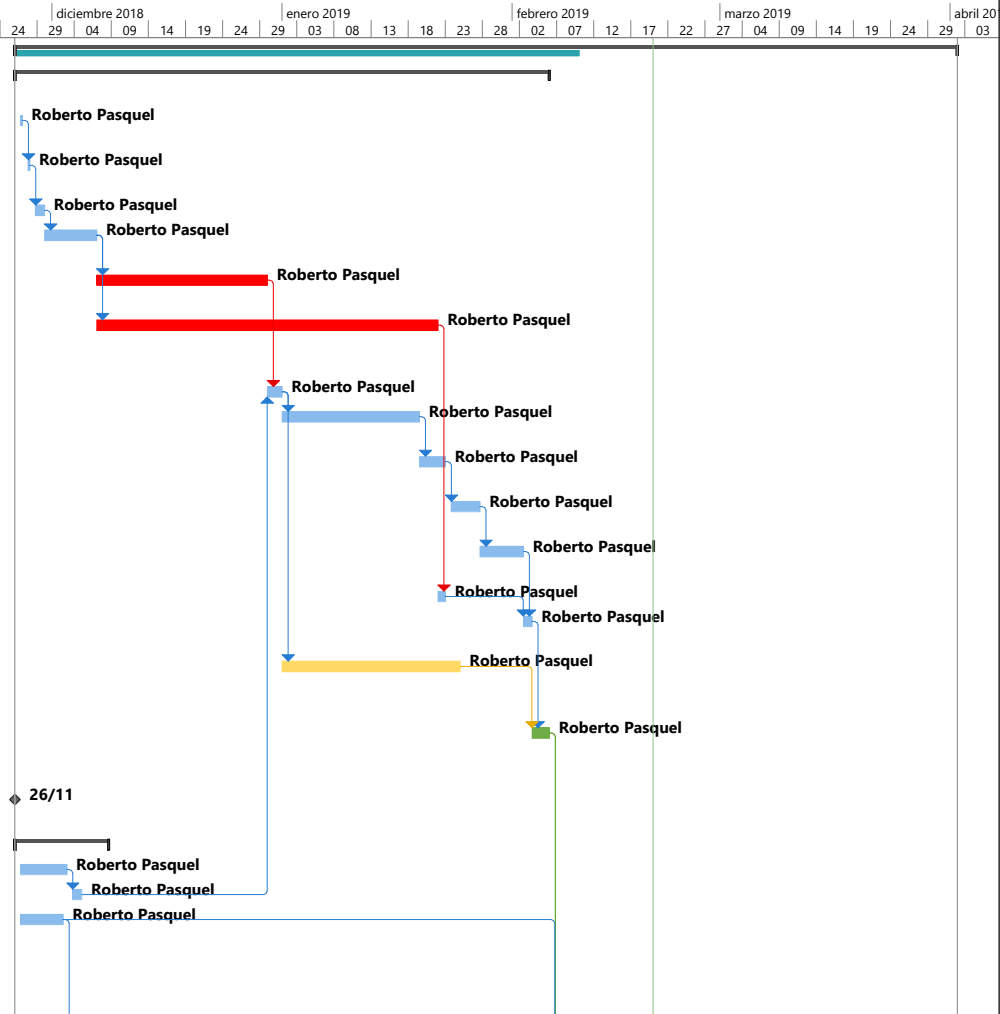
Proforma 4

ElectroDrone			
Direccion:Ambato, Avenida Cevallos y Mera, Atrás del Edificio de la Asociacion de Empleados			
<p>Email:electrodrone_ec@hotmail.com Skype:david_am_ec@hotmail.com Celular/Whatsapp:0984701518 Facebook:electrodrone180</p>			
Codigo	Image	Description	Precio
M001		Arduino UNO R3 + cable usb	\$ 12.00
M002		Arduino Mega R3 + cable usb	\$ 15.50
M003		Arduino Nano V3.0 SIN cable usb	\$ 8.00
P001		Tornillo 8 mm 4 Hilos 300mm Para Cnc Impresora 3d + Tuerca 15cm/20cm/30cm/60cm	14/18/28/38
C001		HC-05 4pin Bluetooth Module(Master+Slave) 0 Stock	\$ 7.00
C002		HC-06 Bluetooth	\$ 9.00

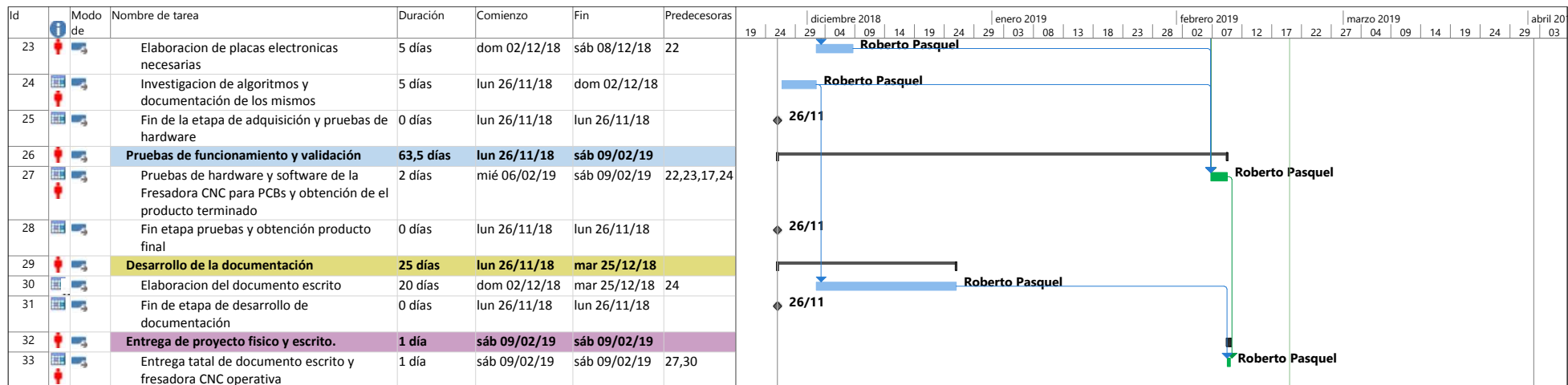
P002		Acople Flexible para tubos de 8mm Entrada de 5mm Diámetro:19mm Altura:25mm	\$ 4.00
P003		Soporte tubos de 8mm	\$ 4.00
P004		Rodamiento Lineal Tubos de 12mm	\$ 4.00
P005		Rodamiento Lineal 8mm Con Agujeros para tornillo	\$ 7.00
P006		Rodamiento tubo 8mm	\$ 4.00
S001		Shield CNC V3	\$ 10.00

ANEXO 4: Cronograma de actividades.

Id	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	diciembre 2018							enero 2019							febrero 2019							marzo 2019							abril 20						
							19	24	29	04	09	14	19	24	29	03	08	13	18	23	28	02	07	12	17	22	27	04	09	14	19	24	29	03							
1		Cronograma PIC Roberto Pasquel	109,63 días	lun 26/11/18	lun 01/04/19																																				
2		Establecer los dispositivos de control para el diseño	61,5 días	lun 26/11/18	mié 06/02/19																																				
3		Investigar los dispositivos eléctricos y electrónicos a utilizar	6 horas	lun 26/11/18	mar 27/11/18																																				
4		Investigar los dispositivos mecánicos a utilizar	6 horas	mar 27/11/18	mié 28/11/18	3																																			
5		Definir el microcontrolador a utilizar	8 horas	mié 28/11/18	jue 29/11/18	4																																			
6		Cotizar los materiales eléctricos electrónicos y mecánicos	6 días	jue 29/11/18	jue 06/12/18	5																																			
7		Adquisición de dispositivos de compra local	20 días	jue 06/12/18	sáb 29/12/18	6																																			
8		Adquisición de dispositivos de importación (Spindle -Husillo de perforación)	40 días	jue 06/12/18	lun 21/01/19	6																																			
9		Pruebas de los dispositivos eléctricos.	2 días	dom 30/12/18	lun 31/12/18	7,21																																			
10		Ensamble de la estructura y soportes mecanicos	15 días	lun 31/12/18	sáb 19/01/19	9																																			
11		Perforacion y ensamble de los motores de paso en las bases.	4 días	sáb 19/01/19	mié 23/01/19	10																																			
12		Alineacion y ajustes de los rodamientos lineales con las bases de soporte	4 días	mié 23/01/19	dom 27/01/19	11																																			
13		Alineacion y ajuste de los tornillos sin fin con las bases de soporte	4 días	dom 27/01/19	sáb 02/02/19	12																																			
14		Revision y pruebas del Spindle	4 horas	lun 21/01/19	mar 22/01/19	8																																			
15		Montaje y alineacion del Spindle en la estructura principal	2 días	sáb 02/02/19	dom 03/02/19	13,14																																			
16		Corrección de fallos y rectificación o cambio de elementos con fallas o no compatibles al proyecto.	20 días	lun 31/12/18	jue 24/01/19	9																																			
17		Realizar pruebas de funcionamiento de los mecanismos y dispositivos eléctricos, con carga.	2 días	dom 03/02/19	mié 06/02/19	15,16																																			
18		Fin de la etapa de diseño e implementación	0 días	lun 26/11/18	lun 26/11/18																																				
19		Definir la interfaz y software	10 días	lun 26/11/18	sáb 08/12/18																																				
20		Descargar los programas necesarios	6 días	lun 26/11/18	dom 02/12/18																																				
21		Descarga librerías necesarias	10 horas	lun 03/12/18	mar 04/12/18	20																																			
22		Diseño de un circuito para pruebas ver viabilidad entre Tarjeta shield de arduino (puente h) o elaboracion de tarjeta puente h, y pruebas en protoboard.	5 días	lun 26/11/18	dom 02/12/18																																				



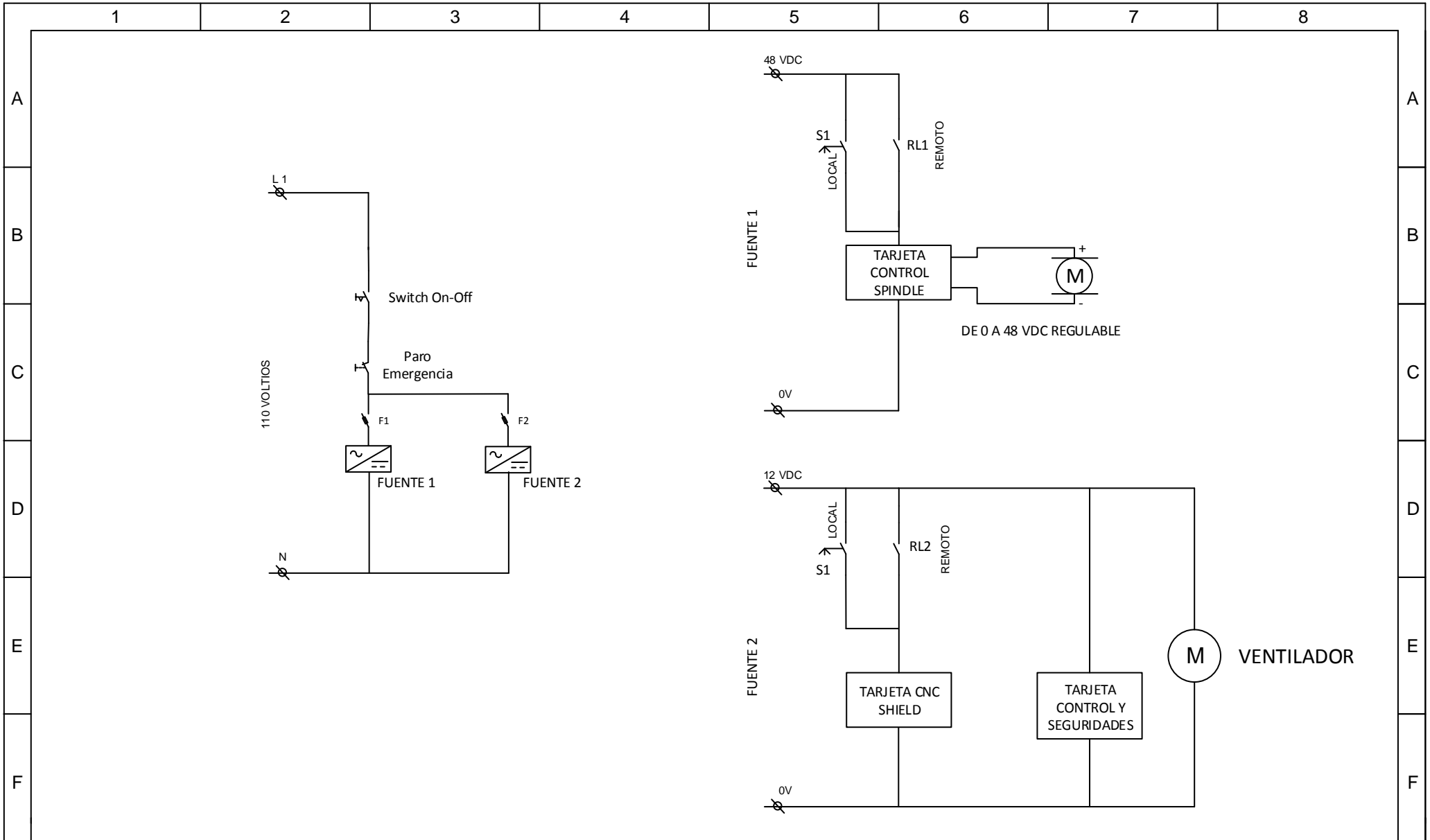
Proyecto: Proyecto1 Fecha: mié 20/02/19	Tarea		Resumen del proyecto		Tarea manual		solo el comienzo		Fecha límite	
	División		Tarea inactiva		solo duración		solo fin		Progreso	
	Hito		Hito inactivo		Informe de resumen manual		Tareas externas		Progreso manual	
	Resumen		Resumen inactivo		Resumen manual		Hito externo			



Proyecto: Proyecto1 Fecha: mié 20/02/19	Tarea		Resumen del proyecto		Tarea manual		solo el comienzo		Fecha límite	
	División		Tarea inactiva		solo duración		solo fin		Progreso	
	Hito		Hito inactivo		Informe de resumen manual		Tareas externas		Progreso manual	
	Resumen		Resumen inactivo		Resumen manual		Hito externo			

ANEXO 5: Planos esquema eléctrico de la CNC.

.



UNIVERSIDAD TECNOLÓGICA
ISRAEL

CONTIENE: DIAGRAMA ELÉCTRICO CNC PCB

UNIVERSIDAD TECNOLÓGICA ISRAEL

PROYECTO: TESIS

APROB: Mg. Rene Cortijo

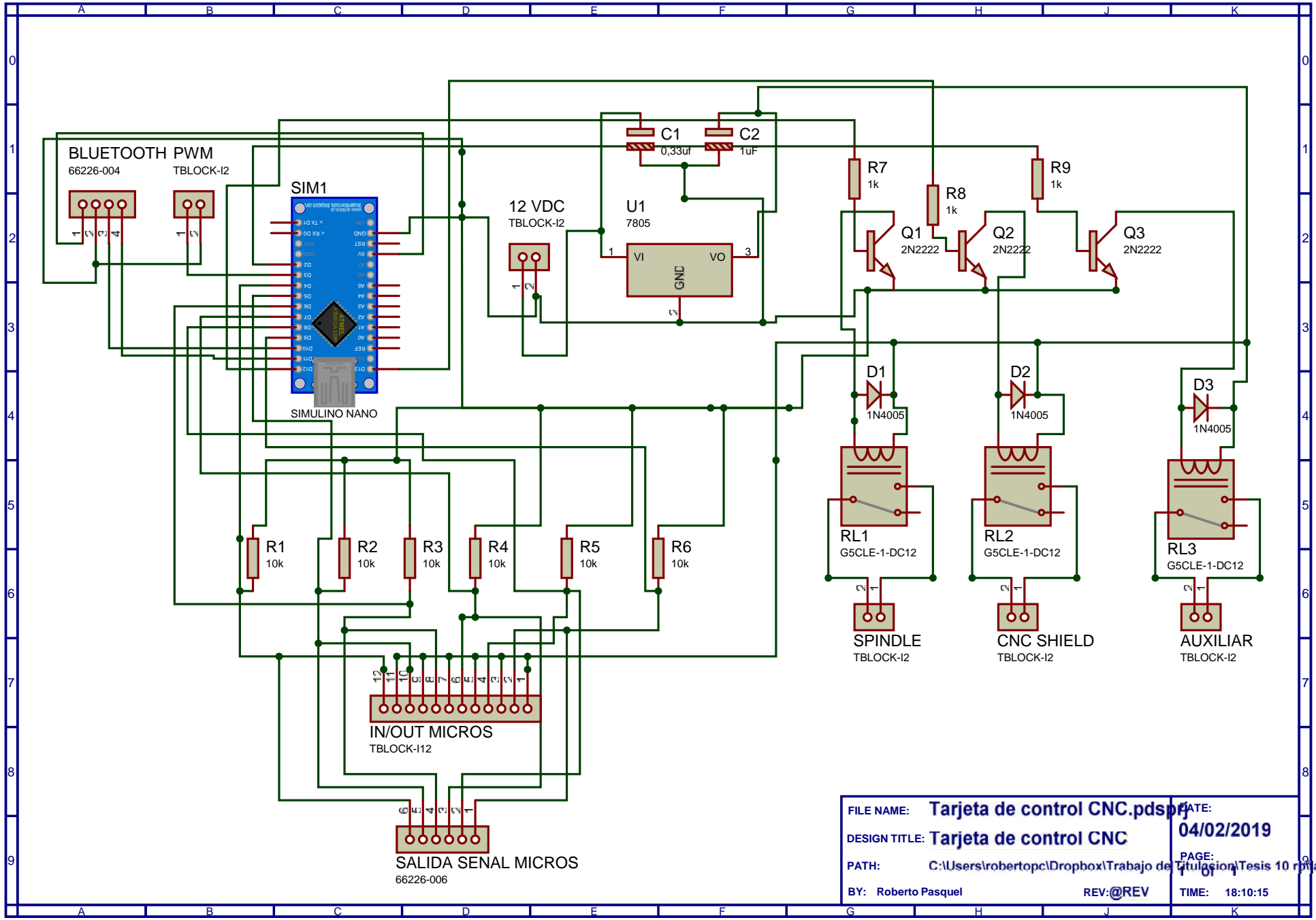
DIBUJADO POR: R. Pasquel.

REALIZADO POR:
R. Pasquel

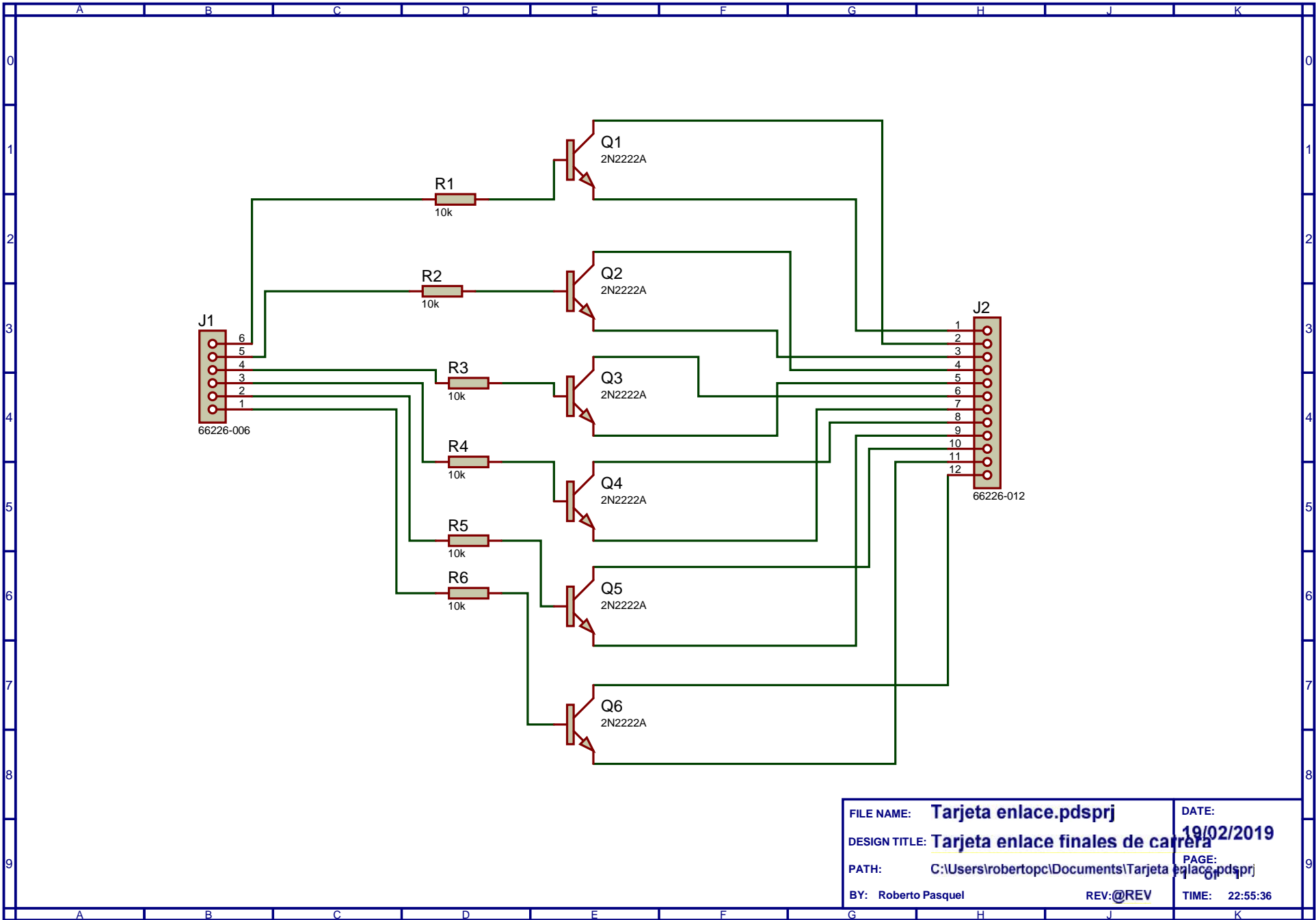
FECHA: 20/01/2019

LAMINA Nº 1

ANEXO 6: Planos esquema electrónico de la CNC.



FILE NAME:	Tarjeta de control CNC.pdsprj	DATE:	04/02/2019
DESIGN TITLE:	Tarjeta de control CNC	PAGE:	1 of 1
PATH:	C:\Users\robertopc\Dropbox\Trabajo de Titulacion\Tesis 10 roba	TIME:	18:10:15
BY:	Roberto Pasquel	REV:	@REV



FILE NAME: Tarjeta enlace.pdsprj	DATE: 19/02/2019
DESIGN TITLE: Tarjeta enlace finales de carrera	PAGE: 1
PATH: C:\Users\robertopc\Documents\Tarjeta enlace.pdsprj	REV: @REV
BY: Roberto Pasquel	TIME: 22:55:36

1

2

3

4

5

6

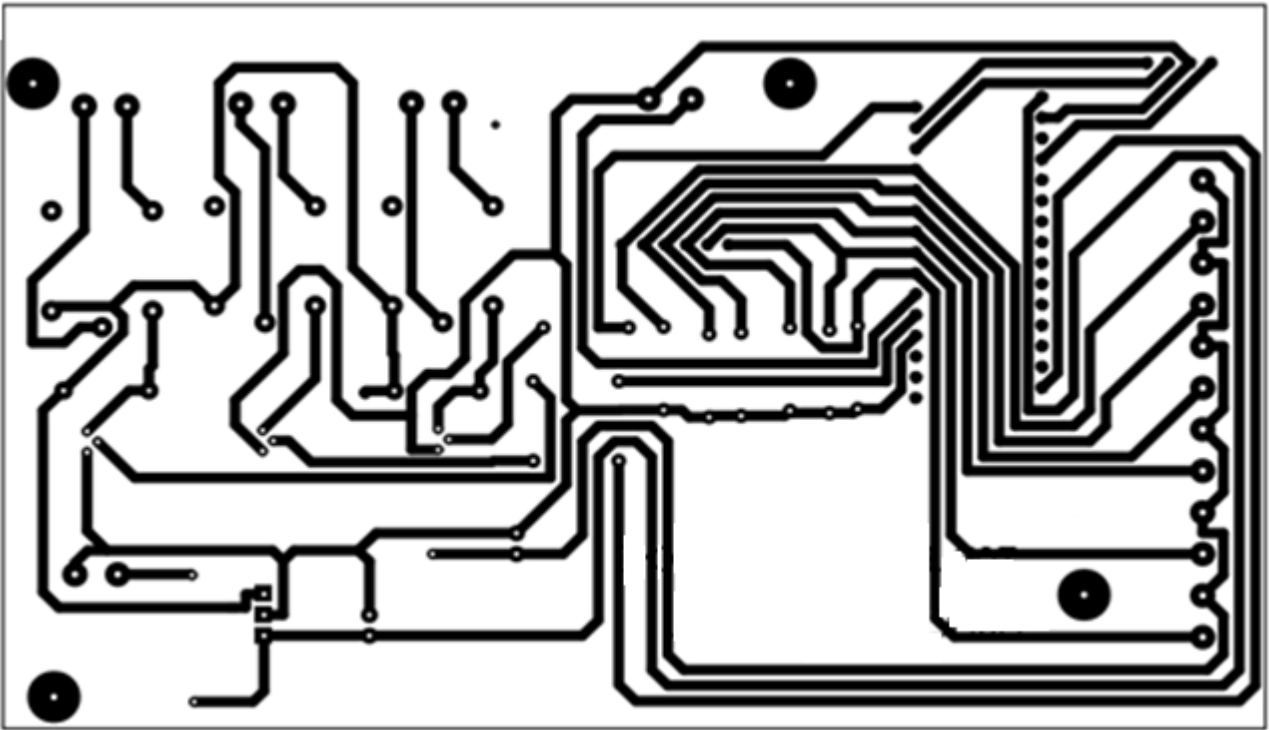
7

8

A

Tarjeta de control seguridades, control PWM, activación remota.

B



C

D

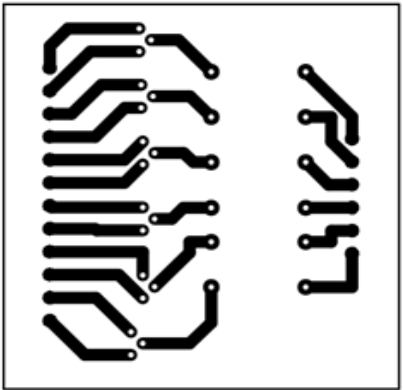
E

F

A

Tarjeta enlace señales micros del Arduino NANO al Arduino UNO

B



C

D

E

F



UNIVERSIDAD TECNOLÓGICA ISRAEL

CONTIENE: Distribución pistas placas control CNC

UNIVERSIDAD TECNOLÓGICA ISRAEL

PROYECTO: TESIS

APROB: Mg. Rene Cortijo

DIBUJADO POR: R. Pasquel.

REALIZADO POR: R. Pasquel

FECHA: 05/01/2018

LAMINA Nº 1

ANEXO 7: Planos mecánicos de la fresadora CNC.

1 2 3 4 5 6 7 8

A

B

C

D

E

F

A

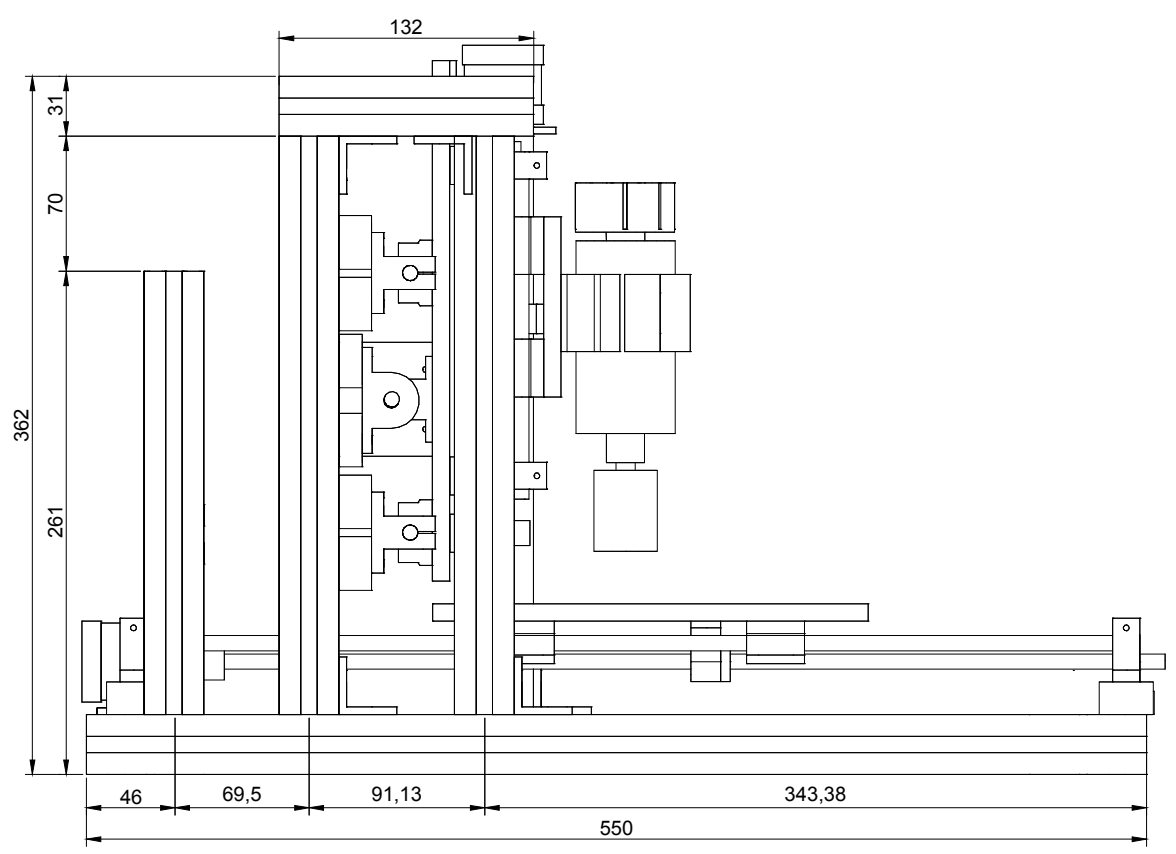
B

C

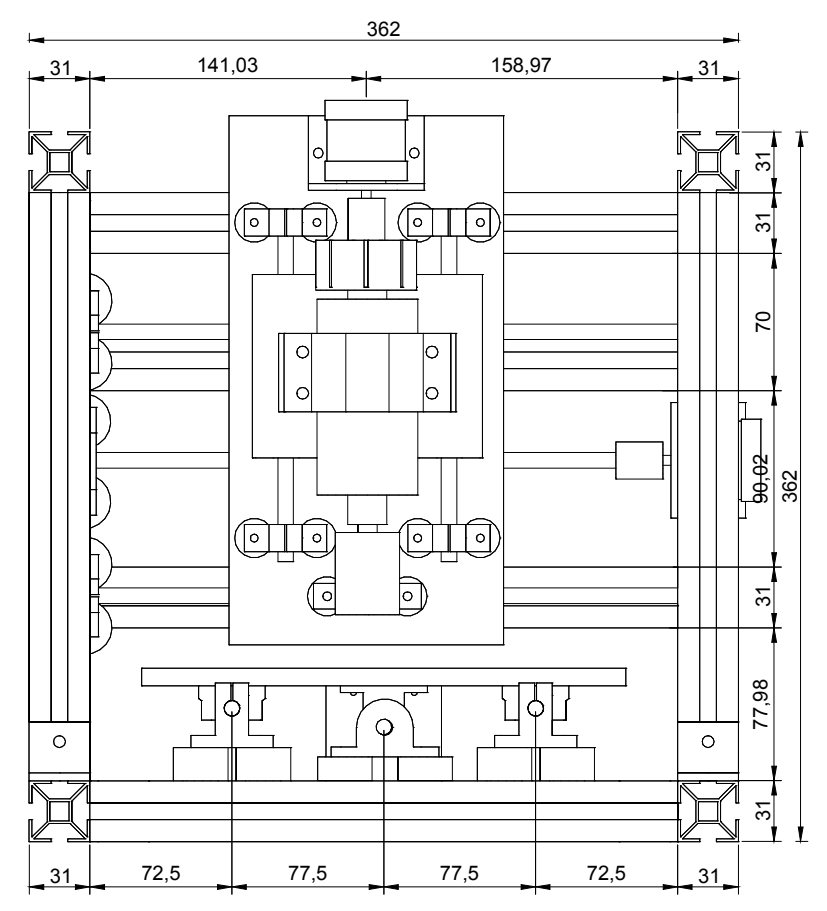
D

E

F



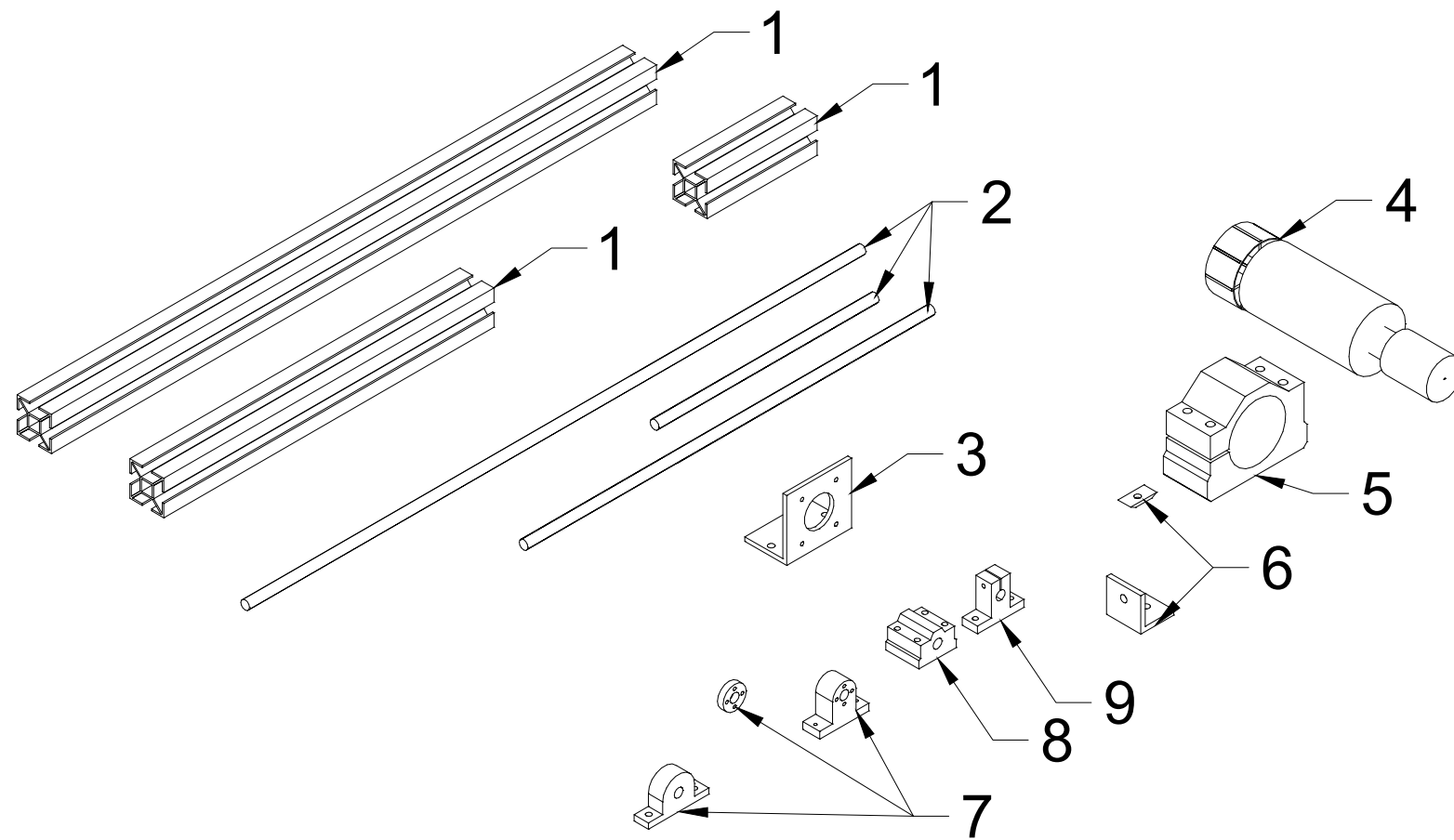
VISTA LATERAL



VISTA FRONTAL

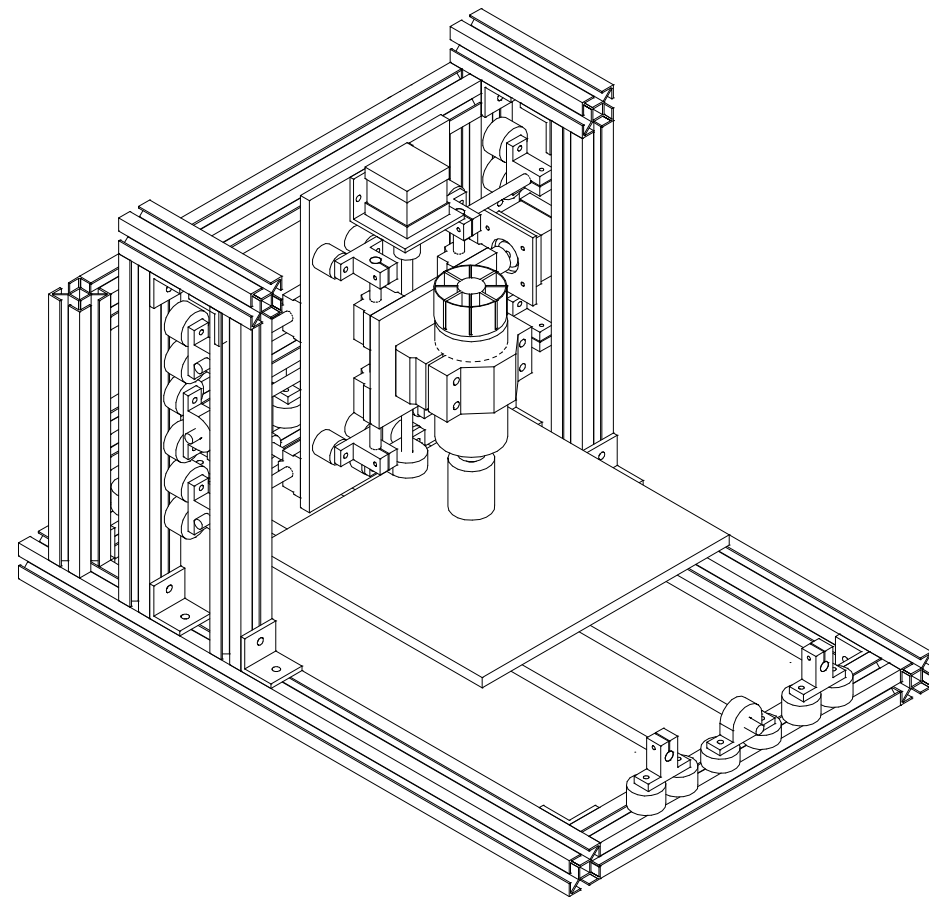
PROYECTO: MAQUINA CNC DISEÑO DE MAQUINA CNC AUTOR: ROBERTO PASQUEL REV. MG RENE CORTIJO	FECHA: 25/01/2019 PAGINA: 1/2
--------------------------------------------------------------------------------------------------	-------------------------------------

1 2 3 4 5 6 7 8



PARTES DE LA ESTRUCTURA:

- 1. SOPORTES
- 2. VARILLAS PARA SOPORTES
- 3. BASE PARA MOTOR A PASOS
- 4. MOTOR SPINDLE
- 5. BASE MOTOR SPINDLE
- 6. UNIONES
- 7. SOPORTES GUÍAS
- 8. RODAMIENTO LINEAL
- 9. BASES VARILLAS



VISTA ISOMETRICA

PROYECTO: MAQUINA CNC	FECHA:
DISEÑO DE MAQUINA CNC	25/01/2019
AUTOR: ROBERTO PASQUEL	PAGINA: 2/2
REV. MG RENE CORTIJO	

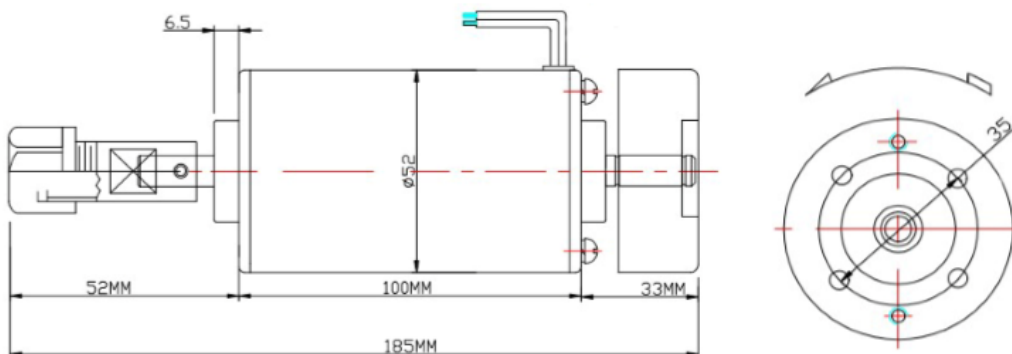
ANEXO 8: Datasheet

CNC Spindle motor DC 400 W



Specifications:

1. Working Voltage :12-48VDC
2. Speed :3000-12000r / min (12V-3000 rpm, 24V-6000 rpm, 36V-9000 rpm, 48V-12000 rpm)
3. Torque: 500mN.m
4. Insulation Resistance:> 2 megohms
5. Dielectric Strength: 400V
6. Diameter: 52mm
7. Chuck Part Length: Arbor parts 35mm, (including nuts and sandwich 43mm)
5. Part chuck Diameter: 16mm
6. Motor Length: 185mm (including gripping parts and motor)
7. Spindle Net Weight: 0.9KG
8. Spindle Runout of about 0.01-0.04.
9. Can be used for engraving metal or nonmetal materials.
10. This comes with air-cooled spindle can work long hours.
11. Suitable for carving PCB, acrylic, drilling



EKT
Electronics Katrangi Trading

www.ekt2.com

Made in China

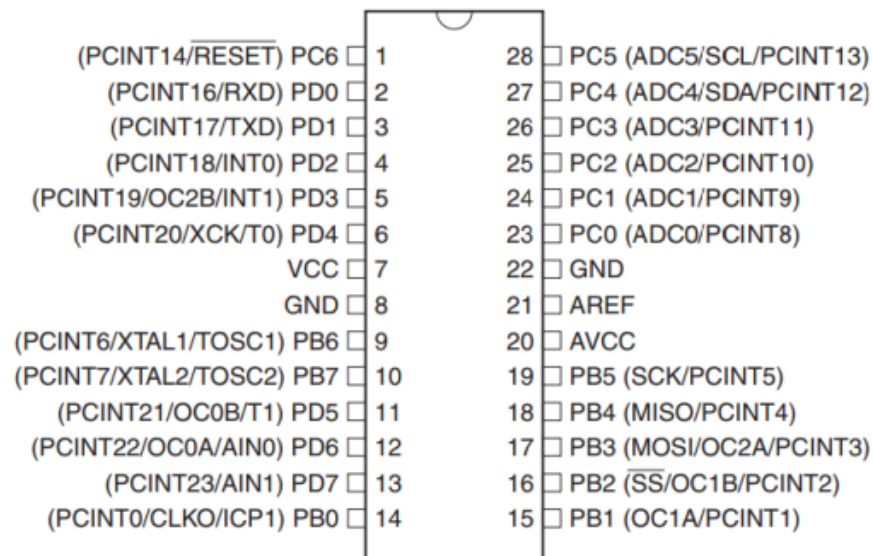
Arduino UNO



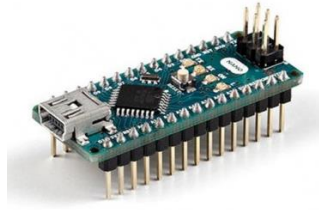
Características técnicas

Microcontrolador	Atmega328
Voltaje de operación	5V
Voltaje de entrada (Recomendado)	7 – 12V
Voltaje de entrada (Limite)	6 – 20V
Pines para entrada- salida digital.	14 (6 pueden usarse como salida de PWM)
Pines de entrada analógica.	6
Corriente continua por pin IO	40 mA
Corriente continua en el pin 3.3V	50 mA
Memoria Flash	32 KB (0,5 KB ocupados por el bootloader)
SRAM	2 KB
EEPROM	1 KB
Frecuencia de reloj	16 MHz

Disposición de pines



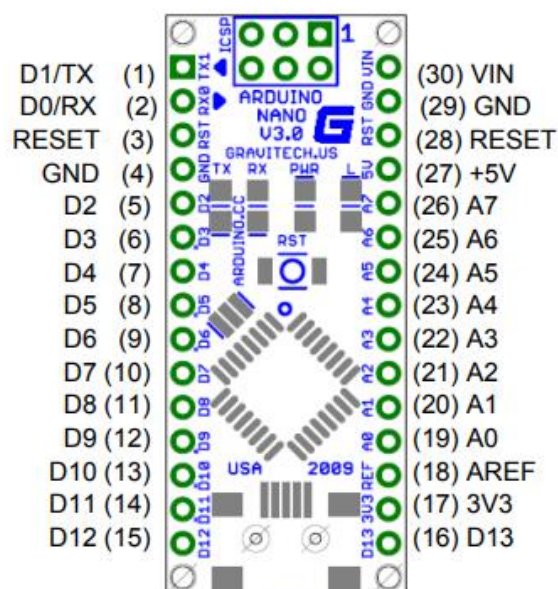
Arduino NANO



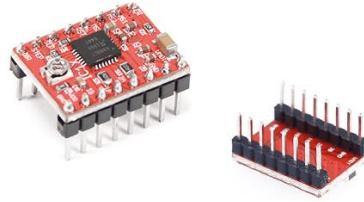
Características técnicas

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

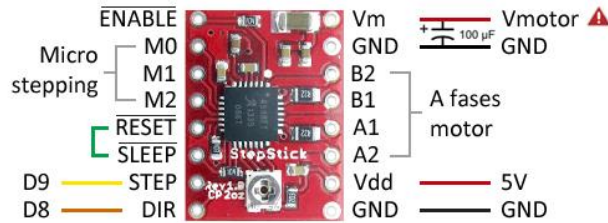
Descripción de pines



Controladores A4988



Descripción de pines



Características técnicas

A4988

DMOS Microstepping Driver with Translator and Overcurrent Protection

ELECTRICAL CHARACTERISTICS¹ at T_A = 25°C, V_{BB} = 35 V (unless otherwise noted)

Characteristics	Symbol	Test Conditions	Min.	Typ. ²	Max.	Units
Output Drivers						
Load Supply Voltage Range	V _{BB}	Operating	8	–	35	V
Logic Supply Voltage Range	V _{DD}	Operating	3.0	–	5.5	V
Output On Resistance	R _{DS(on)}	Source Driver, I _{OUT} = –1.5 A	–	320	430	mΩ
		Sink Driver, I _{OUT} = 1.5 A	–	320	430	mΩ
Body Diode Forward Voltage	V _F	Source Diode, I _F = –1.5 A	–	–	1.2	V
		Sink Diode, I _F = 1.5 A	–	–	1.2	V
Motor Supply Current	I _{BB}	f _{PWM} < 50 kHz	–	–	4	mA
		Operating, outputs disabled	–	–	2	mA
Logic Supply Current	I _{DD}	f _{PWM} < 50 kHz	–	–	8	mA
		Outputs off	–	–	5	mA
Control Logic						
Logic Input Voltage	V _{IN(1)}		V _{DD} × 0.7	–	–	V
	V _{IN(0)}		–	–	V _{DD} × 0.3	V
Logic Input Current	I _{IN(1)}	V _{IN} = V _{DD} × 0.7	–20	<1.0	20	µA
	I _{IN(0)}	V _{IN} = V _{DD} × 0.3	–20	<1.0	20	µA
Microstep Select	R _{MS1}	MS1 pin	–	100	–	kΩ
	R _{MS2}	MS2 pin	–	50	–	kΩ
	R _{MS3}	MS3 pin	–	100	–	kΩ
Logic Input Hysteresis	V _{HYS(IN)}	As a % of V _{DD}	5	11	19	%
Blank Time	t _{BLANK}		0.7	1	1.3	µs
Fixed Off-Time	t _{OFF}	OSC = V _{DD} or GND	20	30	40	µs
		R _{OSC} = 25 kΩ	23	30	37	µs
Reference Input Voltage Range	V _{REF}		0	–	4	V
Reference Input Current	I _{REF}		–3	0	3	µA
Current Trip-Level Error ³	err _I	V _{REF} = 2 V, %I _{TRIPMAX} = 38.27%	–	–	±15	%
		V _{REF} = 2 V, %I _{TRIPMAX} = 70.71%	–	–	±5	%
		V _{REF} = 2 V, %I _{TRIPMAX} = 100.00%	–	–	±5	%
Protection						
Overcurrent Protection Threshold	I _{OCPST}		2.1	–	–	A
Thermal Shutdown Temperature	T _{SD}		–	165	–	°C
Thermal Shutdown Hysteresis	T _{SDHYS}		–	15	–	°C
VDD Undervoltage Lockout	V _{DDUVLO}	V _{DD} rising	2.7	2.8	2.9	V
VDD Undervoltage Hysteresis	V _{DDUVLOHYS}		–	90	–	mV

¹For input and output current specifications, negative current is defined as coming out of (sourcing) the specified device pin.

²Typical data are for initial design estimations only, and assume optimum manufacturing and application conditions. Performance may vary for individual units, within the specified maximum and minimum limits.

³V_{ERR} = [(V_{REF}/8) – V_{SENSE}] / (V_{REF}/8).

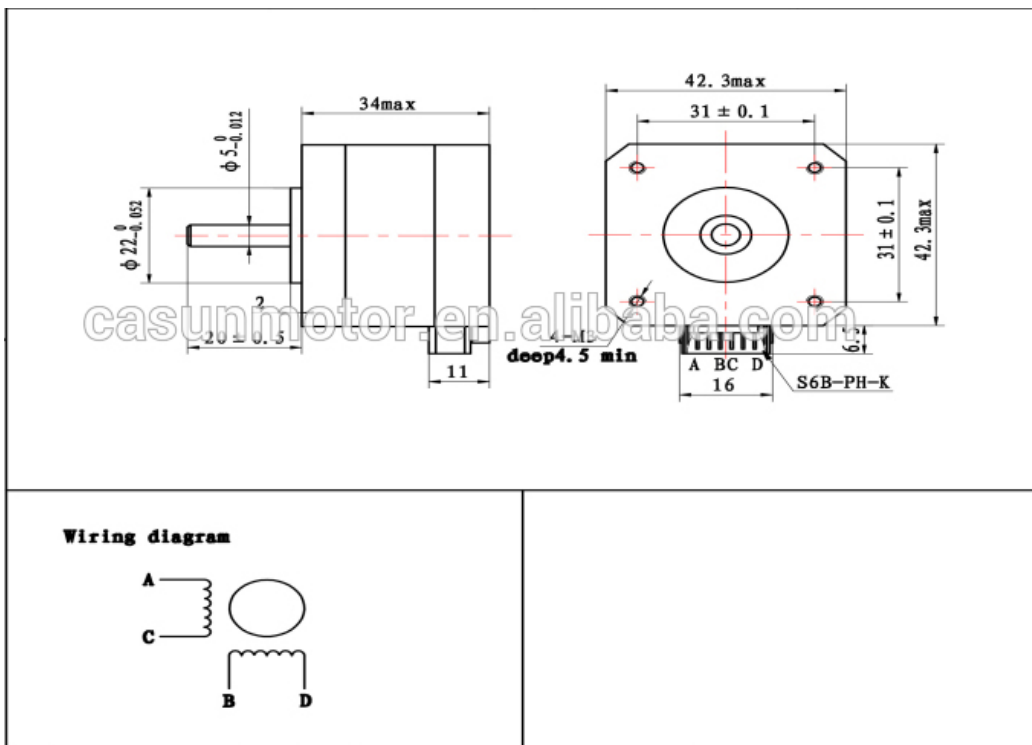
Motores paso a paso



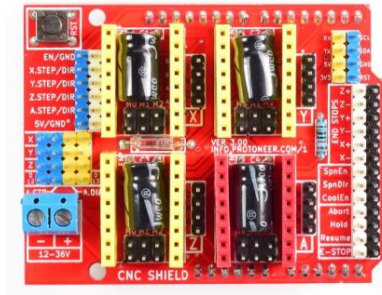
Main characteristics:

- Model: KS42STH40-1204A
- Compatible with NEMA17 standard
- Dimensions: 42.3 x 42.3 x 40 mm
- Motor shaft diameter: 5mm
- Motor shaft length: 22mm
- Phase number: 2
- Step angle: 1.8° / pas (+/- 5%)
- Step count per turn: 200
- Standard Voltage: 3.6V
- Phase current: 1.2A
- Phase resistance: 3 ohm
- Holding torque: 4000 g.cm
- Rotor inertia: 54 g.cm²
- Lead wires: 4
- Weight: 300g

Dimensiones



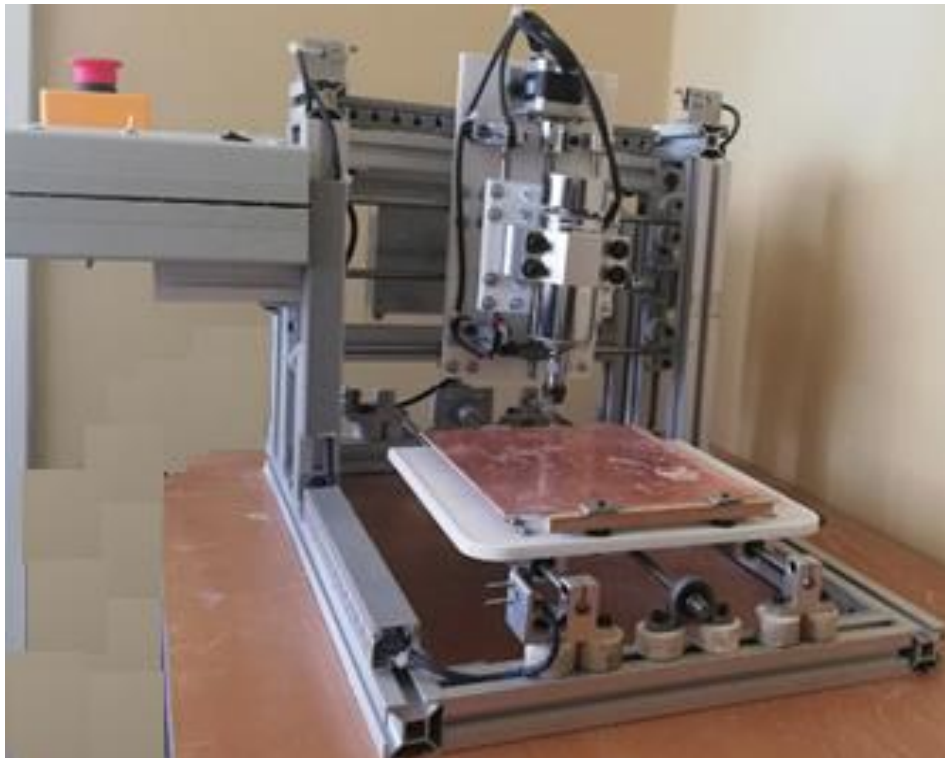
CNC Shield



- Voltaje de Potencia: 12- 36V DC
- Compatible con [GRBL 0.9j](#) (firmware OpenSource para Arduino UNO que convierte código-G en comandos para motores Paso a Paso)
- Soporta 4 Ejes (X, Y, Z y duplicar uno de los anteriores o crear un eje a medida con los pines D12 y D13)
- Fin de carrera por cada eje (6 en Total)
- Habilitador y dirección de *Spindle*
- Habilitador de refrigerante (coolant)
- Soporta drivers Pololu A4988 o DRV8825.
- Jumpers para elegir el micro-stepping de los drivers.
- Diseño Compacto
- Los motores se pueden conectar usando header o moxex hembra de 4 pines

ANEXO 9: Manual de usuario.

Fresadora CNC para PCBs







Manual de usuario

Información general

- Para obtener el máximo rendimiento del equipo el encargado de operar, debe estar familiarizado con cada uno de los dispositivos que contiene la máquina, previo una capacitación.
- El autor de este manual se reserva el derecho de realizar cambios sin previo aviso.

Normas de seguridad

	Usar protección visual
	Usar protección respiratoria
	Usar protección de los oídos
	Usar protección de las manos

Precaución

	No mojar o derramar ninguna clase de líquidos, puede causar daños en el equipo o personales por descargas eléctricas.
	No utilizar líquidos inflamables con el equipo.
	En caso de incendio no usar agua como medio de extinción.
	Riesgo de corte
	Riesgo de atrapamiento.

Manual de usuario

Introducción

El presente equipo es una fresadora CNC para PCBs, debe ser utilizado por personal que haya recibido una previa capacitación sobre el funcionamiento y prestaciones de el mismo.

Características

- La dimensión máxima de trabajo de las PCBs de la fresadora es de 170mm x 140mm.
- El tiempo de acabado de cada PCB varía dependiendo el diseño y la calibración de la máquina.

Funciones

- Realiza perforación (archivo Excellon) de PCBs dependiendo el diseño, las dimensiones de las brocas dependen del trabajo que se realice.
- Realiza pistas (archivo Gerber) de PCBs según el diseño, se utiliza la fresa con Angulo de 10°.
- Realiza el corte (archivo Gerber) de la PCB terminada, se utiliza la fresa con Angulo de 30°.

Nota: Tener en cuenta que para el acabado depende de la calibración de tiempos de cada motor de la máquina.

Especificaciones técnicas

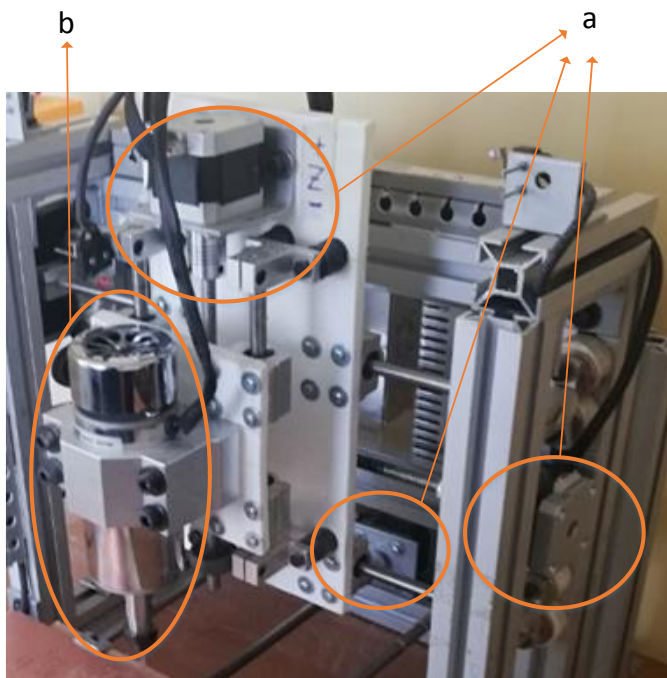
- Fuente de alimentación: 110 V ac 60 HZ
- Tener una laptop con todo el software necesario y detallado en el manual.
- Descargas la App de MIT App Inventor 2 para el control remoto.

Distribución de componentes mecánicos, eléctricos y electrónicos.

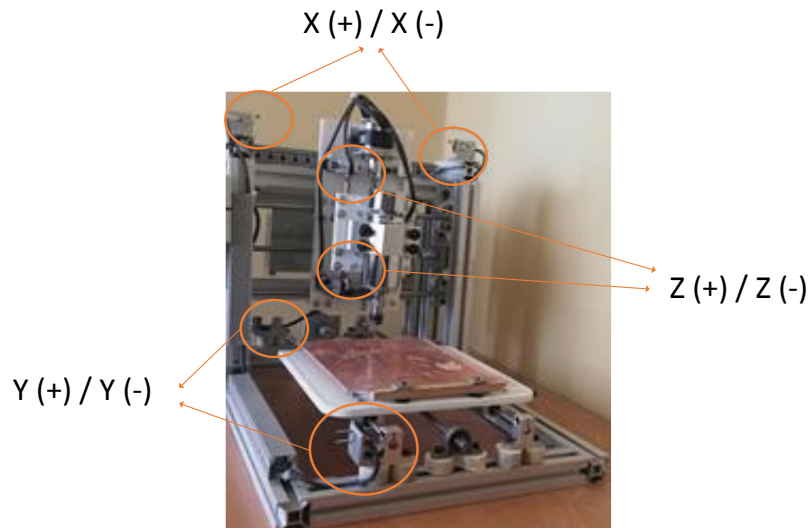


- **Distribución de los motores:**

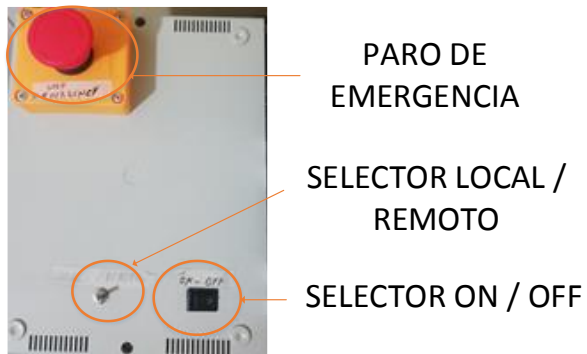
- a) Motores a paso de los ejes X, Y, Z.
- b) Motor *Spindle*.



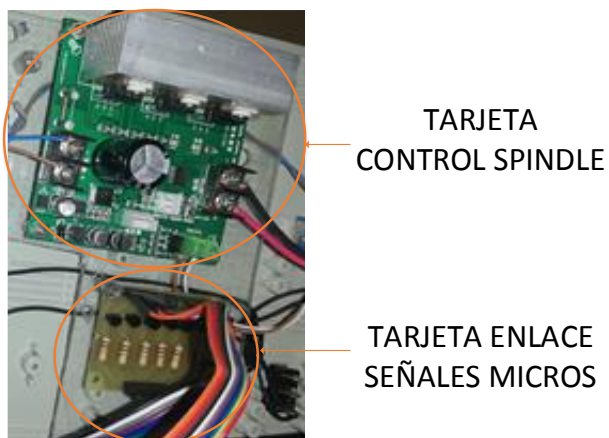
- **Distribución de los micros (finales de carrera):**



- **Distribución panel exterior:**

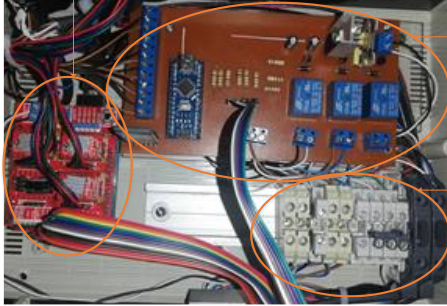


- **Distribución elementos panel interior superior:**



- **Distribución elementos panel interior inferior:**

TARJETA CNC
SHIELD /
ARDUINO UNO



Tarjeta control
seguridades, control
PWM, activación remota

PORTAFUSIBLES,
BORNERS DE
CONEXION

- **Distribución fuentes:**



FUENTE 1 SPINDLE

FUENTE 2 CONTROL
GENERAL

Procedimiento cambio de fresas y brocas

Este trabajo se lo debe realizar con la maquina en paro de emergencia, con el selector en Off y desconectada de la toma eléctrica.

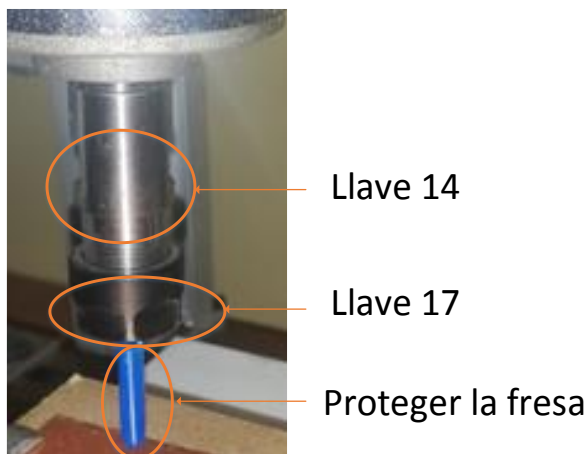


- **Herramientas necesarias**

- a) Llave mixta No. 17
- b) Llave mixta No. 14



Se debe trabajar con la llave 17 y 14, sosteniendo el mandril del Spindle, se afloja sosteniendo con la llave 14 fija en la parte superior y la llave 17 se gira en sentido horario, tener en cuenta que para las fresas se debe tener colocada la protección para evitar daños en la herramienta de corte y daños al operador que esté realizando el trabajo.

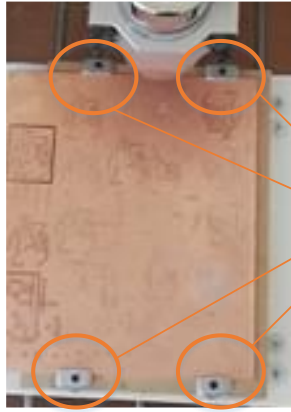


Una vez desmontada la fresa se coloca la nueva, o la broca a usarse dependiendo el trabajo a realizar, el tiempo estimado para esta acción es de 3 minutos.

Procedimiento cambio de PCB

Este trabajo se lo debe realizar con la maquina en paro de emergencia, con el selector en Off y desconectada de la toma eléctrica.

Identificar en la bancada los cuatro pernos M6, los cuales sujetan las placas se debe aflojar por la parte inferior con la ayuda de la llave mixta No. 8 los cuatro pernos sin sacarlos solo dejar flojos, las dimensiones de la placa a instalarse deben ser de 2030 mm x 1500 mm esto quiere decir que una placa virgen de un formato A4, se la debe cortar en 2 y cabe sin ninguna restricción, el tiempo estimado para este trabajo es de 3 minutos, teniendo ya la placa virgen cortada.



Pernos de sujeción PCBs

Procedimiento de operación.

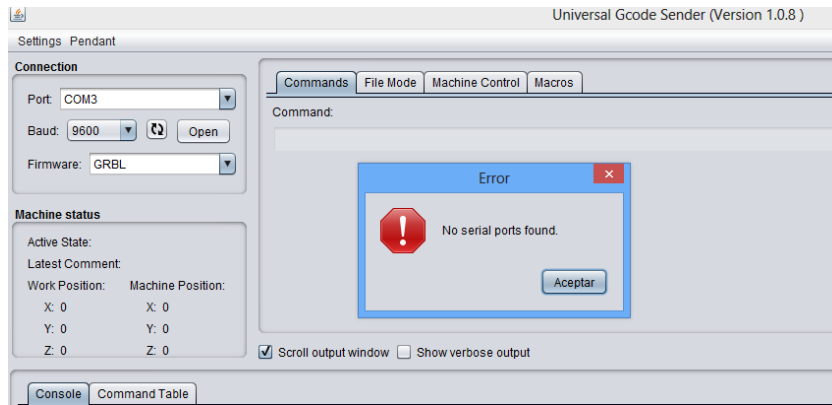
- a) Enchufar la maquina a la toma de 110 V ac, accionar el selector a la posición On, Levantar el paro de emergencia.
- b) Habilitar la computadora abriendo el programa Universal Gcode Sender, realizar el siguiente procedimiento.
- c)

Procedimiento de calibración inicial en Universal Gcode Sender de la fresadora CNC

- a) Abrir el programa Universal Gcode Sender versión 1.0.8.

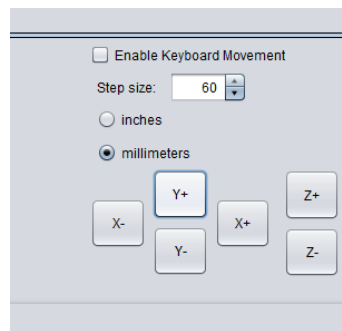


- b) Configurar el puerto en *Port*, se visualiza los puertos COM habilitados se refresca, se coloca en *Baud* 9600 y se abre la interfaz Open, una vez configurado el enlace se puede realizar el procedimiento local / remoto.

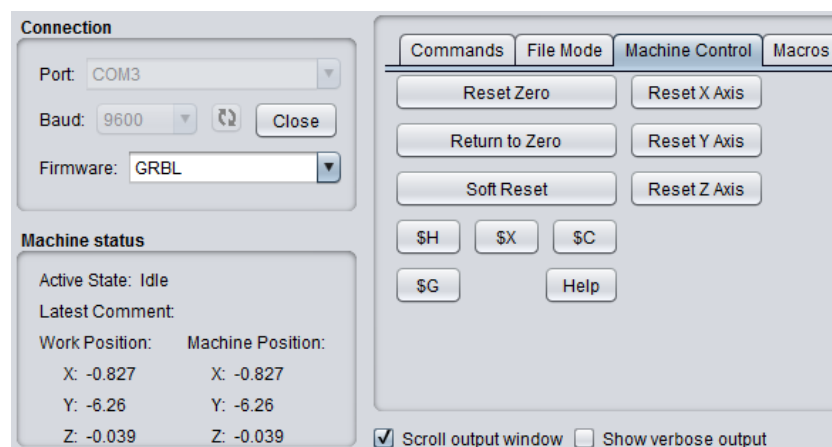


Procedimiento puesta a cero a través de Universal Gcode Sender.

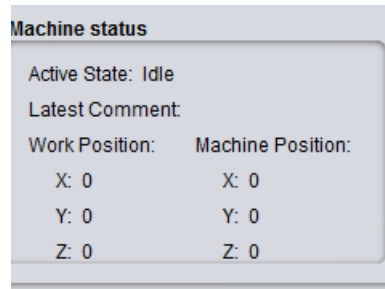
- a) Con los botones de movimiento de los ejes X, Y, Z, ubicar la máquina en el punto requerido.



- b) Presionar el botón *Reset Zero* y la fresadora reconoce el nuevo punto cero.



Antes de presionar

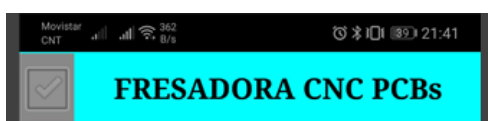


Después de presionar

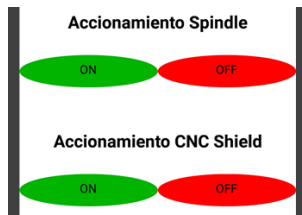
- c) Al realizar esta maniobra se puede manipular a través de los botones los movimientos de los ejes y al presionar el botón *Return to Zero* los ejes vuelven al punto cero fijados.
- d) En el caso de tener un archivo (.ngc) creado se realizara los siguientes pasos caso contrario se realizara el procediendo de generación del archivo Gerber y Excellon detallado más adelante.
- e) Abrir la aplicación del Smartphone realizar la conexión Bluetooth y enlazarse al dispositivo indicado.



- f) Si la conexión ha tenido éxito se observa marcado un visto en el recuadro superior.



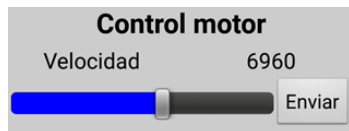
- g) Presionar Control principal y habilitar los relés de las tarjetas Accionamiento CNC Shield y Accionamiento Spindle, tener en cuenta que el selector local / remoto este en remoto, de ser el caso.



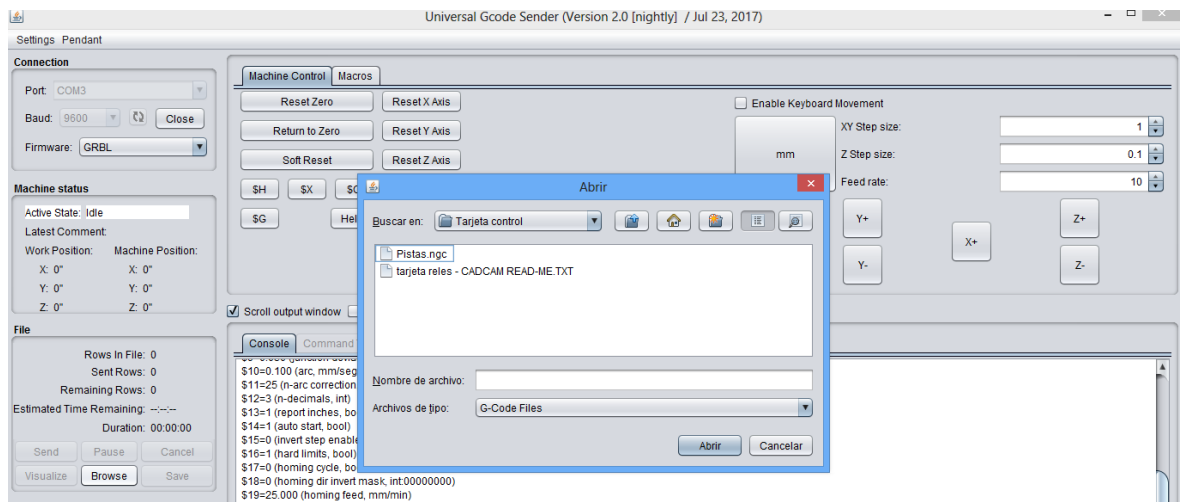
- h) Regresar y abrir la ventana control y alarmas visualizar que no exista ninguna alarma. De existir alarmas revisar que no exista atascamientos o fallos en los micros.



- i) Con el slider se coloca la velocidad deseada y se presiona Enviar esto activara el motor del Spindle.



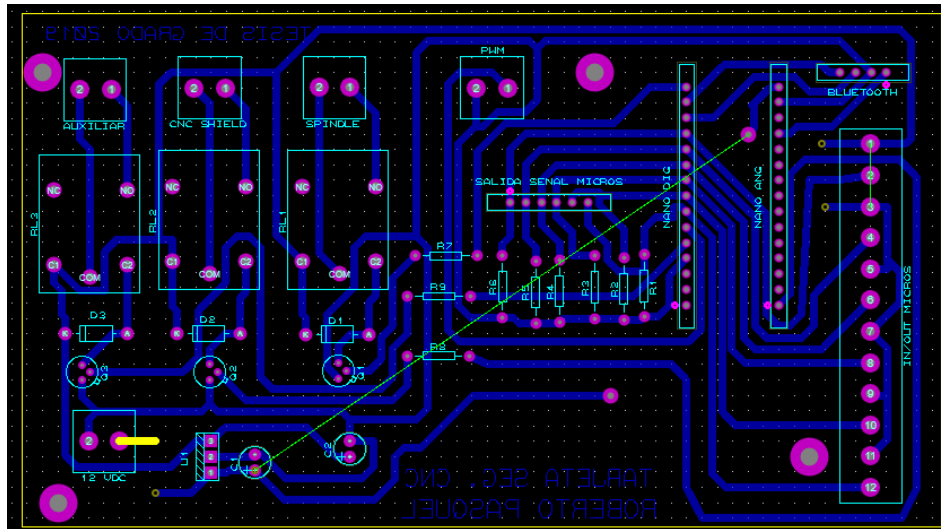
- j) Colocado en el punto cero y activado el Spindle se carga el programa (.ngc).
 k) En la pestaña *Browse* se busca la ubicación del archivo (.ngc) y se presiona abrir esto es para perforación, pistas y corte cada uno por separado.



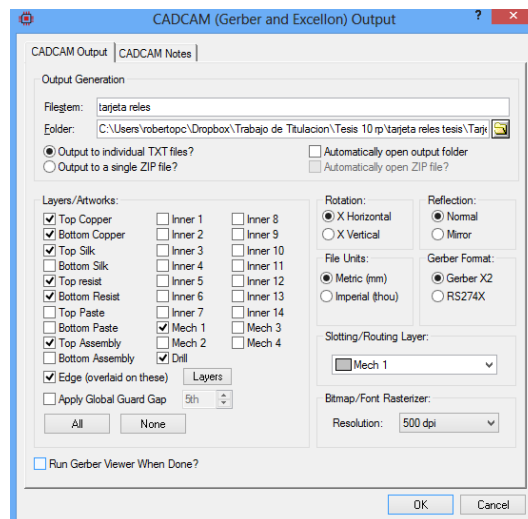
- l) Se presiona el botón Send y la maquina realiza el trabajo cargado.

Procedimiento de generación del archivo Gerber y Excellon

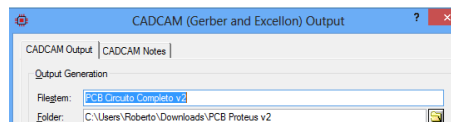
- a) Abrir en Proteus - Ares el circuito electrónico.



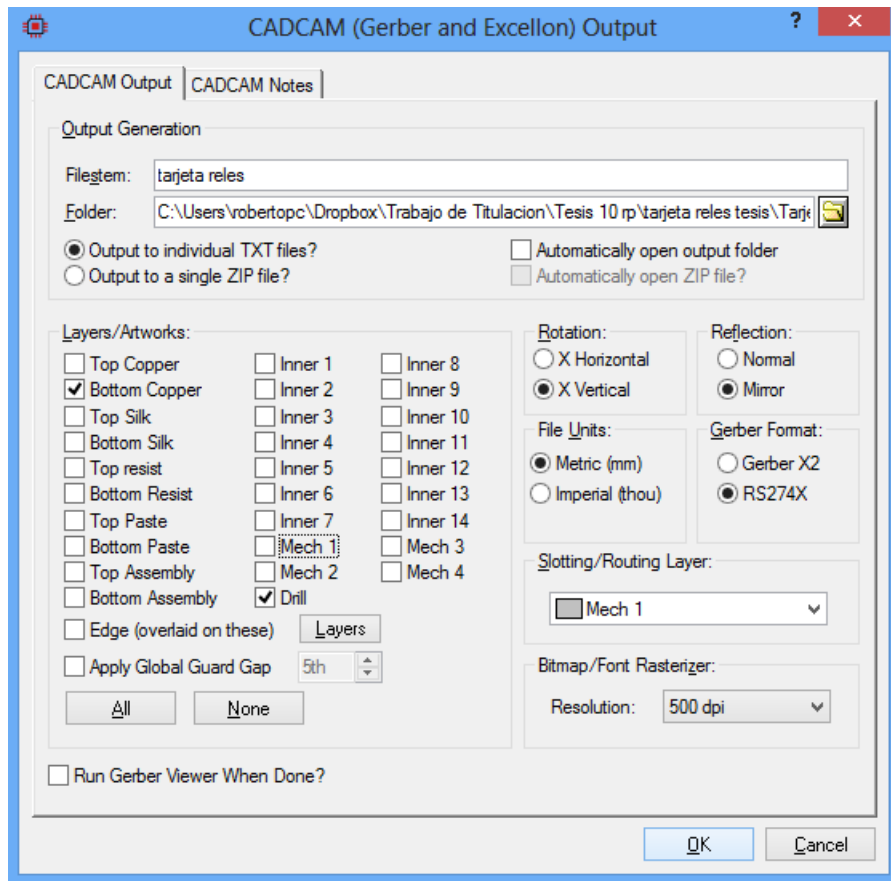
- b) Abrir la pestaña Output – *Generate Gerber / Excellon Files*, presionar, NO. Se abre la siguiente ventana.



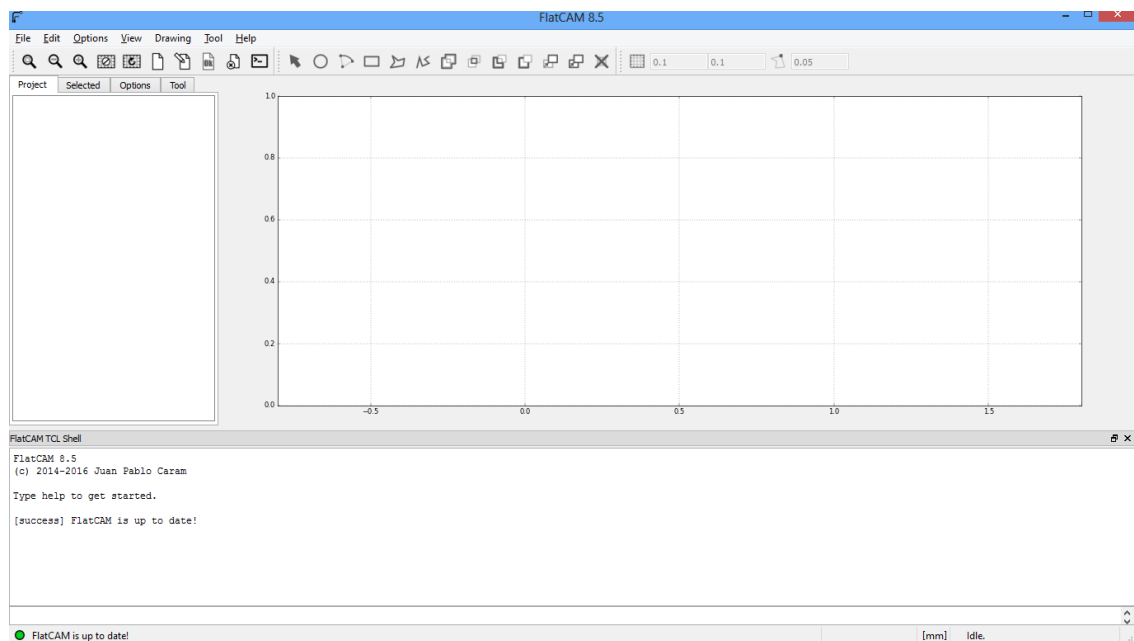
- c) Colocar un nombre al archivo – seleccionar la ubicación para guardar el archivo.



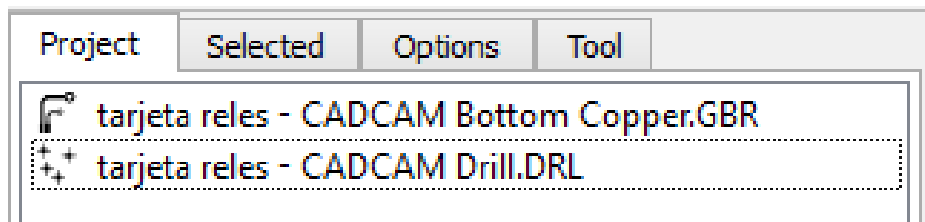
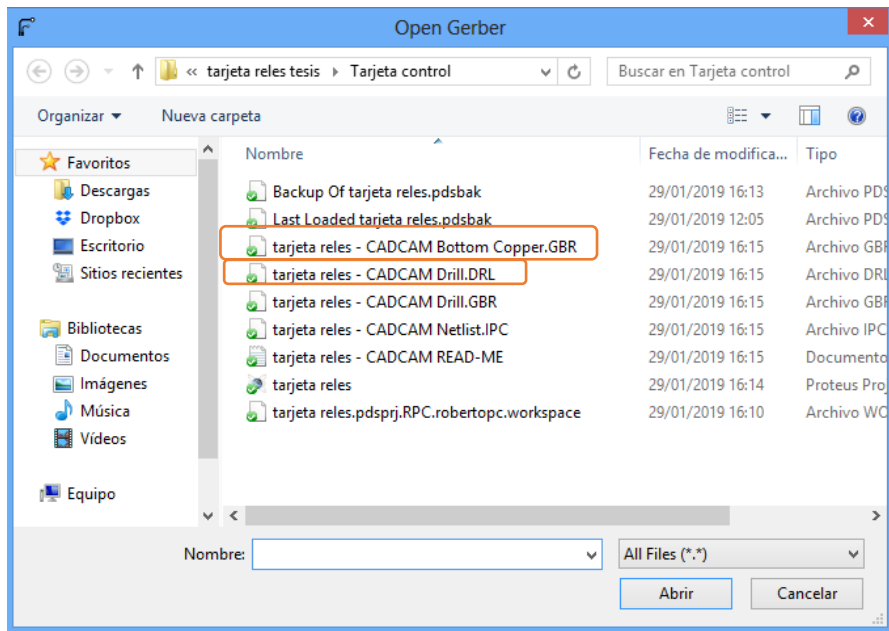
- d) Seleccionar *Layers/Artworks* – *Rotation* – *File Units* – *Reflection* - *Gerber Format* y presionar OK con esto se genera un archivo (.GBR). Por separado se genera 2 archivos seleccionando *Bottom Copper*, *Drill*, los cuales son para las pistas (Gerber) y perforación (Excellon).



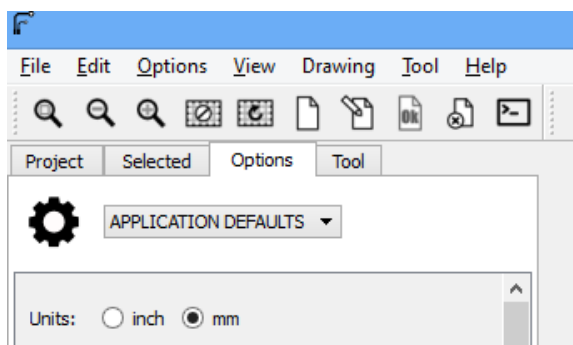
e) Abrir el programa FlatCam 8.5



f) Abrir la pestaña File – Open Gerber o Excellon y buscar la ubicación de donde se guardó y abrir los 2 archivos.



g) Seleccionar la pestaña *Options* – *units: mm*, para cada uno de los archivos.

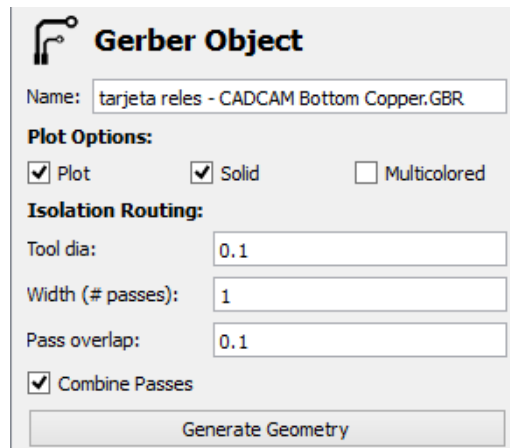


h) Para generar el archivo (.ngc) el cual interpreta el programa Universal Gcode Sender, se parte de los archivos (.GRB) y (.DRL) por separado, seleccionar la pestaña *Selected*, y configurar:

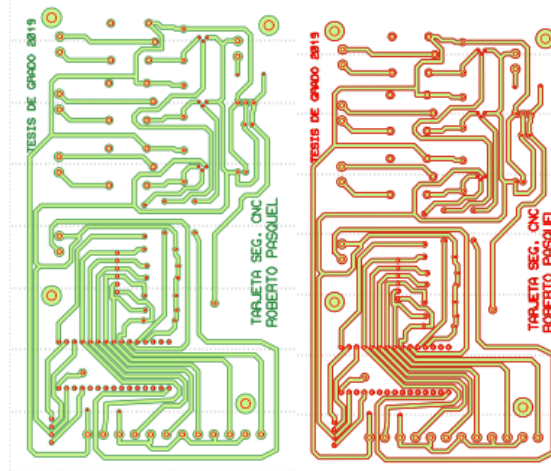
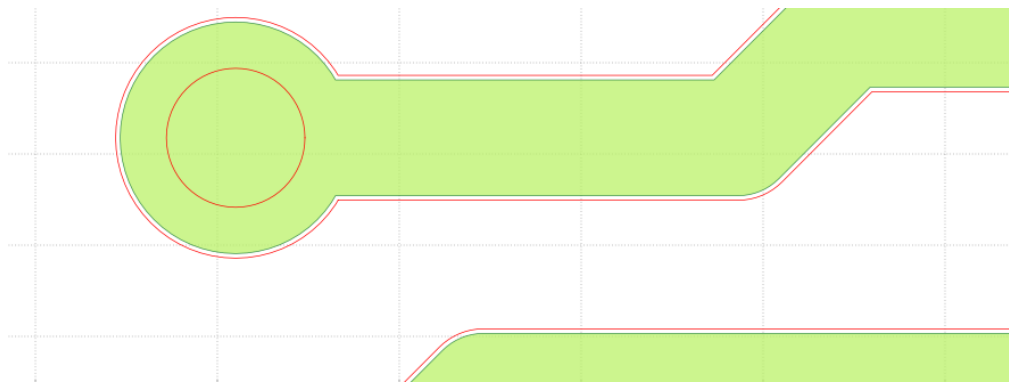
Tool dia: Diámetro de la herramienta.

Width (#passes): Numero de pasadas

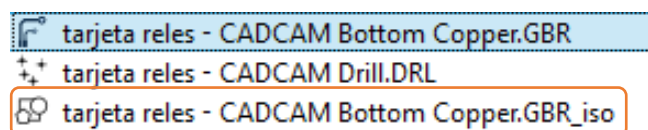
Pass overlap: Profundidad de perforación



- i) Presionar *Generate Geometry*, se observa la generación de la geometría de las pistas y una línea roja que se refiere al número de pasadas.



- j) Se genera un archivo (.GBR_iso).



- k) Se abre dando doble clic se ingresa los parámetros y se da clic en *Generate*.

Cut Z: Desplazamiento de eje Z en negativo

Travel Z: Desplazamiento de eje Z en positivo.

Feed Rate: Velocidad de enrutamiento de las pistas.

Tool dia: Diámetro de la herramienta.

Plot Options:

Plot

Create CNC Job:

Cut Z:

Travel Z:

Feed Rate:

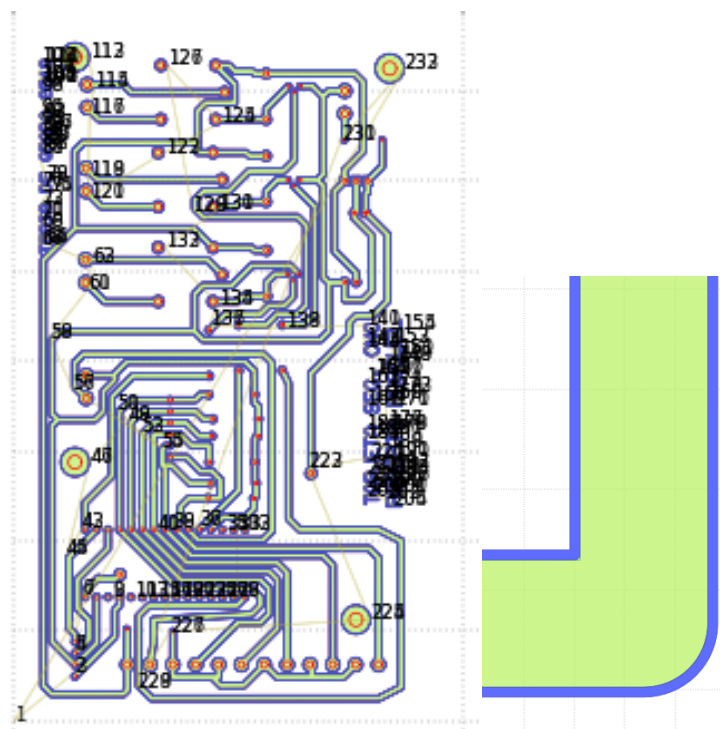
Tool dia:

Spindle speed:

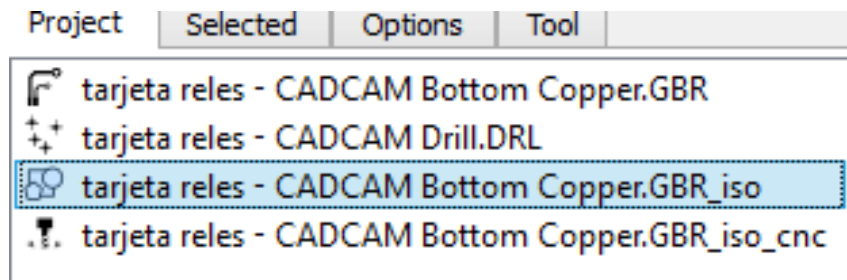
Multi-Depth:

Depth/pass:

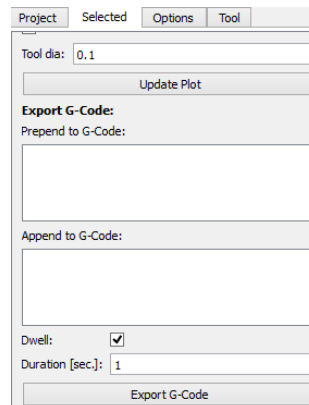
- 1) Se crea la capa definitiva y se observa la cantidad de material a desbastar de color azul.



m) Se crea un nuevo archivo (.GRB_iso_cnc).

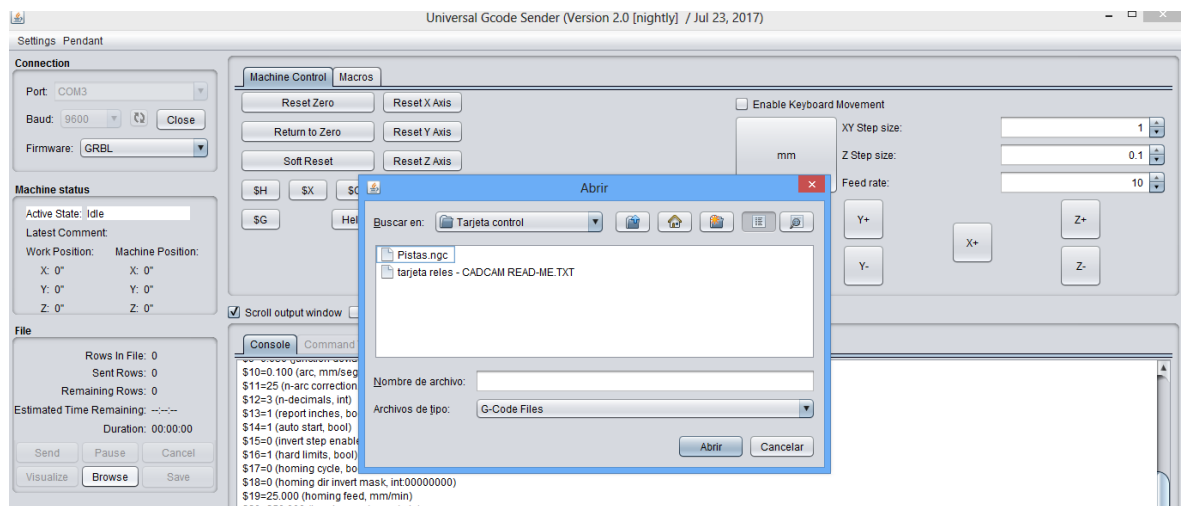


n) Abrir el archivo anterior y presionar en el icono *Export G-code* ubicado hasta abajo.

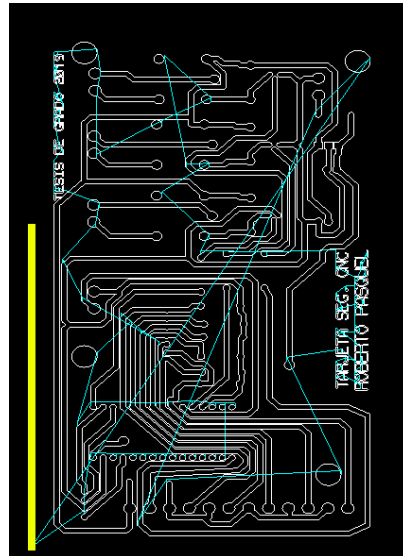


o) Se abre una ventana en la se selecciona una carpeta donde guardar y se coloca un nombre y al final debe ir (.ngc) que es el archivo que reconoce el programa Universal Gcode Sender y presionar Guardar.

p) Se abre el programa Universal Gcode Sender en la pestaña *Browse* se busca la ubicación del archivo (.ngc) y se presiona abrir esto es para perforación, pistas y corte cada uno por separado.



- q) Al hacer clic en el botón *Visualize*, se despliega una ventana en la cual se puede realizar un seguimiento del proceso de maquinado en tiempo real.



- r) Para iniciar el proceso de maquinado se debe considerar la puesta a cero indicado en el inicio de este manual. Posterior se da clic en *Send*, y la fresadora CNC realiza de forma automática el trabajo generado.