UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

MÁQUINA REMOVEDORA DE COBRE Y SIMILARES

TESIS

Que para obtener el título de

Ingeniero Eléctrico-Electrónico

P R E S E N T A

Alan Eduardo Ortega Hernández

DIRECTORA DE TESIS

Ing. Beatriz Eslava Arellanes

Ciudad Universitaria, Cd. Mx., 2019

AGRADECIMIENTOS

A mi familia, a Nacho por enseñarme a ser mejor persona, a Laura por enseñarme a estudiar, a Manene por ser mi hermano mayor, a Javier por enseñarme a subir montañas, a Samuel por soportarme y a Gloria por su amor incondicional, por educarme y por ser mi madre.

A la ingeniera Beatriz Eslava Arellanes por todo el tiempo, apoyo y consejos en el desarrollo de esta tesis.

A Larissa por su amor y compañía.

A mis amigos, por tantas aventuras compartidas.

A mi gato René y a mi perro Aslan por acompañarme en las largas noches de proyectos.

ÍNDICE

ÍNDICE DE FIGURAS

APÉNDICE

ÍNDICE DE TABLAS

CAPÍTULO 3

CAPÍTULO 4

INTRODUCCIÓN

El control numérico por computadora, comúnmente conocido como CNC es un sistema que permite controlar la posición de un elemento físico, en donde normalmente se monta una herramienta para realizar un trabajo específico.

Este control se consigue mediante un programa y un conjunto de órdenes añadidas que en conjunto permiten controlar las coordenadas de posición de un punto respecto a un origen, permitiéndonos trabajar en un plano de múltiples ejes coordenados.

El lenguaje de programación más utilizado en estos sistemas es el conocido como código g, mismo que se emplea para dar a entender a la máquina las ordenes requeridas por el usuario, aunado a esto, es necesario tener un microcontrolador que interprete el código generado y envíe la información de velocidad y sentido a los controladores que se encargaran de accionar los motores a pasos para que estos finalmente puedan realizar el movimiento requerido en función del código g generado.

A lo largo de esta tesis se perseguirá como primera instancia el desarrollo de un sistema físico de tres ejes que mediante el devastado de material sea capaz de entregar como producto final el trazado de alguna imagen proporcionada por el usuario.

El trabajo realizado por esta máquina se busca aplicar directamente en el devastado de cobre para una placa fenólica obteniendo así un circuito listo para realizar el montaje de componentes electrónicos.

Se espera que el sistema sea de bajo coste además de tener versatilidad para el trabajo en otro tipo de materiales para así obtener mejor ganancia en la construcción del mismo.

PLANTEAMIENTO DEL PROBLEMA

A través de este proyecto se busca dar una solución real a un problema en concreto de la ingeniería, siendo este, la falta de equipo de apoyo público en la facultad para la generación de circuitos montados en placas fenólicas.

Este problema limita la creación de proyectos de alto ensamblaje electrónico y la replicación de los mismos, dado que para generar las pistas de forma manual el proceso es tardado y tiende a tener una calidad menor a la que se puede obtener mediante la automatización de un sistema.

Para solventar dicho problema hace falta que la solución abarque a una cantidad considerable de estudiantes, sea de fácil acceso y cuente con una fácil implementación.

ALTERNATIVAS DE SOLUCIÓN

Para solventar este problema, una práctica bastante común entre los estudiantes de electrónica es la serigrafía sobre una placa de cobre virgen ya que con ella se imprimen las pistas del circuito y posteriormente, mediante el sumergimiento en cloruro férrico, se genere una reacción química que barre de la superficie el cobre no deseado, dando como resultado final las pistas de cobre deseadas.

Al ser un proceso artesanal, conlleva tiempo y la calidad no siempre es la deseada. Además de no dar una solución generalizada al problema de escases de equipo de apoyo público en la facultad.

OBJETIVO

Mediante el conocimiento teórico y práctico adquirido en la carrera y en conjunto con habilidades interdisciplinarias, se buscará construir desde cero una máquina que mediante control numérico computacional sea capaz de devastar el cobre en tarjetas fenólicas para así dar paso al diseño de circuitos electrónicos.

OBJETIVOS ESPECÍFICOS

1. Ampliar el diseño de la máquina para poder realizar el devastado en otro tipo de superficies diferentes al cobre.

2. Realizar el devastado deseado con la calidad necesaria para un trabajo semi-profesional.

3. Se busca tener un bajo coste en el desarrollo e implementación de la máquina sin sacrificar la calidad deseable.

4. Utilizar programas libres para los requerimientos del sistema.

5. Brindar una alternativa a los estudiantes en la realización de sus circuitos electrónicos.

METODOLOGÍA

El método que se optará para obtener la solución al problema, es la creación de una máquina que mediante control numérico asistido por computadora sea capaz de devastar el cobre en tarjetas fenólicas para así obtener un circuito listo para el montaje de sus componentes.

Este método al ser automatizado nos permite auxiliar a gran número de personas sin generar desechos en grandes cantidades y por otro lado al ser un proceso estandarizado permite su fácil implementación entre la comunidad universitaria.

Innovar tecnología que cumpla con las especificaciones requeridas no siempre es fácil y mucho menos económico, sin embargo, de aquí nace la fuente de inspiración de nuestro proyecto pues se buscara obtener como producto final, tecnología de bajo costo que pueda ser más accesible obteniendo un buen balance entre costo- beneficio.

ORGANIZACIÓN DE LA TESIS

La presente tesis está estructurada de la siguiente manera:

El capítulo uno busca adentrarse de manera general al mundo de las máquinas de control numérico computacional, partiendo de un contexto histórico hasta llegar a tecnicismos tales como el “código g”.

En el capítulo dos se muestran los alcances de la máquina, dejando claro sus limitaciones, por otra parte, se muestra un panorama general de construcción.

En el capítulo tres se muestra la función que el microcontrolador desarrolla en el sistema, así como un panorama general de las diferentes familias de microcontroladores para que a partir de las necesidades se elija uno que sea capaz de cumplir los requerimientos del proyecto.

En el capítulo cuatro se revisará la función que ejercen los controladores de motores a pasos en el sistema y se elegirá el controlador con el que se trabajará en función a los requerimientos del sistema.

En el capítulo cinco se detallan los programas que se utilizan para la correcta implementación y funcionamiento del sistema.

En el capítulo seis se presenta el proceso de la construcción del sistema mecánico, así como los diagramas de ciertas partes que son clave para un desarrollo del sistema.

En el capítulo siete se presenta el sistema electrónico mostrando sus correspondientes diagramas de conexión.

En el capítulo ocho se presenta la integración del sistema dando una vista general de la máquina construida y los resultados de la misma.

En el capítulo nueve se muestran propuestas a futuro para la mejora del sistema.

CAPÍTULO 1

ESTADO DEL ARTE

1.1. Panorama histórico de las máquinas de control numérico computarizado

Las máquinas de control numérico nacen de la necesidad de automatizar procesos manuales para obtener una mejor calidad de trabajo en el menor tiempo posible.

A continuación, se toma nota de las principales fechas claves en su proceso de evolución, partiendo desde sus inicios, hasta llegar a un posible plano futuro.

Año 1770 d.c

Gracias a la revolución industrial se comienzan a construir y utilizar diversos tipos de máquinas para procesos de producción industriales.

Año 1949-1954 d.c

1949

- El concepto de Control Numérico por Computadora fue desarrollado y estudiado a fondo en el Instituto Tecnológico de Massachusetts.
- El principal objetivo de estos estudios fue crear una fresadora experimental que tuviera integrado un sistema CNC.

1951

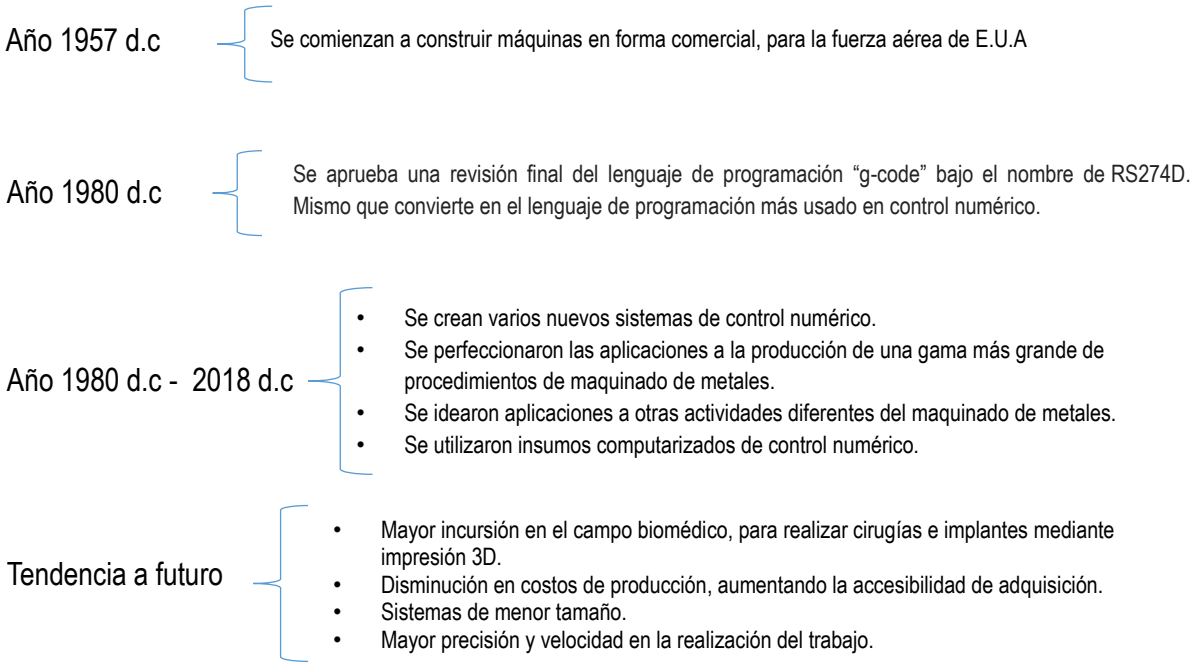
- Se realizó el ensamblaje de las piezas y las primeras pruebas de uso.

1951

- Se obtuvo la información suficiente para poder perfeccionar la máquina.

1954

- Se desarrolló un lenguaje primitivo llamado "Automatically Programed Tool" para la rudimentaria programación de la herramienta.
- Se comienzan a publicar artículos sobre este avance tecnológico.



En la actualidad las máquinas de control numérico son bastante utilizadas en la industria ya que pueden realizar de manera rápida un trabajo de calidad, además de que gracias a su versatilidad tienen un amplio campo de trabajo, llegando a desarrollar trabajos específicos con una buena relación costo/beneficio

1.2. Aplicaciones de las máquinas de control numérico computarizado

Las máquinas de control numérico computarizado tienen bastantes ventajas en comparación con otras máquinas que impulsan el desarrollo de la sociedad. A continuación, se presentan los tres campos más significativos donde estas tienen un impacto positivo.

Aplicaciones en la industria

La tecnología de control numérico computarizado permite que los movimientos de la maquinaria se realicen en forma programada, en lugar de realizarse a mano por el operador como era tradicional, lo cual hace que esta tecnología sea ideal para la producción en serie de piezas y para realizar un mecanizado complejo y/o de alta precisión, ya que libera al

operador de hacer trabajos repetitivos y también elimina los posibles errores en sus movimientos.

Por otro lado, los tornos con tecnología de control numérico computacional son actualmente indispensables en talleres donde se realizan producciones tanto propias como para terceros, reemplazan trabajos de torneado que anteriormente se realizaban con tornos rudimentarios.

Aplicaciones en la medicina

El campo de la medicina es un nicho de mercado en crecimiento que se caracteriza por su gran variedad de productos. A este sector pertenecen tanto los instrumentos y aparatos médicos como las prótesis y los injertos.

Por ello, se necesitan los más diversos procedimientos de producción y maquinaria especializada, tales como los controles numéricos computacionales, ya que con ellos se pueden construir los aditamentos e instrumental de precisión que en este ámbito se requiere.

Esta característica es aplicable tanto a la producción en serie como de piezas individuales que habitualmente tienen unas elevadas exigencias de precisión y calidad de superficie.

Aplicaciones en la construcción

Se ha innovando mucho en el campo de la construcción, puesto que al ser una tecnología que nos permite precisión y automatización se está empleando para la construcción de hogares.

Para lograr esto es necesario contar con maquinaria de grandes dimensiones, así como importantes procesos de potencia, por lo que tienden a ser procesos con costes elevados, pero nos da una idea bastante general de la fuerte inversión que tiene esta tecnología y nos deja ver un poco sobre su futuro próximo.

1.3. Futuro de las máquinas de control numérico computarizado

El mundo de la fabricación industrial se enfrenta actualmente a grandes retos. Un paso decisivo hacia la nueva era de la tecnología, vendrá dada por los avances que se vayan teniendo en los programas que controlen las máquinas de control numérico computarizado.

Para ello, los nuevos controles deberán ofrecer funciones estandarizadas que den respuesta a las necesidades existentes en diferentes áreas como pueden ser la virtualización y la interconexión con agentes de su entorno.

Los fabricantes de controles numéricos deberán adoptar una serie de estándares, creando una base sólida que ofrezca compatibilidad a los diferentes sistemas existentes en el mercado dando así vía libre a una tecnología mucho más abierta.

1.4. Características de las máquinas de control numérico computacional

Las características que presentan las máquinas de control numérico tienden a ser variadas, pero éstas están en función de su aplicación final, lo cual les genera una versatilidad única y les permite tener una gran calidad al desarrollar su objetivo. Por otro lado, también les genera desventajas desde el punto de vista del diseño y construcción de la misma, debido a que no se puede generalizar la construcción de las máquinas sin tener en cuenta el objetivo final para lo que fue diseñada.

Una característica importante a definir en este tipo de máquinas es la precisión con la que puede realizar su trabajo. Por ejemplo si ocuparemos la máquina para el área de la biomedicina, es necesario entender que la precisión es un factor primordial , puesto que es una área donde incluso distancias con errores de micrómetros son bastantes significativos. Por otro lado en el área de corte por láser en madera, no afectara mucho si el corte tiene un error grande, es por eso que cada máquina debe de cumplir con características específicas dependiendo el objetivo de su construcción.

Otra característica importante a considerar en cada una de las máquinas de control numérico computarizado es la potencia con la que realizara su corte. Continuando en el área biomédica, si es que se deseara hacer un corte preciso, posiblemente no es necesario tener

tanta potencia en la herramienta de corte, en cambio para fresar madera la herramienta debería de ser de una potencia mucho mayor a la utilizada en el área biomédica.

Algunas características como el tamaño de la máquina quedan implícitas en el objetivo que persigue la misma. Por ejemplo, si deseamos hacer barrido en cobre, posiblemente el área no tenga que ser tan grande, por otro lado, para el fresado en madera, el tamaño de la máquina tiende a ser bastante más grande en comparación con la anterior aplicación.

Aunque existen varios tipos de lenguaje para la automatización de máquinas en tres ejes, el “código G” es el más utilizado debido a su simplicidad, capacidad, precisión para dar órdenes y su estandarización en todo el mundo.

Es importante entender que estas características tanto variables como fijas se pueden unir para crear un nuevo modelo que integre las mejores características para realizar un mejor trabajo en el ámbito de la construcción de circuitos impresos. Es aquí donde entra la innovación y es un poco lo que se pretende hacer con el diseño y construcción de la máquina para los fines en un principio mencionados

1.5. Código G

El Código G o G-code, es el nombre que habitualmente recibe el lenguaje de programación más usado en control numérico computarizado.

En términos generales, G-code es un lenguaje mediante el cual las personas pueden indicarle a máquinas controladas por computadora qué hacer y cómo hacerlo. Esos "qué" y "cómo" están definidos mayormente por instrucciones que indican a dónde moverse, cuán rápido moverse y qué trayectoria seguir. Las máquinas típicas que son controladas con G-code son fresadoras, cortadoras, tornos e impresoras 3D.

En el presente proyecto se decidió adaptar la máquina a este tipo de lenguaje, puesto que es capaz de solventar los requerimientos de programación que puede presentar la misma y por otro lado su estandarización facilita el uso del mismo y esto permite reducir gastos considerablemente.

El G-Code se almacena en formato texto, es decir, puede leerse y modificarse con un editor de texto, aunque lo más habitual es que se genere y se visualice desde una aplicación de modelado y/o fabricación 3D o alguna herramienta o accesorio específico.

Para que la máquina pueda entender esos “que” y “como” mediante la implementación del G-code, se utilizan líneas en las cuales ya están definidos los procesos esenciales a realizar, tales como su posicionamiento, unidades en las que se trabaja y velocidad de movimiento.

El nombre G viene del hecho de que el programa está constituido por instrucciones Generales que comienzan con la letra G, aunque también existen instrucciones Misceláneas que de igual manera comienzan con la letra M.

El conjunto de instrucciones con el que cuenta este lenguaje es reducido y se muestra en el apartado de apéndice.

CAPÍTULO 2

DESCRIPCIÓN DEL PROYECTO

2.1. Logros y alcances de la máquina

Para que la máquina sea capaz de realizar el trazado de un circuito mediante el devastado de cobre, con la precisión y calidad necesarias para un trabajo semi-profesional se buscará que la distancia mínima de separación entre cada pista del cobre devastado sea de 2mm.

Por otro lado, para obtener la profundidad que podrá realizar el devastado de la máquina se entiende que ésta viene en función de la herramienta de devastado a utilizar y de la altura que exista respecto al “plano XY” y el “eje Z”; Se buscará que la altura mínima del eje z sea de 5cm.

En cuanto a la calidad de la profundidad de devastado que realice la máquina, ésta debe de ser tal que la generación de pistas esta pueda devastar el cobre sin perforar la paca fenólica y a la vez sea capaz de perforar dicha placa cuando se requiera.

Con base en los objetivos planteados el sistema debe poder ser aplicado en diferentes áreas de trabajo para darle otros usos a la máquina, tales como el grabado en madera, lo cual no dista de tener el mismo fundamento mecánico, aunque para obtener dicha incursión en tal campo es necesario tener una máquina robusta y de dimensiones considerables, así como el que la herramienta de devastado sea capaz de tener la potencia necesaria para hacer cortes a la altura de los requerimientos sin perder la calidad de la misma.

2.2. Limitaciones

La máquina tiene limitaciones ligadas al tamaño mínimo de las pistas, puesto que al ser un sistema semiprofesional y no contar con el presupuesto ni los materiales adecuados no se puede obtener una resolución adecuada para pistas menores a 2 mm, con lo que se puede descartar la creación de circuitos para aplicaciones en el área biomédica o donde se requieran pistas con un grosor menor a 1 mm.

Por otro lado, se tiene como parámetro la velocidad a la que la máquina puede trabajar ya que esta va en función a lo rápido que los motores pueden recibir instrucciones sin perder pasos, con un presupuesto acortado se buscará tener una velocidad máxima cercana a los 300 mm / s, misma que se encuentra lejos de una máquina de control numérico computarizado de última generación.

2.3. Diferencias con otras máquinas

La aplicación de máquinas de control numérico es creciente, lo que genera una gran cantidad de las mismas en el mercado y por lo mismo una gran variedad de diferencias entre ellas.

Relación costo/beneficio elevado.

Al ser una máquina prototipo, nos permite tener una ventaja significativa en cuanto a la reducción del costo de refacciones que se llegasen a necesitar.

Simplicidad

Se busca que la estructura de la máquina sea lo más simple posible, esto en parte para reducir costos y por otro lado para que el sistema funcione con los mecanismos necesarios sin llegar a la ostentosidad.

Por otro lado, el hecho de lo simple conlleva a que el proceso de mantenimiento y de refacción sea sencillo y rápido.

Si bien esto trae beneficios, también existe la desventaja de carecer de algunas funciones que podrían mejorar el sistema, tales como el ambiente controlado o la medición del correcto nivel de altura.

Versatilidad

Desde el inicio se planteó que en medida de lo posible la máquina tenía la capacidad de poder incursionar en otros campos mediante la sustitución de la herramienta de devastado por materiales de corte como láser o inclusive, utilizar el esquema para una máquina de impresión en 3D. Lo cual hace a la máquina una excelente herramienta para un gran catálogo de opciones y le da ventaja sobre su competencia.

2.4. Panorama general de construcción.

Hasta el momento poco se ha hablado de cómo se llevará a cabo el proyecto, por lo que a continuación se describirá un panorama general de la construcción.

El proyecto en cuestión requiere que interactúe la programación con el sistema mecánico para poder realizar su trabajo.

Los programas se utilizan para la generación de código G y para poder generar la interfaz de usuario, además de que no se limita en esto, pues cualquier microcontrolador que se utilice para hacer las veces del cerebro de la máquina, requiere programación para poder trabajar como un lector e intérprete del código G que se le proporcione.

Por otro lado, se requiere de un sistema mecánico para poder realizar el trabajo, mismo que no es poco y es tan importante como la programación del sistema. Para empezar, se requiere de un microcontrolador, siendo éste el que recibe el código G, lo interpreta y ejecuta. Posteriormente será necesario el uso de controladores de motores a pasos para servir como acoplo entre el microcontrolador y los motores pasos, no sólo porque el microcontrolador ejecuta el código G como una serie de pulsos con diferente frecuencia y los controladores de motores a pasos son los que deben de indicar la dirección y tiempo en el

que motores a se activen, sino también porque son los que acoplan los diferentes voltajes de trabajo entre el microcontrolador y el sistema de tres ejes. Finalmente se requiere un sistema mecánico de tres ejes conformado por motores a pasos que sea capaz de sostener una herramienta de devastado, misma que se pueda mover libremente en un plano de tres dimensiones sin ningún problema.

La construcción del sistema mecánico no es simple, además la precisión de acople de las piezas mecánicas es crucial para un correcto devastado.

A continuación se muestra un diagrama general de cómo interactúa el sistema mecánico y la programación del sistema. En los capítulos subsecuentes se describirá la implementación de cada uno de los bloques del diagrama.

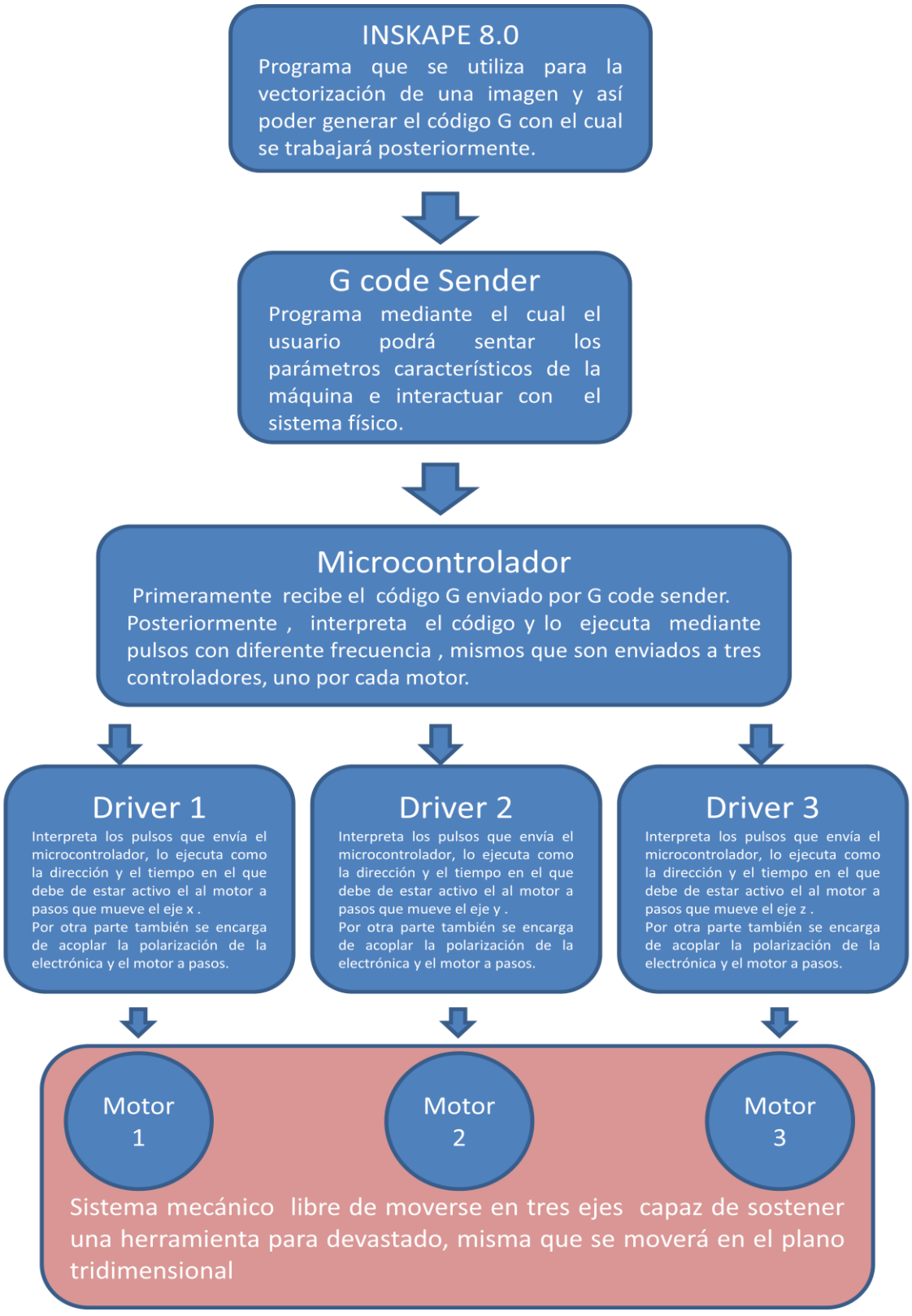


Figura 2.1. Diagrama general de construcción.

CAPÍTULO 3

MICROCONTROLADOR

3.1. Introducción a los microcontroladores

Un microcontrolador es un circuito integrado digital que puede ser usado para diversos propósitos debido a su capacidad de ser programable. Está compuesto por una unidad central de proceso, memorias y líneas de entrada y salida llamados periféricos.

Como su estructura ya viene integrado en un solo chip, para usar un microcontrolador se debe especificar su funcionamiento por código a través de programas que indiquen las instrucciones que el microcontrolador debe realizar.

Un microcontrolador se compone de al menos un microprocesador, periféricos y memoria.

Donde un microprocesador se compone de al menos una unidad aritmética y lógica, registros y una unidad de control.

La unidad aritmética y lógica está compuesta por circuitos electrónicos digitales del tipo combinatorios (compuertas, sumadores, multiplicadores), cuya principal función es el realizar operaciones. Estas operaciones están divididas en tres tipos:

- Lógicas. Son las operaciones básicas que realizan las compuertas lógicas, como la suma lógica, multiplicación lógica y negación. Una operación lógica sólo puede tener como entradas y como salidas una respuesta lógica (0 o 1). Esto dependiendo de los niveles de voltajes de una señal digital.

- Aritméticas. Las operaciones aritméticas son la suma, resta, multiplicación y división. Dependiendo del procesador (8, 16, 32 o 64 bits) será la rapidez con la que se pueden hacer dichas operaciones.

- Misceláneas. En estas operaciones caen todas las demás operaciones como la transferencia de bits.

Los registros son las memorias principales de los procesadores, ya que funcionan a la misma velocidad que el procesador.

La unidad de control es el conjunto de sistemas digitales secuenciales (aquellos que tienen memoria) que permiten distribuir la lógica de las señales.

Un microprocesador también está compuesto por Periféricos y Memoria, siendo esta la principal diferencia entre un microcontrolador y un microprocesador.

Sus componentes se definen de la siguiente manera:

Periféricos. Son los circuitos digitales que nos permiten una interacción con el mundo "exterior" al microcontrolador. Su función es la de poder habilitar o deshabilitar las salidas digitales, comunicación con terminales digitales o sacar señales analógicas de una conversión digital.

- Puertos de entrada/salida. Los puertos están relacionados al tamaño del procesador, es decir que un puerto de 8 bits es porque el procesador es de 8 bits. Un procesador de 64 bits, tiene la capacidad de tener un puerto de 64 bits.

- Puertos seriales. Nos permiten transformar la información digital paralela (bytes de información) en tramas que se pueden transferir por una o varias líneas de comunicación.

- Periféricos analógicos. Como los que convierten señales analógicas a digitales o señales digitales a analógicas o comparadores analógicos.

- Memoria. La memoria está dividida en tres. La memoria para el programa, la memoria para los datos o variables del programa y la memoria para configuraciones o no volátil.

Para darse una idea general de la composición interna de un microcontrolador, se muestra el siguiente diagrama:

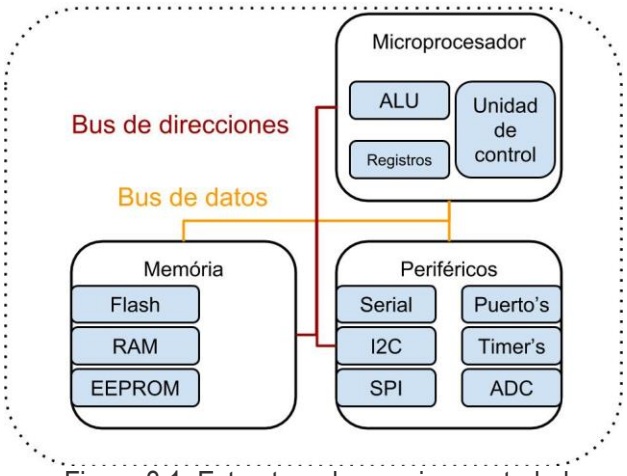


Figura 3.1. Estructura de un microcontrolador.

La arquitectura de un microcontrolador permite definir la estructura de su funcionamiento, las dos arquitecturas principales usadas en la fabricación de microcontroladores son: arquitectura de Von Neumann y arquitectura Harvard.

Para la arquitectura Von Neumann, los *datos* y las instrucciones circulan por el mismo bus ya que estos son guardados en la misma memoria, su principal ventaja es el ahorro de líneas de entrada-salida, pero esto supone una disminución en la velocidad con la que se realizan los procesos.

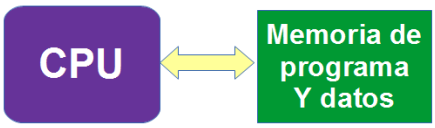


Figura 3.2. Arquitectura Von Neumann.

En la arquitectura Harvard existe una memoria específica para datos y una memoria específica para las instrucciones, de esta forma se usan dos buses bien diferenciados. Con esto se logra trabajar con las dos memorias simultáneamente y en consecuencia se obtiene mucha más velocidad en la ejecución de los programas.

Figura 3.3. Arquitectura Harvard.

En cuanto a los tipos de procesadores que existen se encuentran los siguientes:

Procesador de tipo CISC

Es un procesador que permite manejar un amplio juego de instrucciones es llamada de tipo *CISC* que en español significa "Ordenador con Juego de Instrucciones Complejo", programar en este tipo de arquitectura requiere en algunos casos del dominio de hasta centenares de instrucciones.

Procesador de tipo RISC

Este procesador está diseñado para manejar pocas instrucciones, pero sin afectar las prestaciones del ordenador, es llamado de tipo *RISC* que en español significa "Ordenador con Juego de Instrucciones Reducido", esto permite programar con mucha más facilidad y, por si fuera poco, los circuitos de tipo *RISC* disponen de una estructura que busca como mínimo la instrucción próxima a ejecutar mientras realiza la instrucción actual. Esta estructura permite lograr no solo mayor velocidad de proceso sino también procesar cada instrucción con la misma velocidad.

3.2. Función del microcontrolador en la máquina de control numérico

Como previamente se indicó, al microcontrolador se le puede percibir como el "corazón" de la máquina y esto es porque en general en el recaen 3 trabajos bastante importantes.

1 Lectura de instrucciones.

El primer trabajo del cual se encarga el microcontrolador es de recibir líneas en forma de código G.

Para lograr esto es importante recalcar que primeramente se debe de programar el microcontrolador para que su única función sea la de trabajar con código G.

2 Interpretación de las instrucciones

El segundo trabajo que realiza el microcontrolador es el de interpretar las instrucciones que se le envían, en forma de código G.

3 Enviar la información a los controladores de motores a pasos

El último trabajo que realiza el microcontrolador es el de enviar la información procesada a cada uno de los tres controladores de motores a pasos, uno por cada eje.

Esta información la envía mediante dos señales digitales, mismas que contienen la información dada por el código G.

Estas señales posteriormente serán interpretadas por los controladores de motores a pasos y se reflejarán como la dirección de giro y el tiempo de activación de cada uno de los motores a pasos.

3.3 Elección del microcontrolador

El microcontrolador a elegir debe de ser capaz realizar las tres anteriores tareas, además de cumplir con ciertos requisitos, algunos de ellos simples pero necesarios. Tales como:

- Puertos de entrada necesarios y suficientes.
- Capacidad para recibir e interpretar el código g.
- Fácil adquisición, en caso de reparación.

Revisando las diferentes arquitecturas de microcontroladores, para nuestra aplicación es conveniente utilizar la arquitectura Harvard, pues al tener una memoria específica para datos y otra para las instrucciones tendremos más velocidad al ejecutar el código G y por lo mismo la máquina utilizara un tiempo menor para realizar su trabajo.

En cuanto al procesador, lo mejor sería manejar uno de tipo RISC, para así lograr no solo mayor velocidad de proceso sino también procesar cada instrucción con la misma velocidad, todo esto para que el tiempo de trabajo desarrollado por máquina sea aún menor ya que esto no solo genera un proceso de fabricación corto, sino que también evita el calentamiento de motores a pasos por un trabajo prolongado.

Para tomar una decisión acertada en la elección del microcontrolador se tomaron en consideración los cuatro siguientes modelos:

- MSP430G2553 de Texas Intruments.
- ATmega328 de Atmel.
- PIC16f887 de Microchip.
- MKS Base incorporando el microcontrolador ATmega2560 de Atmel.

Todos son capaces de desarrollar las funciones requeridas por el sistema descritas anteriormente, además de las siguientes características que se describen a continuación:

MICROCONTROLADOR	Voltaje de alimentación	Precio	No bits	Frecuencia de operación	Desarrollo en adaptación de código G
MSP430G2553	1.8[v]-3.3[v]	350 MNX$	16	0[Hz]-16[MHz]	Nulo
ATmega328	1.8[v]-5.5[v]	150MNX$ - 450 MNX$	8	0[Hz]- 20[MHz]	Bastante
PIC16f887	3.3[v]-5[v]	120 MNX$	8	0[Hz]- 20[MHz]	Poco
MKS Base	12[V]-24[v]	820 MNX$	8	0[Hz]-16[MHz]	Bastante

Tabla 3.1 Comparación de 4 modelos de microcontroladores para la implementación en el sistema.

En cuanto al microcontrolador MSP430G2553 podemos observar que cuenta con las características necesarias para continuar trabajando con el mismo, su principal desventaja es la falta de desarrollo en la adaptación de código G.

El microcontrolador ATmega328 cuenta con una buena cantidad de desarrollo en la adaptación del código G y cumple con las especificaciones necesarias, por otro lado, su costo es accesible, obteniendo una ventaja significativa.

El microcontrolador PIC16f887 cuenta con poco desarrollo en la adaptación del código G, pero por otro lado su costo es bajo y cumple con los parámetros necesarios para la implementación en el sistema.

En cuanto a la utilización de la placa MKS Base se puede notar que su precio es elevado, sin embargo, cuenta con bastante desarrollo para la implementación en el sistema, además que tiene la característica que esta optimizada para impresoras 3D lo que da un amplio margen de desarrollo y permite la adaptación para un mayor número de motores en caso de requerirse.

Finalmente se optó por utilizar el microcontrolador ATmega328, esto debido a que además de cumplir con los requerimientos del sistema cuenta con bastante desarrollo para la adaptación de los motores a pasos y su precio no es elevado como el uso de la placa MKS Base.

La siguiente figura muestra la física del microcontrolador ATmega328.

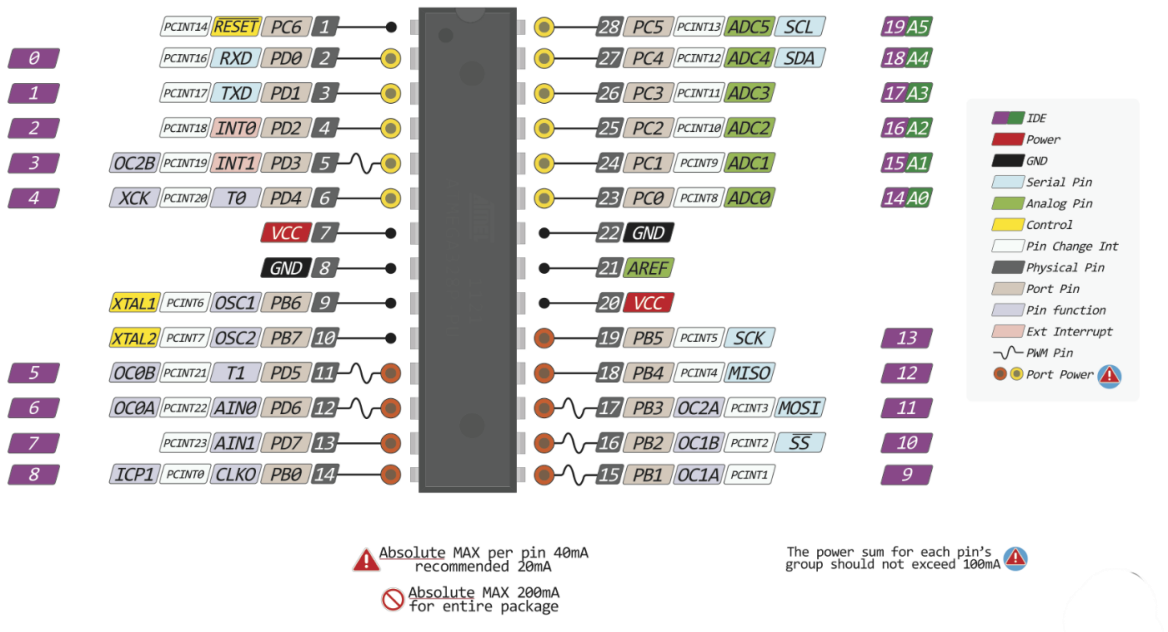


Figura 3.4. Microcontrolador Atmega328.

Se optó por trabajar con la tarjeta desarrollo “Arduino UNO”, donde ya se encuentra instalado el microprocesador Atmega328 esto para conseguir que el tamaño de la electrónica se reduzca y tener un montado más simple, además de permitirnos en un futuro poder remplazarlo por algún otro microcontrolador que pudiera cumplir las veces que hace el ATmega328. Esta última idea se siguió hasta la construcción final de la máquina dejando el espacio necesario y una PCB con el diseño desmontable.

A continuación se muestra la siguiente figura donde se aprecia el aspecto físico de la tarjeta de desarrollo “Arduino uno”.

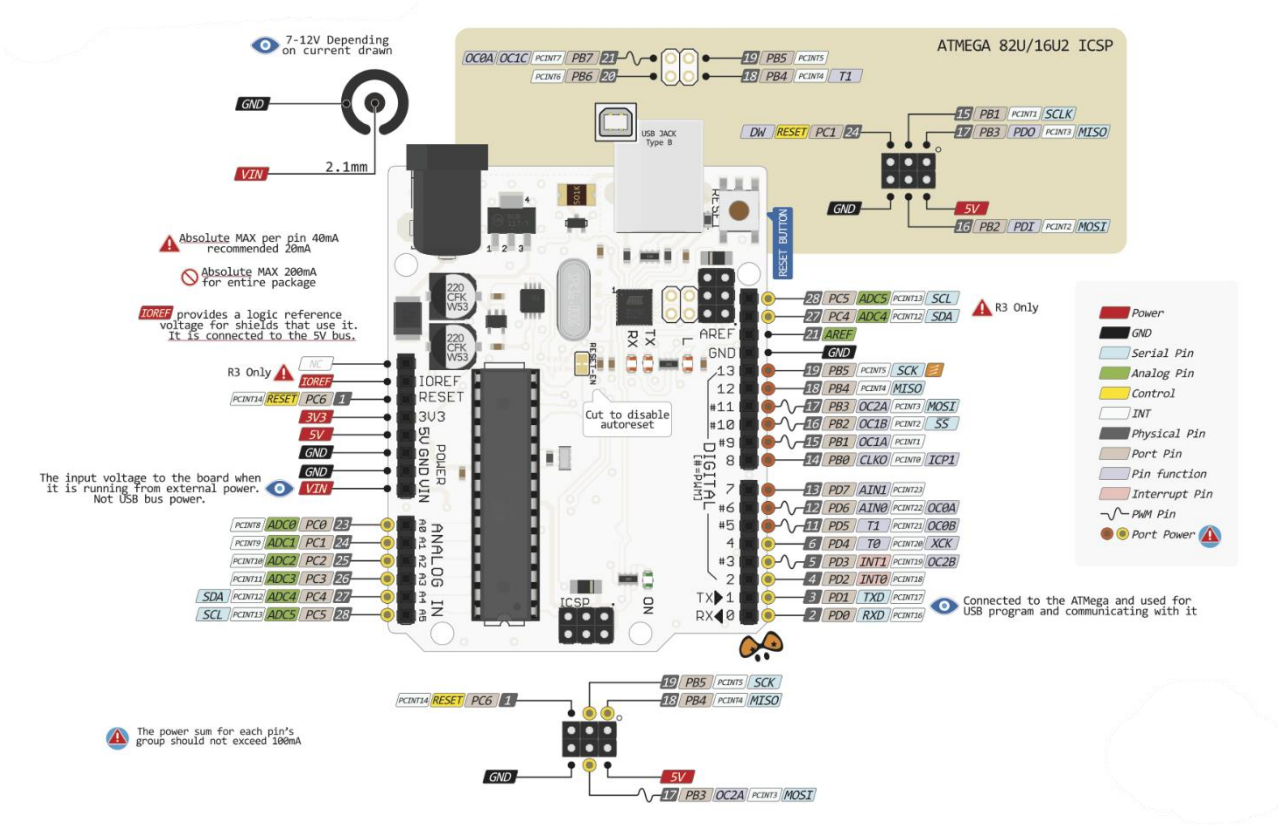


Figura 3.5. Tarjeta de desarrollo Arduino UNO.

CAPÍTULO 4

CONTROLADORES DE MOTORES A PASOS

4.1. Introducción a los controladores de motores a pasos

Los controladores de motores a pasos permiten proporcionar la potencia y dar el sentido y la velocidad con la que girarán los motores a pasos. Gracias a estos los motores son capaces de convertir la energía eléctrica en un movimiento mecánico deseado y preciso.

Los controladores de motores a pasos son los primeros en recibir las instrucciones de dirección de giro e interpretan el pulso generado por el microcontrolador como el número de pasos que debe dar el motor.

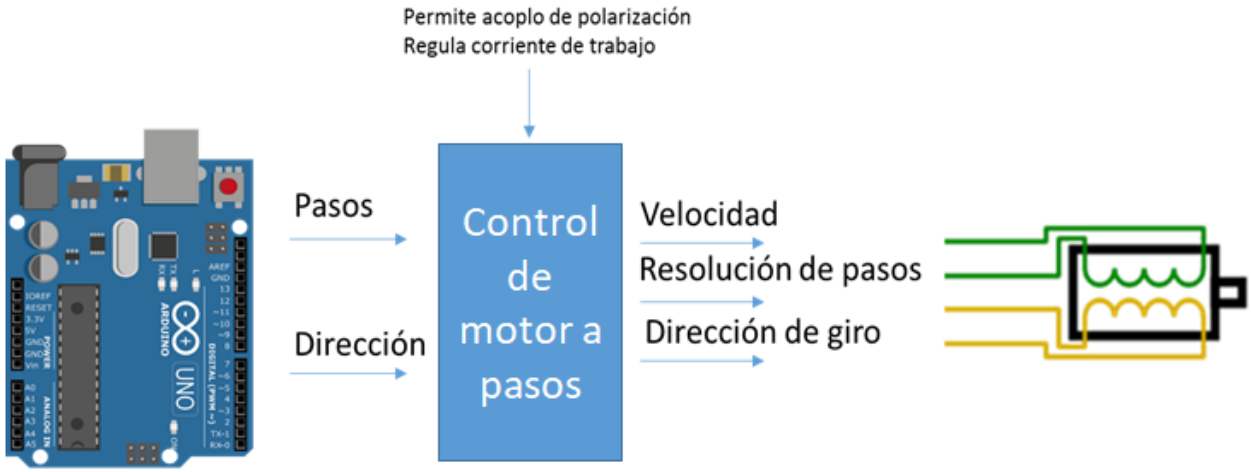


Figura 4.1. Interconexión de elementos.

Al ser una máquina con tres ejes coordenados, permitiendo el desplazamiento de cada uno mediante el accionar de un motor, la máquina debe de contar con tres controladores de motores a pasos, uno por cada eje, permitiendo así el libre desplazamiento en el plano tridimensional.

4.2. Función del controlador del motor a pasos en la máquina de control numérico

El controlador tiene 5 tareas importantes en la estructura del sistema, siendo estas las siguientes:

Interfaz

Primeramente, hace las veces de la interfaz entre el microcontrolador y los motores a pasos.

Acoplamiento

Sirve como acoplador de las diferencias de potencial entre la polaridad de los motores a pasos, el microcontrolador y su propia alimentación lógica.

Regulador

Regula la corriente de trabajo de los motores a pasos

Resolución de pasos del motor

Adecua la configuración de la resolución de los pasos del motor

Interpretación

Por último interpreta la señal que le envía el microcontrolador a los motores a pasos, tanto como del sentido en el que deben de girar los mismos, la velocidad que esto deben tomar y la cantidad de pasos que estos deben de dar.

4.3. Elección del controlador de motor a pasos

El controlador a elegir, además ser capaz de desarrollar sus funciones en el sistema, debe de cumplir con ciertas características para poder acoplarse con los motores a pasos seleccionados, tales como:

- Ser compatible con distintos microcontroladores, esto por si en un futuro se desea extender los alcances de la máquina.
- Ser funcional para motores a pasos bipolares.
- Manejar el rango de voltaje y amperaje correcto para la alimentación de los motores a pasos.
- Tener la capacidad de seleccionar la resolución de los pasos del motor.

Para tomar una decisión acertada en la elección del controlador se tomaron en consideración los tres siguientes modelos:

- DRV8825 de Pololu
- A4988 de Pololu
- TB6600HG de Toshiba

Todos son capaces de desarrollar las funciones requeridas por sistema descritas anteriormente, además de las siguientes características que se describen a continuación:

Controlador de motor a pasos	Voltaje máximo de alimentación para los motores a pasos.	Amperaje máximo de alimentación para los motores a pasos.	Resolución de pasos.	Funcional para motores a pasos bipolares	Compatibilidad con diferentes microcontroladores	Precio
DRV8825	45 [v]	2.2 [a]	$1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$	Sí	Sí	45 MXN$
A4988	35 [V]	2 [a]	$1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$	Sí	Sí	25 MXN$
TB6600HG	50 [v]	5 [a]	$1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$	Sí	Sí	400 MXN$

Tabla 4.1. Comparación de tres diferentes modelos de controladores de motores a pasos.

Basándonos en la comparativa anterior se optó por utilizar el controlador de motor a pasos A4988 de Pololu , debido a que este cumple con todas las especificaciones necesarias y su precio es el más accesible del mercado , lo que aporta a cumplir el objetivo de desarrollar un sistema económico. Sin embargo el controlador DRV8825 se presenta como una opción viable por no tener tan elevado precio y su mayor capacidad de manejo de voltaje y amperaje, así como una mayor resolución de pasos. Por otro lado el controlador TB6600HG se encuentra con un exceso de capacidad y precio, por lo que para en el actual sistema se descarta como opción.

Las características completas y pines de conexión del controlador elegido se presentan a continuación:

Voltaje de operación mínimo	8 V
Voltaje de operación máximo	35 V
Mínima corriente por fase	1 A
Mínima corriente por fase	2 A
Voltaje lógico mínimo	3 V
Voltaje lógico máximo	5 V
Resolución de pasos	Completo , 1/2, 1/4, 1/8, 1/16
Temperatura de trabajo	De -20°c a 80°c

Tabla 4.2. Características del circuito integrado a4988

Figura 4.2. Terminales del controlador a4988.

Para evitar la pérdida de pasos en el motor se optó por una resolución completa ya que esta evita en lo posible que la velocidad disminuya, en la siguiente figura extraída de las especificaciones del fabricante del controlador, se puede apreciar que para esta resolución la corriente tiene un máximo de 70.71 % siendo un dato importante a tomar en cuenta en la regulación de corriente de los motores.

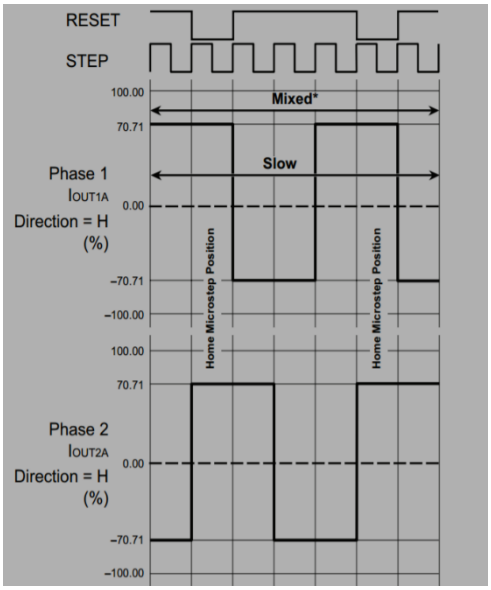


Figura 4.3. Variación de la corriente en la resolución completa de pasos.

Para configurar la resolución de pasos elegida se utilizó la siguiente tabla, también extraída de las especificaciones del fabricante.

Ms1	Ms2	Ms3	Resolución de Pasos
Bajo	Bajo	Bajo	Paso completo
Alto	Bajo	Bajo	Medio paso
Bajo	Alto	Bajo	Cuarto de paso
Alto	Alto	Bajo	Octavo de paso
Alto	Alto	Alto	Dieciseisavo de paso

Tabla 4.3. Tabla de verdad para la resolución de los pasos del motor

A continuación, se presenta un esquema general de las conexiones que se tendrían tomando en cuenta la implementación de dicho controlador.

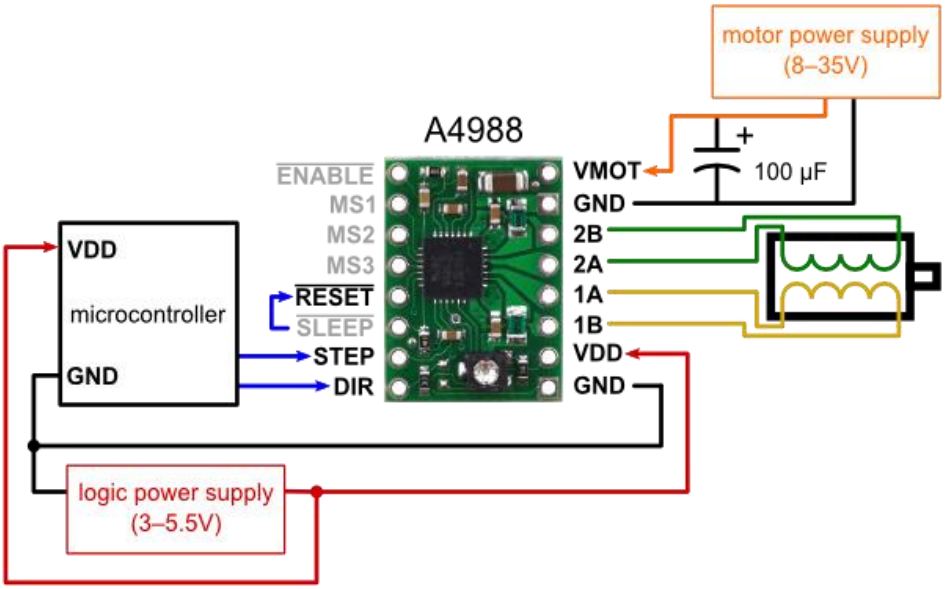


Figura 4.4. Diagrama de conexión del controlador a4988 de Pololu.

Al ser un controlador convencional para propósitos generales, la regulación de corriente con la que trabajaran los motores a pasos se hace de manera manual.

El proceso de regulación de corriente se describe a continuación:

1.- Primeramente, se establecen los parámetros de funcionamiento para los motores a pasos, en este caso los valores convencionales de polarización de los motores en voltaje son de 12 volts a 1.2 amperes. Para este ejemplo se tomarán los valores escritos previamente.

2.- De la siguiente ecuación obtenida de las especificaciones del fabricante se puede obtener el voltaje de referencia "Vref" con el que la regulación de corriente de los motores a pasos estará completa.

$$I\,\text{trip} = \frac{V\,ref}{8\,Rs} \qquad \textit{Ecuación 4.1}$$

Donde:

$I\ trip$ es la corriente de trabajo de los motores a pasos.

$Vref$ es el voltaje de referencia

Rs es la resistencia de censado, misma que se incluye en el integrado del controlador

Para obtener la corriente de trabajo de los motores $I\ trip$ se aplica la siguiente formula

$$Itrip = (\%I\max/100)\ (\ I\ max) \qquad \textit{Ecuación 4.2}$$

Donde:

$I\ max$ es la corriente máxima de trabajo de los motores

$\%\ Imax$ es el porcentaje de corriente alcanzado en nuestra resolución de pasos

El valor de la resistencia de censado Rs varía en función del fabricante del circuito integrado, pero se sitúa en la misma posición y de esta manera podemos obtenerla, su valor debe de ser expresado en ohms.

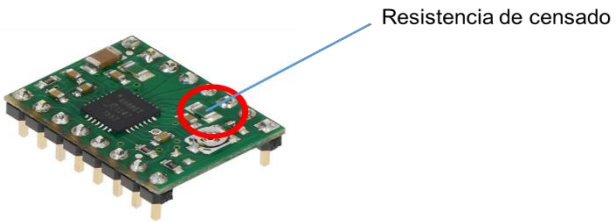


Figura 4.5. Ubicación de la resistencia de censado.

3.- Se obtiene el valor de $I\ trip$ sustituyendo valores en la ecuación 4.2

$$Itrip = \left(\frac{70.71\%}{100}\right)(1.2[a]) \qquad \textit{Ecuación 4.3}$$

$$Itrip = 0.84852[a] \qquad \textit{Ecuación 4.4}$$

3.- Se procede a despejar $Vref$ de la ecuación 4.1 para posteriormente sustituirlos valores de $Itrip$ y Rs para así encontrar el valor de $Vref$, para este ejemplo tomaremos el valor de Rs como 0.1 ohm

Despejando $vref$ de la ecuación 4.1 se obtiene

$$Vref = Itrip(8Rs) \qquad \textit{Ecuación 4.5}$$

$$Vref = (0.84852[a]\,)\,(8\,(0.1\,[\Omega])) \qquad \textit{Ecuación 4.6}$$

$$Vref = 0.678816\,[v] \qquad \textit{Ecuación 4.7}$$

4.- Se polariza el circuito integrado sin conectar el motor a pasos ni su alimentación.

5.- Se ubica el tornillo de regulación mismo que nos indicara el voltaje de referencia $vref$.

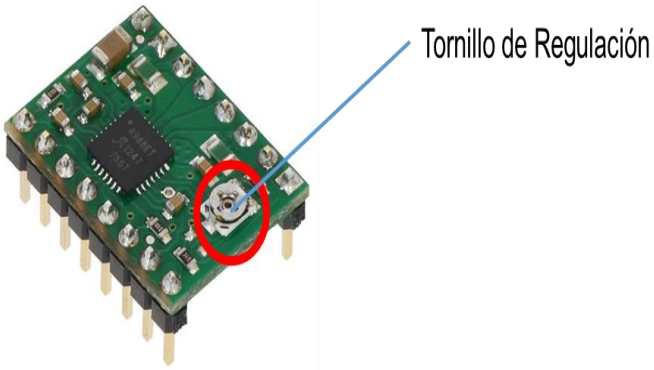


Figura 4.6. Ubicación del pin de $Vref$ y el Tornillo de Regulación.

6.- Con un voltímetro y el valor de voltaje en $Vref$, se procede a ajustar el tornillo de regulación hasta llegar al voltaje de $Vref$ requerido para la correcta polarización del motor a pasos.

CAPÍTULO 5

PROGRAMACIÓN DEL SISTEMA

5.1. Visión general de los programas del sistema

En el presente proyecto no se busca el desarrollo de nuevo software, sino la implementación del ya existente y cuenta con los siguientes requerimientos:

1.-Generar código G que cumpla con las especificaciones del usuario.
2.-Programa para que el microcontrolador acepte el código G y lo procese.
3.- Adecuar el código g a las características físicas de la máquina.
4.- Interfaz gráfica entre el usuario y la máquina.

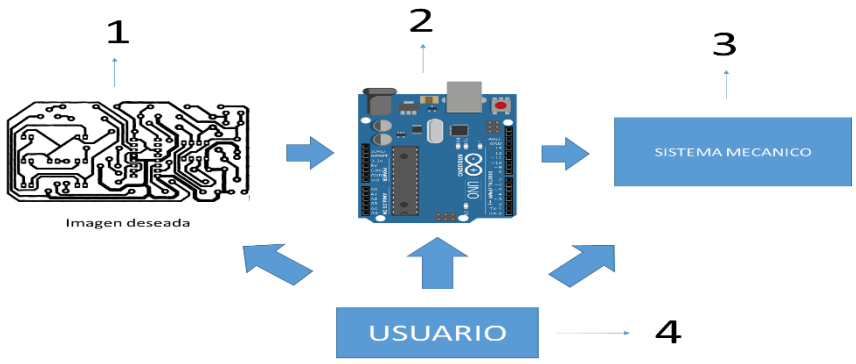


Figura 5.1. Requerimientos de programación.

Con esto en mente podemos descartar algunas cosas menos relevantes como la compañía desarrolladora de la paquetería de programas y la necesidad de que esta sea de coste, pues consideramos que lo más importante para este caso es el hecho de que esta paquetería se limite a cumplir los objetivos anteriores.

5.2. Elección de los programas del sistema

El programa más adecuado para cumplir con los requerimientos del primer punto de requerimientos de programación, es el llamado Inskape, esto debido a que es una paquetería libre y robusta. Este programa cuenta con la opción de generar el código G a partir de una imagen o texto dado por el usuario, además de tener actualizaciones constantes.

Para cumplir con los requerimientos del segundo punto el programa deber ser implementado en el microprocesador seleccionado (atmega328).

Se encontró que existe un programa libre llamado Grbl que mediante código g permite al microcontrolador establecer el movimiento de los motores a pasos de las máquinas y éstas puedan desarrollar un trabajo específico. Este programa está optimizado para el microcontrolador atmega328, además de que es utilizado en la gran mayoría de máquinas fresadoras, de corte láser y de impresión en 3D de software libre, dando una versatilidad al sistema, por lo que se optó por trabajar con el mismo.

Para hacer uso del programa Grbl e implementarlo en el controlador, el desarrollador presenta una serie de instrucciones, mismas que se detallan en el apartado de apéndice.

Para cumplir con los objetivos de los puntos tres y cuatro se optó por utilizar el programa Universal Gcode Sender mismo que es recomendado por el desarrollador del programa Grbl.

Universal Gcode Sender cuenta con una interfaz para interactuar con el usuario, además de que permite configurar y acoplar las características físicas de la máquina para que las instrucciones dadas por el usuario sean ejecutadas de la manera esperada.

En el apartado de apéndice se describe el proceso de configuración de las características físicas de la máquina, así como una tutorial básico de uso del programa Universal Gcode Sender.

5.3. Programa de control para motores a pasos

Grbl es un programa gratuito que sirve para controlar el movimiento en motores a pasos, este se encuentra optimizado para él microcontrolador atmega328. Actualmente cuenta con tres versiones principales, la versión 0.8 ,0.9 y 1.1, la última acaba de ser lanzada en julio del 2018, aún sigue en desarrollo y puede ser inestable según el desarrollador.

Para el desarrollo del proyecto se trabajó con la versión 0.8 y la 0.9 esto debido a que ambas versiones cuentan con diferentes opciones de configuración para el sistema mecánico por lo que fue necesario encontrar la versión que más se adecue al sistema.

Para instalar ambas versiones de GRBL en el microcontrolador atmega328 el desarrollador marca una serie de opciones entre las cuales destaca el proceso de carga de datos en la memoria del microcontrolador, lo que permite a este configurarse para recibir código G e interpretarlo, este proceso se describe en el apartado de apéndice.

Primeramente, se trabajó con la versión 0.8, para esto, se desarrolló su plantilla de circuito impreso y se implementó en una tarjeta fenólica, para así poder tener un espacio destinado a las terminales de los motores a pasos, la alimentación del circuito y ubicar a los controladores de motores a pasos en los pines correspondientes del microcontrolador atmega328. El proceso de desarrollo e implementación de la tarjeta fenólica se detalla en el apartado de apéndice.

Una vez ya instalada la versión 8.0 de Grbl, y conectando la tarjeta fenólica desarrollada, se procedió a configurar las características mecánicas de la máquina mediante Universal G code Sender para así poder controlar el sistema mecánico.

Los resultados de este primer proceso no fueron los esperados, en primera instancia los motores perdían pasos, emitían ruidos no deseados, iban lentos y el diseño hecho en código G no se cumplía, esto se debía a la inexperiencia para configurar parámetros, la falta de varios de ellos, así como la inestabilidad del programa.

Posteriormente se trabajó con la versión 0.9 de Grbl, para esto se utilizó la misma placa fenólica desarrollada para la versión anterior, esto debido a que la configuración de los pines sigue siendo la misma.

El desarrollador recomienda la utilización de su propia placa de instalación, llamada CNC Shield con la cual trata de minimizar el espacio de la electrónica, tener un aspecto más profesional y no perder de vista todas las utilidades de su programa. Al ser un producto económico (55MXN) se decidió adquirir la misma, aunque el sistema bien puede trabajar con la placa fenólica desarrollada previamente.

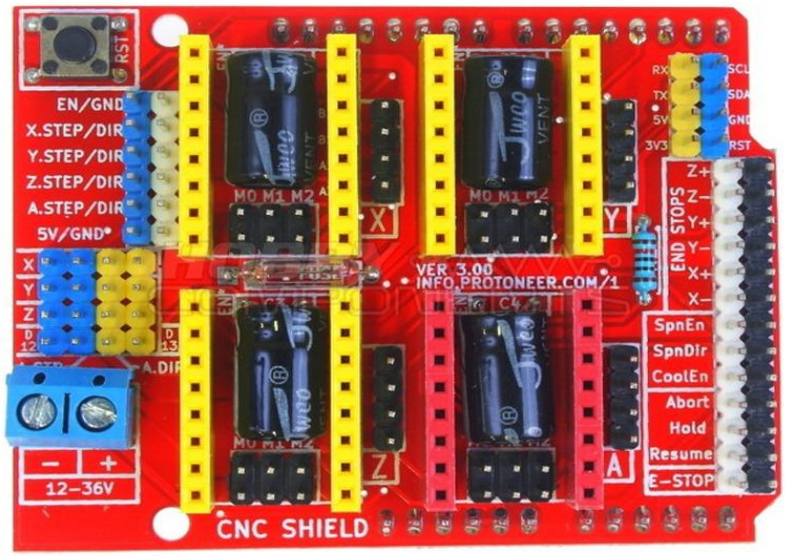


Figura 5.2. CNC Shield.

El resultado al implementar esta nueva versión del programa fue diferente a la anterior, ya que con la experiencia adquirida y con las configuraciones añadidas, los motores dejaron de perder pasos y moverse más fluidamente, lo que permitió que el código G se ejecutara de la manera deseada.

CAPÍTULO 6

SISTEMA MECÁNICO

6.1. Descripción del sistema mecánico

La estructura física del proyecto cuenta con cuatro partes fundamentales que en conjunto permiten el movimiento libre de la herramienta de devastado o corte en tres ejes coordenados.

Tres de estas estructuras tienen como función sostener un eje de movimiento, mientras que la cuarta es el espacio destinado para la ubicación de los componentes electrónicos de la máquina y a su vez se utilizará como base de montaje de las estructuras que sostienen los ejes.

A continuación, se muestra una figura que pretende dar una idea más concreta de la física de las estructuras.

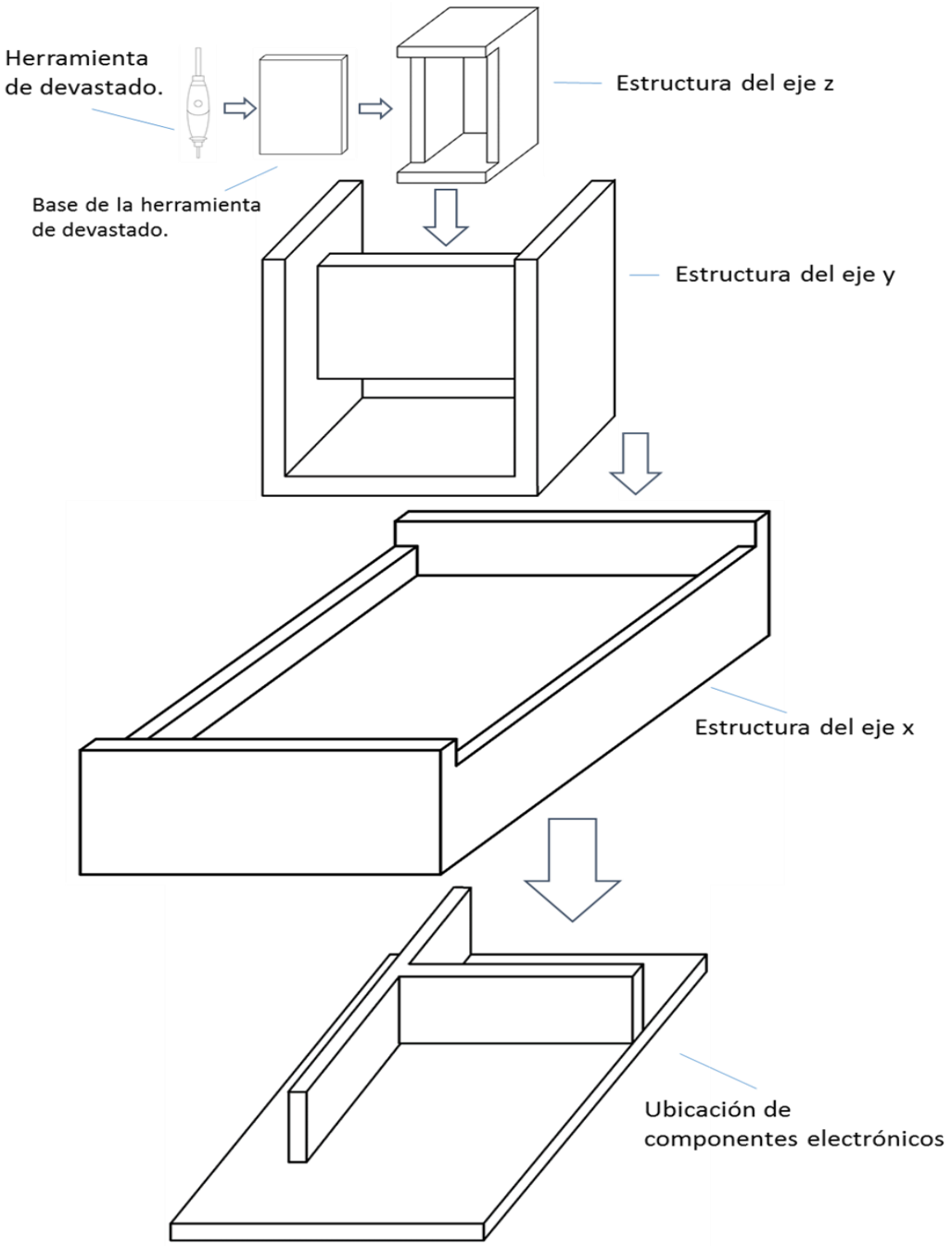


Figura 6.1. Panorama general de la estructura de la máquina.

Para que la máquina pueda desplazarse libremente, cada estructura que sostiene un eje coordenado cuenta con un mecanismo de movimiento capaz de sostener tanto la base de su eje como a los ejes subsecuentes.

Esto se logra gracias a la implementación de un mecanismo basado en el acoplamiento de un motor a pasos con una varilla roscada, donde la varilla a su vez va dentro de un husillo que sostiene el eje coordenado, permitiendo así el movimiento en uno de los tres ejes coordenados.

En el eje coordenado z será montada la herramienta de devastado, siendo posible cambiar la misma dependiendo de las necesidades de corte.

En lo subsecuente del capítulo se profundizará en cada una de las estructuras antes mencionadas, así como en la mecánica de movimiento de los ejes y del montaje de herramientas de devastado.

6.2. Ubicación de componentes electrónicos

El proyecto en cuestión requiere de una variedad de equipo electrónico, no basta con seleccionarlo y trabajar con él, también es necesario ubicarlo físicamente en una parte estratégica del sistema.

A partir de esto se diseñó una estructura en donde se montaron los componentes y se pueda maniobrar libremente en cualquier momento.

Se buscó que dicha estructura estuviera dentro del sistema y sea tanto visible para trabajar los componentes montados.

Para tener una idea más general de la estructura se enumeraron los componentes y sistemas electrónicos que irían montados sobre ella, los cuales se listan a continuación:

- Fuente de poder
- Microcontrolador
- Controlador de motor a pasos por cada eje (3 en total)
- Contacto a corriente
- Interruptor de encendido
- Entrada de USB
- Cable para conexiones entre elementos

La primera parte de la estructura es de la siguiente manera:

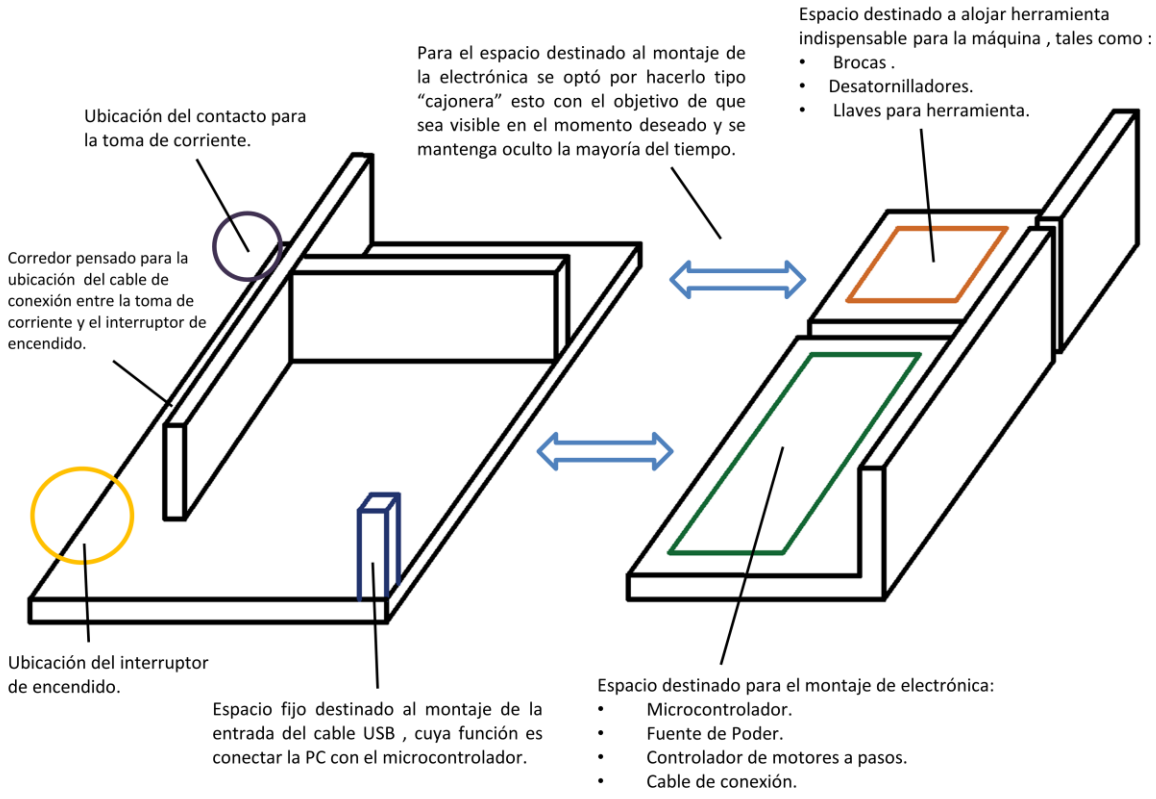


Figura 6.2. Primera parte de la ubicación de los componentes electrónicos.

En la segunda parte se encuentran montados el interruptor de encendido y el contacto para la toma de corriente, a su vez es la base para sostener el eje coordenado “X” y los ejes subsecuentes.

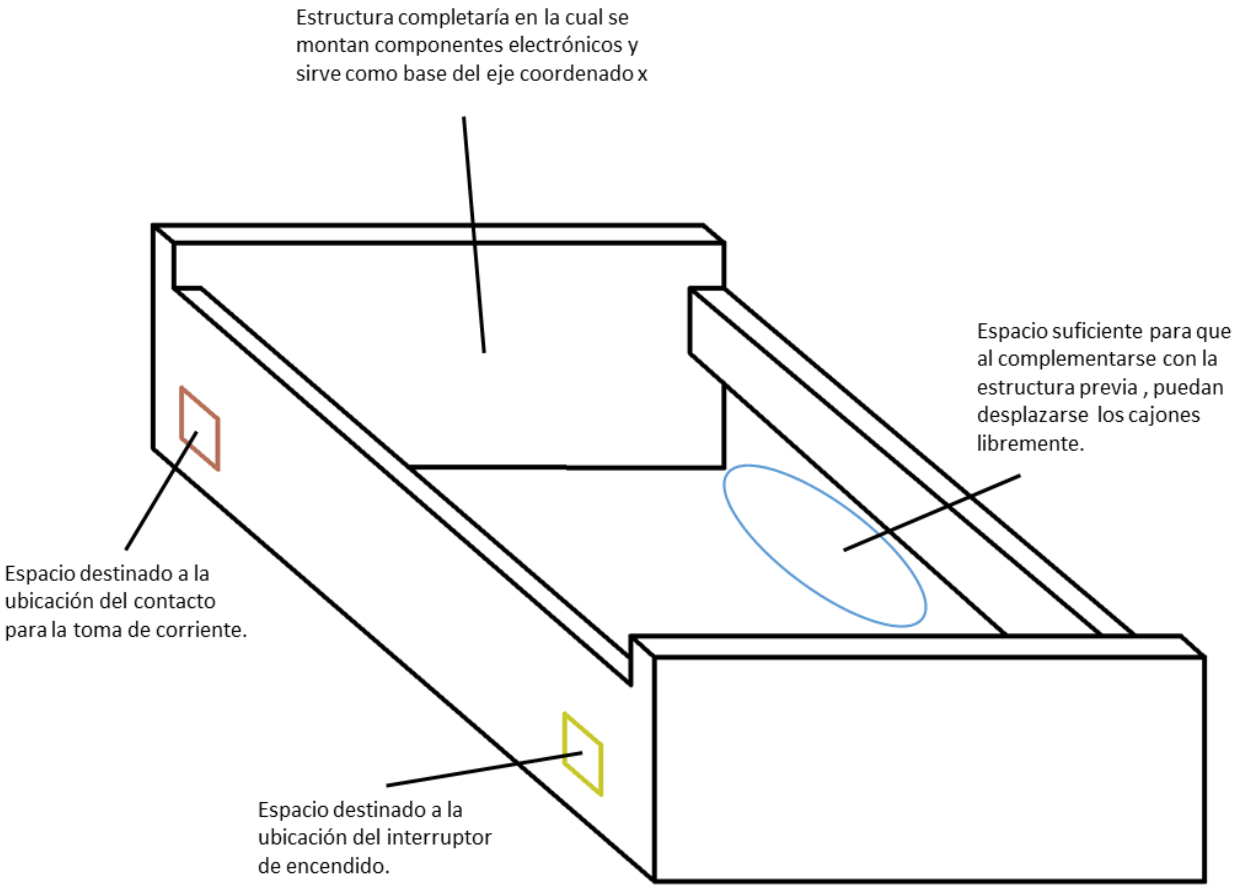


Figura 6.3. Segunda parte de la ubicación de los componentes electrónicos.

La unión de ambas partes da paso a la estructura donde se monta la electrónica y sirve como base de los ejes coordenados, la siguiente figura muestra cómo se realiza la unión entre ambas partes.

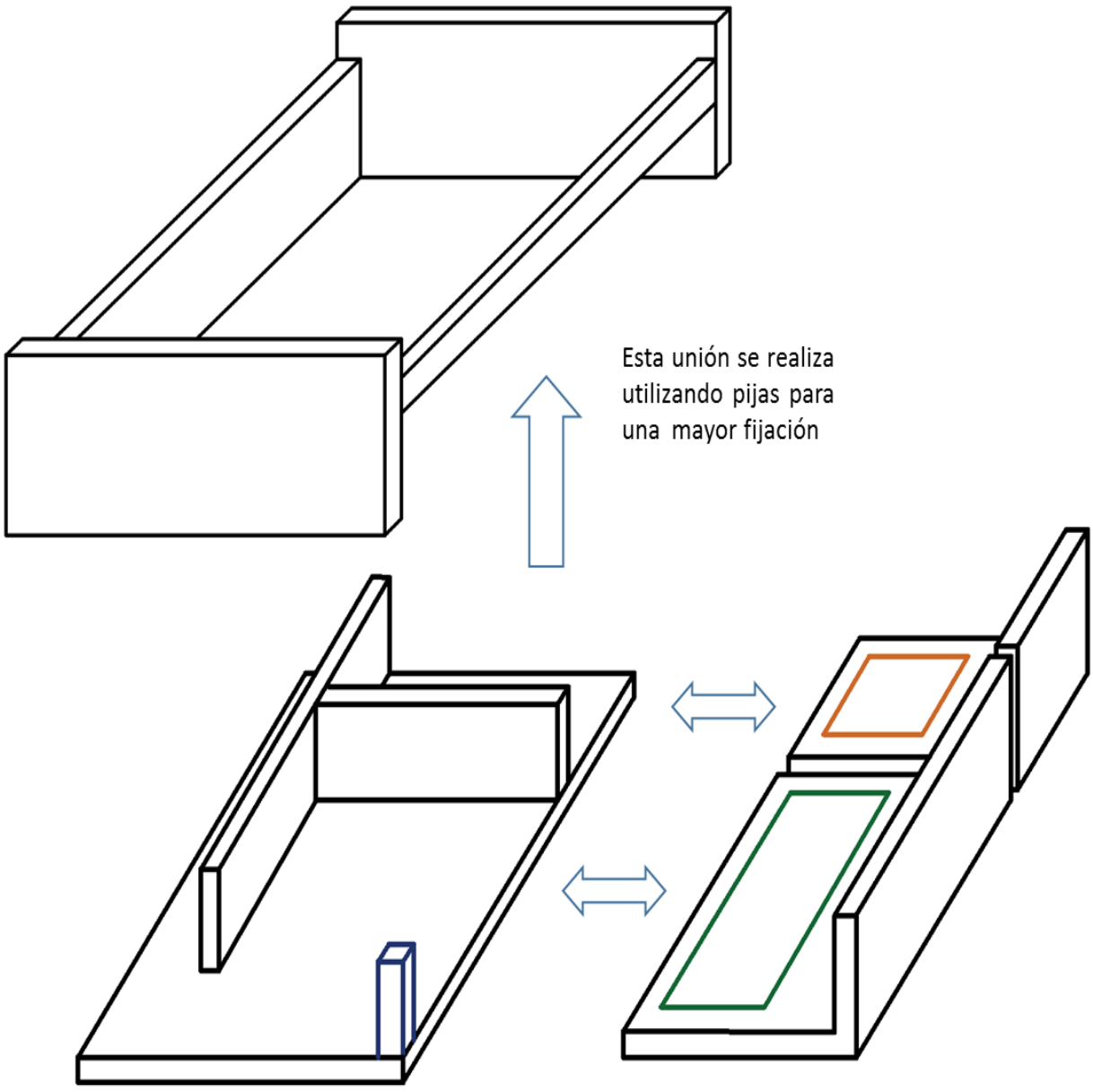


Figura 6.4. Tercera parte de la ubicación de los componentes electrónicos.

Los finales de carrera son dispositivos físicos que ayudan a que el sistema mecánico encuentre un punto específico del plano tridimensional de trabajo, esto para que dicho punto se utilice como el origen del sistema coordenado y es útil para marcar el punto en el que se comienza a realizar el trabajo.

Figura 6.5. Final de carrera.

Los finales de carrera no fueron considerados en el sistema, esto debido a que al ser una máquina que busca devastar diferentes tipos de materiales, su origen no siempre seria el mismo, por lo que sería de poca utilidad tener un punto de inicio preestablecido.

6.3. Mecánica de movimiento en los ejes

La mecánica de movimiento en los ejes es un proceso fundamental para que la máquina se pueda desplazar libremente en cada uno de sus ejes coordenados.

Para facilitar la construcción del proyecto se decidió estandarizar el mecanismo de movimiento y replicarlo para cada uno de los ejes, todo es sin dejar de lado la calidad y objetividad en cada una de sus funciones.

El proceso en cuestión se dividió en dos partes: Movimiento y guía, ambas complementarias entre sí.

6.3.1. Movimiento

La primera parte busca principalmente el acoplamiento de un motor a pasos con una varilla roscada, donde la varilla a su vez va dentro de un husillo que sostiene el eje coordenado y de esta manera se produce movimiento, directamente controlado por la velocidad y dirección de giro del motor.

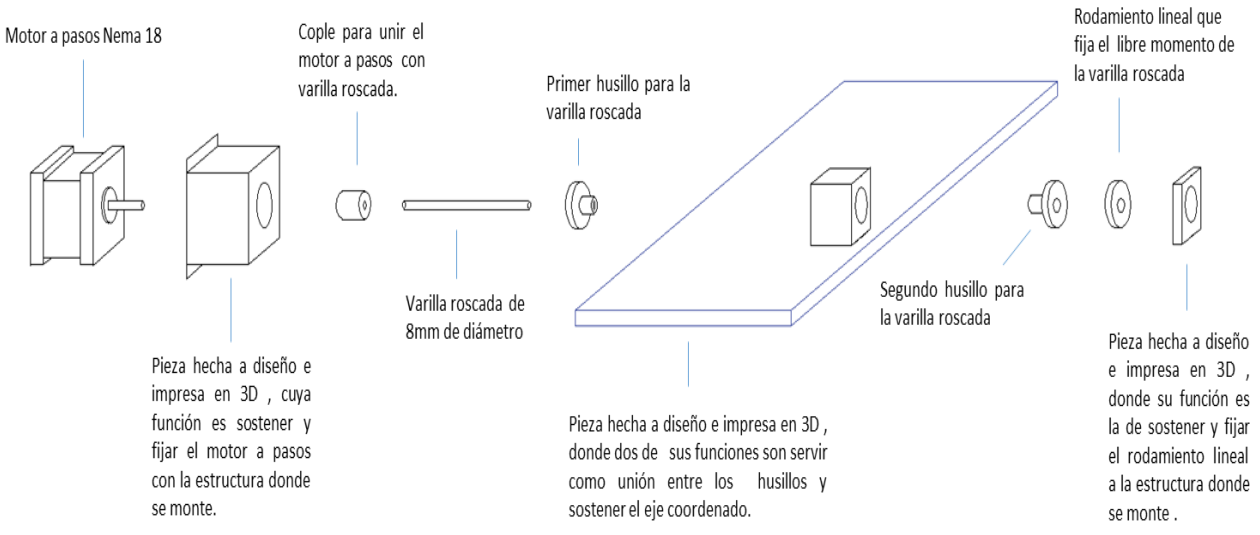


Figura 6.6. Diagrama general de la mecánica de movimiento.

Los componentes que conforman el proceso de movimiento fueron seleccionados o diseñados en función de su capacidad para brindar soluciones a las necedades del sistema.

A continuación se describen a detalle los mismos:

Motor Tipo Nema 17

Características

- Tamaño: 42.3×48mm, sin incluir el eje (NEMA 17)
- Peso: 350 gramos (13 oz)
- Diámetro del eje: 5 mm "D"
- Longitud del eje: 25 mm
- Pasos por vuelta: 200 (1,8º/paso)
- Corriente: 1.2 Amperios por bobinado
- Tensión: 12 V
- Resistencia: 3.3 Ohm por bobina
- Torque: 3.2 kg/cm (44 oz-in)
- Inductancia: 2.8 mH por bobina

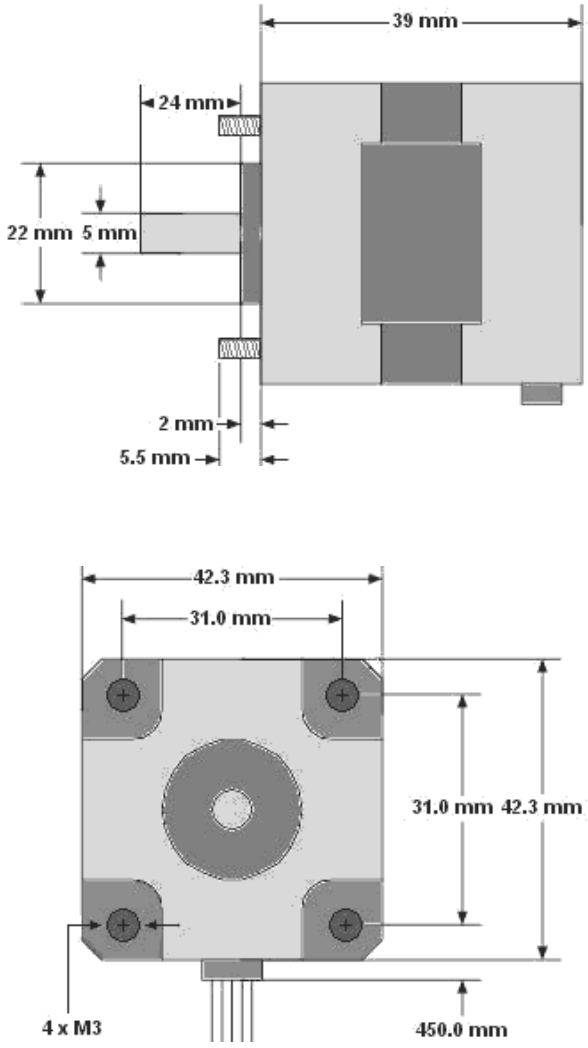


Figura 6.7. Dimensiones del motor a pasos tipo Nema 17.

Soporte de motor a pasos

El objetivo de esta pieza es el de sostener y fijar el motor a la estructura donde se planee montar.

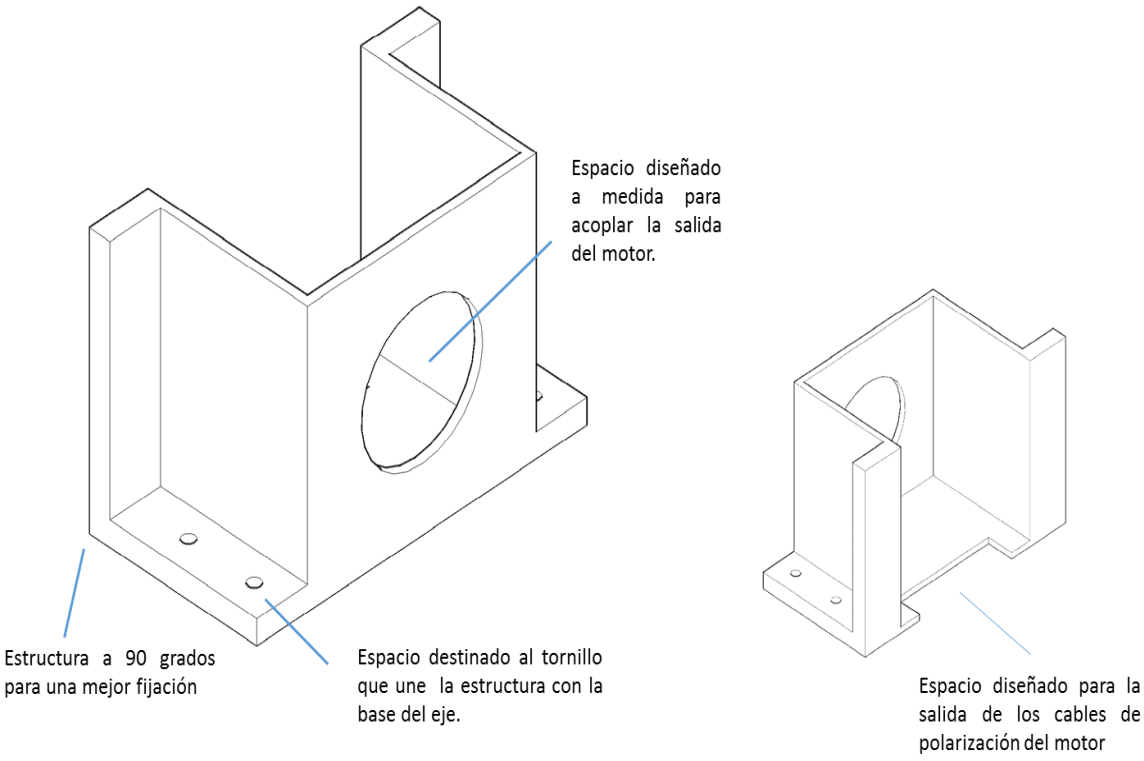


Figura 6.8. Soporte de motor a pasos.

Ya que el motor a pasos es el mismo para todos los ejes, el diseño de esta pieza se puede replicar en cada eje, facilitando así la construcción general del proyecto.

Cople

El cople es el encargado de la unión del motor a pasos con la varilla roscada, debido a esto tiene dos diámetros interiores diferentes, así como un mecanismo de presión para evitar su desacople con el motor y la varilla respectivamente.

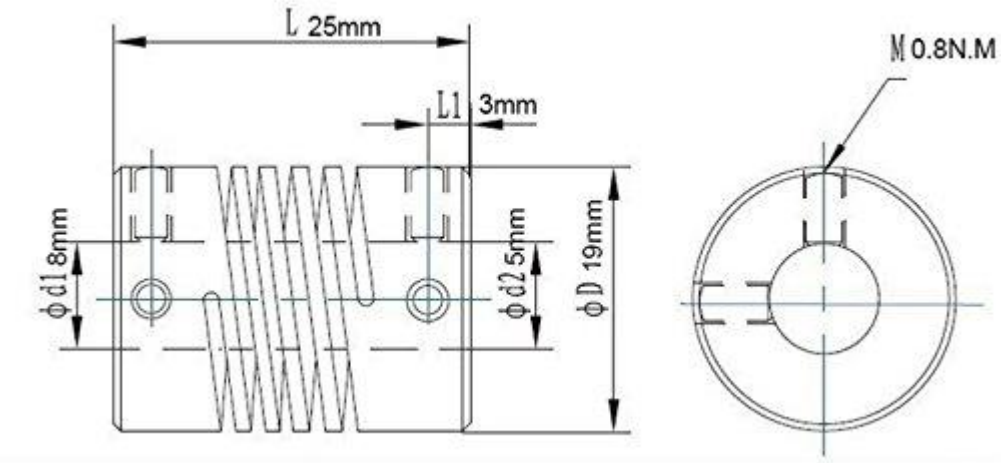


Figura 6.9. Dimensiones del cople.

Varilla roscada 8mm

La varilla roscada permite que el movimiento sea lineal y establece el esqueleto del mecanismo de movimiento.

La distancia entre cada hilo, da precisión a la posición deseada por la máquina.

Figura 6.10. Varilla roscada.

La buena calidad de la misma es indispensable debido a que sin ésta, el desplazamiento se desarrollaría de forma incorrecta y el posicionamiento del eje sería erróneo.

Su longitud varía en cada eje dependiendo la necesidad del mismo.

Husillo

El husillo sirve como unión entre la varilla roscada y el sostén del eje, su estructura permite el libre movimiento por toda la varilla.

Figura 6.11. Husillo.

Tiene un solo diámetro interior que se adecua a los nudos de la varilla roscada y cuatro orificios con el objetivo de ser incrustado en el sostén del eje.

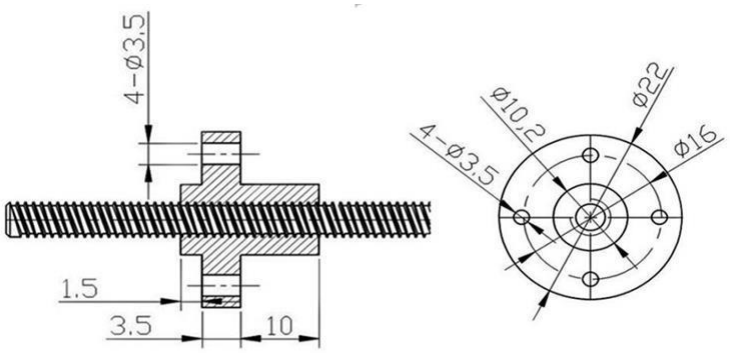


Figura 6.12. Dimensiones del husillo.

Sostén del eje de movimiento

Esta pieza es de diferente tamaño para cada eje coordenado debido a que cada eje tiene características físicas diferentes.

Cada una de estas piezas fue diseñada, modelada e impresa en 3D.

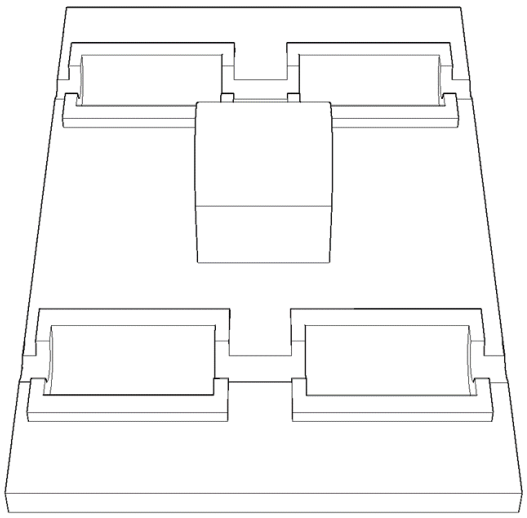


Figura 6.13. Sostén del eje de movimiento.

Su función es cumplir con los siguientes tres objetivos:

1.-Sostener al eje coordenado en el que se implementara el movimiento

2.- Acoplar al husillo con el eje coordenado.

3.- Hacer de vínculo entre el movimiento y la guía del eje, completando la mecánica del movimiento de los ejes.

A continuación, se presenta una figura, describiendo cada una de las partes esenciales que conforman a esta pieza.

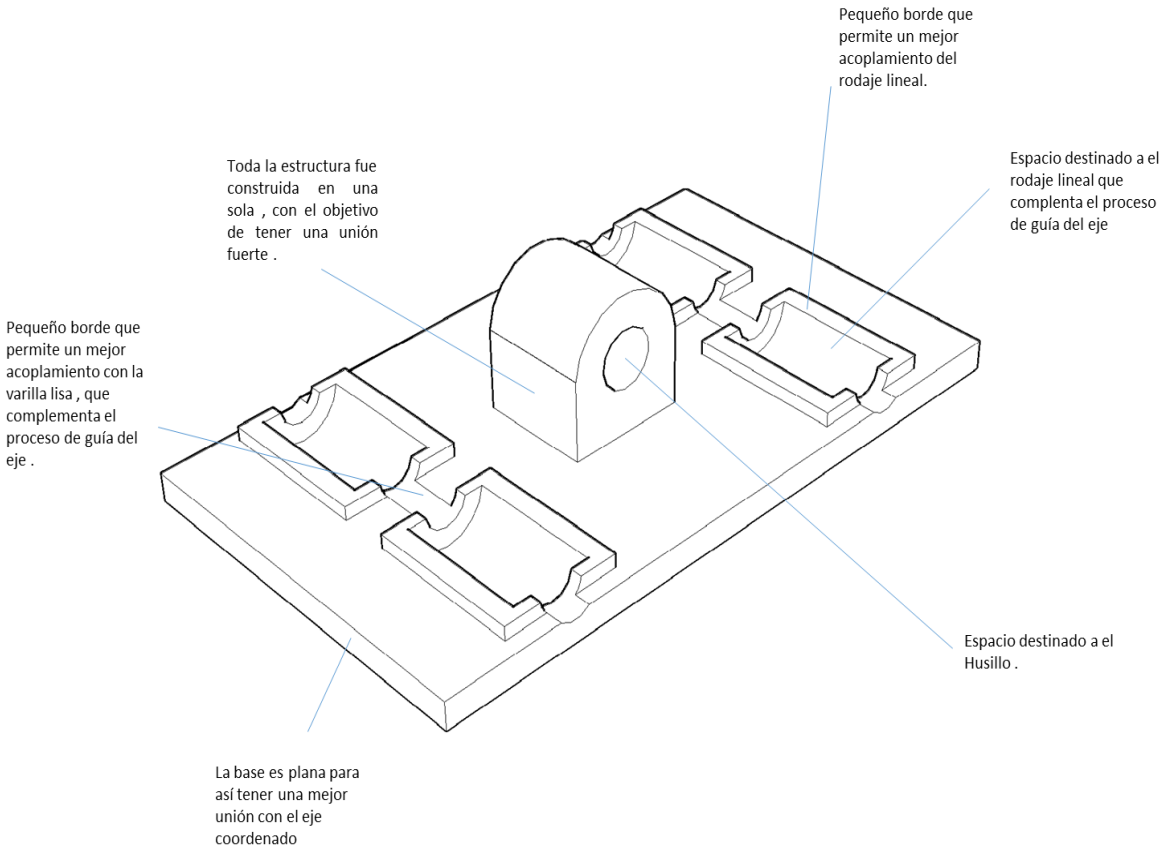


Figura 6.14. Descripción del sostén del eje de movimiento.

Rodamiento Plano

Esta pieza sostiene a la varilla roscada, además de que permite el libre movimiento giratorio de la misma.

Tiene dos diámetros, uno exterior y otro interior, cabe destacar que el interior tiene la medida exacta del diámetro de la varilla, esto permite la unión entre los mismos.

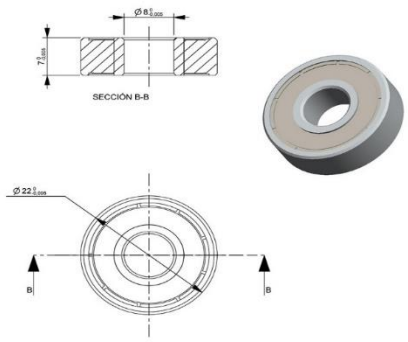


Figura 6.15. Dimensiones del Rodamiento Plano.

Se compone por una serie de balines que permiten un movimiento giratorio en su diámetro interior.

Figura 6.16. Rodamiento Plano.

Soporte del rodamiento plano

Esta pieza cumple con dos funciones, la primera es sostener al rodamiento plano y la segunda incrustarlo en el lugar fijo, siendo así la última pieza de componentes que conforman el proceso de movimiento.

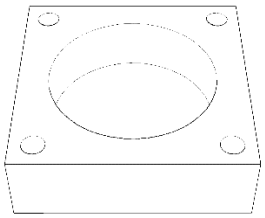
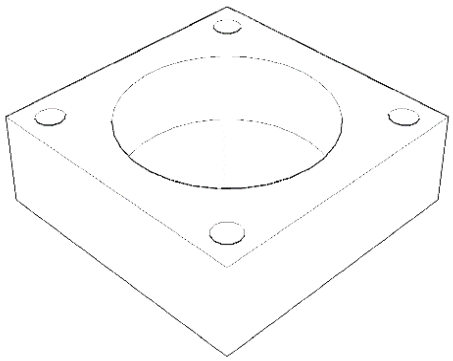


Figura 6.17. Soporte del rodamiento plano.

Esta pieza es la misma para cada uno de los ejes coordenados y aunque ya existen en el mercado una variedad de productos que cumplen su función, se decidió diseñar modelar e imprimir la pieza para abatir costos.



La pieza cuenta con un hueco interior con la forma del rodamiento plano y es aquí donde el rodamiento se montará.

También cuenta con cuatro orificios destinados a tornillos que serán los que se encarguen fijarlo en su superficie destinada.

Figura 6.18. Soporte del rodamiento plano.

6.3.2. Guía de movimiento

La segunda parte tiene como objetivo servir como guía para la dirección de desplazamiento y complementar la sujeción del husillo con el eje coordenado.

Una fuerte unión es vital en el sistema ya que en todos los ejes se registra una considerable carga.

La siguiente figura muestra de manera gráfica los componentes que sirven como guía para la dirección de desplazamiento en el eje coordenado

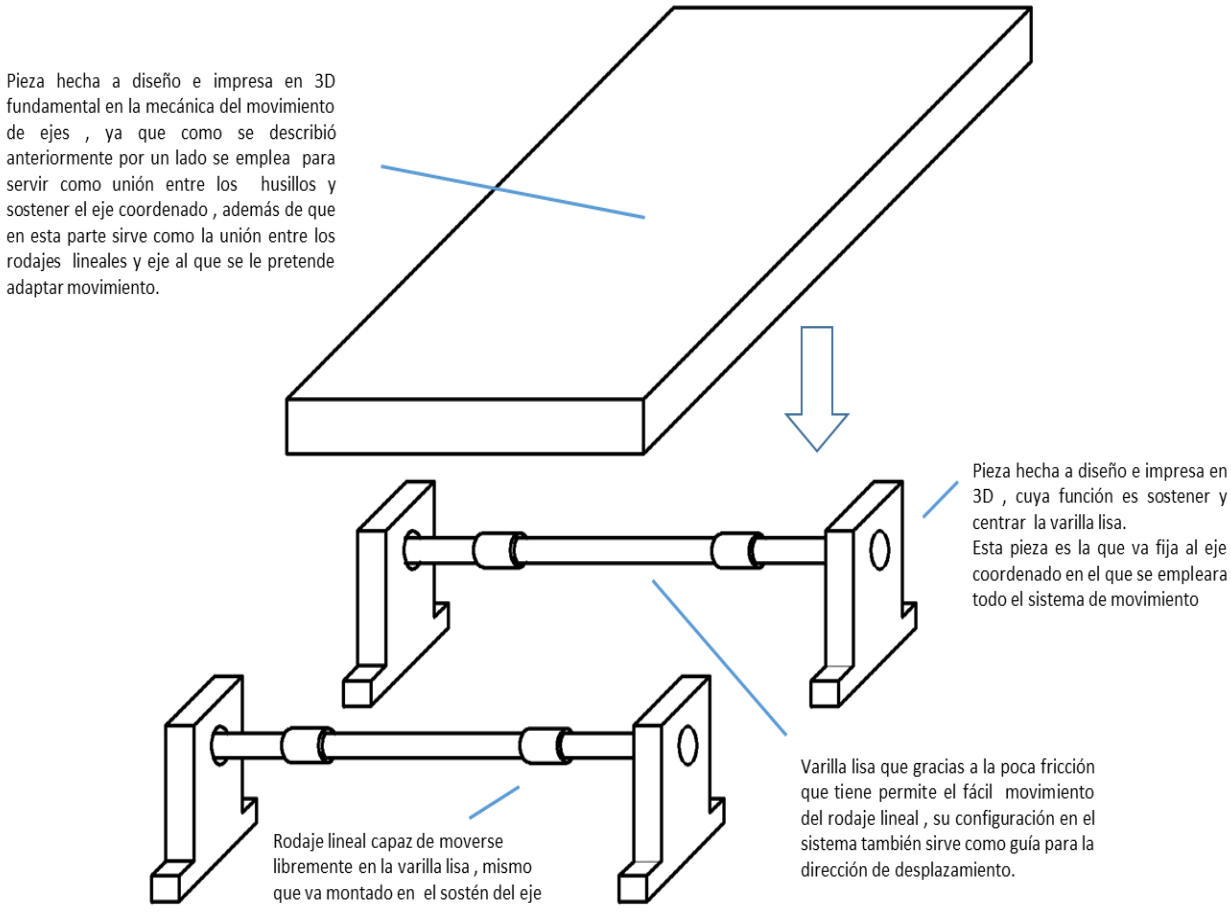
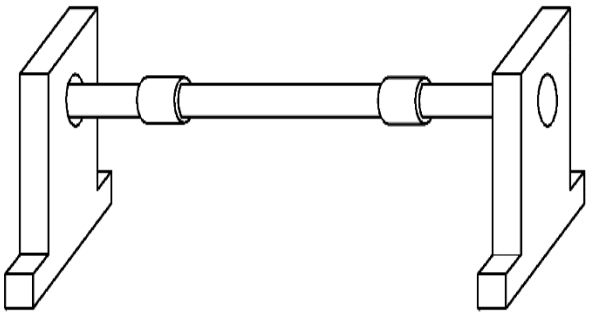


Figura 6.19. Mecánica de desplazamiento.



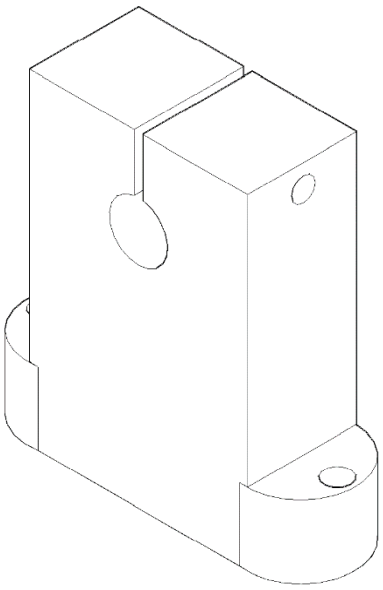
Los componentes que conforman el proceso de guía son dos estructuras paralelas, en éstas se monta el sostén del eje.

Para cada eje la estructura es idéntica en cuanto a mecánica se refiere, pero de diferente tamaño.

Figura 6.20. Estructura de desplazamiento.

Esto debido a que debe de tener el tamaño correcto para que el eje pueda desplazarse la distancia correspondiente al diseño

Cada estructura está conformada por varias piezas que se describen a continuación.



Soporte

Cada estructura está conformada por dos piezas diseñadas, modeladas e impresas en 3d que sirven como soporte de la guía.

Sobre esta pieza serán montadas las varillas lisas, por donde se desplazarán los rodajes lineales que posteriormente se unirán con el soporte del eje.

Figura 6.21. Soporte.

El siguiente diagrama describe cada una de las partes esenciales que conforman a esta pieza.

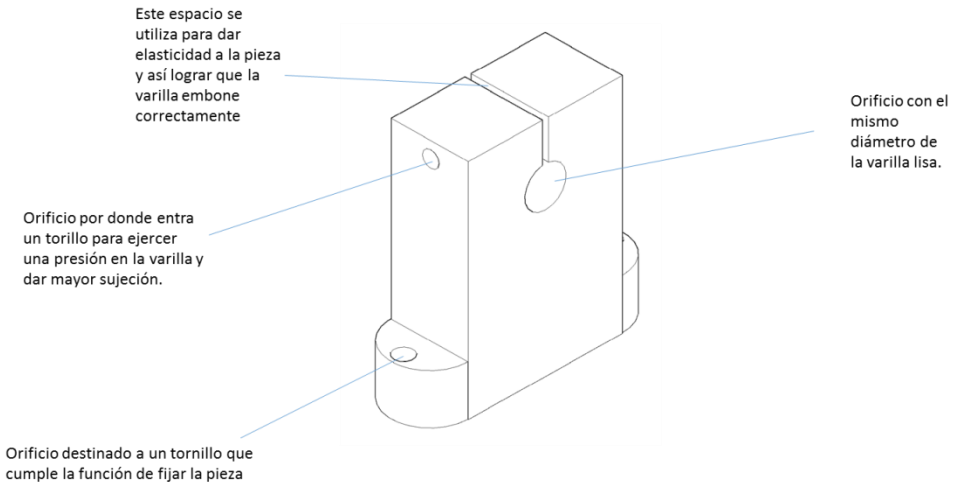


Figura 6.22. Estructura de la pieza soporte.

Varilla lisa

La varilla lisa es el esqueleto de la guía, siendo esta necesariamente lisa, para disminuir la ficción que ejerce el rodamiento lineal al desplazarse sobre ella.

Al igual que las varillas roscadas, éstas son de diferente largo para cada eje coordenado, esto dependiendo de las necesidades que presente el sistema.

Su diámetro se mantiene igual para su aplicación en todos los ejes, siendo este el justo para acoplarse a la pieza diseñada que sostiene la varilla.

Figura 6.23. Varilla lisa.

Rodamiento lineal lm8uu

Esta pieza fue considerada desde el principio para la mecánica de movimiento, a partir de sus medidas se diseñó la estructura del soporte de los ejes.

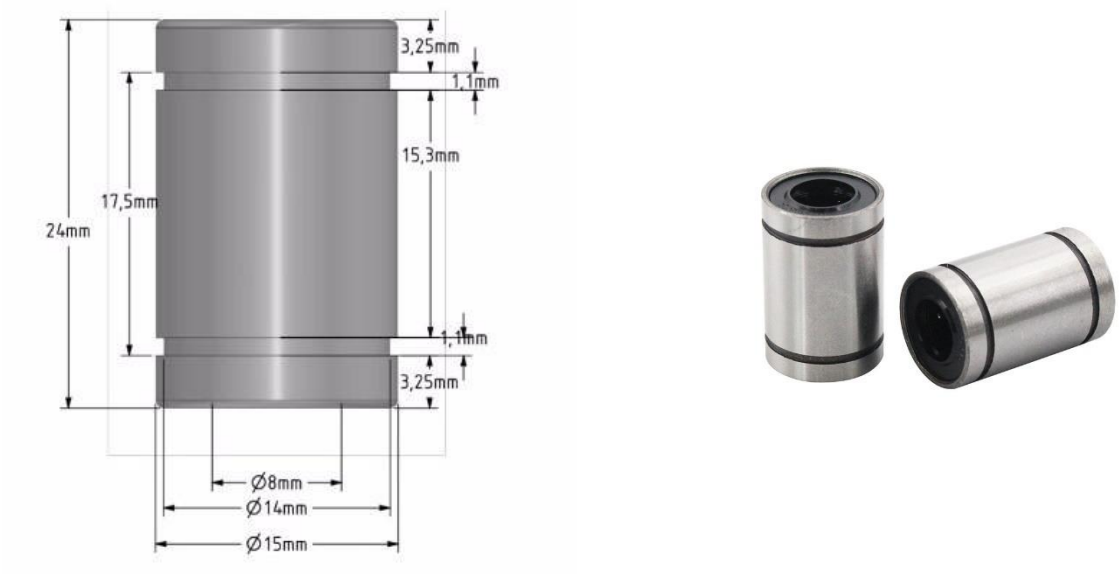
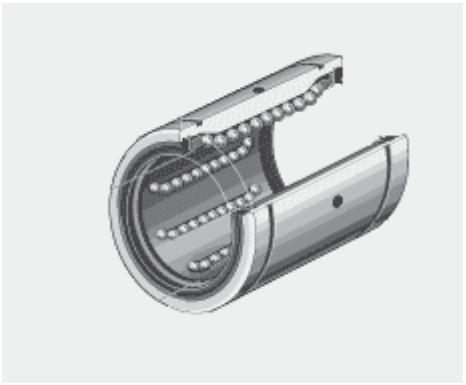


Figura 6.24. Dimensiones y aspecto del rodamiento lineal lm8uu.



La pieza, tiene un diámetro interno de 8mm, igual al de la varilla lisa, lo que permite que se pueda desplazar evitando una presión indeseable.

Por dentro tiene una serie de canicas que periten su libre desplazamiento en la varilla lisa

Figura 6.25. Mecánica rodamiento lineal lm8uu.

6.4. Eje X

El eje X es en el cual se requiere más precisión en su construcción, puesto que es en donde se sientan las bases para la construcción de los subsecuentes ejes coordenados.

En caso de que se presentara un error de construcción, éste se propagaría a los demás ejes, dando como resultado un error significativo que repercutiría indudablemente en el proceso final.

Para una completa cohesión de la máquina, se buscó que su base fuera montada sobre la estructura donde se ubican los componentes electrónicos.

6.4.1. Diseño del eje X

Con esto en mente se añadieron dos columnas idénticas para que se pueda sostener dicho eje, todo esto teniendo en mente que el eje debería de estar nivelado correctamente a un ángulo de cero grados para evitar propagar errores posteriores.

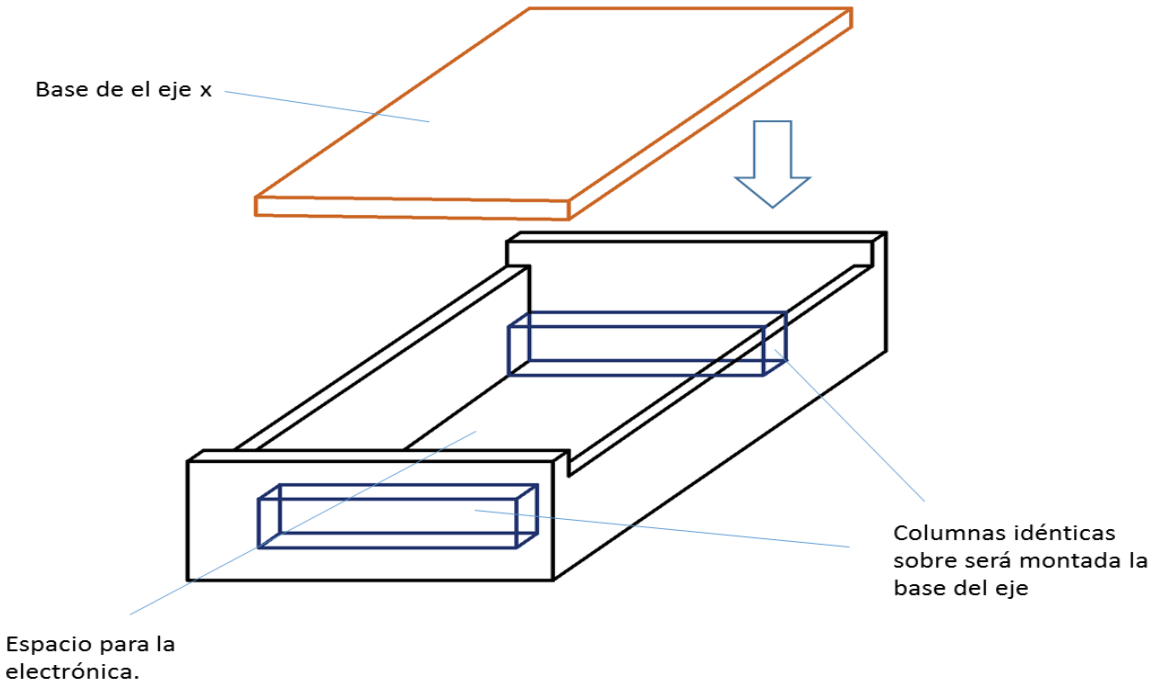


Figura 6.26. Diseño del eje X.

El eje X debe de contar con su mecanismo de movimiento para poder desplazarse libremente y a su vez debe de ser capaz de sostener a los ejes Y y Z, así como a la herramienta de corte, por lo que analizando un poco la situación se concluye que lo mejor es que sea el eje más grande de todos y que este limitara directamente la dimensión máxima, respecto a lo largo que puede ser el área de trabajo final.

El siguiente diagrama indica cómo es que se montará la mecánica de movimiento a la base del eje X.

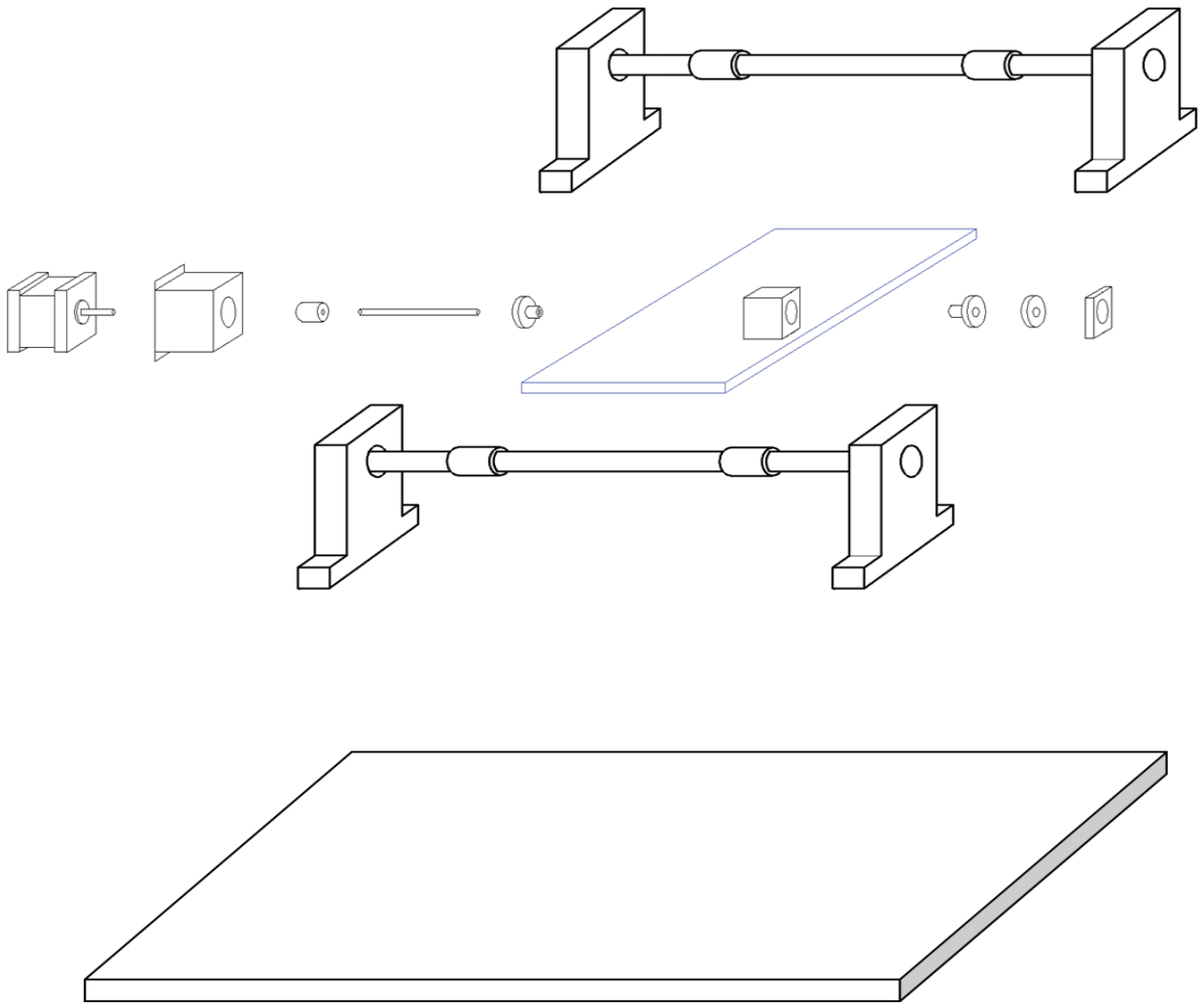


Figura 6.27. Montaje de la mecánica de movimiento en el eje X.

Una vez el eje X de encuentre instalado, éste será montado sobre la electrónica, esto para reducir el volumen de la máquina y mantener todo compacto sin afectar su funcionamiento.

La siguiente figura muestra cómo es que se pretende realizar el montado del eje X hasta este momento.

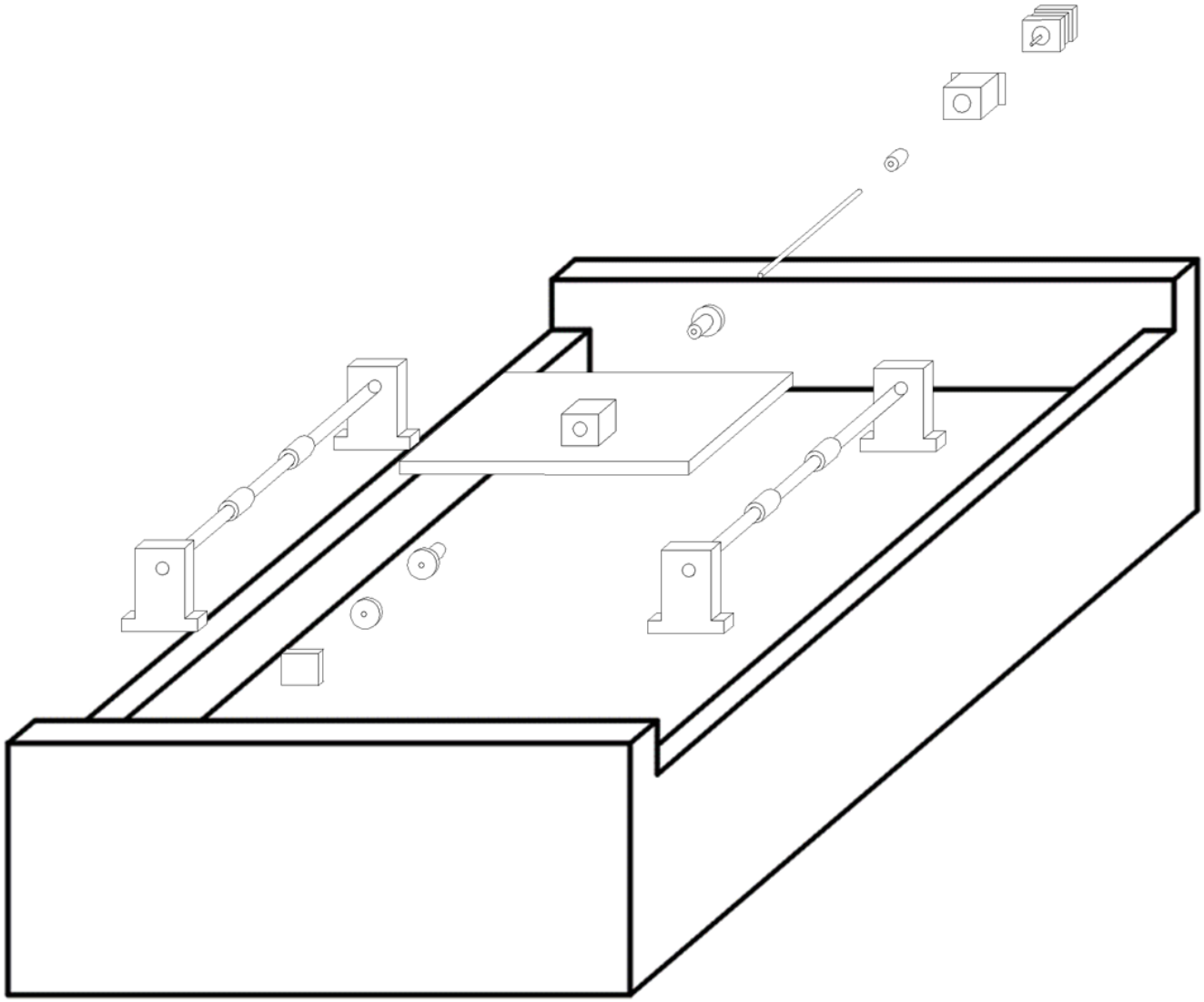


Figura 6.28. Eje X.

6.5. Eje Y

Sobre la estructura del eje Y será montado el eje Z y posteriormente la herramienta de devastado, por lo que será de vital importancia la robustez del mismo y lo bien montado que este se encuentre sobre el eje X.

El eje Y es el encargado de proporcionar la anchura máxima que puede obtener el área del producto final, por lo que es importante tener la suficiente dimensión para maximizar el área de trabajo.

6.5.1. Diseño del eje Y

El eje Y sirve como interfaz entre los ejes X y Z, esto se logra mediante una estructura en forma de “H” donde por la parte inferior se une a la estructura del eje X y por la parte superior sostiene a el eje Z.

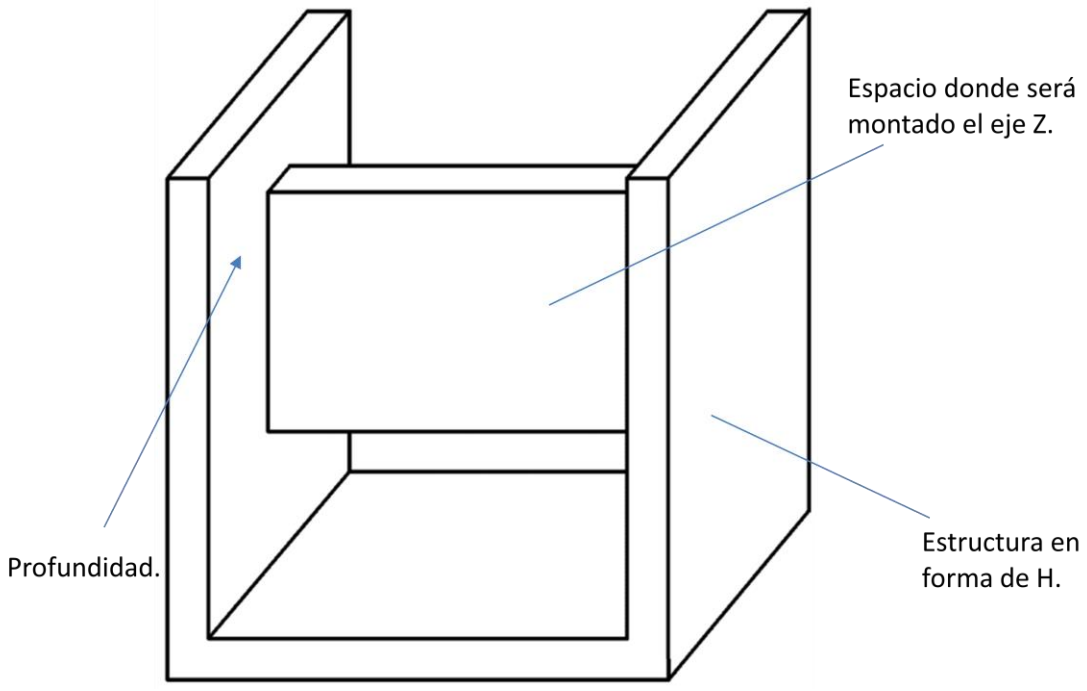


Figura 6.29. Diseño del Eje Y.

Se tuvo en cuenta que este eje debe de tener la capacidad física para sostener al eje Z y a la herramienta de devastado, junto con las piezas que esto implica, por lo que en el diseño de la estructura se le dio profundidad al diseño, para permitir una correcta instalación de la electrónica.

Al igual que el eje X, el eje Y debe de contar con su mecanismo de movimiento para poder desplazarse libremente y a su vez debe de ser capaz de sostener a al eje Z, así como a la herramienta de corte.

En la siguiente figura podemos observar cómo es que se pretende montar la mecánica de movimiento al eje Y.

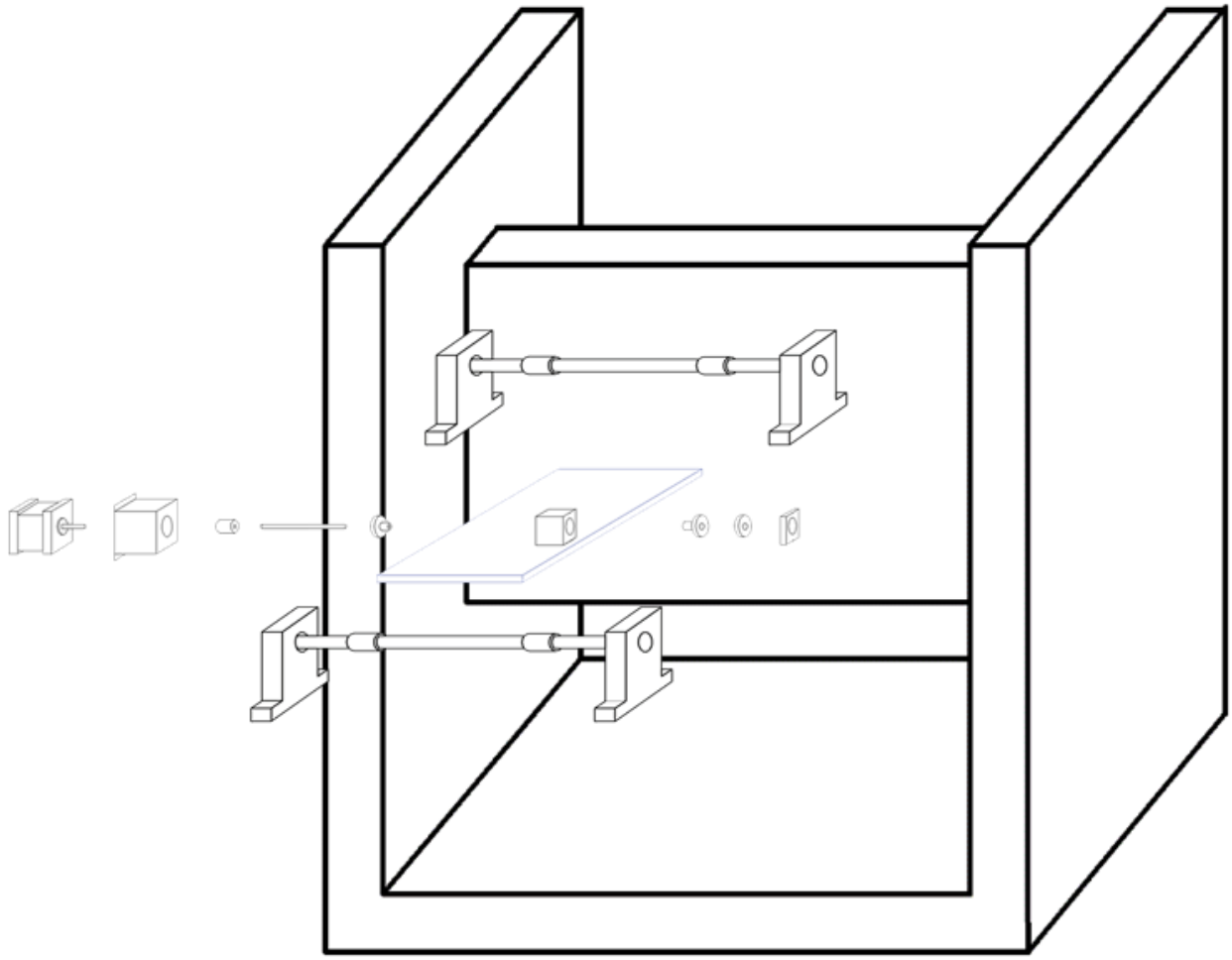


Figura 6.30. Montaje de la mecánica de movimiento en el eje Y.

Cuando el eje Y se encuentre instalado, éste será montado sobre la estructura del eje X como se muestra a continuación.

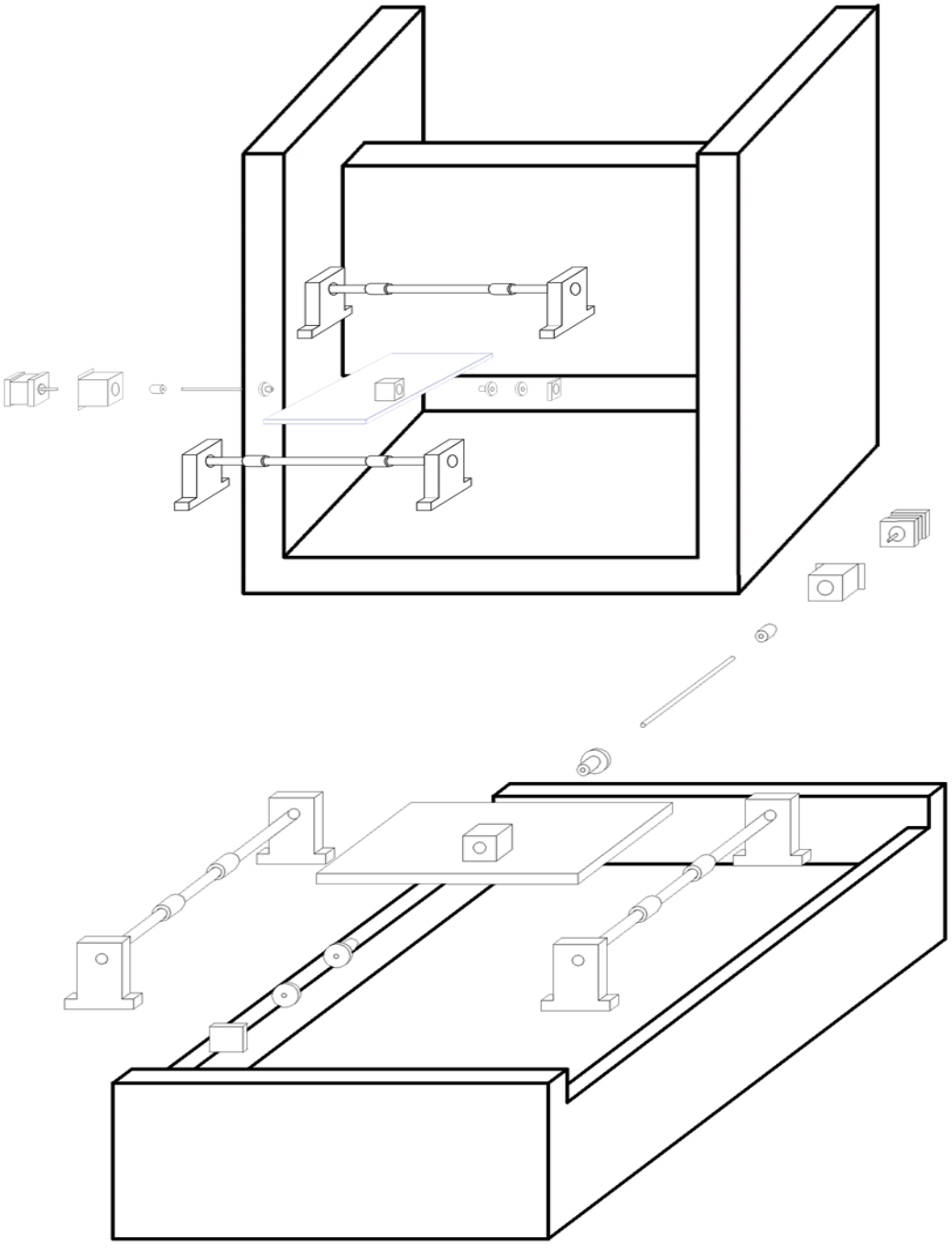


Figura 6.31. Eje Y.

6.4. Eje Z

El eje Z es el último de los ejes y por la física de la máquina este eje será el más pequeño, limitando su área a las dimensiones que se adecuen a la herramienta de devastado, misma que será montada sobre dicho eje.

El eje Z es el encargado de proporcionar la altura máxima que puede tener el objeto donde se realiza el trabajo de devastado, al tener en mente que éste será aplicado para placas de cobre, esta altura no debe de ser tan grande para obtener un producto con buena calidad, sin embargo, no es mala idea tener una superficie de altura sobrada, para así poder realizar el trabajo de devastado en otras superficies diferentes al cobre, como lo es la madera o mdf.

6.4.1. Diseño del eje Z

Tomando en cuenta que este eje será el más pequeño se decidió diseñar al eje como una pequeña caja donde serán montada la electrónica de movimiento y posteriormente la herramienta de devastado, por lo que su diseño es el más simple y objetivo de todos.

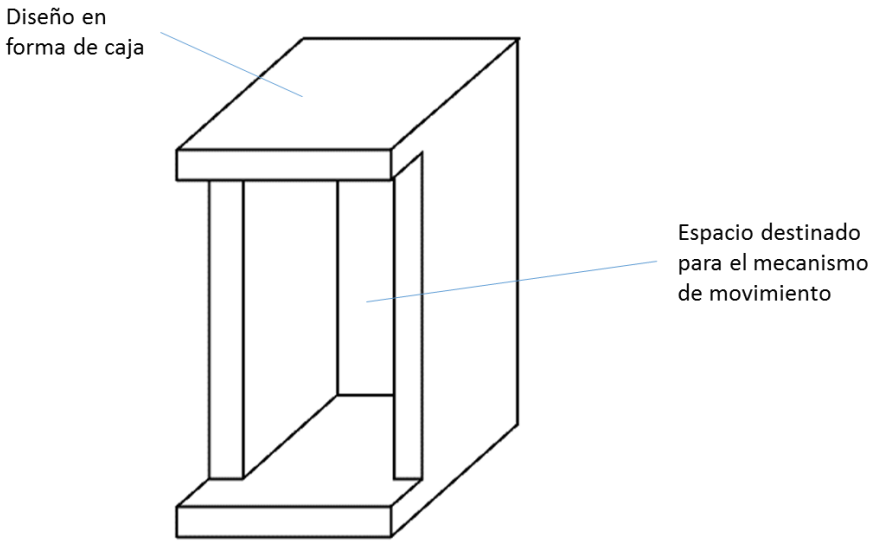


Figura 6.32. Diseño del Eje Z.

Como en los demás ejes, el eje Z debe de contar con su mecanismo de movimiento para poder desplazarse libremente y ser capaz de sostener y herramienta de corte.

Al ser el más pequeño de los ejes, estará lleno de detalles a tomar cuenta como la forma en la que será montada la herramienta de devastado para que esta pueda ser removible y su armado será más complicado que los anteriores.

En la siguiente figura podemos observar cómo es que se pretende montar la mecánica de movimiento al eje Z.

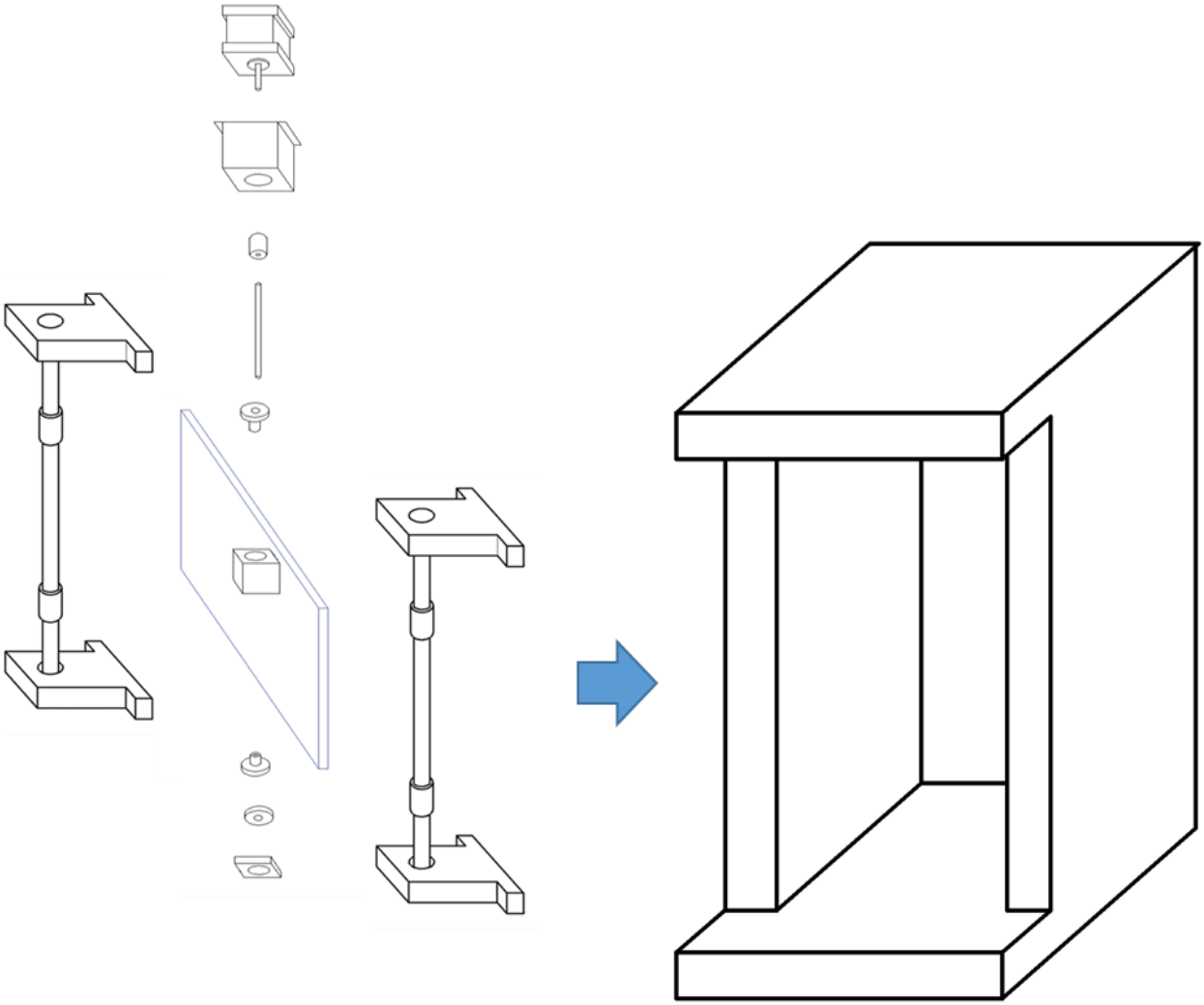


Figura 6.33. Montaje de la mecánica de movimiento en el eje Z.

Cuando el eje Z se encuentre instalado, éste será montado sobre la estructura del eje Y como se muestra a continuación.

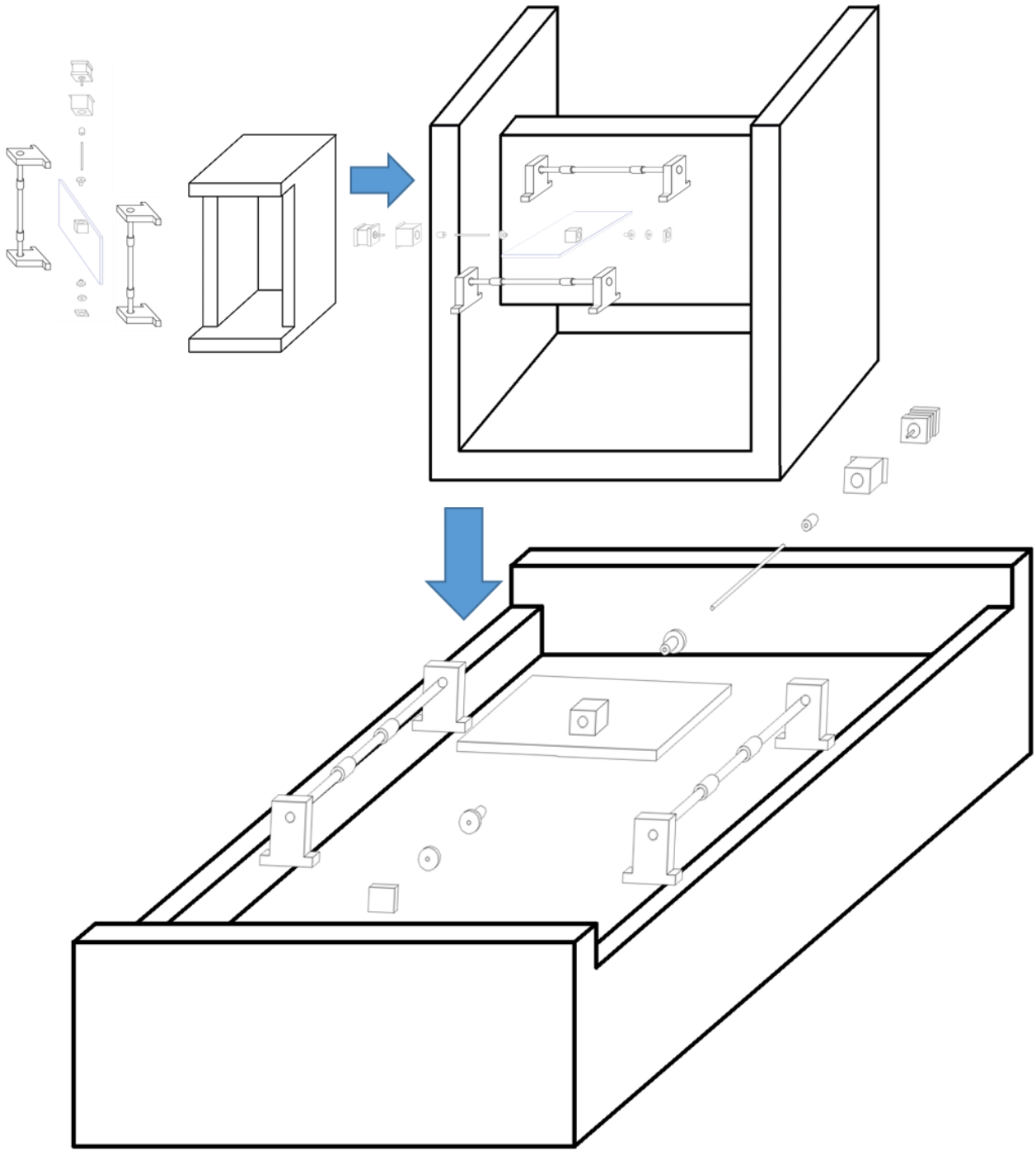


Figura 6.34. Eje Z.

6.7. Herramienta de devastado

La herramienta de devastado es la encargada de barrer el cobre en la placa fenólica, por lo que es importante que esta cuente con la potencia necesaria para realizar adecuadamente el trabajo.

La herramienta que se decidió utilizar para realizar el trabajo es el Dremel 3000, esto gracias a que cuenta con las suficientes revoluciones por minuto (5.000 rpm - 32.000 rpm) para poder devastar la placa de cobre, además de que tiene la capacidad de intercambiar sus piezas dependiendo de las necesidades, lo que resulta muy práctico a la hora de utilizar la máquina para otras aplicaciones, como el devastado a madera.

Figura 6.35. Dremel 3000.

Caracteristicas descritas por el fabricante:

1.-Rosca ergonomica para el cambio de accesorios, sin necesidad de llaves.

2.-Botón para el rápido cambiado de broca.

3.-Motor universal.

4.- Velocidad variable entre 5000rpm - 32000 rpm.

5.- Piezas reemplazables.

Para montar al dremell en el eje Z y evitar en lo posible vibraciones se utilizó una pieza impresa en 3D que sirve como sujetador del mismo.

Al no ser un diseño especifico el diseño de la pieza fue descargado de un repositorio gratuito de archivos .stl para así poder imprimirse en 3D.

Figura 6.36. Soportes de dremel 3000.

Las brocas utilizadas varían en función del trabajo a realizar. A continuación, se muestra una figura donde se puede apreciar que broca utilizar para cada trabajo específico.

Figura 6.37. Diferentes tipos de broca.

Para dar más versatilidad al sistema mecánico y las funciones que este puede desempeñar se montó sobre el eje Z un sujetador de lápiz, para que en lugar de devastar material se pueda dar paso al dibujo.

La pieza donde se montara el lápiz no requere un diseño especifico, por lo que también fue descargada de un repositorio gratuito de archivos .stl para asi poder imprimirse en 3D.

CAPÍTULO 7

SISTEMA ELECTRÓNICO

7.1 Integración del sistema electrónico

Con una idea más concreta del sistema mecánico, de los requerimientos electrónicos de la máquina, y del espacio destinado para que estos sean montados se puede desarrollar la integración de los elementos que componen la electrónica, la siguiente figura muestra las terminales utilizadas por la placa de desarrollo arduinoUNO.

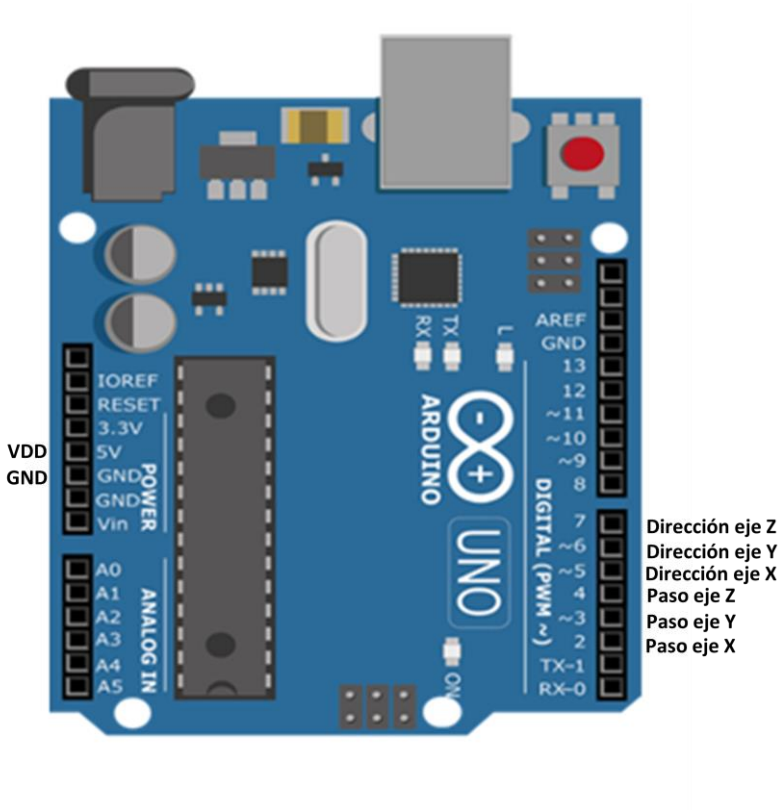


Figura 7.1. Terminales utilizadas por la placa de desarrollo arduinoUNO.

El diagrama completo de conexión entre la placa de desarrollo arduinoUNO y los controladores de motores a pasos se muestra a continuación:

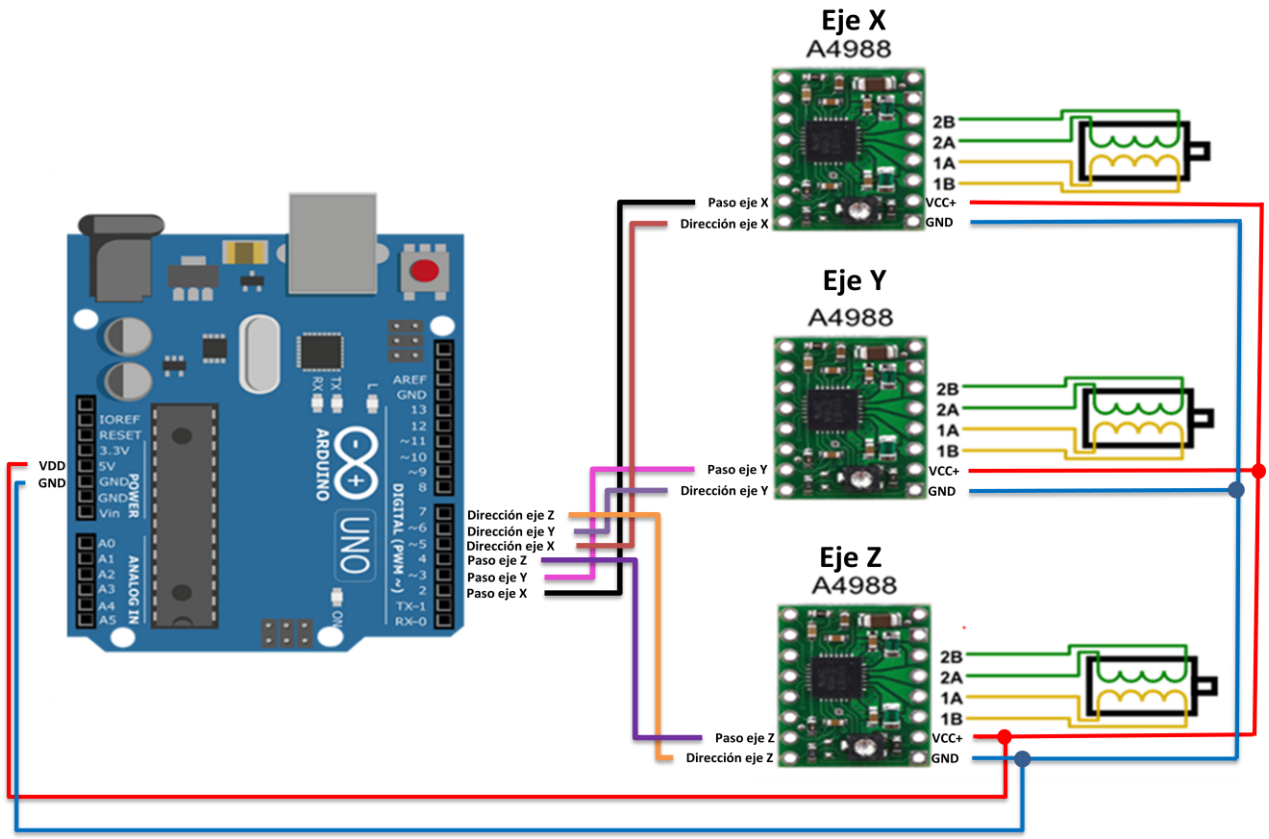


Figura 7.2.Diagrama de conexión entre microcontrolador, controladores y motores a pasos.

A continuación, se muestra un diagrama con una integración más completa del sistema, incluyendo el voltaje de polarización de los motores.

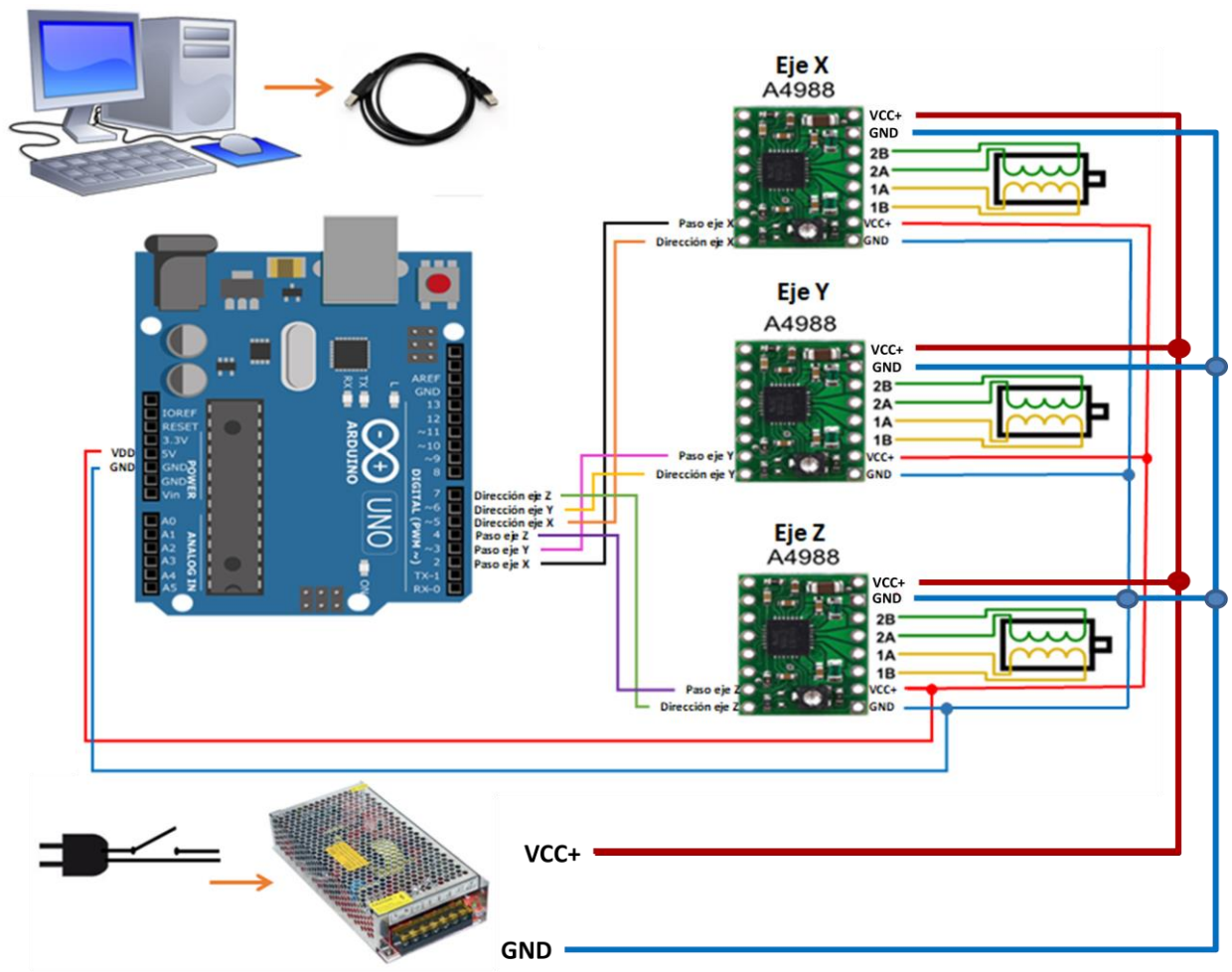


Figura 7.3. Integración completa del sistema.

Para una integración más estable del sistema se optó por montar los controladores de motores a pasos, el microcontrolador y sus terminales en una tableta fenólica, su diagrama la plantilla de circuito impreso y las fotografías de su implementación en tarjetas fenólicas se muestran a continuación.

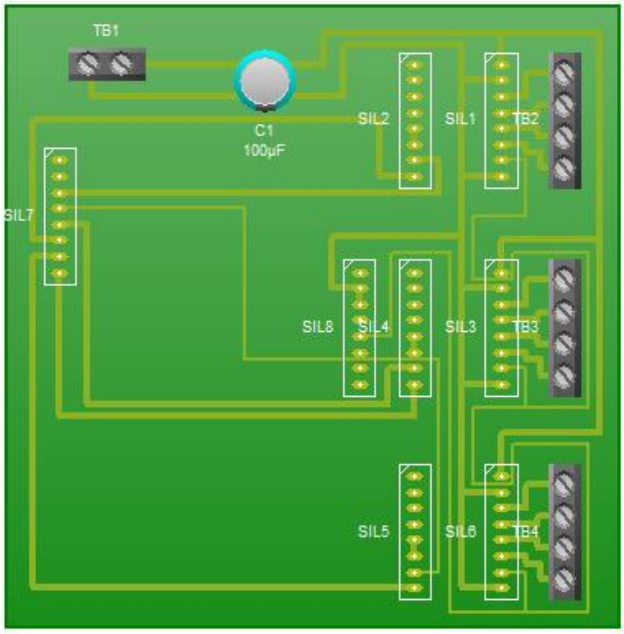


Figura 7.4. Prototipo tableta fenólica.

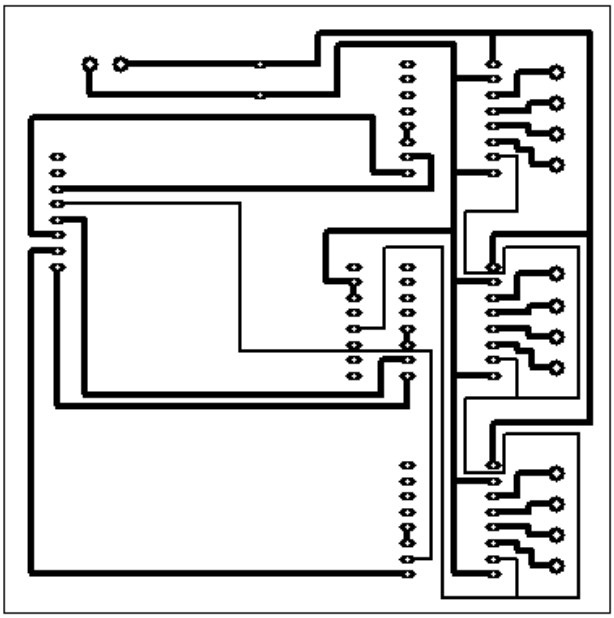


Figura 7.5. Plantilla de circuito impreso.

Al hacer uso de la placa CNC Sheld se facilito de manera conciderable el montado de la electronica , asi como que se le dio un aspecto mas profesional al sistema.

La siguiente figura muestra como luce la placa CNC Sheld con los los controladores de motores a pasos montados y la tarjeta de desarrollo ArduinoUNO.

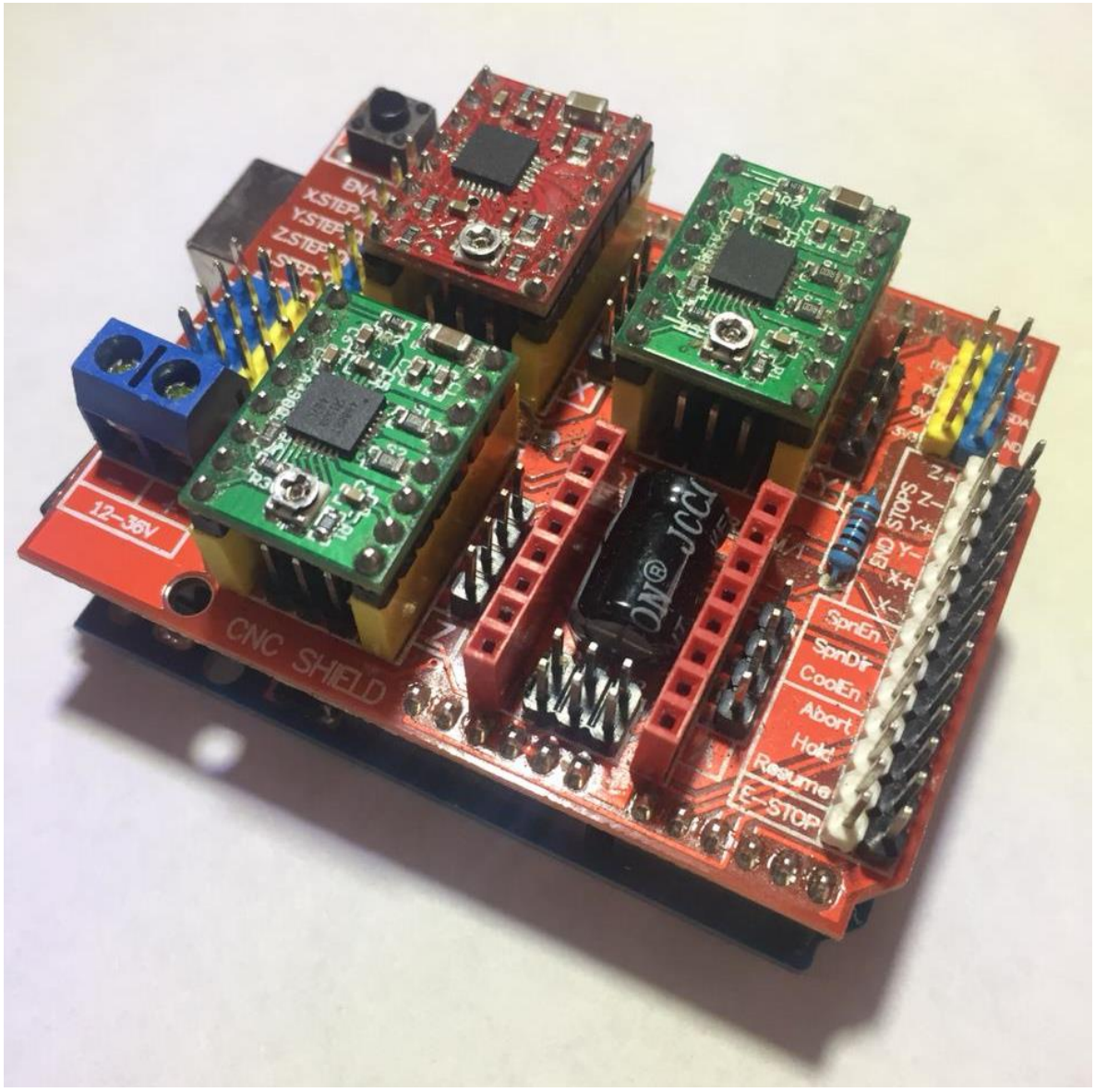


Figura 7.6. Placa Cnc Sheld con sus respectivos componentes montados.

CAPÍTULO 8

INTEGRACIÓN DEL SISTEMA

8.1. Vista general de la máquina ya construida.

A continuación se muestra un diagrama de la integración total del sistema mecánico junto con su herramienta de devastado.

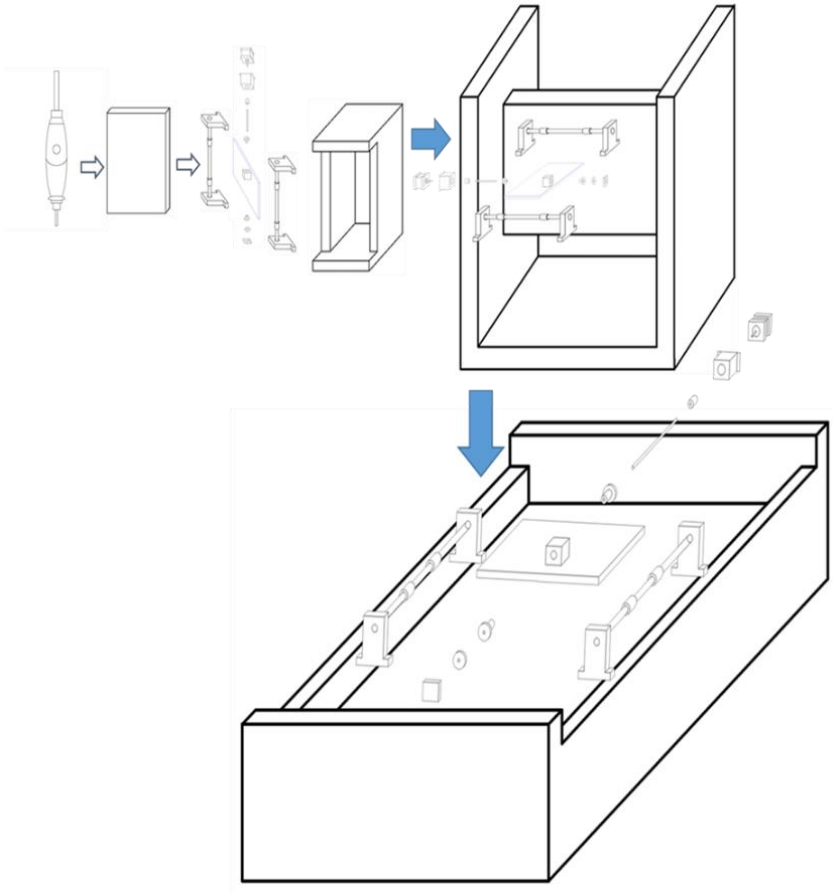


Figura 8.1. Integración total del sistema.

La máquina terminada se presenta en la figura 8.2:

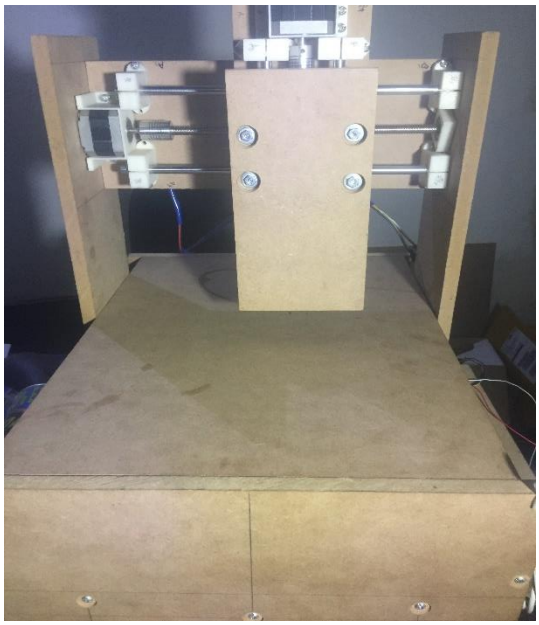


Figura 8.2. Vista física de la máquina.

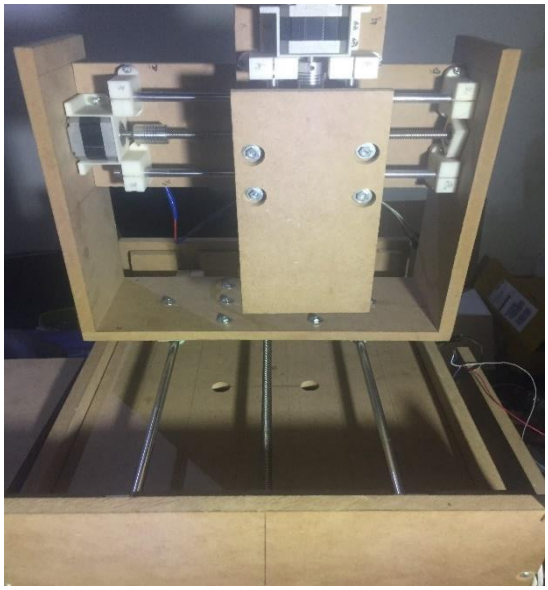


Figura 8.3. Vista Física de la máquina.

8.2. Resultados de funcionamiento.

El funcionamiento del sistema mecánico en cada uno de sus ejes es bueno, el desplazamiento es fluido y sin contratiempos, por lo que se afirma que su diseño fue acertado. Las piezas impresas en 3D fueron una excelente adición al mismo, pues al ser diseñadas a la medida, evitan que este particular sistema tenga contratiempos.

Por otro lado, la falta de nivelación en la base hace diferente la profundidad de penetración de la broca a lo largo de la superficie, sin embargo, al ser un área reducida la de trabajo, esto pasa casi desapercibido, sin ser realmente significativo en el trabajo final.

Al trabajar con las diferentes versiones del programa de control de los motores a pasos se obtuvieron resultados distintos. Por una parte, se consiguieron malos resultados con la primera versión y resultados más adecuados para la segunda.

En la siguiente figura se muestran los resultados de trabajar con la primera y segunda versión del programa de control de los motores a pasos.

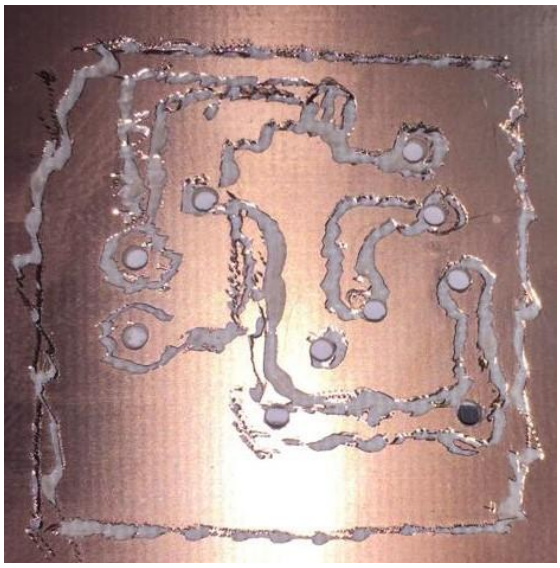


Figura 8.4. Trabajo con Versión 0.8 de GRBL.

Figura 8.5. Trabajo con Versión 0.9 de GRBL.

CAPÍTULO 9

TRABAJO FUTURO

9.1. Propuestas a futuro

Con base en a los resultados y el aprendizaje obtenido se describen a continuación varias propuestas con vista a una implementación a futuro.

Sensor de nivelación

Una de las primeras propuestas es adaptar un sensor para la nivelación del área de trabajo, ya que con esto y unas modificaciones al código del programa se obtendría una mejor calidad a la hora del devastado puesto que la calibración manual quedaría de lado y se tendría una calibración más profesional.

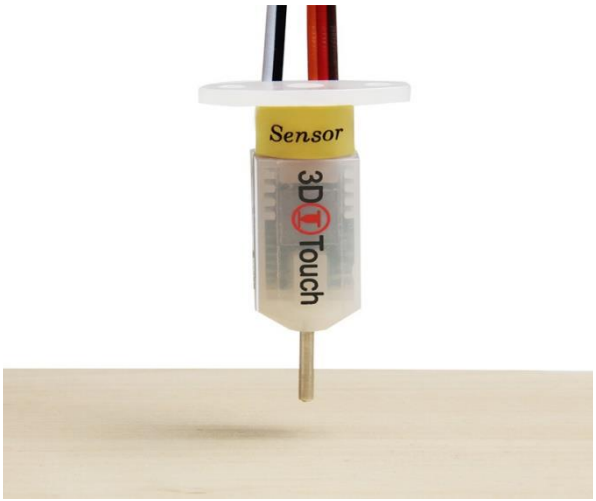


Figura 9.1. Sensor de nivelación.

Cambio de material para la construcción

Cambiar el MDF que se utilizó para la construcción de los ejes ayudaría a disminuir considerablemente el peso y el tamaño de la máquina, además de un posible aspecto estético más profesional.

Las opciones más viables serian el aluminio o el acrílico e incluso se podría llegar a pensar, dependiendo de su aplicación en la construcción de uno o varios ejes mediante la impresión en 3D.

Cambio en el diseño de los ejes

En la actual máquina los ejes se fueron apilando unos a otros. Una manera para mejorar la mecánica y evitar que los ejes soporten mucho peso entre ellos sería la de modificar la forma que están distribuidos.

Se podría pensar en diseñar al eje X sin la necesidad de que sostenga a ejes subsecuentes, independizando su movimiento, los ejes subsecuentes podrían estar montados sobre columnas que solo cumplan el propósito de sostenerlos.

El software no sería obstáculo ya que con un poco de modificaciones al código no habría problema en obtener los resultados esperados.

Integrar láser

Aprovechando el sistema mecánico de la máquina se podría montar en el eje Z un láser, esto con el objetivo de realizar cortes en la superficie de trabajo, para ello sería necesario modificar el programa. Los desarrolladores de programa de control para motores a pasos también ofrecen una versión para trabajar con este dispositivo.

Figura 9.2. Láser de 5 watts.

Impresora 3D

Otra buena modificación que se puede realizar es la implementación del sistema como una impresora 3D, para esto sería necesario agregar el control de otro motor a pasos para sustituir la herramienta de devastado por un extrusor de filamento, agregar control de temperatura, un área de impresión y modificar el programa de trabajo.

Sujetadores

Como una mejora adicional se podría sugerir el agregar dispositivos para presionar las placas en el área de trabajo, esto con el objetivo de sujetar el material de trabajo y evitar desplazamientos involuntarios.

CONCLUSIONES

El presente proyecto logra cumplir con el objetivo de diseñar y construir una máquina de control numérico computarizado capaz generar pistas para circuitos electrónicos en placas de cobre. Esto se logró mediante el control e implementación de un sistema mecánico de tres ejes coordenados con la posibilidad de poder situarse en diferentes puntos específicos de un espacio tridimensional.

La implementación del código G como lenguaje de programación y la interpretación del mismo a cargo del microcontrolador atmega328 dieron los resultados esperados ya que la máquina en conjunto con su mecánica de movimiento es capaz de realizar cualquier desplazamiento deseado dentro de su superficie de trabajo sin estar limitada por instrucciones.

Los controladores de motores a pasos a4988 de pololu dieron buenos resultados, permitiendo regular por corriente a los motores a pasos nema 17 y la resolución de los pasos del motor, además de brindar una opción de compatibilidad si en el futuro se desea trabajar con otros motores y microcontroladores.

Al utilizar el dremell 3000 como máquina de devastado se amplía la posibilidad de realizar el trabajo en diferentes superficies, tales como madera, MDF y acrílico, dando así versatilidad al sistema, además de que el eje Z, al tener una superficie plana, en un futuro se puede montar un láser o un extrusor de impresora 3D y así dar paso a una máquina más completa.

El utilizar programas libres ayudó a que los costos de producción no superaran los 4500 MXN$, siendo este un precio bajo en comparación al precio de las máquinas de este tipo. Además de que no limitó los alcances de la máquina, puesto que ésta puede realizar cualquier trabajo sin necesidad de invertir en programa de paga y gracias a los donantes, los programas libres se actualizan periódicamente.

La mecánica de movimiento y las piezas impresas cumplieron con las necesidades del sistema para poder desplazarse libremente en un plano tridimensional, lamentablemente por la ubicación de los motores la superficie final de trabajo se limitó drásticamente, esto nos lleva a replantearnos una nueva mecánica para sistemas futuros, reubicando los motores y optimizando el espacio destinado a la mecánica de movimiento.

APÉNDICE

SET DE INSTRUCCIONES A CÓDIGO G

A continuación se muestran los códigos generales y misceláneos más utilizados en este lenguaje, así como su interpretación.

Códigos Generales

G00: Posicionamiento rápido (sin maquinar)

G01: Interpolación lineal (maquinando)

G02: Interpolación circular (horaria)

G03: Interpolación circular (anti horaria)

G04: Compás de espera

G10: Ajuste del valor de offset del programa

G20: Comienzo de uso de unidades imperiales (pulgadas)

G21: Comienzo de uso de unidades métricas

G28: Volver al home de la máquina

G32: Maquinar una rosca en una pasada

G36: Compensación automática de herramienta en X

G37: Compensación automática de herramienta en Z

G40: Cancelar compensación de radio de curvatura de herramienta

G41: Compensación de radio de curvatura de herramienta a la izquierda

G42: Compensación de radio de curvatura de herramienta a la derecha

G70: Ciclo de acabado

G71: Ciclo de maquinado en torneado

G72: Ciclo de maquinado en frenteado

G73: Repetición de patrón

G74: Taladrado intermitente, con salida para retirar virutas

G76: Maquinar una rosca en múltiples pasadas

G96: Comienzo de desbaste a velocidad tangencial constante

G97: Fin de desbaste a velocidad tangencial constante

G98: Velocidad de alimentación (unidades/min)

G99: Velocidad de alimentación (unidades/revolución)

Códigos Misceláneos

M00: Parada opcional

M01: Parada opcional

M02: Reset del programa

M03: Hacer girar el husillo en sentido horario

M04: Hacer girar el husillo en sentido anti horario

M05: Frenar el husillo

M06: Cambiar de herramienta

M07: Abrir el paso del refrigerante B

M08: Abrir el paso del refrigerante A

M09: Cerrar el paso de los refrigerantes

M10: Abrir mordazas

M11: Cerrar mordazas

M13: Hacer girar el husillo en sentido horario y abrir el paso de refrigerante

M14: Hacer girar el husillo en sentido antihorario y abrir el paso de refrigerante

M30: Finalizar programa y poner el puntero de ejecución en su inicio

M31: Incrementar el contador de partes

M37: Frenar el husillo y abrir la guarda

M38: Abrir la guarda

M39: Cerrar la guarda

M40: Extender el alimentador de piezas

M41: Retraer el alimentador de piezas

M43: Avisar a la cinta transportadora que avance

M44: Avisar a la cinta transportadora que retroceda

M45: Avisar a la cinta transportadora que frene

M48: Maquinar exclusivamente con las velocidades programadas

M49: Cancelar M48

M62: Activar salida auxiliar 1

M63: Activar salida auxiliar 2

M64: Desactivar salida auxiliar 1

M65: Desactivar salida auxiliar 2

M66: Esperar hasta que la entrada 1 esté en ON

M67: Esperar hasta que la entrada 2 esté en ON

M70: Activar espejo en X

M76: Esperar hasta que la entrada 1 esté en OFF

M77: Esperar hasta que la entrada 2 esté en OFF

M80: Desactivar el espejo en X

M98: Llamada a subprograma

M99: Retorno de subprograma

Con el fin de entender mejor este lenguaje se tomará como ejemplo el trazado de un círculo mediante código G y se comentaran las acciones que este realiza mediante la descripción simple del código.

```
% <<< Comienzo del programa >>>
(Header)
(Generated by gcodetools from Inkscape.)
(Using default header. To add your own header create file "header" in the output dir.)
M3 <<< Hacer girar el husillo en sentido horario >>>
(Header end.)
G21 <<< Comienzo de uso de unidades métricas >>>
(Footer)
M5 <<<Frenar el husillo >>>
G00 X0.0000 Y0.0000 <<< Posicionamiento rápido >>>
M2 <<< Reset del programa >>>
(Using default footer. To add your own footer create file "footer" in the output dir.)
(end)
% <<< Fin del programa >>>
```

Inkscape

Inkscape es un programa libre y de código abierto, optimizado para Windows, Mac OS y GNU/Linux, utilizado para crear una gran variedad de gráficos como ilustraciones, iconos, logos y diagramas, mismos que se pueden exportar a diferentes formatos de archivo, tales como el código G.

A continuación se presenta un tutorial básico de cómo manejar dicho programa para generar código G a partir de una imagen.

1. Lo primero es tener el archivo de lo que se pretende crear en formato de imagen .jpg o .png, para este ejemplo utilizaremos la siguiente figura en formato .png

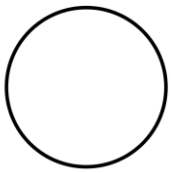


Figura A.1. Ejemplo de una imagen de círculo.

2. Abrimos nuestro programa, donde primeramente encontraremos la siguiente interfaz:

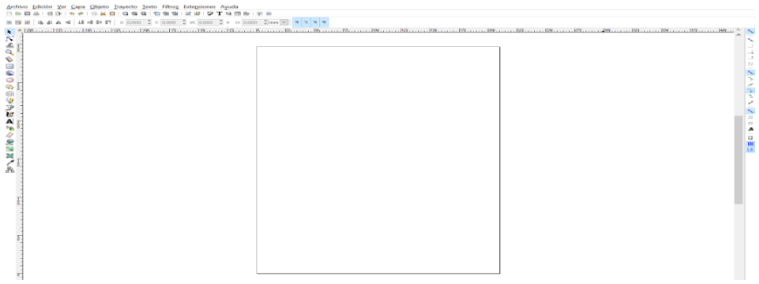


Figura A.2. Interfaz del software Inskape 8.0.

Dicha paquetería trae opciones preestablecidas que al no ser de relevancia se evitará modificar en este tutorial y solo se centrara en obtener el código g a partir de una imagen.

3. Importaremos nuestra imagen a nuestro programa mediante la pestaña de archivo > importar y seleccionamos la ubicación de la imagen con la que trabajaremos.

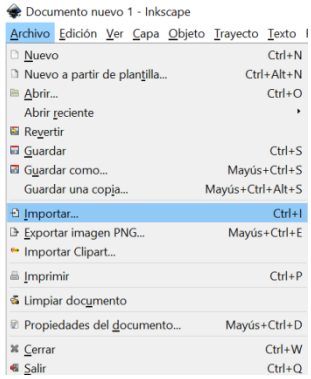


Figura A.3. Proceso para importar una imagen.

3.1 En la siguiente ventana, presionamos continuar sin hacer modificación alguna.

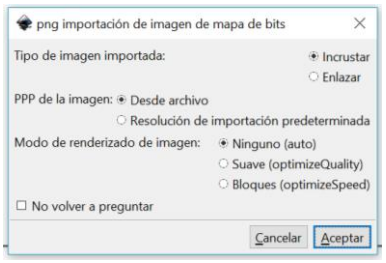


Figura A.4. Proceso para importar una imagen.

4. Escalamos y ubicamos la imagen entro del plano

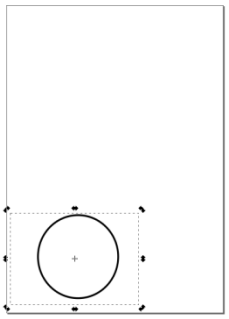


Figura A.5. Ubicamos una imagen dentro del plano.

5. Convertimos la imagen a modo vector mediante la pestaña Trayecto > Vectorizar mapa de bits.

Posteriormente se procede a aceptar los valores predeterminados que se presentan a continuación y cerramos la pantalla.

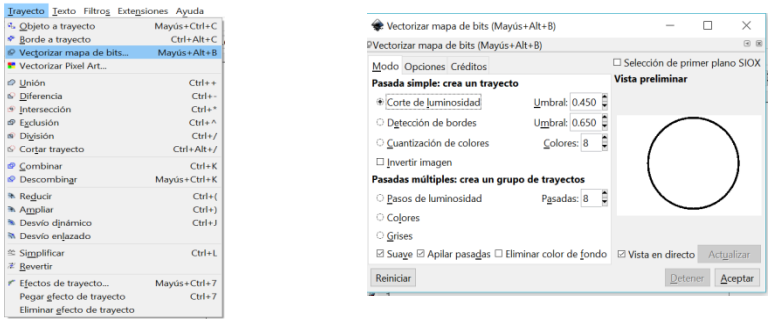


Figura A.6, A.7. Proceso de vectorización de una imagen.

6. Con esto tendremos dos imágenes, la nueva esta encima de la original, la separamos y utilizaremos la nueva, borraremos la original, quedándonos solo con la nueva imagen ya vectrorizada.

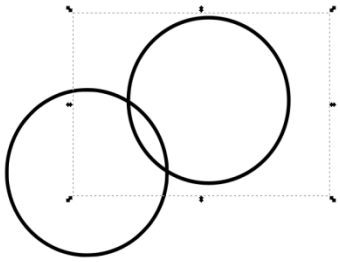


Figura A.8. Separación de imágenes.

7. Seleccionada la imagen, buscamos la opción Trayecto > Objeto a trayecto.

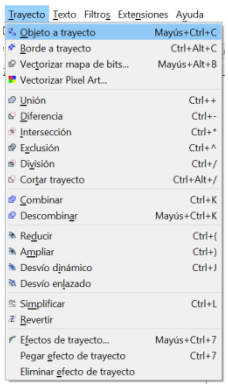


Figura A.9. Ventana para seleccionar el objeto a trayecto.

8. Posteriormente hacemos un desvió dinámico de la imagen mediante la pestaña Trayecto > Desvió dinámico.

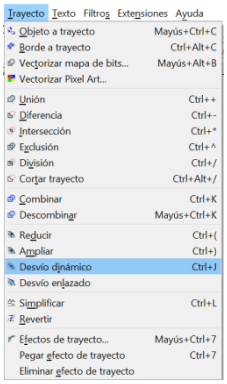
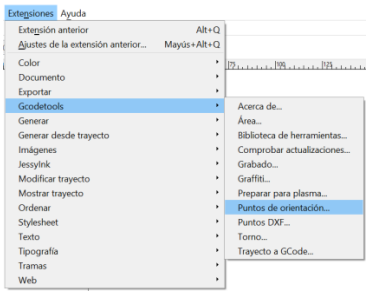


Figura A.10. Ventana para realizar un desvió dinámico de nuestra imagen.

9. Seleccionando la imagen y mediante la pestaña Extensiones > Gcode Tools > Puntos de orientación, podemos personalizar nuestra orientación en el mapa y las características que esta puede tener.



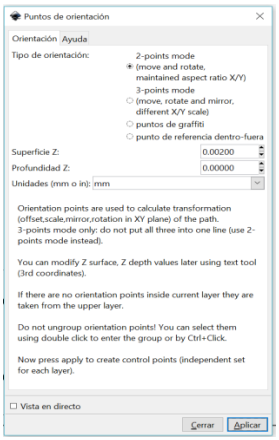


Figura A.11, A.12. Proceso de orientación de una imagen.

10. Seleccionando la imagen podemos definir qué tipo de forma tendrá nuestra herramienta que hará el devastado de la figura.

Para esto accedemos a la pestaña Extensiones > Gcode Tools > Biblioteca de herramientas.

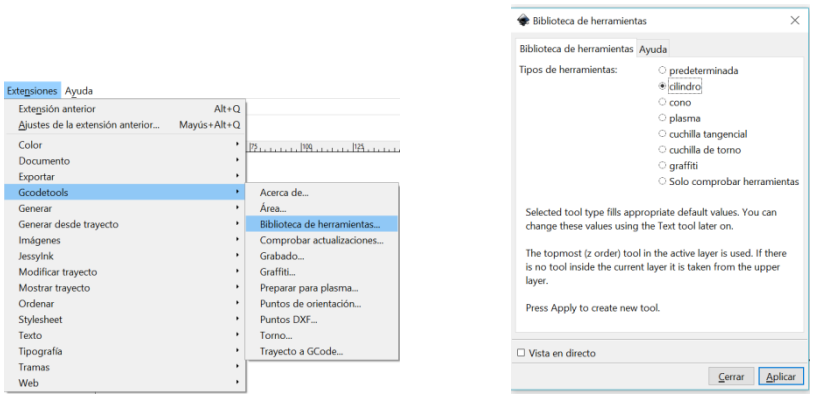


Figura A.13, A.14. Proceso de selección de herramienta de devastado.

En este momento nos saldrá una pantalla verde misma que cambiará dependiendo la herramienta a utilizar. En dicha pantalla verde podremos modificar características importantes de la herramienta a utilizar, tales como su diámetro y ángulo en el que esta devasta.

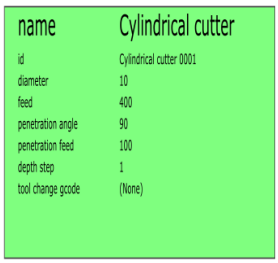


Figura A.15. Ventana donde se cambian parámetros importantes de la herramienta de devastado.

11. Posteriormente veremos las opciones disponibles para la generación de código G, para esto es necesario acceder a la pestaña Extensiones > Gcode Tools > Trayecto a G code.

En la siguiente pantalla, es importante localizarse en la pestaña de Preferencias donde podemos modificar el nombre del archivo de código G, el directorio donde será guardado y las unidades con las que éste trabajará.

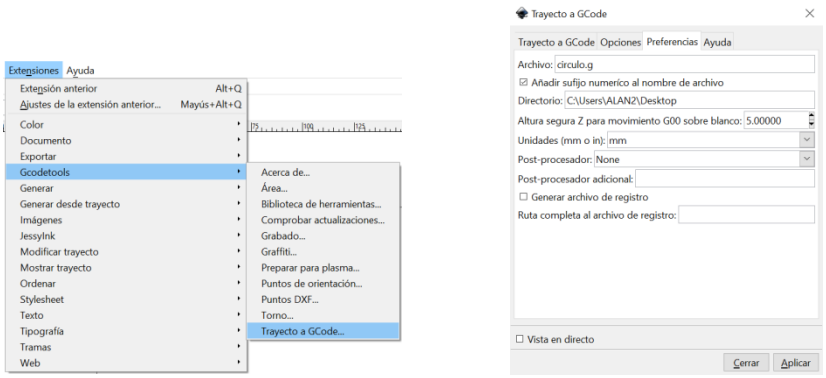


Figura A.16, A.17. Proceso para seleccionar características del código G.

Para terminar de forma correcta con la generación del código es importante regresar a la pestaña de “Trayecto a Gcode” y posteriormente aplicar los cambios.

12. Terminado este proceso, nuestro archivo a Gcode se encontrará listo en el directorio que elegimos en el punto anterior.

A continuación, se agregan las líneas del código G generado por el software Inskape 8.0

```
%
(Header)
(Generated by gcodetools from Inkscape.)
(Using default header. To add your own header create file "header" in the output dir.)
M3
(Header end.)
G21 (All units in mm)
(Footer)
M5
G00 X0.0000 Y0.0000
M2
(Using default footer. To add your own footer create file "footer" in the output dir.)
(end)
%
```

Universal Gcode Sender

Universal Gcode Sender es un programa libre utilizado para interactuar con Grbl presentando una interfaz gráfica en la cual el usuario puede configurar diferentes aspectos físicos referidos al sistema mecánico de la máquina que se desea controlar.

Su interfaz es la siguiente:

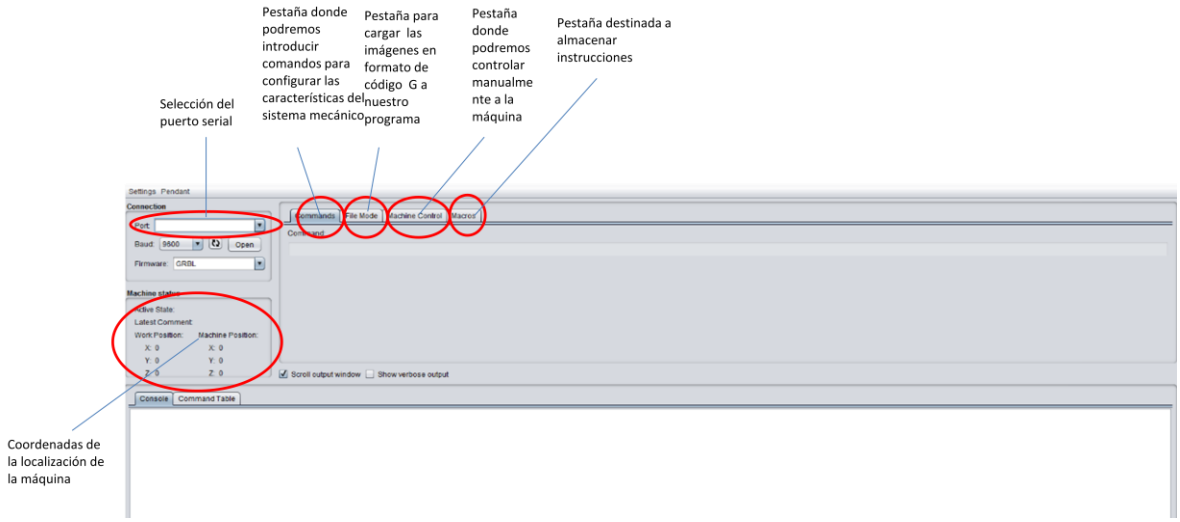


Figura A.18. Interfaz de Universal Gcode Sender.

Para trabajar con el programa lo primero que debemos de hacer es conectar el microcontrolador, esto lo hacemos en la siguiente sección.

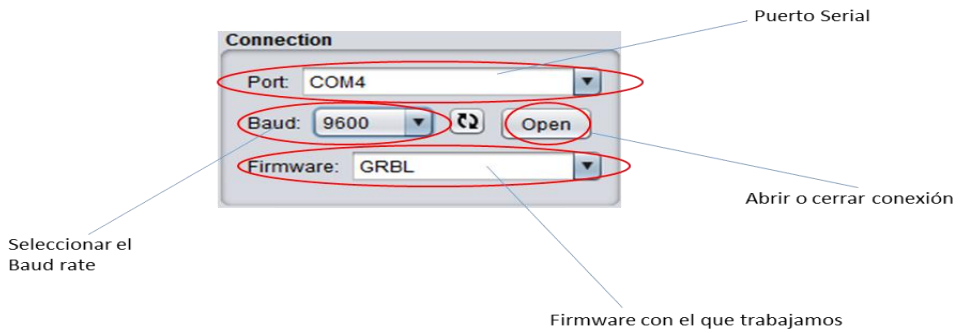


Figura A .19. Conexión del microcontrolador.

1.- Seleccionamos el puerto donde trabajaremos.
2.- Seleccionamos el Baud Rate de trabajo (9600 para la primera versión, 115200 para última versión).
3.- Corroboramos que trabajamos con el firmware GRBL.
4.- Abrimos conexión.

Una vez esto hecho el microcontrolador estará conectado con la computadora y mediante la interfaz de Universal Gcode Sender podremos hacer las siguientes tareas:

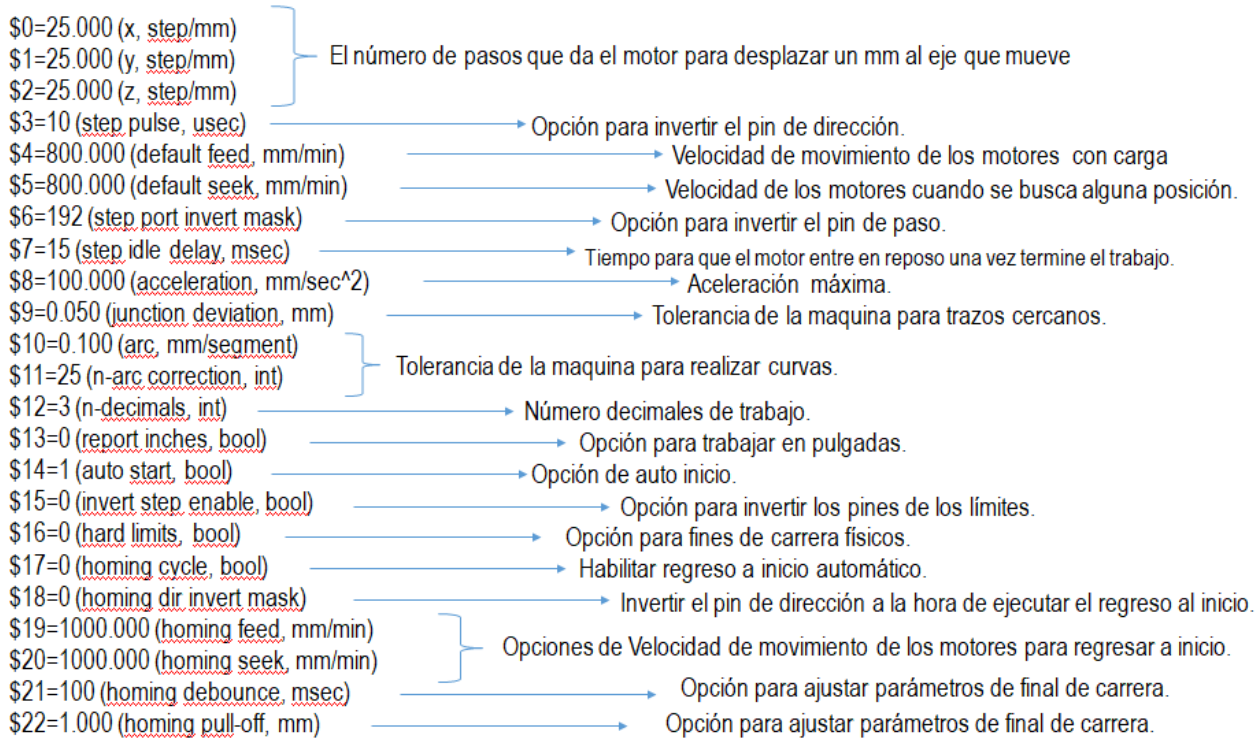
1.- Acoplar la configuración de Grbl con las características mecánicas de la máquina.
2.- Controlar la máquina para posicionarla en cualquier punto de su espacio tridimensional de trabajo.
3.-Enviar archivos en código G al microcontrolador para que la máquina los ejecute.

A continuación, se describirán los pasos para poder realizar las tres tareas anteriormente mencionadas.

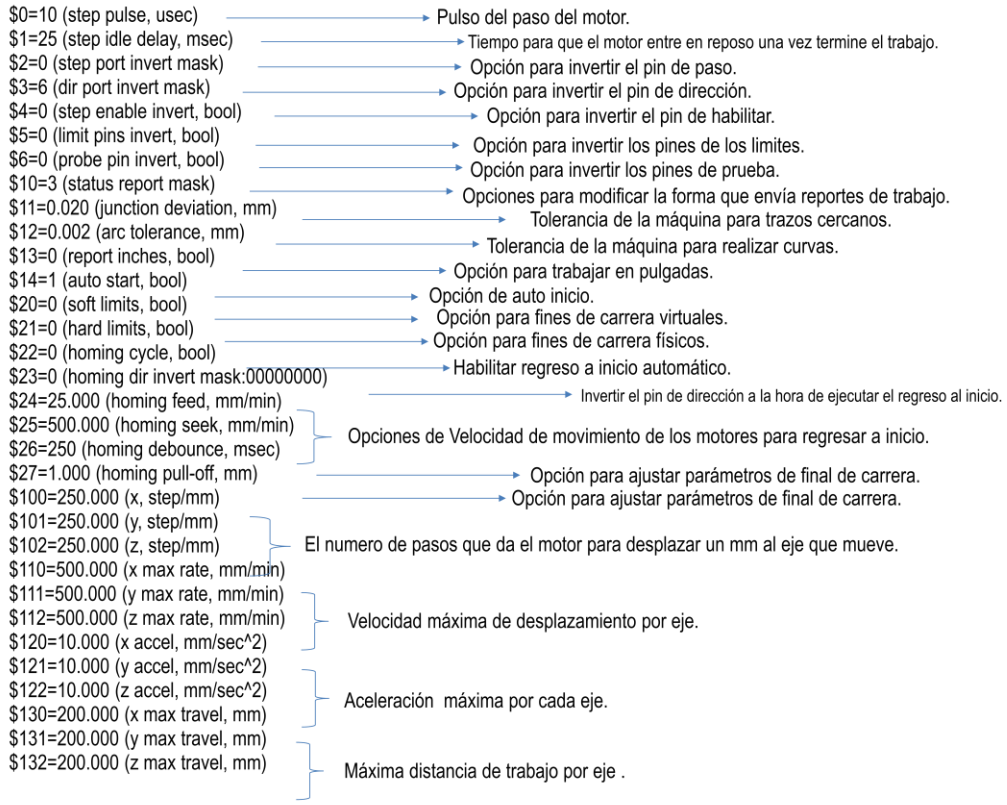
Acoplar la configuración de Grbl con las características mecánicas de la máquina.

Para configurar las características mecánicas de la máquina debemos de ir a la pestaña commands donde mediante el comando “$$” podremos desglosar un menú donde se muestran las especificaciones actuales de máquina.

Para la versión de 0.8 de Grbl son las siguientes:



Para la versión de Grbl 0.9 podemos configurar las siguientes características:



En ambas versiones la interacción con Universal Gcode Sender es la misma, puesto que la versión del programa no es la que cambia, solo cambia la versión de firmware con el que interactúa.

Para cambiar cualquier configuración de los parámetros solo es necesario teclear en la barra de comando la siguiente estructura que varía dependiendo del parámetro a configurar.

$#=value

Donde

$ Es el carácter para indicar que vamos a elegir cierto parámetro para modificar.

Es el número que representa el valor a representar.
= Es el carácter para asignarle un valor al parámetro a modificar.
Value es el valor que tomará el parámetro a modificar.

Por ejemplo, para modificar a 250 el valor del número de pasos que da el motor para desplazarse un mm en el eje Z en la versión Grbl 0.8 escribiríamos $2=250 y presionaríamos enter.

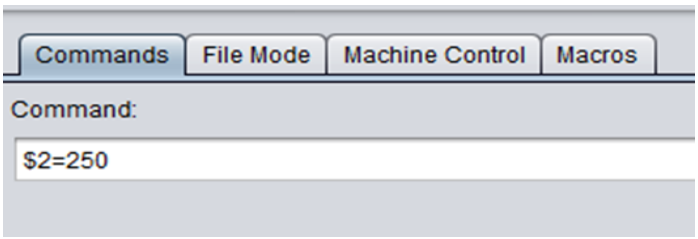
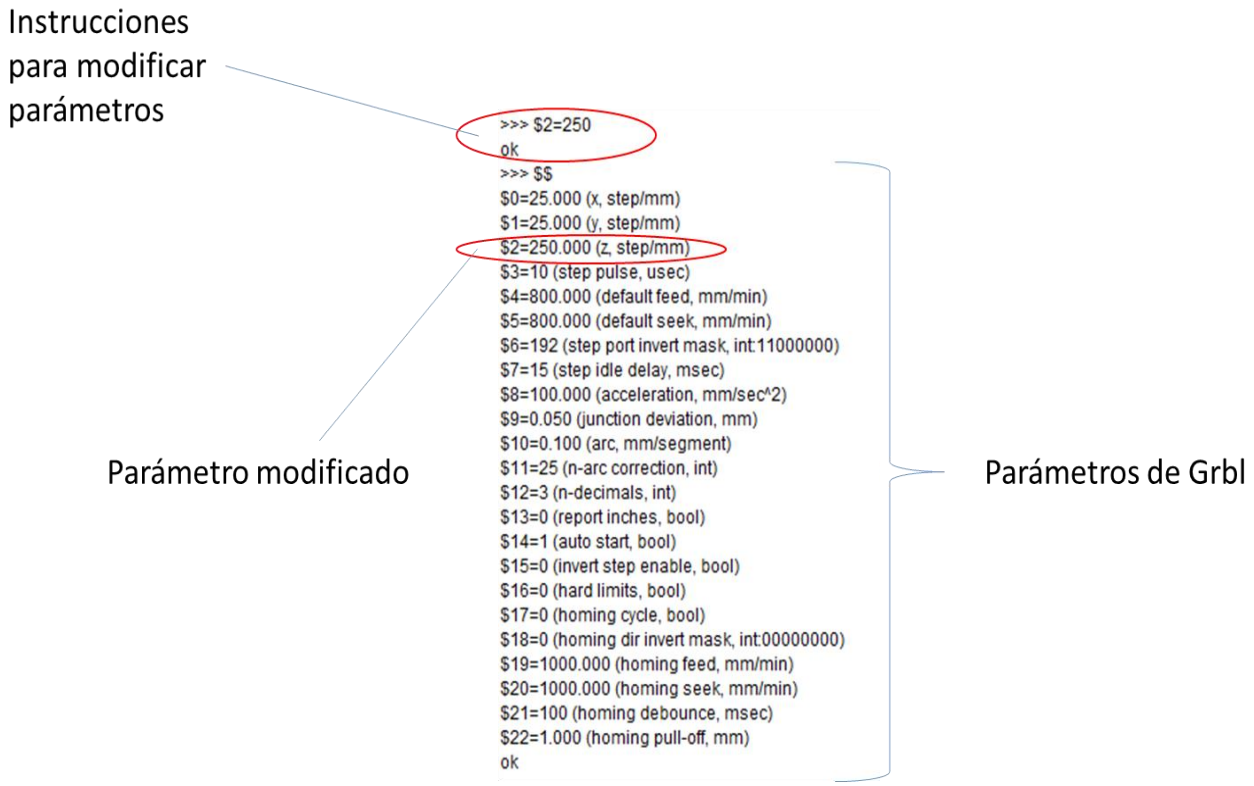


Figura A.20. Ejemplo de cambio parámetros.

Posteriormente podemos confirmar en la consola que las instrucciones fueron modificadas



Cuando conectamos la máquina por primera vez nos encontramos con parámetros preestablecidos, es importante modificar los mismos en función a las características físicas de nuestro sistema, de no ser así los motores se pueden calentar, perder pasos, ir lentos o no funcionar.

Controlar la máquina para posicionarla en cualquier punto de su espacio tridimensional de trabajo.

Para lograr posicionar la máquina en cualquier punto de su plano tridimensional trabajamos en la pestaña “machine control”. Una vez en ella podemos mover la máquina de manera en cada uno de sus ejes hasta posicionarla en un punto de conveniencia, esto nos permite tener libre albedrio para seleccionar el punto de inicio de trabajo.

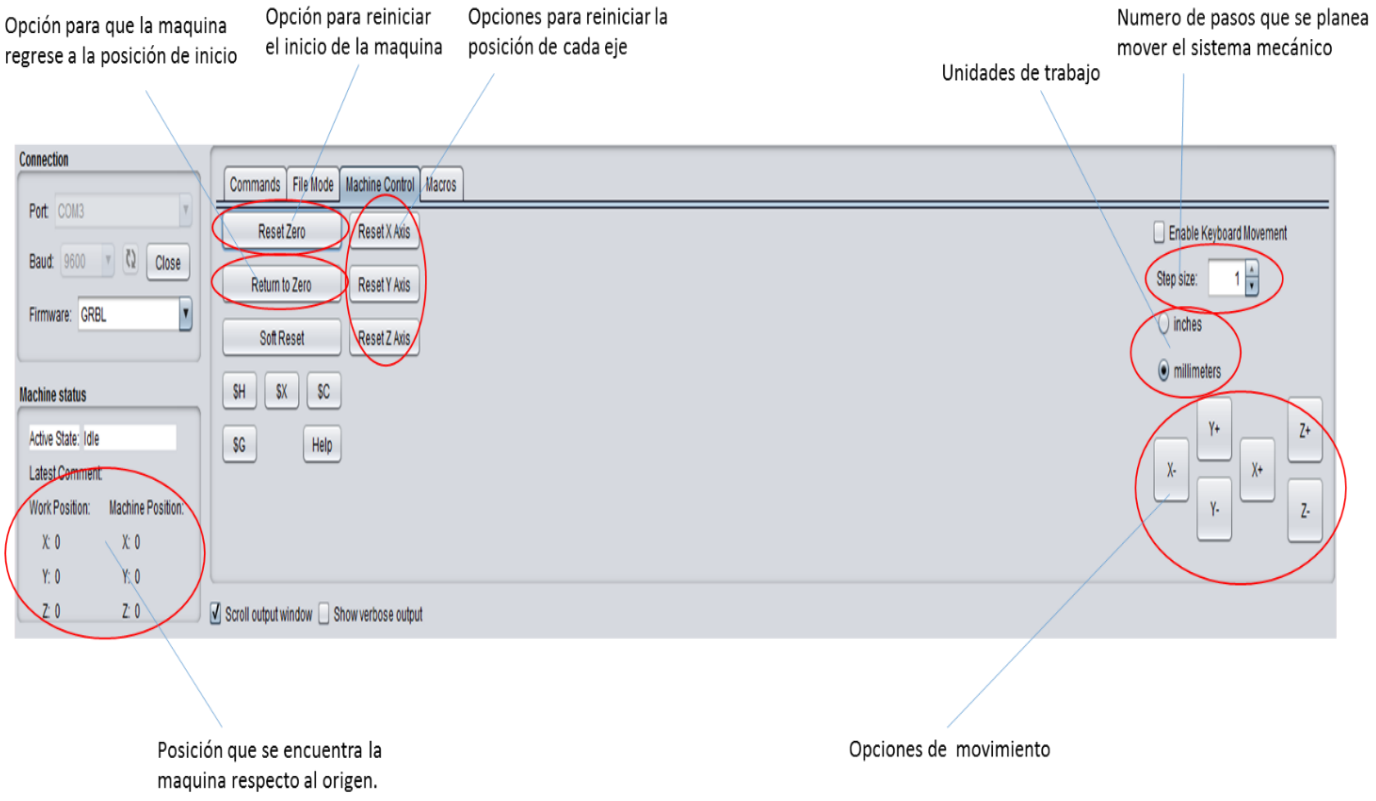


Figura A.21. Posicionamiento manual de la máquina.

Enviar archivos en código G al microcontrolador para que la máquina los ejecute.

Para enviar archivos en código G al microcontrolador entramos a la pestaña “file mode”

Ya en la pestaña, podemos observar un botón llamado browse, el mismo nos permitirá seleccionar de nuestra computadora un archivo en formato código G, mismo con el que se trabajará.

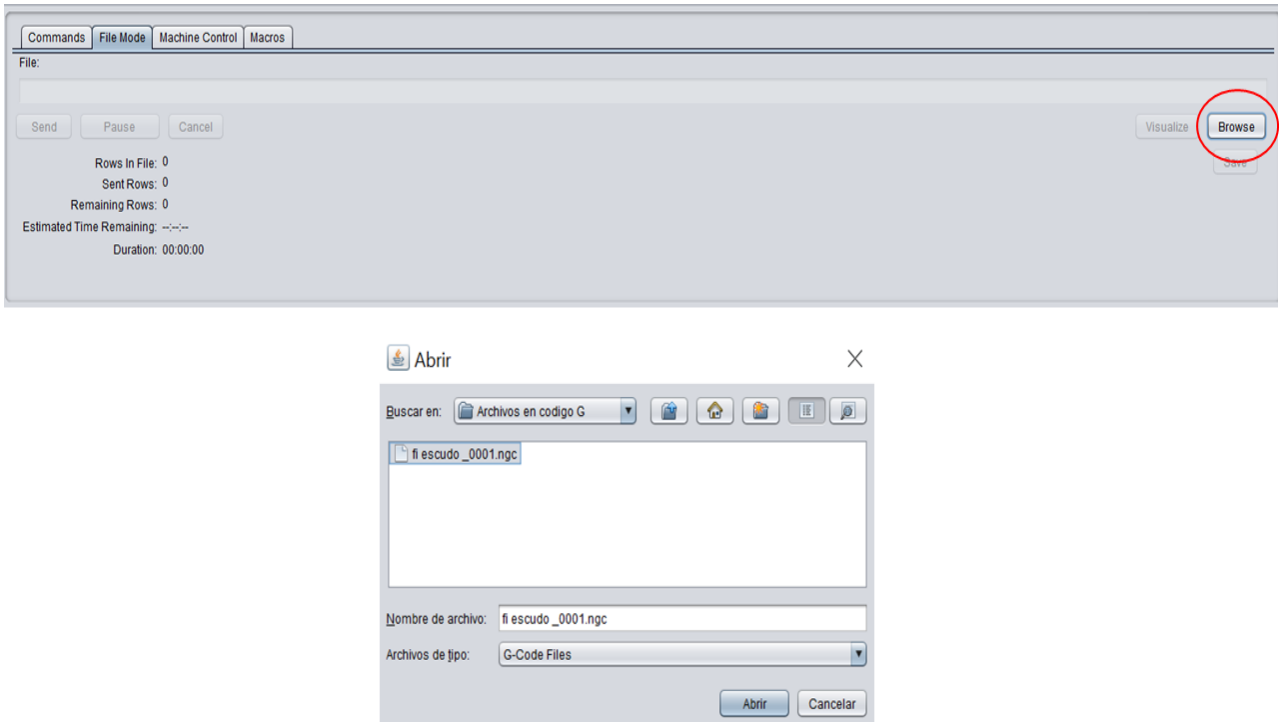


Figura A.22. Proceso de selección de archivo en formato código G.

Posteriormente se nos habilitará un botón llamado “visualize” que nos permitirá visualizar el archivo de código G.

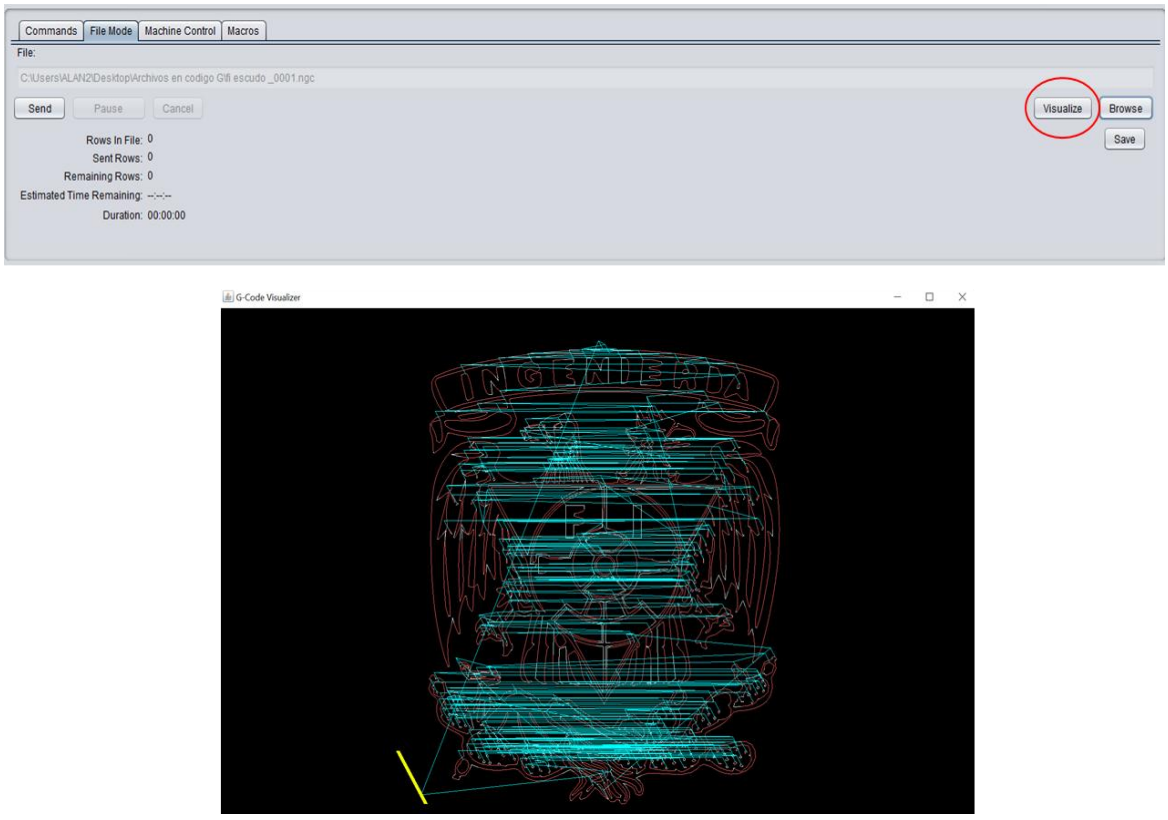


Figura A.23. Visualización de una imagen en código G con Universal Gcode Sender.

Instalar GRBL en el microcontrolador Atmega328

Para las versiones 0.8 y 0.9 de GRBL, la forma de instalación es la misma. Ésta comienza seleccionando y descargando la versión deseada del programa mediante un archivo hexadecimal en la página del desarrollador.

https://github.com/grbl/grbl

Posteriormente el desarrollador nos presenta la opción de descarga de un programa llamado Xloader mediante su sitio.

https://github.com/xinabox/xLoader

Una vez descargado y descomprimido el archivo, aparecerá el programa, éste permitirá cargar los datos al microcontrolador y hacer uso del programa GRBL, para lograr esto se realizan los siguientes pasos:

1.-Ejecutar el programa

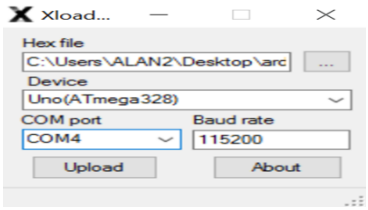


Figura A.24. Interfaz del programa Xloader.

2.-Seleccionar la ubicación del archivo .hex descargado.

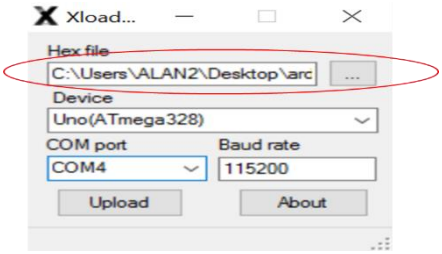


Figura A.25. Selección del archivo .hex.

3.-Seleccionar el microcontrolador con el que se trabajará.

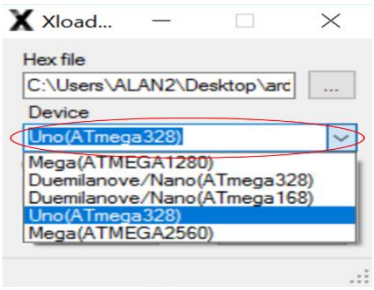


Figura A.26. Selección del microcontrolador.

4.- Seleccionar el puerto serial al que está conectado el microcontrolador y el Baud Rate, para la versión 0.8, éste se recomienda de 9600 y para la versión 0.9 de 115200.

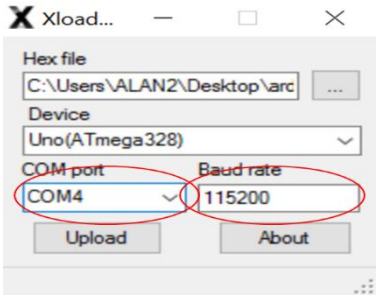


Figura A.27. Selección del puerto serial y del Baud Rate.

5.- Después de realizar estos pasos, solo basta con presionar el botón “upload”, para que Xloader envié el programa al microcontrolador y éste sea capaz de procesar código G.

Encontrar los pasos por mm que recorre nuestro sistema mecánico.

Los pasos por cada mm que recorre el sistema mecánico van en función con la varilla roscada utilizada y de las especificaciones del motor a pasos, el proceso para obtener este dato es el siguiente:

1.- Obtener el número de grados por paso que da nuestro motor (este dato se puede encontrar en la hoja de especificaciones del fabricante), en este caso el motor que utilizamos para el proyecto tiene el valor de 1.8° por paso.

2.- Aplicar la siguiente fórmula para encontrar el número de pasos que da el motor para dar una vuelta completa.

$$\#Pasos = \frac{360°}{Grados\ PP} \qquad \textit{Ecuación A.1}$$

Donde

#Pasos es el número de pasos que necesita dar el motor para dar una vuelta completa.

Grados PP son los grados por paso que da el motor.

Para nuestro ejemplo, sustituyendo valores tenemos:

$$\#Pasos = \frac{360°}{1.8°} \qquad \textit{Ecuación A.2}$$

$$\#Pasos = 200 \qquad \textit{Ecuación A.3}$$

3.- Después de obtener este valor, buscamos la distancia que recorre la varilla roscada por cada vuelta, en este caso tenemos que la varilla que utilizamos, al ser una varilla de cuatro hilos, recorre 8mm por vuelta, con este valor podemos calcular el número de pasos que se requieren para recorrer un mm de distancia, para esto aplicamos la siguiente ecuación.

$$\#Ppmm = \frac{\#Pasos}{mmpv} \qquad \textit{Ecuación A.4}$$

Donde:

#Ppmm es el número de pasos requeridos para que el sistema mecánico recorra un mm.

#Pasos es el número de pasos que necesita dar el motor para dar una vuelta completa.

mmpv son los milímetros por vuelta que da la varilla roscada.

Para nuestro ejemplo sustituyendo valores, tenemos lo siguiente:

$$\#Ppmm = \frac{200}{8} \qquad \textit{Ecuación A.5}$$

$$\#Ppmm = 25 \text{ [pasos/mm]} \qquad \textit{Ecuación A.6}$$

Este valor se debe de calcular para cada eje de movimiento, debido que en el proyecto se utilizó el mismo tipo de varilla y motor, este cálculo vale para todos los ejes del sistema mecánico.

REFERENCIAS

BIBLIOGRAFÍA

[1] A.Albert, Understanding CNC Routers, FPInovations.

[2] Geoff Williams Robotica CN, Mc Graw Hill 2003

[3] F.Rivera Roman, Pracricas de torno de CNC, Universidad de cordova 2002

[4] H.A.Morales Rios << Diseño mecanico de la estructura de un router CNC>>
Que para optar el grado de maestro de ingenieria, UNAM 2012

MESOGRAFÍA

[1] Allegro, Microsistems DMOS Microstepping Driver with Translator And Overcurrent Protection [En linea]
https://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf

[2] S.Svale Skogsrud y S.K. Jeon, GitHub <<Grbl>> [En línea]
https://github.com/grbl/grbl

[3] Pololu, Robotics and electronics A4988 Stepper Motor Driver Carrier [En línea]
https://www.pololu.com/product/1182

[4] Mouser, 8-bit Microcontroller ATmega328
https://www.mouser.com/pdfdocs/Gravitech_ATMEGA328_datasheet.pdf

[5] Statcboards GRBL a fondo [En línea]
https://www.staticboards.es/blog/dominar-motor-paso-a-paso-con-grbl/