

**IMPLEMENTACIÓN DE MÁQUINA CNC PARA EL LABORATORIO DE  
MECATRÓNICA INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO**

**GUSTAVO ALBERTO CALY GOMEZ  
ANDRES GIL RUIZ**

**INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO  
FACULTAD DE INGENIERÍA  
TECNOLOGÍA EN SISTEMAS MECATRÓNICOS  
MEDELLÍN  
2017**

**IMPLEMENTACIÓN DE MÁQUINA CNC PARA EL LABORATORIO DE  
MECATRÓNICA INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO**

**GUSTAVO ALBERTO CALY  
ANDRES GIL RUIZ**

**Trabajo de grado para optar al título de tecnólogo en sistemas  
mecatrónicos**

**Asesor  
Carlos Mario Moreno Paniagua  
Ingeniero Electricista**

**INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO  
FACULTAD DE INGENIERÍA  
TECNOLOGÍA EN SISTEMAS MECATRÓNICOS  
MEDELLÍN  
2017**

## Contenido

	Pág.
Resumen.....	9
Introducción .....	12
1. Planteamiento del problema.....	13
1.1 Descripción.....	13
1.2 Formulación.....	13
2. Justificación .....	14
3. Objetivos .....	15
3.1 Objetivo General .....	15
3.2 Objetivo específicos .....	15
4. Marco Teórico.....	16
4.1 CNC.....	16
4.1.1 Características de la CNC.....	16
4.1.2 Ventajas .....	17
4.1.3 Desventajas. ....	17
4.2 Ejes de la CNC .....	17
4.4.2 Desventajas.....	19
4.5 Interfaz de potencia .....	20
4.6 Arduino.....	20
4.6.1 Har .....	21
4.6.2 Software de Arduino.....	22
4.7 G-Código .....	22
4.8. Control numérico.....	23
4.8.1 Ambito de aplicación del control numérico.....	24
4.8.2 Ventajas del control numérico.....	24
4.8.3 Introducción al control numérico computarizado CNC.....	25
5. Metodología .....	26
5.1 Tipo de proyecto.....	26
5.2 Método.....	26
5.3 Instrumentos de recolección de la información.....	26

5.3.1 Fuentes primarias. Observación directa, entrevistas.....	26
5.3.2 Fuentes secundarias. Libros técnicos, catálogos e internet.....	26
6. Resultados Del Proyecto .....	27
6.1 Construcción de la estructura de la CNC .....	27
6.1.1. Lista de componentes mecánicos: .....	27
6.1.2 Paso 1.....	28
6.1.3 Paso 2.....	30
6.1.4 Paso 3.....	30
6.1.5 Paso 4.....	32
6.1.6 Paso 5.....	33
6.1.7 Paso 6.....	34
6.1.8 Paso 7.....	36
6.1.9 Paso 8.....	37
6.1.10 Paso 9.....	38
6.1.11 Paso 10.....	40
6.1.12 Paso 11.....	41
6.1.13 Paso 12.....	42
6.1.14 pasó 13.....	43
6.1.14 Configuración de los micropasos.....	44
6.1.15 Configuración del cuarto eje.....	46
6.2 Equipamiento del sistema del control de potencia .....	48
6.2.1 GRBL Shield. ....	48
6.2.2 Drivers. ....	50
6.2.3 Motores. ....	51
6.2.4 Accionamiento de la fresadora. ....	52
6.2.5 Alimentación.....	54
6.2.6 Disipación del calor. ....	55
6.2.7 Interferencias electromagnéticas. ....	56
6.2.8 pequeña guía de instalación del driver de potencias.....	57
6.3 sistema de comunicaciones.....	59
6.3.1 Programación de Arduino.....	60

6.3.2 Instalación del Sketch de GRBL para Windows. ....	62
6.4 software de control numérico .....	63
6.4.1 Interface máquina – ordenador. ....	63
6.4.2 instalación del controlador.....	66
6.4.3 Configuración de GRBL Controller. ....	71
7. Conclusiones .....	73
8. Recomendaciones.....	74
8. Referencias bibliográficas .....	75
Referencias .....	75
10. Bibliografía .....	76
11. Anexos .....	77

## Lista de figuras

	Pág.
Figura 1. CNC .....	16
Figura 2. Ejes .....	18
Figura 3. Huesillo.....	18
Figura 4. Tambor de herramientas .....	19
Figura 5. Interfaz de potencia .....	20
Figura 6 Arduino.....	21
Figura 7 Hardware Arduino .....	21
Figura 8 Software Arduino.....	22
Figura 9 Código G.....	23
Figura 10 Control numérico.....	23
Figura 11 Listado de piezas.....	28
Figura 12 Base eje x.....	28
Figura 13 Tornillo de ensambles.....	29
Figura 14 Acoples angulares.....	29
Figura 15 Estructura eje Y .....	30
Figura 16 Eje X y Y con su respectivos acoples angulares.....	31
Figura 17 Eje X y Y acopladas .....	31
Figura 18 Pieza lateral derecha para el eje Y .....	32
Figura 19 Pieza lateral izquierda para el eje Y .....	32
Figura 20 Tornillos de ensambles con su respectiva tuerca.....	33
Figura 21 Pieza lateral derecha ajustado al eje y .....	33
Figura 22 Pieza lateral izquierda ajustada al eje y .....	34
Figura 23 Eje de rodamiento 8mm eje x .....	34
Figura 24 Eje de rodamiento 8mm eje y .....	35
Figura 25 Eje de rodamiento 8mm eje z .....	35
Figura 26 Eje de rodamiento ensamblado.....	36
Figura 27 Estructura para el montaje de motor de fresado .....	36
Figura 28 Ensamble completo de la estructura del motor de fresado .....	37
Figura 29 Piezas laterales para la estructura del eje y.....	37

Figura 30 Estructura terminada.....	38
Figura 31 Motor stepper atornillado .....	38
Figura 32 Motor stepper con arandelas largas .....	39
Figura 33 Motor stepper ensamblado en la estructura del eje z.....	39
Figura 34 Ensamble de la estructura del eje z a la del eje y.....	40
Figura 35 Ensamble de las dos estructuras a la base metálica .....	40
Figura 36 Piezas de laterales para el montaje de la cama de fresado.....	41
Figura 37 Ensamble del eje de rodamiento a la cama de fresado .....	42
Figura 38 Otro Angulo para la vista de la cama de fresado con el eje.....	42
Figura 39 Total montaje de la cama de fresado con el eje de rodamiento .....	43
Figura 40 Ensamble completo de la estructura CNC .....	43
Figura 41 Ensamble completo de la estructura los motores stepper .....	44
Figura 42 Configuración micropasos shield GRBL.....	45
Figura 43 Cuarto eje como eje X .....	46
Figura 44 Cuarto eje como eje Y .....	47
Figura 45 Cuarto eje como eje Z.....	47
Figura 46 Cuarto eje controlado por los pines 12 y 13 .....	48
Figura 47 Shield GRBL .....	49
Figura 48 Esquemático shield GRBL .....	49
Figura 49 Detalle pin Vref A4988 .....	50
Figura 50 Relé fresadora .....	53
Figura 51 Conexiones fuente de alimentación .....	55
Figura 52 Detalle disipador driver .....	56
Figura 53 Driver con cable de alimentación .....	57
Figura 54 Conexión driver – relé .....	57
Figura 55 Empalme de cables de terminales de la fuente .....	58
Figura 56 Cables de la fuente conectados al relé .....	58
Figura 57 Motor de fresado conectado a la fuente.....	59
Figura 58 Conectando los motores stepper a los driver de potencia.....	59
Figura 59 Código de control CNC .....	60
Figura 60 Esquemático de control de los ejes de la CNC .....	61

Figura 61 Conexión de los driver de los tres ejes .....	62
Figura 62 Configuración de interfaz_ordenador .....	63
Figura 63 Interfaz del software de control .....	64
Figura 64 Conexión arduino- ordenador .....	66
Figura 65 Instalación del driver de comunicación .....	66
Figura 66 Software de control del código g CNC.....	67
Figura 67 Vista de cómo debe aparecer el mensaje de comunicación .....	67
Figura 68 Acceso a configuración.....	68
Figura 69 Configuración del eje z.....	69
Figura 70 Línea de código g.....	70
Figura 71 precisión de los ejes X, Y, Z.....	70
Figura 72 Ajuste de direccionamiento del eje Y.....	71
Figura 73 Configuración GRBL .....	72



## Lista de tablas

	Pág.
Tabla 1. Componentes mecánicos.....	27
Tabla 2. Configuración dirver Pololu A988.....	45
Tabla 3. Confugración drever Pololy DRV8825.....	46
Tabla 5 Motores Nema 23.....	51
Tabla 6 Características del relé .....	53
Tabla 7 Fuente de poder.....	54
Tabla 8 compatibilidad del software .....	65

## **Resumen**

### **TÍTULO COMPLETO DEL PROYECTO DE GRADO**

**Gustavo Alberto Caly Gómez**

**Andrés Ricardo Gil Ruiz**

Este proyecto surge de la necesidad de crear los prototipos y materia prima para proyectos pequeños, Enfocados de manera creciente, tales como moldes y serie de piezas específicas de prototipos. Creados para lograr una mayor eficiencia de la producción en serie, ya sea de operaciones de fresado, mandrilado y perforado. Y así sumada a numerosos y nuevos requerimientos, que día a día aparecen forzando la utilización de nuevas técnicas, que reemplazarán al operador humano en casos donde hay la necesidad de fabricar productos, que no se podían conseguir en cantidad y calidad suficientes. Logrando no recurrir a la automatización del proceso de fabricación o piezas que en su estructura, fueran casi imposibles o muy difíciles de realizar por ser excesivamente complejas, para ser controladas por un operador humano. Durante este proyecto logramos desarrollar e implementar una CNC de hardware y software libre para el laboratorio de mecatrónica.

*Palabras Claves: CNC, Arduino, Driver, Ejes, Software, Hardware*

## **Abstract**

This Project begin from need for create prototypes and raw material for small projects, centred exponentially, such as molds and a variety of parts specific of prototypes.

Designed for achieve efficient of production in mass. Either of milling operations, boring and perforated. And the summing to variety and new requirements that each day appear for use new skills, that replace to the operator human, in situations that have need to build products that no is possible get in mass and large quantities and quality, and so not need to automate the process to building of part that in your structure are almost impossible or very hard make for be much complex for be controlled for a human operator. During this project we were able to develop and implement a CNC of hardware and free software for the mechatronics laboratory.

*Keywords: CNC, arduino, driver, Axis, software, hardware*

## Glosario

**Acoples:** se llama acoplamiento mecánico a una serie de enlaces rígidos, con ligamentos que forman una cadena cerrada o una serie de cadenas cerradas.

**Arduino:** es una plataforma de prototipos electrónica de código abierto

**CNC:** control numérico computarizado

**Disipadores de calor:** un disipador es un instrumento que se utiliza para bajar la temperatura de algunos componentes electrónico

**Driver:** controlador, rutina o programa que enlaza un dispositivo periférico al sistema operativo.

**Ejes:** es un elemento constructivo destinado a guiar el movimiento de rotación a una pieza o de un conjunto de piezas

**Ejes de rodamientos:** es un elemento mecánico que reduce la fricción entre un eje y las piezas conectadas a este por medio de una rodadura, que le sirve de apoyo y facilita su desplazamiento.

**Fresadoras:** es una herramienta para realizar trabajos mecanizados por arranque de viruta

**Lenguaje de programación:** un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras.

**Mototool:** Estas es una herramienta neumática para desbaste y rectificado de diferentes materiales.

**Motores stepper:** El motor paso a paso conocido también como motor de pasos es un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamientos angulares discretos

## **Introducción**

La CNC es una máquina de control numérico computarizado, que desde sus inicios fue uno de los logros más importantes en la industria, por precisión, y desarrollo de proyectos industriales mecanizados, que facilitan la fabricación de moldes con un alto grado de complejidad y exactitud al momento de su fabricación.

Por eso en la pedagogía es importante enseñar a controlarla, y comprender su funcionamiento, para una competencia elevada en el campo laboral respecto, a estas máquinas de control numérico.

El presente proyecto hace referencia a la construcción y equipamiento de esta máquina. Para equipar el laboratorio de mecatrónica, logrando así que los estudiantes y profesores se beneficien de esta máquina de control numérico, para complementar en sus competencias en el campo laboral.

El proyecto está constituido por varias partes, primero se presenta la razón del porque se elaboró este proyecto, su justificación, logrando mostrar luego sus objetivos generarles como específicos, este proyecto también muestra más adelante, un marco teórico, donde se da conocimientos técnicos de los elementos que conforman el proyecto. Por último se da a conocer, el desarrollo del proyecto paso a paso con sus especificaciones.

## **1. Planteamiento del problema**

### **1.1 Descripción**

La Institución Universitaria Pascual Bravo es una institución de educación superior adscrita al municipio de Medellín, cuenta con 5000 estudiantes aproximadamente, 23 tecnologías, 2 ingenierías y una especialización.

En esta institución se observa que en las prácticas realizadas en el laboratorio de mecatrónica no hay suficientes máquinas para la enseñanza de un grupo completo en el área de dispositivos CNC, es por eso que los estudiantes no pueden realizar las prácticas en forma individual. Dichos estudiantes cuando salen al campo laboral no tienen todas las habilidades completas para manejar las máquinas de fresado CNC.

La mayoría de las veces todos los alumnos hacen la práctica en una sola máquina y uno solo aprende a controlarla y manipularla; esta incomodidad hace que los estudiantes queden con vacíos en el mantenimiento, control y supervisión de dispositivos CNC.

### **1.2 Formulación**

¿Es posible implementar una maquina CNC con hardware y software libre, para el laboratorio de mecatrónica para que los futuros egresados se desempeñen mejor en el campo laboral?

## **2. Justificación**

La Institución Universitaria Pascual Bravo cuenta con tecnología en sistemas mecatrónicos, tecnología en mecánica automotriz, tecnología electrónica, tecnología en electromecánica, ingeniería eléctrica entre otras.

Donde se ha observado que no todos los estudiantes de estas carreras, saben manipularla y tienen conocimientos básicos respecto a la CNC.

Todos los estudiantes y profesores deben saber controlar estas máquinas, para su buen desempeño en el campo laboral y académico. Por esta razón, la universidad debe capacitar al 100% a sus estudiantes.

Las empresas nacionales están buscando alternativas, que le permitan asegurar las ventas de sus productos. Logrando brindarle al cliente un servicio óptimo. Esto significa invertir en tecnología, capacitación e investigación.

El laboratorio de mecatrónica, está en busca, no solo de disminuir los tiempos de producción y aumentar la calidad en sus dispositivos electrónicos, sino también de adquirir conocimientos en nuevos y mejores procesos de fabricación de hardware necesarios, para sus desarrollos tecnológicos. Así la CNC tendrá más funciones, como hacer moldes en madera, para objetos de fibra de vidrio (altamente necesario en empresas), como ruteadora o para la fabricación de circuitos impresos.

La CNC se destaca por su precisión, rendimiento y confiabilidad característicos, que la han convertido en la mejor alternativa, a la hora de las necesidades relacionadas, con este tipo de procesos. Por esta razón, integrar el talento humano, y las diferentes herramientas tecnológicas que el mundo ofrece, se convierte en un elemento valioso, y diferenciador para competir en el mercado actual.

### **3. Objetivos**

#### **3.1 Objetivo General**

Implementar una máquina CNC, para el laboratorio de mecatrónica, de la Institución Universitaria Pascual Bravo. Donde permita mejorar las condiciones del laboratorio.

#### **3.2 Objetivo específicos**

Construir la estructura de la CNC, que permita darle la estabilidad requerida e implementar el sistema de desplazamiento de cada eje.

Equipar el Sistema de control de potencia, que permita suministrarle la energía suficiente a la CNC.

Proveer un Sistemas de comunicaciones, que logre enlazar un ordenador con la CNC.

Implementar un software que permita leer los datos numéricos necesarios, para el control de funcionamiento de la CNC.



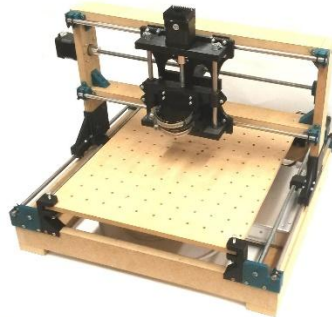
## 4. Marco Teórico

### 4.1 CNC

El control numérico por computadora (CNC), es un sistema que permite controlar en todo momento la posición de un elemento físico. Normalmente es una herramienta que está montada en una máquina. Esto quiere decir, que mediante un software y un conjunto de órdenes, controlaremos las coordenadas de posición de un punto (la herramienta) respecto a un origen (0,0,0 de máquina), o sea, una especie de GPS pero aplicado a la mecanización, y muchísimo más preciso.

La CNC controla todos los movimientos de la herramienta cuando estamos fabricando, y no solo controla las coordenadas que hemos visto, sino también, la manera de desplazarse entre ellas, su velocidad, y algunos parámetros más. Una CNC es un equipo totalmente integrado dentro de máquinas-herramienta de todo tipo, de mecanizado, por láser y por cortadoras.

La pregunta lógica, ¿para qué sirve la CNC? Pues como hemos dicho, nos permite controlar en todo momento cuales son los movimientos de una herramienta, así que nos servirá para obtener piezas con determinadas medidas, para crear programas que nos repitan con gran precisión piezas iguales, también se utiliza, y mucho, para verificar las medidas de algo que ha sido fabricado. (Cadcamcae, 2017)



*Figura 1. CNC*

Fuente: extraído de <http://onlineshop.iniciativas3d.com/Iniciativas3D/maduixa-cnc>

4.1.1 Características de la CNC. El diseño adecuado, de las estructuras en las máquinas y herramientas, requieren el análisis de factores como: la forma, material de las estructuras, esfuerzos, peso, consideraciones de fabricación y rendimiento.

Es una medida cualitativa de la resistencia, a las deformaciones elásticas producidas por un material, que contempla la capacidad de un elemento estructural, para soportar esfuerzos sin adquirir grandes deformaciones. (Taringa, 2017)

4.1.2 Ventajas. Las ventajas del control numérico computarizado, son la facilidad de operación, programación sencilla, mayor exactitud, adaptabilidad y menos costos de mantenimiento, la combinación del diseño con computadora, mayor productividad. (Taringa, 2017)

4.1.3 Desventajas. La desventaja, es que las condiciones que influyen, en las decisiones con la automatización, son los crecientes costos de producción, alto porcentaje de piezas rechazadas, demoras en la producción, escasez de mano de obra y condiciones peligrosas de trabajo. Y los factores que se deben estudiar con cuidado son, el alto costo inicial del equipo, los problemas de mantenimiento y el tipo de producto. (Taringa, 2017)

## 4.2 Ejes de la CNC

Los tornos CNC constan de 03 ejes

Eje Z: Corresponde al recorrido en el eje longitudinal. Se utiliza en las operaciones de cilindrado.

Eje X: Este realiza en un movimiento transversal y corresponde a las operaciones de refrenado, ranurado y tronzado.

Eje Y: Es perpendicular a los otros dos ejes y sirve para determinar la altura de las herramientas respecto a la bancada de la máquina.

Estos ejes tienen incorporada la función de interpolación, es decir, pueden moverse de manera simultánea, pudiéndose conseguir mecanizados cónicos y esféricos, de acuerdo a la geometría de las piezas. (buenastareas, 2011)

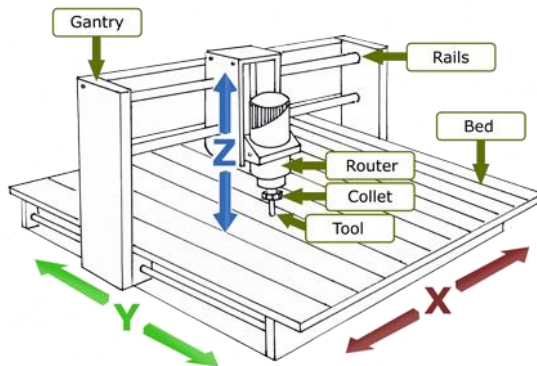


Figura 2. Ejes

Fuente: extraído de <http://www.taringa.net/posts/hazlo-tu-mismo/18799298/Construi-una-maquina-fresadora-CNC-y-te-enseno-como-hacerlo.html>

#### 4.3 Husillo

Son 2 bolas templadas y rectificadas, asegurando una gran precisión en los movimientos de la máquina.

Estos husillos funcionan de tal manera, que un tornillo sin fin, tiene un acoplamiento en los carros.

Cuando el tornillo gira, el carro se desplaza longitudinalmente por las guías de la bancada. Giran así en ambos sentidos (buenastareas, 2011)



Figura 3. Husillo

Fuente: extraído de <http://www.linearguide.es/2-6-ball-screw.html>

#### 4.4 Tambor Porta Herramientas

Dependiendo del tamaño, pueden ir de 06 a 20 herramientas. El cambio de herramientas es controlado mediante el programa de mecanizado. Cuando acaba el mecanizado de la herramienta, los carros retroceden a su posición inicial. (Buenastareas, 2011)



*Figura 4.* Tambor de herramientas

Fuente: extraído de <https://es.slideshare.net/AriadneBC/cnc-004-53666033>

##### 4.4.1 Ventajas.

Permite mayor precisión.

- \* Mecanizar piezas complejas.
- \* El cambio de pieza es sencillo.
- \* Reducción de errores de operarios
- \* El precio no es muy alto. (buenastareas, 2011)

##### 4.4.2 Desventajas.

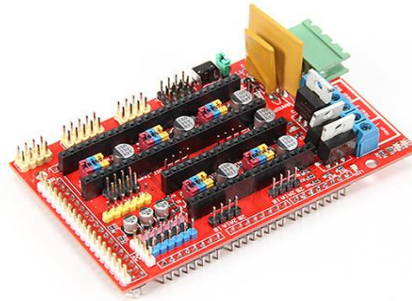
Tener la necesidad de realizar un programa antes del mecanizado.

Coste de accesorios y herramientas.

Existen en pares e impares de acuerdo a las herramientas fijas y móviles. (buenastareas, 2011)

## 4.5 Interfaz de potencia

La interfaz de potencia, son dispositivos intermedios entre nuestro microcontrolador, y aquellos aparatos que requieran cantidades de corriente, mayores a los que pueda suministrar el microcontrolador, como motores paso a paso, motores DC, o servomotores, etc. (Universitat Pompeu Fabra, 2017)



*Figura 5.* Interfaz de potencia

Fuente: extraído de [http://www.evoltapc.cl/catalog/product\\_info.php/products\\_id/1204](http://www.evoltapc.cl/catalog/product_info.php/products_id/1204)

## 4.6 Arduino

Arduino, es una plataforma de prototipos de electrónica, de código abierto, (open-source), basada en hardware y software flexibles, y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquier interesado en crear objetos o entornos interactivos.

Arduino, puede sentir el entorno mediante la recepción de entradas, desde una variedad de sensores, y puede afectar a su alrededor, mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa, usando el Arduino Programming Language (basado en Wiring), y el Arduino Development Environment (basado en Processing). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar, con software en ejecución en un ordenador (por ejemplo con Flash, Processing, MaxMSP, etc.).

Las placas se pueden ensamblar a mano o encargarse preensambladas; el software se puede descargar gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que es libre de adaptarlas a tus necesidades.

Arduino recibió una mención honorífica en la sección Digital Communities del Ars Electronica Prix en 2006. (arduino, 2017)



Figura 6 Arduino

Fuente: extraído de <https://es.wikipedia.org/wiki/Arduino>

#### 4.6.1 Hardware de Arduino.

Consiste en una placa de circuito impreso con un microcontrolador. Usualmente Atmel AVR, puertos digitales y analógicos de entrada/salida, los cuales pueden conectarse a placas de expansión (shields), que amplían las características de funcionamiento de la placa Arduino. Así mismo, posee un puerto de conexión USB, desde donde se puede alimentar la placa, y establecer comunicación con el computador. (Arduino, 2017)

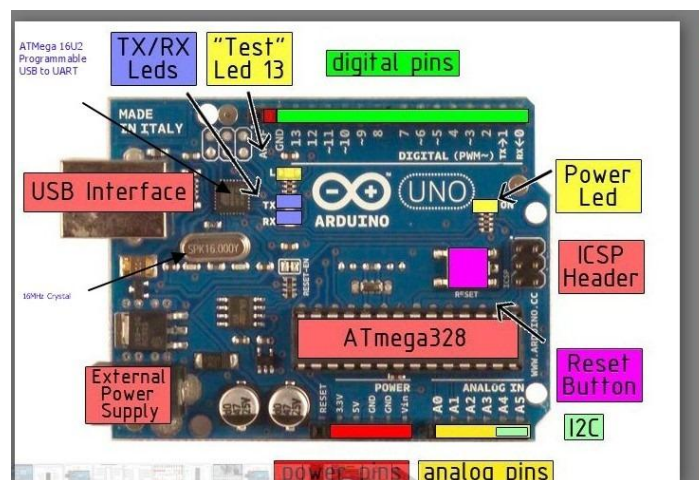


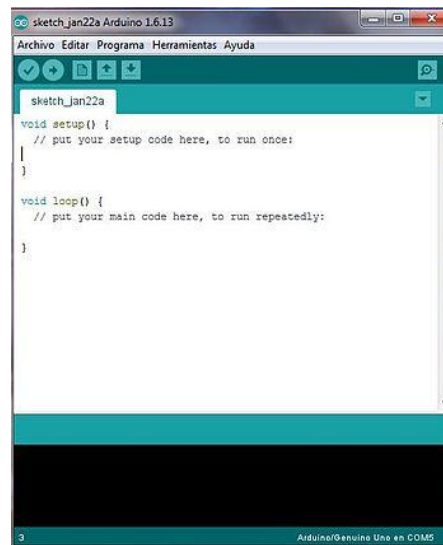
Figura 7 Hardware Arduino

Fuente: extraído de <https://alselectro.wordpress.com/tag/arduino-uno-guide/>

#### 4.6.2 Software de Arduino.

Consiste en un entorno de desarrollo (IDE), basado en el entorno de Processing, y lenguaje de programación basado en Wiring, así como en el cargador de arranque (bootloader), que es ejecutado en la placa.

El microcontrolador de la placa, se programa mediante un computador, usando una comunicación serial, mediante un convertidor de niveles RS-232 a TTL serial (arduino, 2017)



*Figura 8* Software Arduino

Fuente: extraído de <https://es.wikipedia.org/wiki/Arduino>

#### 4.7 G-Código

El G-code, es el nombre que habitualmente, recibe el lenguaje de programación más usado en Control numérico (CNC), el cual posee múltiples implementaciones. Usado principalmente en automatización. Forma parte de la ingeniería asistida por computadora. Al G-code, se le llama en ciertas ocasiones, lenguaje de programación G.

En términos generales, G-code, es un lenguaje, mediante el cual las personas pueden decir a máquina controladas por computadora, qué hacer y cómo hacerlo. Esos "qué" y "cómo", están definidos mayormente por instrucciones, sobre adónde moverse, cuán rápido moverse y qué trayectoria seguir (arduino, 2017)

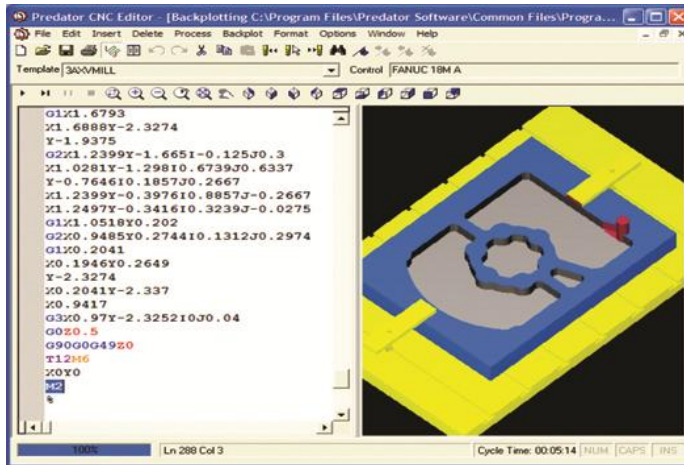


Figura 9 Código G

Fuente: extraído de <http://fabricatupropioroutercnc.com/blog/aprende-a-generar-un-codigo-g.html>

#### 4.8. Control numérico

Definición general: Se considera control numérico, a todo dispositivo capaz de dirigir, posicionamientos de un órgano mecánico móvil, en el que las órdenes relativas a los desplazamientos del móvil, son elaboradas en forma, totalmente automática, a partir de informaciones numéricas definidas, sea manualmente o por medio de un programa.(Slideshare, 2017)



Figura 10 Control numérico

Fuente: extraído de <https://www.logismarket.com.mx/ambar-control-numerico/control-numerico/2043882556-1929219521-p.html>



#### 4.8.1 Ambito de aplicación del control numérico.

Como ya se mencionó, las cuatro variables fundamentales, que inciden en la bondad de un automatismo son: productividad, rapidez, precisión y velocidad. De acuerdo con estas variables, vamos a analizar qué tipo de automatismo es el más conveniente, de acuerdo al número de piezas a fabricar y series de fabricación. Las grandes series (mayor a 10.000 piezas), son producciones que están cubiertas en la actualidad, por las máquinas transfert, realizadas por varios automatismos, trabajando simultáneamente en forma sincronizada. Las series medias (entre 50 y 10.000). Son aquellas, en las que existen varios automatismos, que cubren esta gama, entre ellos los copiadore y los controles numéricos. La utilización de estos automatismos, dependerá de la precisión, flexibilidad y rapidez exigidas. El control numérico, será especialmente interesante, cuando las fabricaciones se mantengan en series, comprendidas entre 5 y 1.000 piezas, que deberán ser repetidos varias veces durante el año. En las series pequeñas (menores a 5 piezas), la utilización del control numérico, suele no ser rentable, a no ser que la pieza sea lo suficientemente compleja, como para justificarse su programación, con ayuda de una computadora. Pero en general, para producciones menores a cinco piezas, la mecanización, en máquinas convencionales, resulta ser más económica.

#### 4.8.2 Ventajas del control numérico.

Las ventajas, dentro de los parámetros de producción, explicados anteriormente son: Posibilidad de fabricación de piezas imposibles o muy difíciles.

Gracias al control numérico, se han podido obtener piezas muy complicadas, como las superficies tridimensionales, necesarias en la fabricación de aviones. Otra de las ventajas es la seguridad. El control numérico, es especialmente recomendable, para el trabajo con productos peligrosos. La Precisión es una ventaja mas, Esto se debe a la mayor precisión de la máquina de control numérico, respecto de las clásicas y el aumento de productividad de las máquinas. Esto debido a la disminución del tiempo total de mecanización, en virtud de la disminución de los tiempos de desplazamiento, en vacío y de la rapidez de los posicionamientos, que suministran los sistemas electrónicos de control. La reducción de controles y desechos es también una ventaja. Reducción, es debida fundamentalmente a la gran fiabilidad y repetitividad de una máquina, con control numérico. Esta reducción de controles, permite

prácticamente, eliminar toda operación humana posterior, con la subsiguiente reducción de costos y tiempos de fabricación. (Slideshare, 2017)

#### 4.8.3 Introducción al control numérico computarizado CNC.

Tuvo su origen, a principios de los años cincuenta, en el Instituto de Tecnología de Massachusetts (MIT), en donde se automatizó, por primera vez una gran fresadora. En esta época las computadoras, estaban en sus inicios, y eran tan grandes, que el espacio ocupado por la computadora, era mayor que el de la máquina.

Hoy día las computadoras, son cada vez más pequeñas y económicas, con lo que el uso de la CNC se ha extendido, a todo tipo de maquinaria sea tornos, rectificadoras, eletroerosionadoras, máquinas de coser, etc.

En una CNC, a diferencia de una máquina convencional o manual. una computadora controla la posición y velocidad de los motores, que accionan los ejes de la máquina. Gracias a esto, puede hacer movimientos, que no se puedan lograr manualmente, como círculos, líneas diagonales y figuras complejas tridimensionales. Las CNC, son capaces de mover, la herramienta al mismo tiempo, en los tres ejes, para ejecutar trayectorias tridimensionales, como las que se requieren, para el maquinado de complejos moldes y troqueles.

En la CNC, una computadora controla el movimiento de la mesa, el carro y el husillo. Una vez programada la máquina, ésta ejecuta todas las operaciones por sí sola, sin necesidad de que el operador esté manejándola. Esto permite aprovechar mejor el tiempo del personal, para que sea más productivo. El término "control numérico", se debe a que las órdenes dadas, a la máquina, son indicadas mediante códigos numéricos. (Slideshare, 2017)

## **5. Metodología**

### **5.1 Tipo de proyecto**

**Descriptivo.** Porque es un conocimiento, obtenido por medio de observaciones, sin intervención, de un proceso de medición.

### **5.2 Método**

**Deductivo.** Es parte de una teoría que ya existe, para solucionar un problema, que se presenta con las CNC

### **5.3 Instrumentos de recolección de la información.**

**5.3.1 Fuentes primarias.** Observación directa, entrevistas.

**5.3.2 Fuentes secundarias.** Libros técnicos, catálogos e internet

## 6. Resultados Del Proyecto

### 6.1 Construcción de la estructura de la CNC

#### 6.1.1. Lista de componentes mecánicos:

*Tabla 1.*  
Componentes mecánicos

Elementos	Unidades
Placa de aluminio	1
Acoplamientos	3
Cara de extremo de cojinete fijo	3
Córner	1
Tuerca de cobre	5
Soporte	5
Cilindro de cobre	12
Tabla de aluminio(23*12cm)	1
Perfiles de longitud larga	4
Perfiles de longitud corta	3
Tornillos de rodamiento 8mm	3
Varilla de guía	6
Motor	1
Motor stepper	3

Fuente: Diseño propio



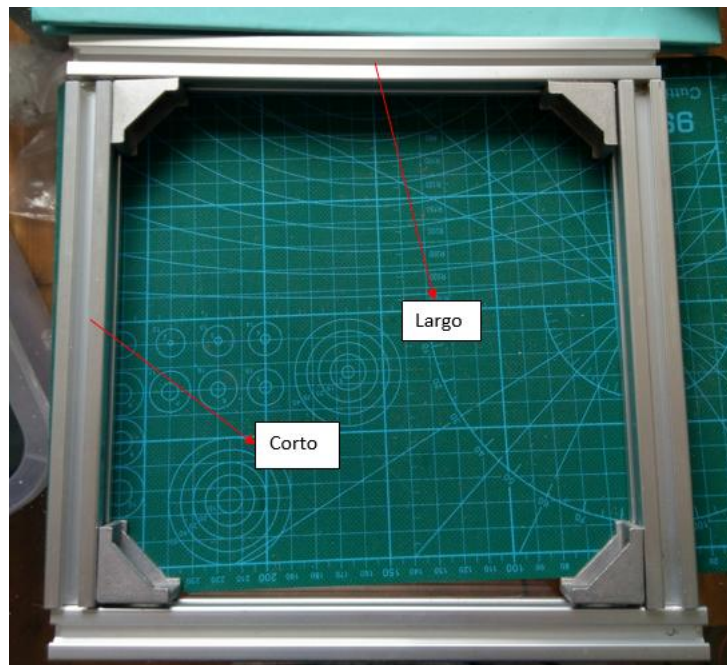
*Figura 11* Listado de piezas  
Fuente: Foto tomada por el autor

### 6.1.2 Paso 1.

Construcción de la estructura.

Base del eje x:

Primero comenzamos, con la base de la maquina (eje x). Para esto necesitamos, 4 largueros de aluminio, 2 de 250mm y 2 de 210mm. Se instalan según la siguiente figura.



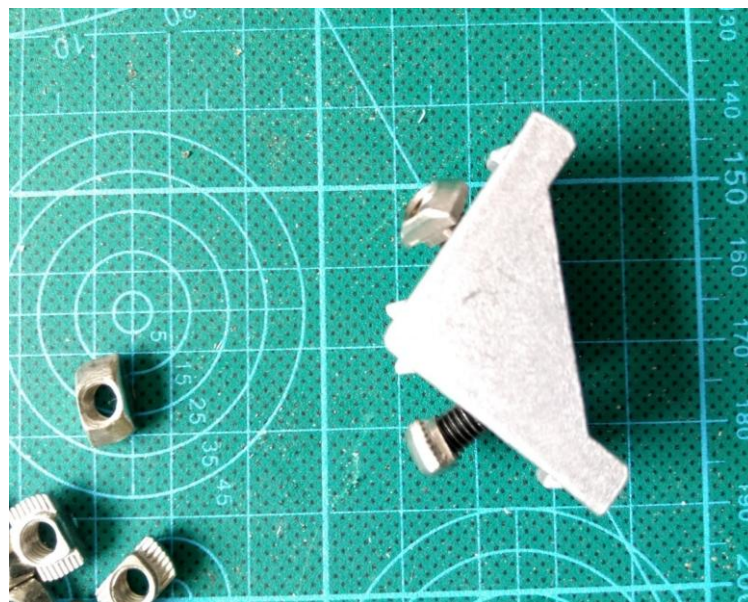
*Figura 12* Base eje x  
Fuente: Foto tomada por el autor

Se construye de esta manera, para que sea un estructura cuadrada.

Para acoplar los 4 largeros, se necesita tuercas, de tornillos M5, que se muestran en la siguiente figura.



*Figura 13* Tornillo de ensambles  
Fuente: Foto tomada por el autor



*Figura 14* Acoples angulares  
Fuente: Foto tomada por el autor

Esta pieza es un barco con tuerca de tornillo, con la forma de realización de la figura previamente señalada. Es fácil de instalar, pero en la instalación debe observarse que en el

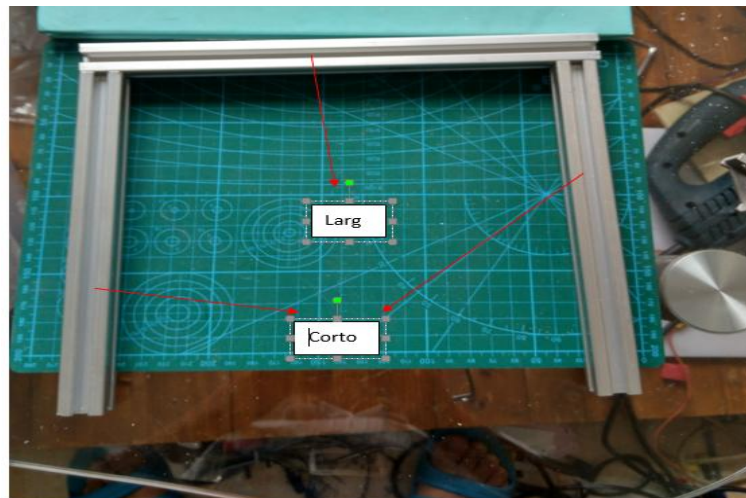
barco, debe hacer que la tuerca se gire, o no se puede fijar, en la ranura, de la aleación de aluminio.

### 6.1.3 Paso 2.

Construcción de la estructura del eje y:

Para este paso, se necesita 3 largueros, 1 larguero de 250mm y 2 de 210mm.

El cual va armado de la siguiente manera, según la figura:



*Figura 15* Estructura eje Y  
Fuente: Foto tomada por el autor

Solo se utiliza, estos tres largueros en vez de 4, como la figura anterior. solo es asi, porque éste, va encima de la la estructura de la base del eje x, como lo veremos mas adelante en los siguientes pasos

### 6.1.4 Paso 3.

Uniremos las dos estructuras metálicas.

Para la unión de las estructuras metálicas, se necesitan, los mismos tornillos M5, y los barcos de aluminio, que se utilizaron para unir las estructuras individuales al acoplarlas.

Se monta la estructura del paso 2, que tiene tres largueros de aluminio encima de la estructura cuadrada, posteriormente se monta en una esquina de ésta.



Figura 16 Eje X y Y con su respectivos acoples angulares  
Fuente: Foto tomada por el autor

Se hace el montaje, para que quede las dos estructuras unidas, una sobre la otra.

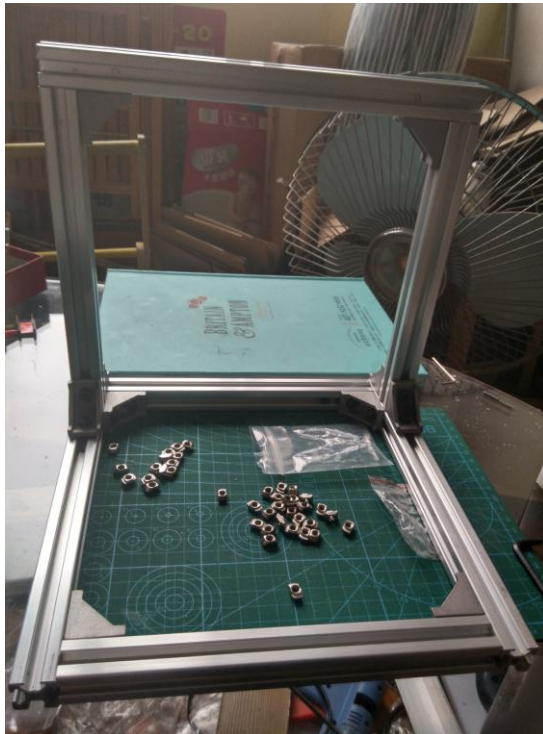


Figura 17 Eje X y Y acopladas  
Fuente: Foto tomada por el autor

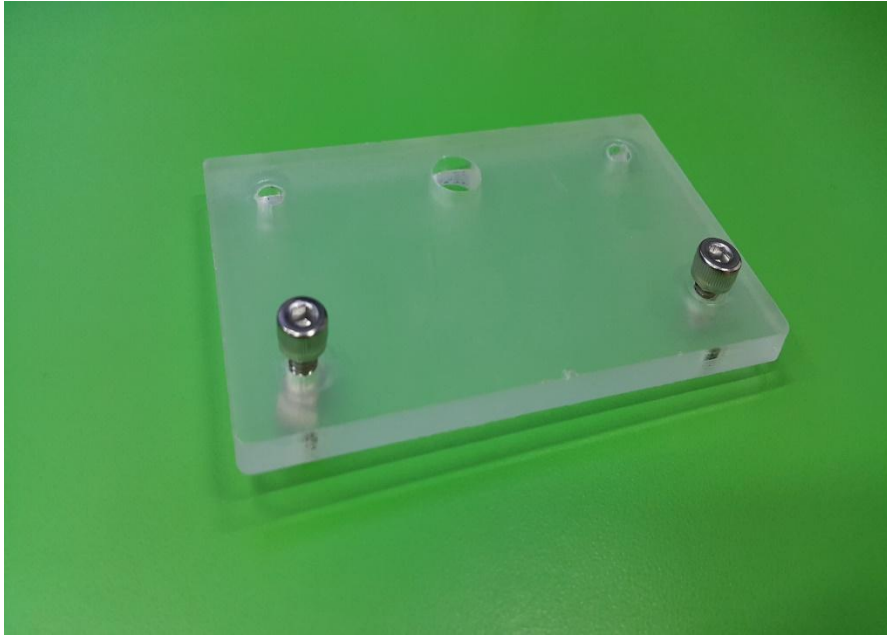


#### 6.1.5 Paso 4.

Montaje de la estructura del eje z.

Vea a continuación la placa de montaje de soporte, en esta placa de aluminio

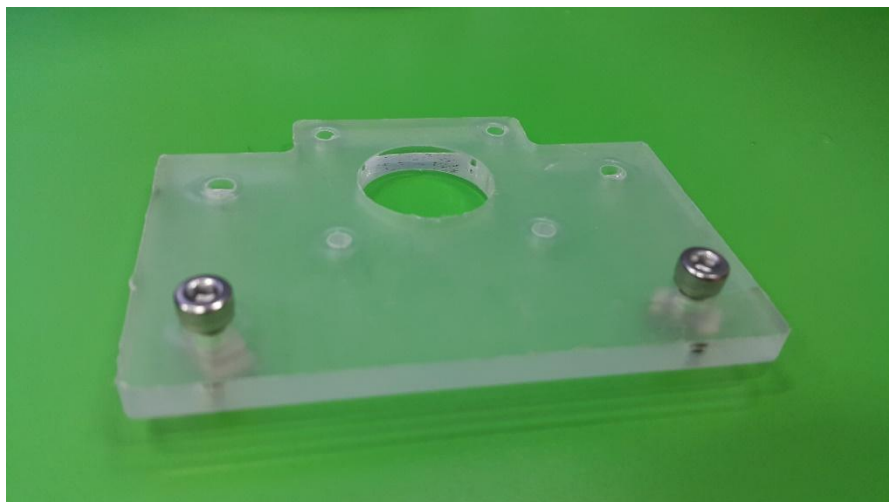
Se utiliza un tornillo M4 \* 10 de espesor.



*Figura 18* Pieza lateral derecha para el eje Y

Fuente: Foto tomada por el autor

Se utiliza dos placas del mismo tamaño y dimensiones. Con los mismo respectivos tornillos de acoplamiento



*Figura 19* Pieza lateral izquierda para el eje Y

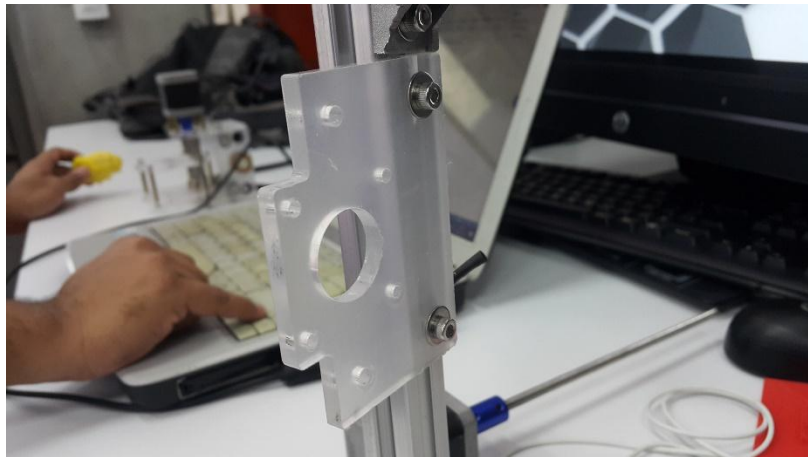
Fuente: Foto tomada por el autor



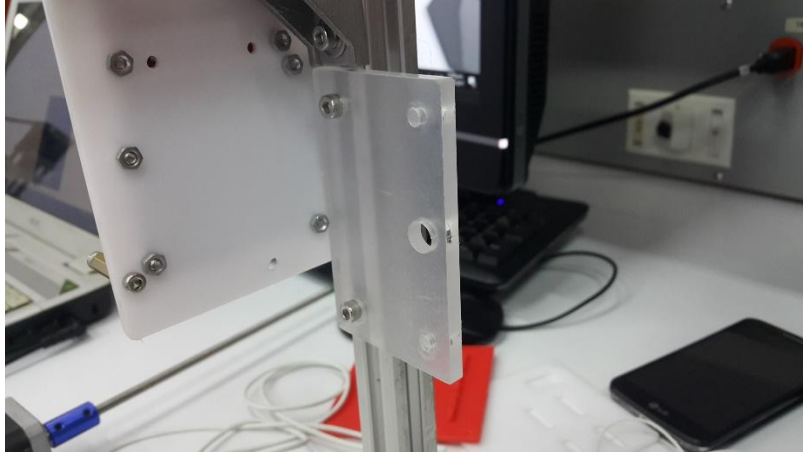
*Figura 20* Tornillos de ensambles con su respectiva tuerca  
Fuente: Foto tomada por el autor

#### 6.1.6 Paso 5.

Se ajusta las dos placas, a la parte superior del eje Y, y al tener listas las dos placas con su respectivos tornillos de acoplamiento, se procede a insertarlas o ajustarlas, al eje y de la estructura metálica



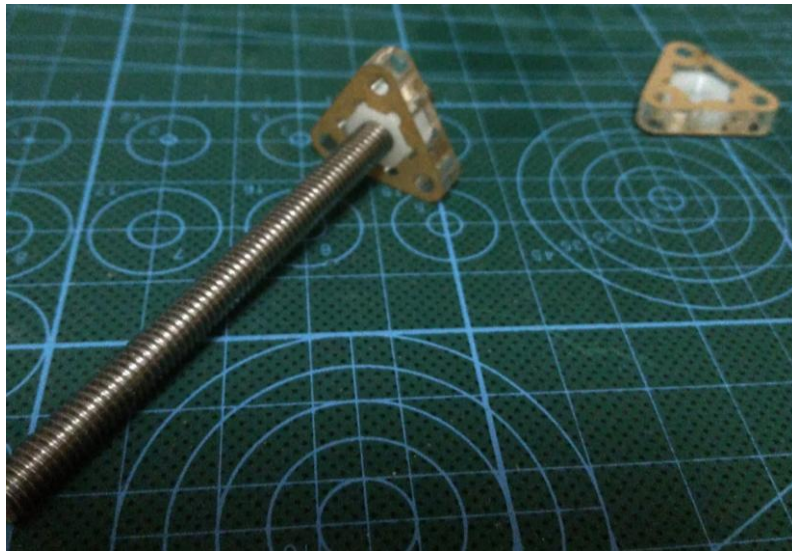
*Figura 21* Pieza lateral derecha ajustado al eje y  
Fuente: Foto tomada por el autor



*Figura 22* Pieza lateral izquierda ajustada al eje y  
Fuente: Foto tomada por el autor

#### 6.1.7 Paso 6.

Se arma los ejes de los de cada estructura, de las dimensiones X, Y, Z, los cuales van acoplados, a los motores, y estos son los que permiten, el movimiento tridimensional de la máquina, para su óptimo funcionamiento



*Figura 23* Eje de rodamiento 8mm eje x  
Fuente: Foto tomada por el autor



*Figura 24* Eje de rodamiento 8mm eje y  
Fuente: Foto tomada por el autor

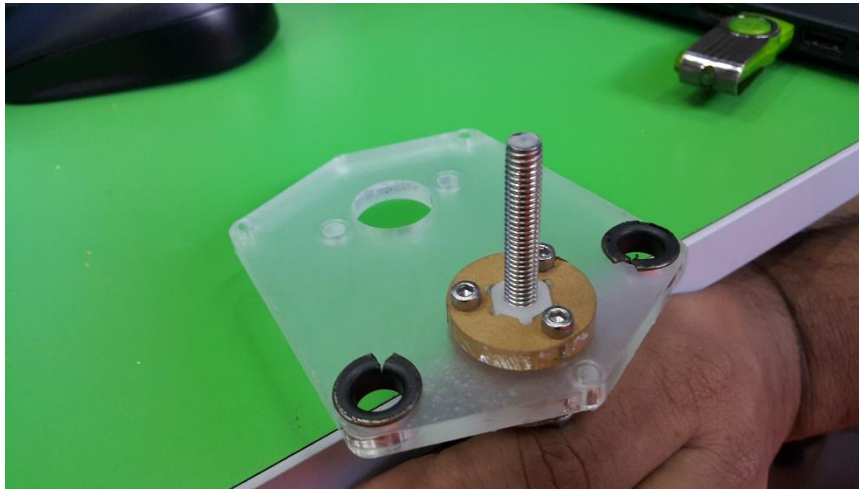
Son soportes de tuerca de dos delantales triangulares, a rotación lenta, el cual empuja el tornillo de mano, sintiendo ningún hueco, mientras gira el tornillo, y preferiblemente no tiene gran sensación de resistencia.



*Figura 25* Eje de rodamiento 8mm eje z  
Fuente: Foto tomada por el autor

### 6.1.8 Paso 7.

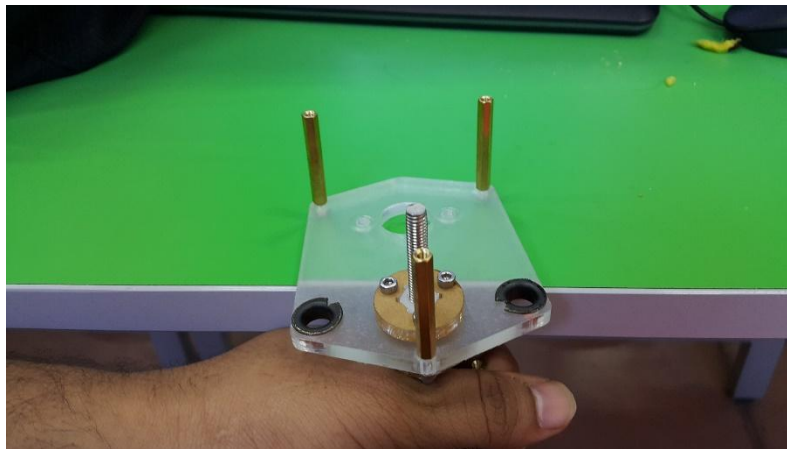
Montaje del eje z, en la estructura que da soporte, para el motor de fresado



*Figura 26* Eje de rodamiento ensamblado

Fuente: Foto tomada por el autor

Luego de montar el eje, tal cual como aparece en la figura anterior, colocamos pilares o arandelas largas para tornillos M3



*Figura 27* Estructura para el montaje de motor de fresado

Fuente: Foto tomada por el autor

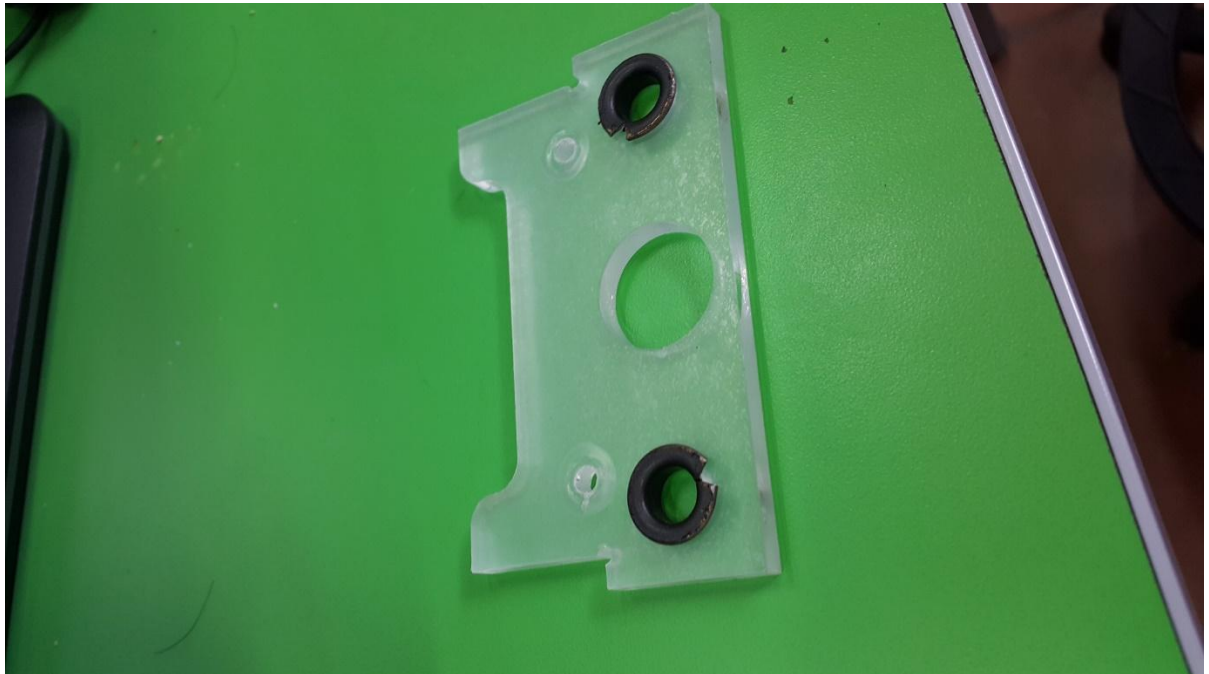
Se termina de armar todo el caparazón o la estructura, que va a llevar o a soportar el motor de fresado, y que va a permitir el movimiento del eje z



*Figura 28* Ensamble completo de la estructura del motor de fresado  
Fuente: Foto tomada por el autor

#### 6.1.9 Paso 8.

Se arma toda la estructura para el eje Y, que soporta la estructura del motor de fresado, para que se pueda mover por horizontalmente.



*Figura 29* Piezas laterales para la estructura del eje y  
Fuente: Foto tomada por el autor

Estas dos piezas laterales, son las paredes de la estructura, del movimiento horizontal del eje Y

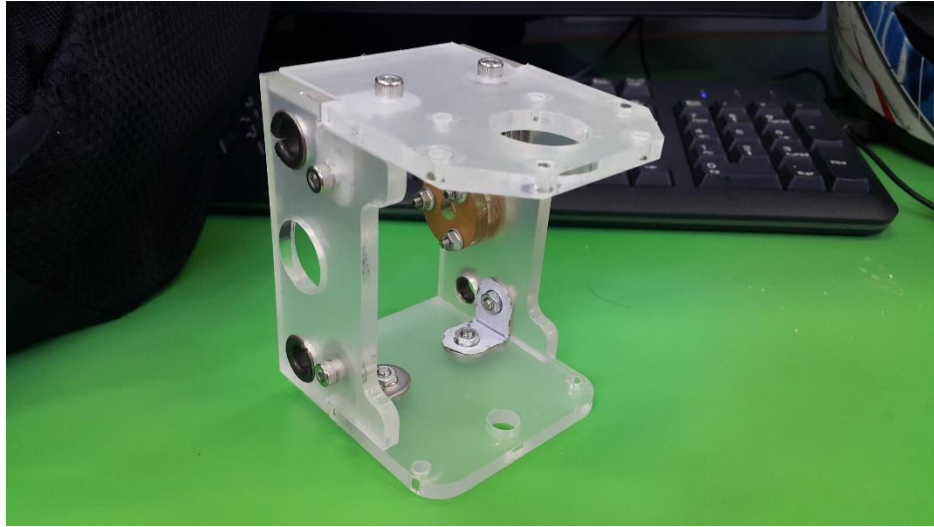


Figura 30 Estructura terminada  
Fuente: Foto tomada por el autor

#### 6.1.10 Paso 9.

Luego de tener las estructuras del motor de fresado, y la que va al eje, se procede a colocar o montar, el primer motor mena, que es el responsable del movimiento de la estructura.



Figura 31 Motor stepper atornillado  
Fuente: Foto tomada por el autor

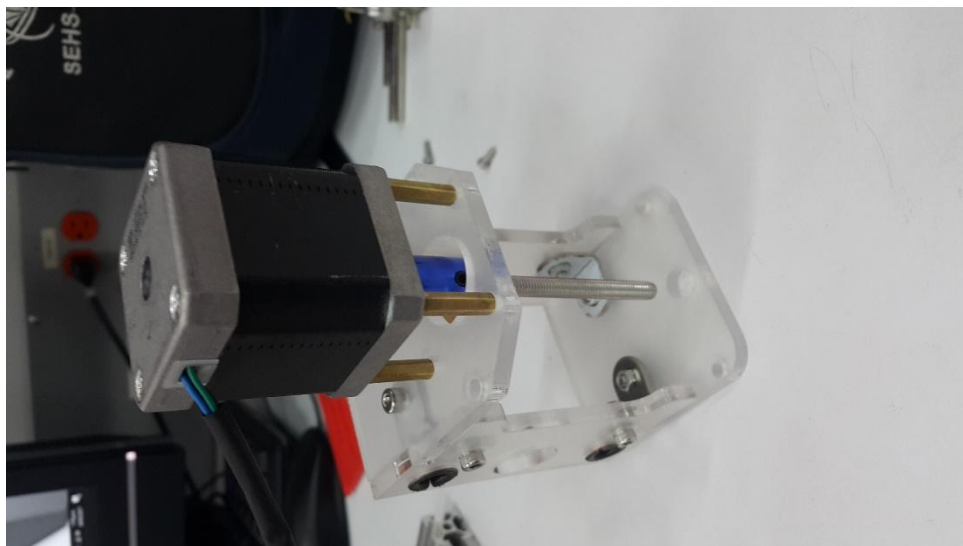
Luego de atornillar, los cuatro tornillos sin cabeza, en los agujeros del motor, se procese a colocarles las tuercas largueras, en cada tornillo sin cabeza,



*Figura 32* Motor stepper con arandelas largas

Fuente: Foto tomada por el autor

Al tener el motor con sus soportes listos, se procede a montarlo o acoplarlo en la estructura.



*Figura 33* Motor stepper ensamblado en la estructura del eje z

Fuente: Foto tomada por el autor



#### 6.1.11 Paso 10.

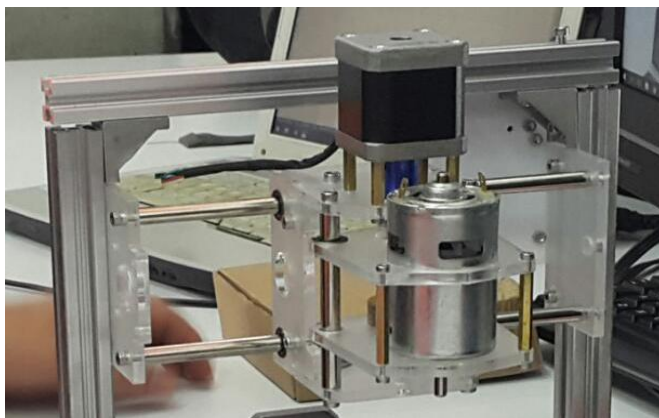
Ensamblamos las estructuras del motor de fresado, con la estructura del eje Y.

En este paso vamos a ensamblar, respecto a la figura, estas dos piezas o estructuras metálicas, son una parte importante para la CNC



*Figura 34* Ensamble de la estructura del eje z a la del eje y  
Fuente: Foto tomada por el autor

Luego de tenerlas ensambladas, procedemos a colocarla en la estructura principal, respecto al eje Y de la máquina.



*Figura 35* Ensamble de las dos estructuras a la base metálica  
Fuente: Foto tomada por el autor

#### 6.1.12 Paso 11.

Construiremos y ensamblaremos la cama de fresado.

En este momento colocaremos, las dos piezas de aluminio que quedan, a cada lado de la estructura, de la parte del eje X, en el ancho de la estructura como tal, exactamente en la parte central de cada larguero.



*Figura 36* Piezas de laterales para el montaje de la cama de fresado

Fuente: Foto tomada por el autor

Al colocarla, procedemos a colocar el ultimo eje, en la cama de fresado, para el movimiento de dicha cama.



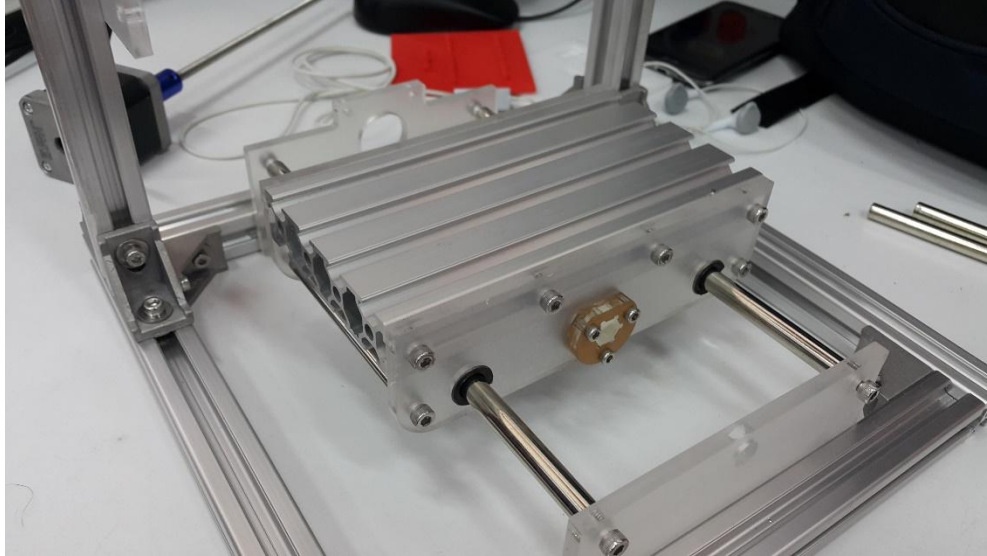
*Figura 37 Ensamble del eje de rodamiento a la cama de fresado*  
Fuente: Foto tomada por el autor



*Figura 38 Otro Angulo para la vista de la cama de fresado con el eje*  
Fuente: Foto tomada por el autor

#### 6.1.13 Paso 12.

Al tener la cama de fresado lista, se procede a montarla en la estructura, para terminar con los movimientos de los 3 ejes X, Y, Z



*Figura 39* Total montaje de la cama de fresado con el eje de rodamiento  
Fuente: Foto tomada por el autor

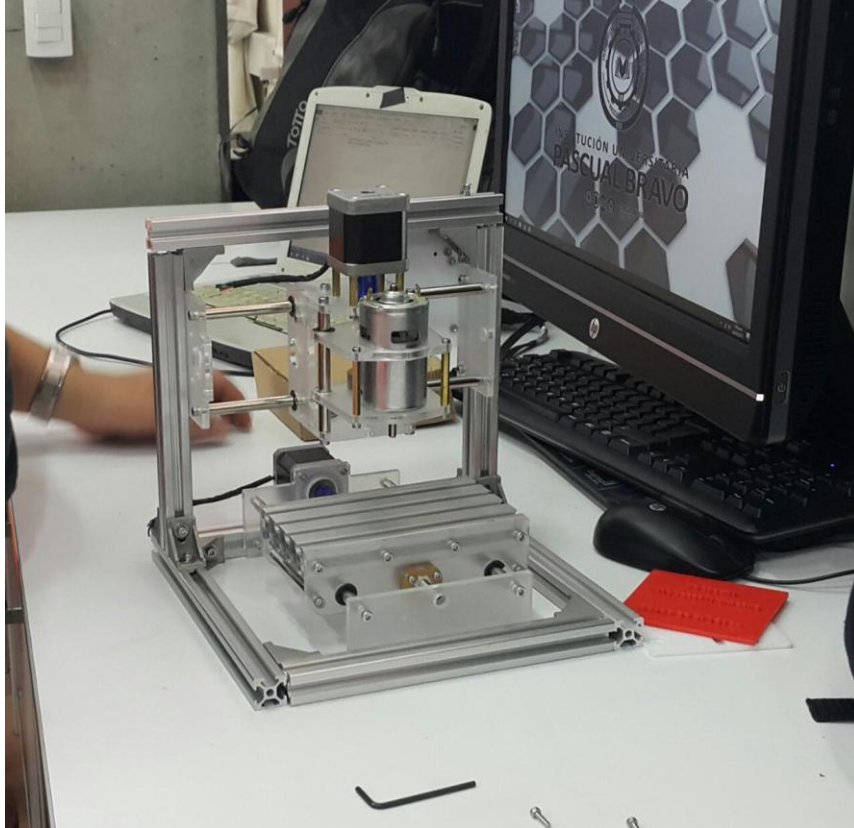
En el momento del montaje, de la cama de fresado, podemos empezar a acoplar los 2 motores mena que faltan



*Figura 40* Ensamble completo de la estructura CNC  
Fuente: Foto tomada por el autor

6.1.14 pasó 13.

Ensamblamos y acoplamos los 2 motores del eje X y Y.



*Figura 41* Ensamble completo de la estructura los motores stepper  
Fuente: Foto tomada por el autor

#### 6.1.14 Configuración de los micropasos.

Cada eje, puede ser configurado mediante diferentes micropasos.

Es recomendable, que todos los ejes, tengan la misma configuración. Los jumper deben de ser situados en las siguientes posiciones:

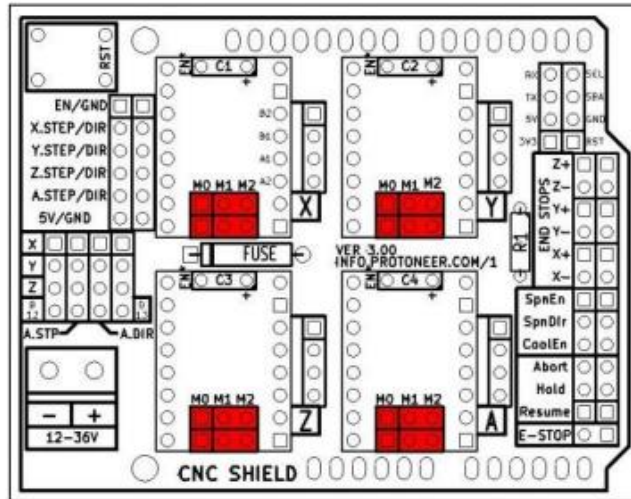


Figura 42 Configuración micropasos shield GRBL

Fuente: extraído de <https://electronilab.co/tienda/cnc-grbl-shield-v3-driver-a4988-impresora-reprap-3d/>

La siguiente tabla, muestra la configuración para el driver Pololu A4988:

Tabla 2. Configuración dirver Pololu A988

MS0	MS1	MS2	RESOLUC IÓN
BAJO	BAJO	BAJO	1
ALTO	BAJO	BAJO	1/2
BAJO	ALTO	BAJO	1/4
ALTO	ALTO	BAJO	1/8
ALTO	ALTO	ALTO	1/16

Fuente: Diseño propio

La siguiente tabla muestra la configuración para el driver Pololu DRV8825:

Tabla 3. Configuración driver Pololu DRV8825

MS0	MS1	MS2	RESOLUCION
BAJO	BAJO	BAJO	1
ALTO	BAJO	BAJO	1/2
BAJO	ALTO	BAJO	1/4
ALTO	ALTO	BAJO	1/8
BAJO	BAJO	ALTO	1/16
ALTO	BAJO	ALTO	1/32
BAJO	ALTO	ALTO	1/32
ALTO	ALTO	ALTO	1/32

#### 6.1.15 Configuración del cuarto eje.

Usando dos jumper se puede clonar el eje X, Y o Z. Realmente también se puede usar como un eje independiente usando el pin digital 12 como STEP y el pin digital 13 como DIR, pero GRBL no soporta más que 3 ejes.

Cuarto eje como eje X

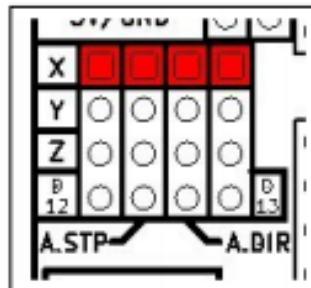


Figura 43 Cuarto eje como eje X

Fuente: extraído de <https://electronilab.co/tienda/cnc-grbl-shield-v3-driver-a4988-impresora-reprap-3d/>

Cuarto eje como eje Y

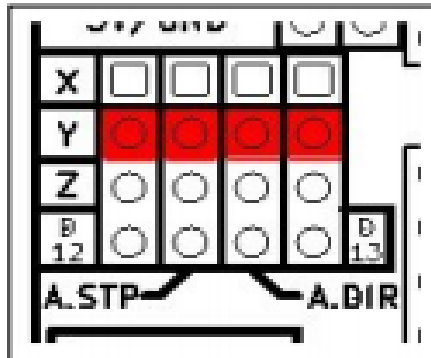


Figura 44 Cuarto eje como eje Y

Fuente: extraído de <https://electronilab.co/tienda/cnc-grbl-shield-v3-driver-a4988-impresora-reprap-3d/>

Cuarto eje como eje Z

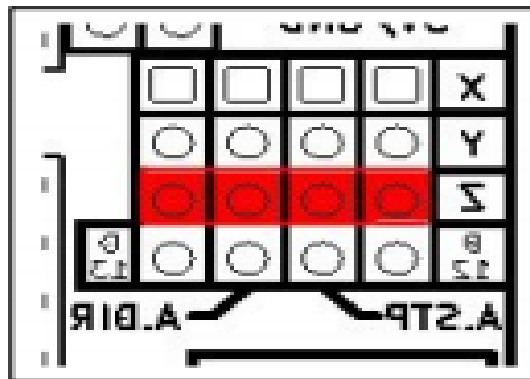
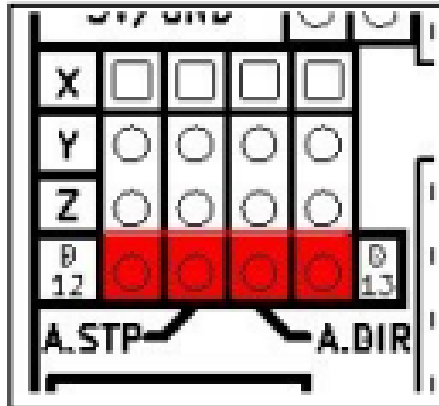


Figura 45 Cuarto eje como eje Z

Fuente: extraído de <https://electronilab.co/tienda/cnc-grbl-shield-v3-driver-a4988-impresora-reprap-3d/>



Cuarto eje controlado por los pines 12 y 13



*Figura 46* Cuarto eje controlado por los pines 12 y 13

Fuente: extraído de <https://electronilab.co/tienda/cnc-grbl-shield-v3-driver-a4988-impresora-reprap-3d/>

## 6.2 Equipamiento del sistema del control de potencia

### 6.2.1 GRBL Shield.

Con GRBL cargado, ya se está en condición de implementar a nivel de hardware el conexionado de los drivers con el controlador. Tras probar las diferentes posibilidades, la solución óptima ha resultado en adquirir un shield específicamente diseñado para GRBL con el fin de lograr un dispositivo compacto y facilitar el conexionado.

Se trata de un diseño hardware libre elaborado por Bertus Kruger de Protoneer. Existen diferentes versiones, la usada en este proyecto es la 3.00, que tiene este aspecto:

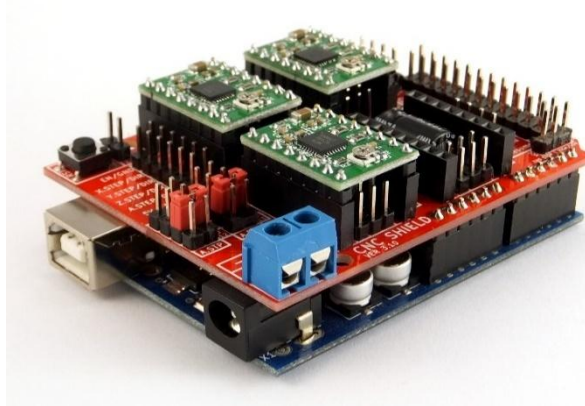


Figura 47 Shield GRBL

Fuente: extraído de <https://github.com/grbl/grbl/wiki/Connecting-Grbl>

GRBL solamente es compatible con controles numéricos de 3 ejes, por lo que el cuarto driver se puede emplear para duplicar la fuerza motriz en un eje. En este proyecto no se ha considerado necesario, por lo que solo se hace uso de 3 drivers.

En el siguiente esquemático se pueden observar todos los pines.

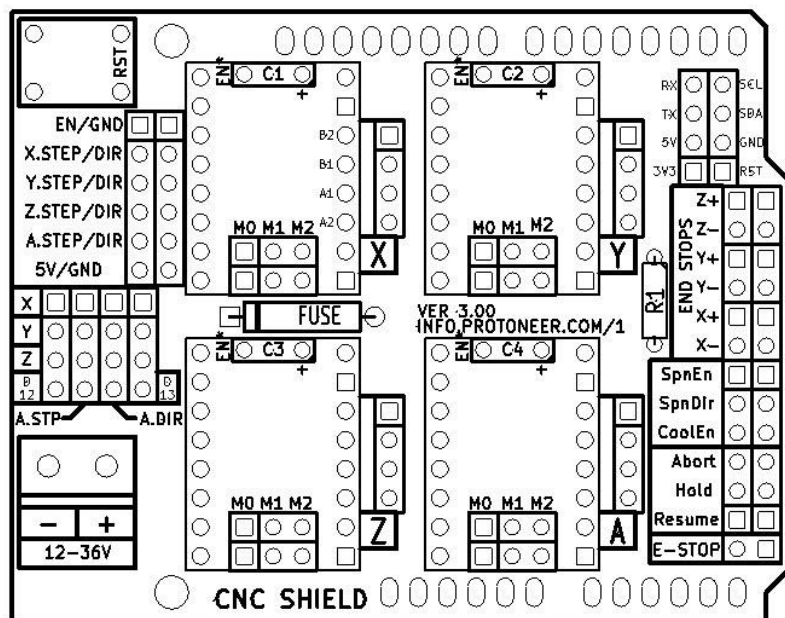


Figura 48 Esquemático shield GRBL

Fuente: extraído de <https://electronilab.co/tienda/cnc-grbl-shield-v3-driver-a4988-impresora-reprap-3d/>

Teóricamente la placa está diseñada para trabajar con hasta 36V, pero hay que llevar cuidado ya que si los drivers no están preparados para soportar esta alimentación serán destruidos cuando se alimente. En este trabajo se han empleado los POLOLU A4988, fijando la alimentación en 12V.

La forma de configurar algunos parámetros de la placa es mediante el uso de jumpers y no mediante programación. En concreto, los jumpers son empleados para configurar el uso del cuarto eje (duplicación de un eje ya existente para aumentar la potencia motriz), micropasos de los drivers y los finales de carrera.

### 6.2.2 Drivers.

Ubicar los drivers es una tarea realmente sencilla, simplemente hay que conectarlos con los pines adecuadamente. El driver presenta un potenciómetro que permite regular la intensidad que suministra a cada bobina. La forma hacer esto no resulta sencilla y por eso se ha considerado necesario aclarar cómo hacerlo en este documento. En primer lugar hay que observar un terminal relativamente escondido:

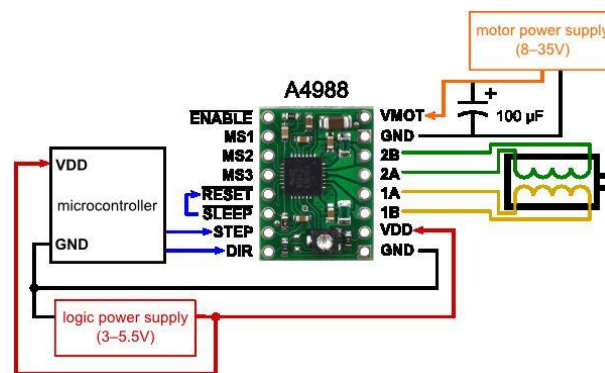


Figura 49 Detalle pin Vref A4988

Fuente: extraído de <http://www.instructables.com/id/Drive-a-Stepper-Motor-with-an-Arduino-and-a-A4988-/>

Se trata del pin Vref. Para ajustar la intensidad, hay que medir la tensión que hay entre este pin y masa, y aplicar la siguiente fórmula:

$$\text{corriente} = V_{\text{ref}} * 25$$

### *Ecuación 1 corriente de driver*

En este proyecto, y como se verá más adelante, se ha llevado el driver al máximo, extrayendo de él 2A, por lo que en el terminal Vref se debe de medir 0.8V. Obviamente, en estas circunstancias, ha sido fundamental dotar de disipador al driver. Además, y debido a la cercanía de los tres drivers, se ha dotado de un sistema de refrigeración activa mediante ventilador. Más adelante se detallará el mecanismo de extracción de calor.

### 6.2.3 Motores.

El conexionado de los motores Nema 23 también resulta sencillo, pero hay que tener en cuenta un aspecto que puede llevar a confusión, ya que el autor del shield de GRBL no utiliza la misma nomenclatura en los nombres de los cables que en los drivers. Es decir, en vez de usar B2, B1, A1, A2, hay que usar 2B, 2A, 1A, 1B, donde:

*Tabla 4*  
*Motores Nema 23*

2B	Rojo
2 <sup>a</sup>	Azul
1 <sup>a</sup>	Verde
1B	Negro

Aunque en este momento presentar este apartado pueda resultar precipitado (más adelante se hablará sobre los parámetros constructivos de la máquina), es necesario para llegar a entender la precisión que dota el conjunto de motor paso a paso y tornillo roscado. El desplazamiento angular del motor se traduce en un desplazamiento lineal de la herramienta mediante el husillo. Esto proporciona una gran precisión así como una gran potencia en el desplazamiento. Como principal inconveniente, la velocidad lineal se ve enormemente atenuada, la cual depende obviamente de la velocidad de giro del motor, pero también del paso de la rosca. Los tornillos empleados son de paso 1mm. No se ha empleado un paso

mayor con el fin de no comprometer la precisión de la máquina. Esto significa que por cada vuelta completa que efectúe el motor, el eje se desplazará 1mm.

#### 6.2.4 Accionamiento de la fresadora.

Una solución posible a la hora de realizar un trabajo es presionar el botón de encendido de la fresadora y apagarla manualmente cuando el proceso termina, pero la idea es automatizar el sistema lo máximo posible. Mediante las funciones de control numérico M3 y M5 se puede actuar sobre un pin digital de Arduino para encender y apagar la herramienta. Esto, además de aumentar la comodidad, permite ahorrar energía en movimientos rápidos sobre la pieza, apagando la máquina, así como otorgando robustez en los ciclos de mecanizado. Desgraciadamente, no se puede controlar automáticamente la velocidad de giro de la herramienta, ya que dispone de un potenciómetro que se debe de accionar físicamente. Por suerte, la mayoría de trabajos se pueden realizar a la máxima velocidad de giro, por lo que no será necesario de modificar constantemente, salvo cuando se busque la máxima precisión y rendimiento. Al usar como herramienta la fresadora, existe un problema de incompatibilidad de potencia en la señal de control. Arduino proporciona 5V, mientras que la fresadora funciona a 230. En este instante están coincidiendo la etapa de control y la de potencia. Para efectuar este nexo, la solución inmediata es el uso de un relé. El relé es un dispositivo electromecánico. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes. Dado que el relé es capaz de controlar un circuito de salida de mayor potencia que el de entrada, puede considerarse, en un amplio sentido, como un amplificador eléctrico. La gran ventaja de los relés electromagnéticos es la completa separación eléctrica entre la corriente de accionamiento, la que circula por la bobina del electroimán, y los circuitos controlados por los contactos, lo que hace que se puedan manejar altos voltajes o elevadas potencias con pequeñas tensiones de control. También ofrecen la posibilidad de control de un dispositivo a distancia mediante el uso de pequeñas señales de control. Para este proyecto, se ha optado por emplear un módulo de relé comercial que incluye toda la etapa de acondicionamiento, que principalmente se trata de un diodo que permite la libre circulación de corriente en el bobinado del relé cuando la

señal de control desaparece (de este modo se evitan posibles y peligrosas fugas de corriente hacia el controlador), así como un transistor para aumentar la ganancia de la intensidad.



*Figura 50* Relé fresadora

Fuente: extraído de <http://www.prometec.net/reles/>

Sus principales características son:

*Tabla 5*  
Características del relé

Señal de control	3.5-12V
Salida	N.C. y N.A.
Señal de potencia	VDC (30V) y VAC (250V)
Dimensiones	40x27x18mm

Por seguridad, cabe destacar un importante fallo de diseño que tiene este relé. Al otro lado de la PCB, están las soldaduras. Las más grandes son las correspondientes a la regleta de tres terminales para conectar la alimentación de la herramienta. No es difícil, si se está manipulando el relé mientras está conectado, poner las manos sobre estas soldaduras que están a 230V. Es necesario situar el relé sobre una base segura que no permita en ningún momento acceder con las manos a la parte trasera de la PCB.

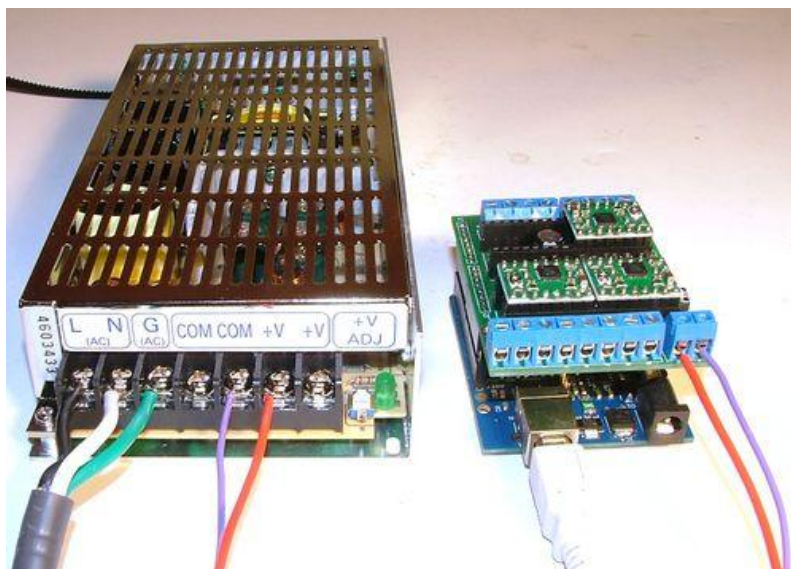
### 6.2.5 Alimentación.

Para alimentar todo el conjunto se ha optado por emplear una fuente de alimentación de ordenador de tipo ATX. Esta solución es perfecta ya que a un precio muy razonable, se disponen de multitud de salidas a diferente tensión y además se dispone de ventilador para contribuir a la refrigeración de los drivers. En concreto se ha empleado el modelo TMXPT0500 de TECNIMAX, que ofrece hasta una potencia de 500W, y además ofrece protección frente a sobrecarga y cortocircuito, perfecto para las primeras fases de la ejecución del proyecto. Esta se encarga de alimentar el shield de GRBL (los drivers), el ventilador y el relé. La alimentación de la fresadora se hace directamente a 110V.

Sus principales características son:

*Tabla 6*  
Fuente de poder

Tensión de entrada	110VAC
Frecuencia de entrada	60Hz
Eficiencia	>70% a plena carga
Temperatura de funcionamiento	0-40°C
Protecciones	OVP (Sobre voltaje) OCP (Sobretensión) SCP (Cortocircuito)



*Figura 51* Conexiones fuente de alimentación

Fuente: extraído de <https://www.shapeoko.com/wiki/index.php/Electronics> (SO1)

Esta imagen contiene mucha información. Además de observar las conexiones desde la fuente de alimentación, se puede apreciar el Arduino UNO soportando el shield GRBL con tres drivers, el relé, y el ventilador. A falta de los finales de carrera, y de la seta de emergencia, contiene toda la parte de control del CNC. Aunque dos drivers son de color blanco y otro de color negro, ambos corresponden a la versión BLACK de los A4988. La única diferencia, es que se adquirieron desde otro proveedor y parece que no se pusieron de acuerdo con el color de la máscara de soldadura. Obsérvese el detalle del interruptor de la fuente de alimentación. Permite apagar fácilmente el CNC sin necesidad de desconectar ningún cable.

El shield sobre Arduino no es capaz de alimentar al microcontrolador, por lo que este aunque sea alimentado, Arduino requiere también de alimentación externa. La solución es más que sencilla ya que se alimenta directamente desde el puerto USB.

Por ello, para que la máquina funcione, debe de estar conectada permanentemente al ordenador.

#### 6.2.6 Disipación del calor.

Motores y drivers son sinónimo de calor. En este proyecto, donde todo el sistema electrónico va a estar embebido en una caja cerrada y donde se van a efectuar largos ciclos de



mecanizado, es crucial planificar e idear un método para la correcta refrigeración del sistema. Ya se han presentado los dos ventiladores que van a contribuir a la ventilación forzada. Aunque no es objetivo de este apartado presentar la estructura mecánica, es necesario adelantar una imagen para entender mejor el sentido de caudal de aire que se pretende conseguir con la disposición de los ventiladores.

Ventilación La entrada de aire la produce el ventilador situado sobre los drivers. En concreto, se trata de un ventilador de 70mm con un consumo de intensidad de 0.15 amperios con una tensión de 12 voltios. Este ventilador proporciona un caudal muy similar al que dispone la fuente de alimentación, uno de 80mm, que se encarga de expulsar el aire por el lazo izquierdo. Sobre los drivers, tal y como recomienda el fabricante, se han situado disipadores de aluminio de 13x13x7 milímetros con un adhesivo termoconductor.



*Figura 52* Detalle disipador driver

Fuente: extraído de <http://www.ledcontrols.com.mx/ver.php?modelo=747>

Los resultados de este conjunto han sido excepcionales, tras periodos de mecanizado de 120 minutos, no se ha apreciado en ningún momento un calentamiento notable.

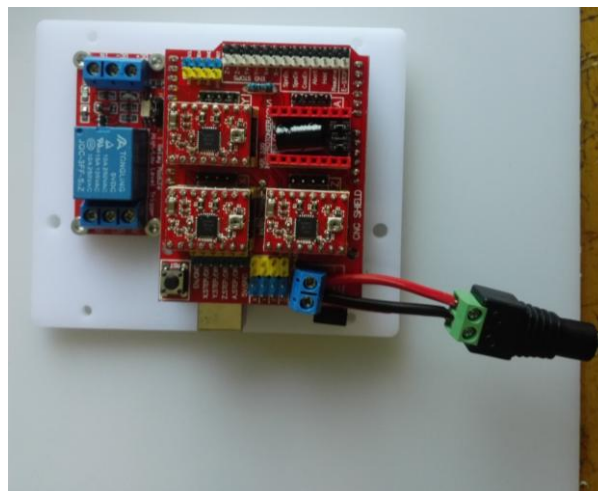
#### 6.2.7 Interferencias electromagnéticas.

Algunos usuarios de GRBL han tenido dificultades con este al quedar constantemente congelado mientras se estaba enviando el código G. Esto no es un problema del software,

Si no más bien del entorno. Este problema se produce por las interferencias electromagnéticas que producen los dispositivos de alta potencia, como por ejemplo, los cables de la fresadora. Para evitar esto, todo el cableado va a ser ordenado, empleando un cable USB blindado y dispuesto lo más lejos posible de todas las fuentes de ruido electromagnético.

#### 6.2.8 pequeña guía de instalación del driver de potencias.

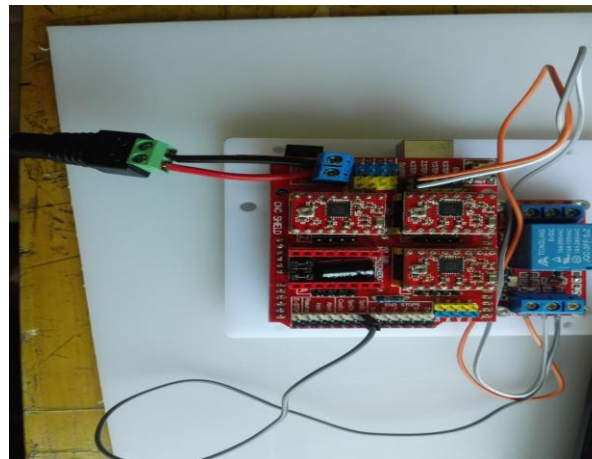
Se conecta uno de las terminales de la fuente al driver de potencia



*Figura 53* Driver con cable de alimentación

Fuente: Foto tomada por el autor

Luego se instala el relé en el driver



*Figura 54* Conexión driver – relé

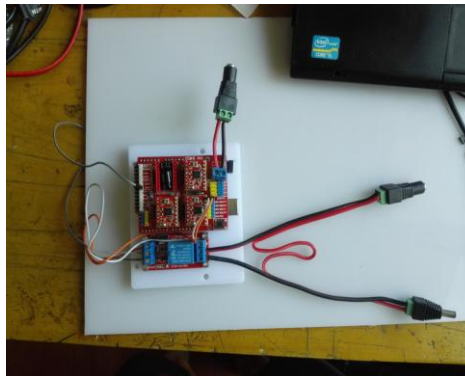
Fuente: Foto tomada por el autor

Las dos terminales restantes de la fuente se conectan entre si para darle paso a la alimentación del motor al ser accionado el relé



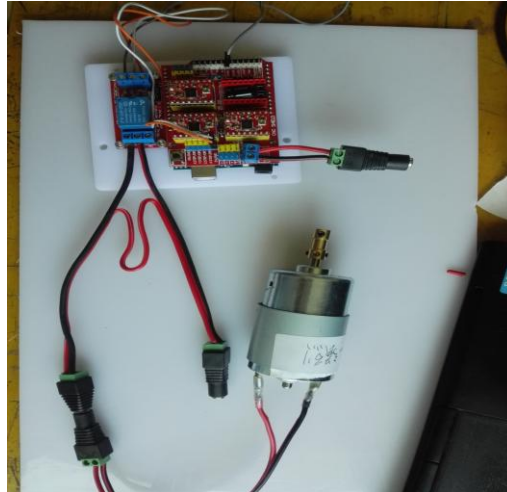
*Figura 55* Empalme de cables de terminales de la fuente  
Fuente: Foto tomada por el autor

Y se conectan de la siguiente manera positivo con positivo y negativo a un terminal del relé sea el NA y el otro negativo al NO o viceversa



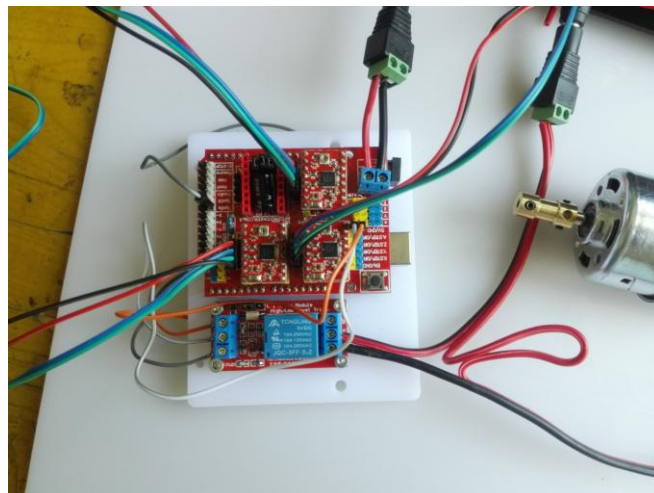
*Figura 56* Cables de la fuente conectados al relé  
Fuente: Foto tomada por el autor

Luego de esto se conectan al motor de fresad para darle la alimentación suficiente  
Para su óptimo funcionamiento



*Figura 57* Motor de fresado conectado a la fuente  
Fuente: Foto tomada por el autor

Luego cada eje es alimentado según el diagrama anteriormente mencionado



*Figura 58* Conectando los motores stepper a los driver de potencia  
Fuente: Foto tomada por el autor

### 6.3 sistema de comunicaciones

Para la implementación de sistema de comunicaciones en este proyecto utilizaremos un control de sistemas muy común y fácil de manejo además es de software libre como es el Arduino.

### 6.3.1 Programación de Arduino.

El sketch introducido en la placa Arduino es la versión 0.8c de GRBL, ya que su funcionamiento está totalmente garantizado. Se puede acceder a la configuración de GRBL en el fichero Config.h desde el IDE de Arduino. La empleada en este proyecto ha sido:

```
//cartesian bot pins
#define X_STEP_PIN 4
#define X_DIR_PIN 5
#define X_ENABLE_PIN 10
#define X_MIN_PIN 0
#define X_MAX_PIN 0

#define Y_STEP_PIN 2
#define Y_DIR_PIN 3
#define Y_ENABLE_PIN 9
#define Y_MIN_PIN 0
#define Y_MAX_PIN 0

#define Z_STEP_PIN 7
#define Z_DIR_PIN 6
#define Z_ENABLE_PIN 11
#define Z_MIN_PIN 0
#define Z_MAX_PIN 0

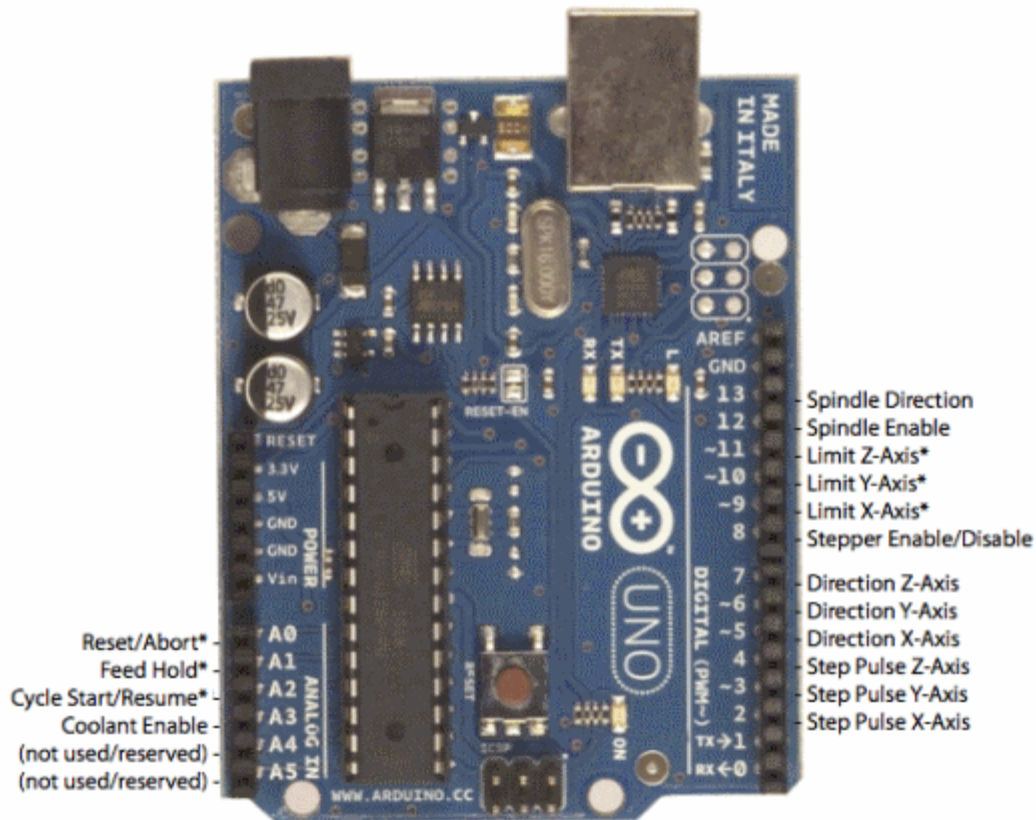
//extruder pins
// NOTE - USING Timer1 FOR STEPPER TIMER SO CAN'T USER PINS 9 OR 10 FOR PWM
// OUTPUT (EXTRUDER_MOTOR_SPEED_PIN, EXTRUDER_HEATER_PIN, OR EXTRUDER_FAN_PIN)
#define EXTRUDER_MOTOR_SPEED_PIN 0
#define EXTRUDER_MOTOR_DIR_PIN 0
#define EXTRUDER_HEATER_PIN 0
#define EXTRUDER_FAN_PIN 0
#define EXTRUDER_THERMISTOR_PIN 0 //a -1 disables thermistor readings
#define EXTRUDER_THERMOCOUPLE_PIN 0 //a -1 disables thermocouple readings

<
```

*Figura 59* Código de control CNC

Fuente: extraído de <https://www.ikkaro.com/como-hacer-fresadora-cnc-casera-parte-4/>

Esta configuración no es trivial y más adelante se detallará el principal motivo por el que se ha elegido: adaptar la tarjeta de Arduino al shield específico de GRBL. A continuación se muestra una leyenda gráfica de los pines anteriormente configurados:



\* - Indicates input pins. Held high with internal pull-up resistors.

Figura 60 Esquemático de control de los ejes de la CNC

Fuente: extraído de <http://blog.protoneer.co.nz/grbl-arduino-library/>

Como se puede apreciar, los drivers que soporta son los que disponen de señales STEP, DIR y ENABLE. Es importante destacar que los pines de ENABLE de todos los drivers son comunes y se conectan al pin 8 de Arduino. Este conexionado abarca todos los pines de que dispone el Arduino UNO. Obviamente los pines 1 y 2 correspondientes a transmisión Tx y Rx no se pueden emplear ya que Arduino está en todo momento comunicándose por el puerto serial. Los pines A4 y A5 están reservados y probablemente se les añada en un futuro nuevas funcionalidades. Existen multitud de formas de conexionado de los drivers. Una de ellas es conectar directamente los drivers de forma individual, como pueden ser los EasyDriver V4.4.

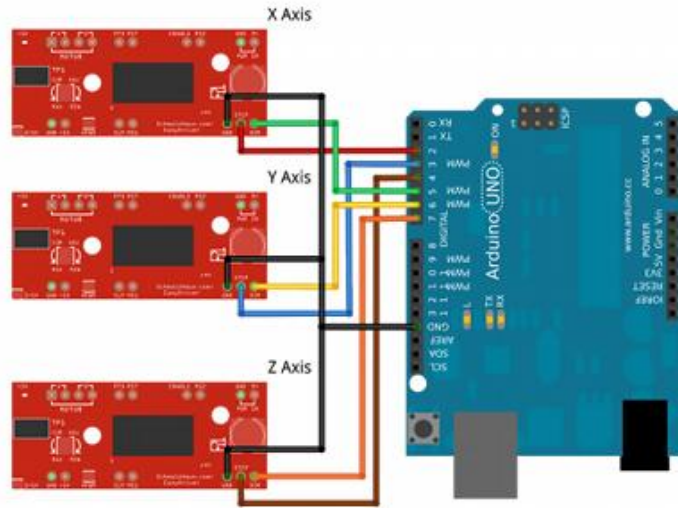


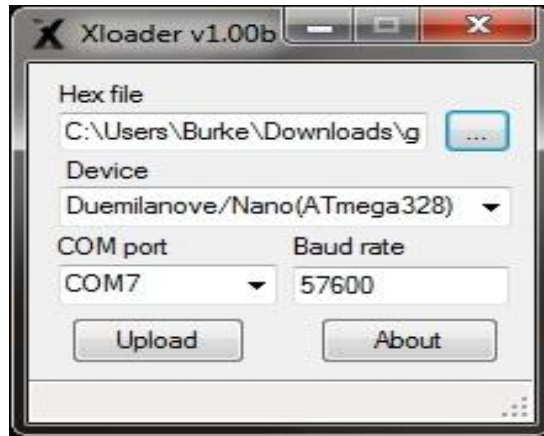
Figura 61 Conexión de los driver de los tres ejes

Fuente: extraído de <http://comeandmakeit.blogspot.com.co/2015/12/wiring-your-own-cnc-yes-it-is-that.html>

También se puede optar por conectar un sistema de control de motores embebido en una PCB, como pueden ser los proyectos de synthetos.com (grblShield o TinyG), pero lo más recomendable, y lo realizado en este proyecto ha sido el uso de un shield específicamente diseñado para GRBL. De este modo, se aprovechan todas las funcionalidades tanto de GRBL como de los drivers de una forma muy compacta. Más adelante se detallarán los aspectos de este shield.

### 6.3.2 Instalación del Sketch de GRBL para Windows.

El programa, o sketch de GRBL se encuentra dentro de un fichero hexadecimal, por lo que no se puede introducir directamente desde el IDE de Arduino al microcontrolador. Para realizarlo correctamente, se debe de utilizar un programa específico para flashear sobre el microcontrolador el firmware. En este proyecto, se ha empleado XLoader v1.00. No obstante, hay que tener presente que antes de flashear Arduino, los drivers de Arduino UNO tienen que estar instalados. Cuando se ejecuta XLoader, la pantalla se debe de configurar de la siguiente forma:



*Figura 62 Configuración de interfaz\_ordenador*

Fuente: extraído de <https://www.shapeoko.com/wiki/index.php/Grbl>

Hex file: directorio donde se encuentra el archivo de código fuente de GRBL en hexadecimal. v Device: GRBL solamente es compatible con el microcontrolador ATmega 328. Se recomienda flashear únicamente Arduino UNO, que obviamente dispone de este microcontrolador.

COM port: puerto donde está conectada la tarjeta de Arduino. v Baud rate: se recomienda una velocidad de 115200, de forma que la transferencia sea lo más rápida posible.

Una vez configurada la ventana, se hace click en “Upload”. Si el puerto está correctamente seleccionado GRBL comenzará a volcarse en la tarjeta y comenzará a parpadear. Una vez finalizado este proceso, XLoader no volverá a hacer falta. Es importante destacar que si el shield de GRBL se encuentra conectado a la tarjeta Arduino este proceso no se puede llevar a cabo, ya que los pines 0 y 1 (Tx y Rx) se deben de encontrar libres cuando se programa la tarjeta desde el ordenador.

## 6.4 software de control numérico

### 6.4.1 Interface máquina – ordenador.

Se trata del programa encargado de mostrar información sobre el estado de la máquina en una pantalla, así como permitir una comunicación bidireccional entre ambos.



Existen multitud de soluciones. La más famosa es sin duda Mach3 (Mach4 actualmente en desarrollo) totalmente compatible con cualquier versión desde Windows 2000. Para usuarios de Linux la más recomendable es EMC2, proyecto totalmente “Open Source”, aunque existen otras muchas como KCAM, solución perfecta para usuarios de Windows 98. La solución optada en este proyecto es GRBL, otro trabajo totalmente Open Source optimizado para trabajar sobre Arduino, soportado por una comunidad muy preparada que lo mejora día a día. Los autores de este software lo definen como: “Grbl is an opensource software that turns your Arduino development board into a full blown G-Code interpreter.” Para comunicar el controlador con el ordenador se ha optado por GRBL Controller 3.6.1, un software libre y gratuito producido por el maravilloso autor de Zapmaker.org. El programa puede ser descargado gratuitamente desde esa misma web. Su interfaz es la siguiente:

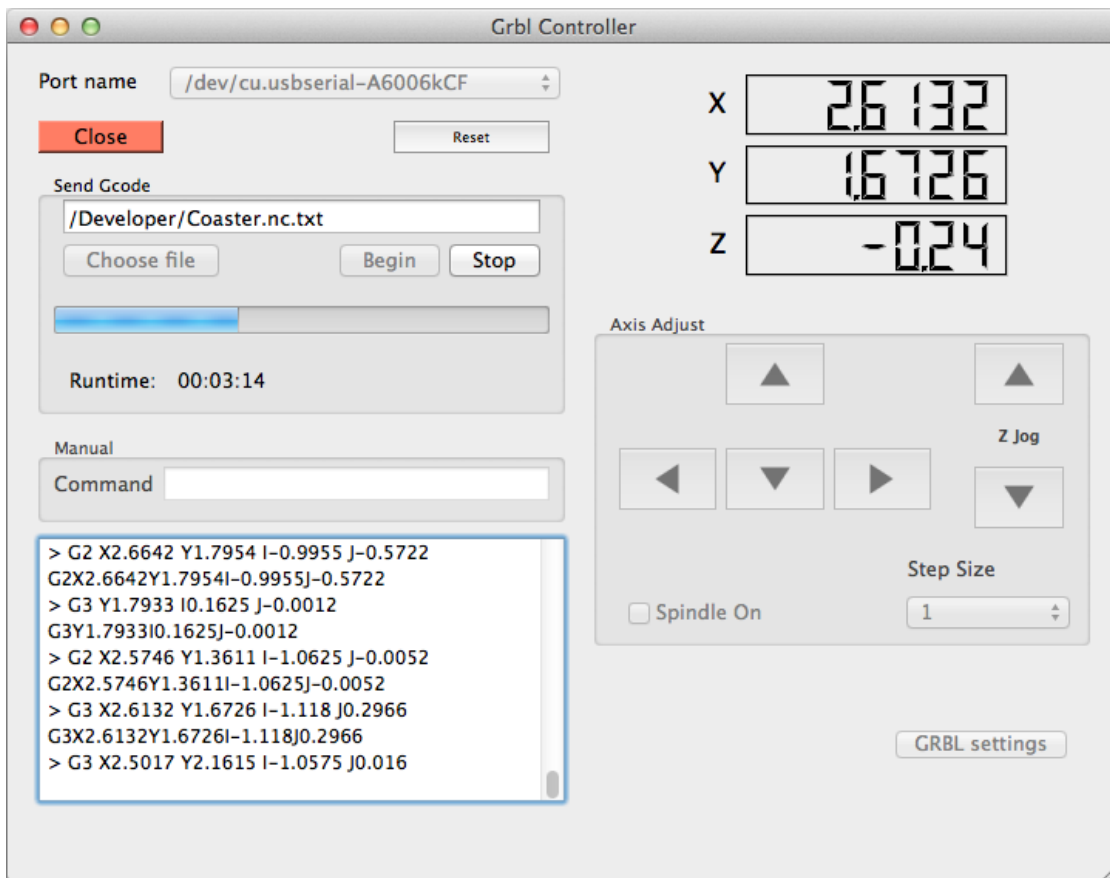


Figura 63 Interfaz del software de control

Fuente: extraído de <http://zapmaker.org/projects/grbl-controller-3-0/>

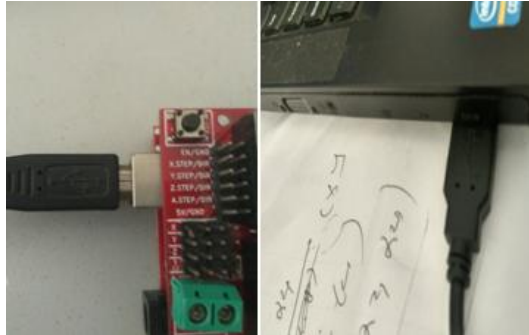
El programa empleado es GRBL Controller, totalmente compatible con GRBL. Se trata de un software diseñado para enviar código G a dispositivos de control numérico de 3 ejes. Aunque no es el más inteligente, dispone de herramientas más que suficientes para enviar a la máquina cualquier programa G previamente creado con post-procesadores TinyG por ejemplo. Este software está escrito usando las librerías multiplataforma de Qt, así como la librería QextSerialPort para simplificar el proceso para elegir el puerto serie USB correcto. Este desarrollo multiplataforma le permite trabajar sobre Windows, Mac y Linux. Este programa ha recibido multitud de actualizaciones en el último año, y continúa recibiendo. La última, fue lanzada tan solo un mes antes de comenzar a redactar el proyecto. A continuación se muestra una tabla con la compatibilidad de cada versión con las diferentes plataformas.

*Tabla 7*  
compatibilidad del software

COMPATIBILIDAD			
VERSION	WINDOW	MAC	LINUX
3.0	SI	SI	SI
3.2	SI	SI	X
3.3	SI	SI	X
3.3.1	SI	X	X
3.3.2	SI	SI	X
3.3.3	SI	X	X
3.3.4	SI	X	X
3.3.5	SI	X	X
3.3.9	SI	X	X
3.4	SI	X	X
3.4.2	SI	X	X
3.4.4	SI	SI	X
3.4.5	SI	X	X
3.4.5	SI	SI	X
3.5	SI	SI	X
3.5.1	SI	X	X
3.6.1	SI	X	X

#### 6.4.2 instalación del controlador.

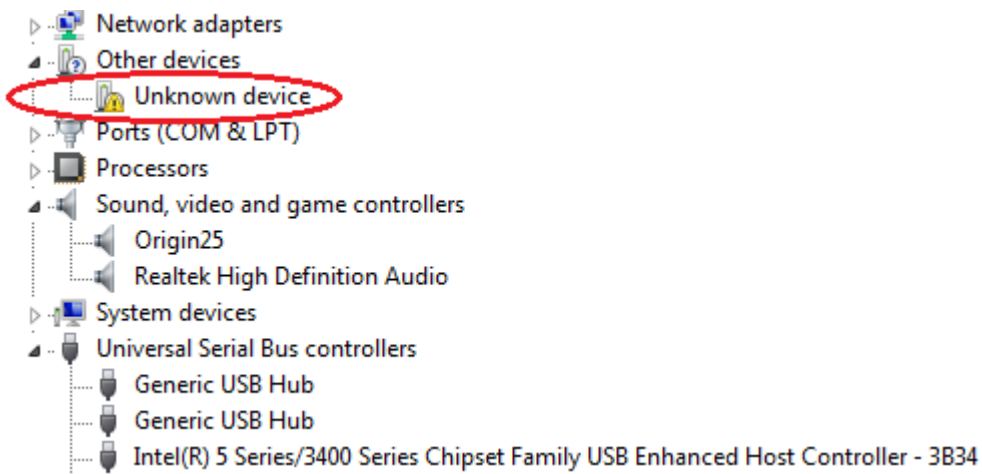
Se conecta el cable serial UBS del arduino al ordenador



*Figura 64* Conexión arduino- ordenador

Fuente: Foto tomada por el autor

En la primera instalación, el controlador necesita ser instalado, el programa de instalación en el actual paquete, el nombre del archivo HL340. Después de la instalación, el hardware se muestra debajo del cuadro rojo aparecerá en el ordenador. Esto indica que la unidad está instalada.



*Figura 65* Instalación del driver de comunicación

Fuente: extraído de: <https://support.native-instruments.com/hc/es/articles/210293205-Mi-ordenador-no-reconoce-un-dispositivo-USB-Windows->

Antes de que el controlador de clase se ha instalado: el controlador de dispositivo está instalado en el hardware de mi PC se pueden encontrar en el puerto correspondiente

Después de instalar el controlador, abrir el archivos GRBL Controller en el paquete o CONTROL GRBL

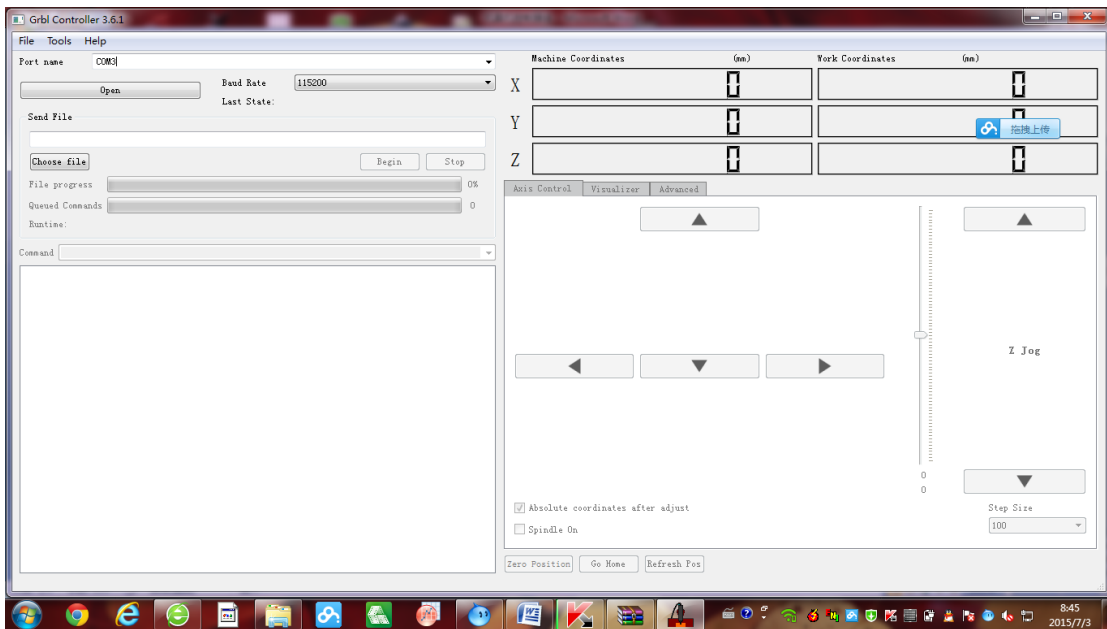


Figura 66 Software de control del código g CNC

Fuente: Foto tomada por el autor

En el mapa, seleccione los números de puerto correspondientes, visto en Mi PC y seleccione la velocidad de transmisión de 115.200, y haga clic en Abrir

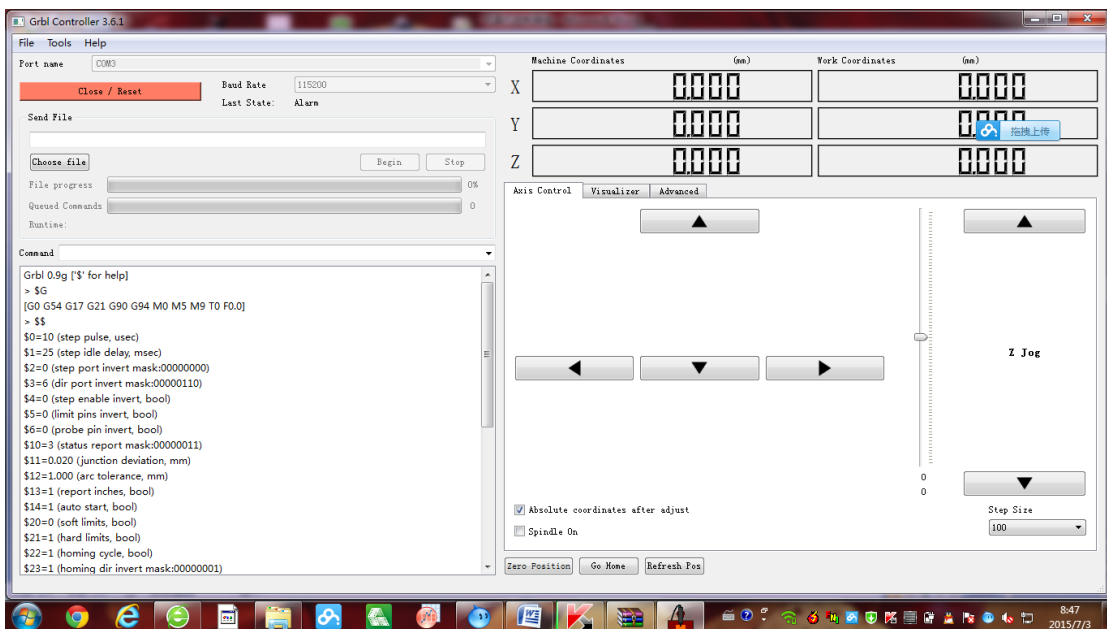


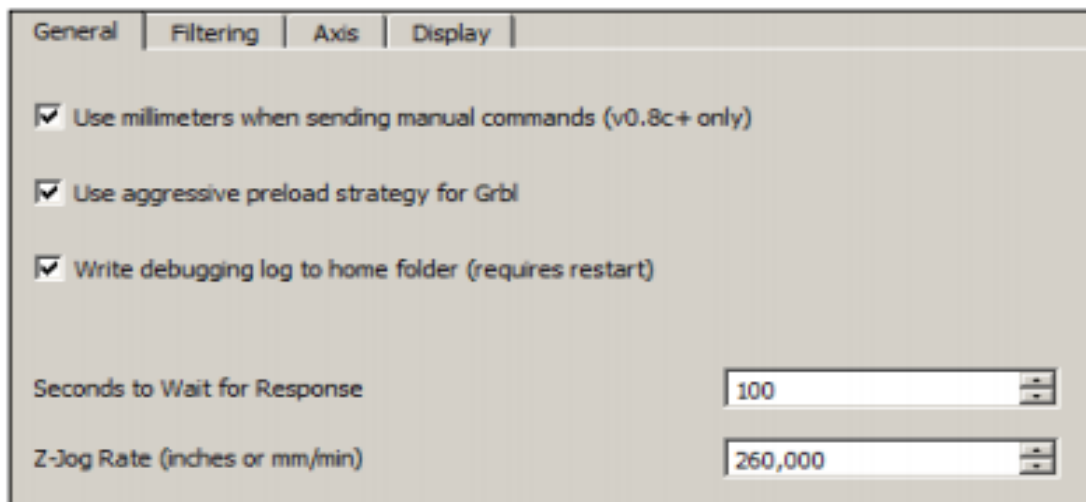
Figura 67 Vista de cómo debe aparecer el mensaje de comunicación

Fuente: Foto tomada por el autor

Si el mensaje que aparece en la parte inferior izquierda, aparece para indicar una comunicación exitosa.

Una vez que se ejecuta el programa, aparece una ventana sobre la que apenas se puede interactuar. Si el controlador está correctamente conectado, con el puerto serial correctamente asignado y con el sketch de GRBL introducido, al pinchar sobre “Open”, el programa accede a la programación del controlador, la carga.

En esta ventana se pueden apreciar cinco partes totalmente diferenciadas: Barra de herramientas: más adelante se explicarán todas las opciones de que dispone GRBL. Inicialización de GRBL: tras seleccionar la velocidad de comunicación en baudios, el puerto serial y conectar Arduino, al pulsar abrir el programa inicializa. Ventana de comandos y explorador de ficheros G: desde esta ventana se pueden enviar comandos a GRBL, así como cargar un programa de control numérico. También se muestran los comandos que se han ejecutado y en el caso de haberlas, las salidas que proporcionan esos comandos. Coordenadas de la máquina y de trabajo. Control manual. Visualizador. Acceso a configuración. Desde la pestaña “herramientas” de la barra de herramientas, se tiene acceso a cuatro opciones de configuración. La primera ventana es la de opciones generales y tiene este aspecto:



*Figura 68* Acceso a configuración

Fuente: extraído de <http://zapmaker.org/projects/grbl-controller-3-0/>

Si la primera opción no está seleccionada, las coordenadas que se van a mostrar en la ventana principal, estarán en pulgadas. Si está seleccionada, se mostrarán en milímetros. Si la carga agresiva de GRBL no está activada, GRBL va a esperar una respuesta del fichero después de ejecutar cada comando. Si está activada, el fichero va a enviar comandos hasta llenar el buffer de GRBL, y se va a encargar de que el buffer esté en todo momento lleno. Esto permite una mayor velocidad en la ejecución de las instrucciones, pero tiene un problema. Si se pulsa el botón de parada (dentro de GRBL Controller, no la seta de emergencia), la parada no es inmediata. Primero se deben de ejecutar todas las instrucciones que hay en el buffer hasta llegar a la orden de paro. La mejor solución para pararlo es simplemente pulsando la seta de emergencia en caso de que surjan problemas, y dejar de este modo la carga agresiva activada. La siguiente opción, escribe un fichero de depuración llamado GrblController.log en la carpeta por defecto del computador con todos los mensajes de diagnóstico y estado. Este archivo se borra siempre GRBL Controller se reinicia. Permite depurar problemas si ocurre algo extraño, e incluso se puede enviar a la web del autor para que analice lo que ha ocurrido y solucionar fallos en un futuro. La siguiente opción “segundos de espera hasta respuesta” se encarga de abortar un trabajo con un fichero si GRBL no responde tras el tiempo configurado. Es conveniente que este tiempo sea relativamente grande en el caso de realizar trayectorias largas. Por último, “Z-Jog Rate” es la velocidad a la que se va a mover el eje Z cuando se acciona mediante el slider de la ventana principal llamado Z Jog. La siguiente pestaña, llamada filtro, es la siguiente:

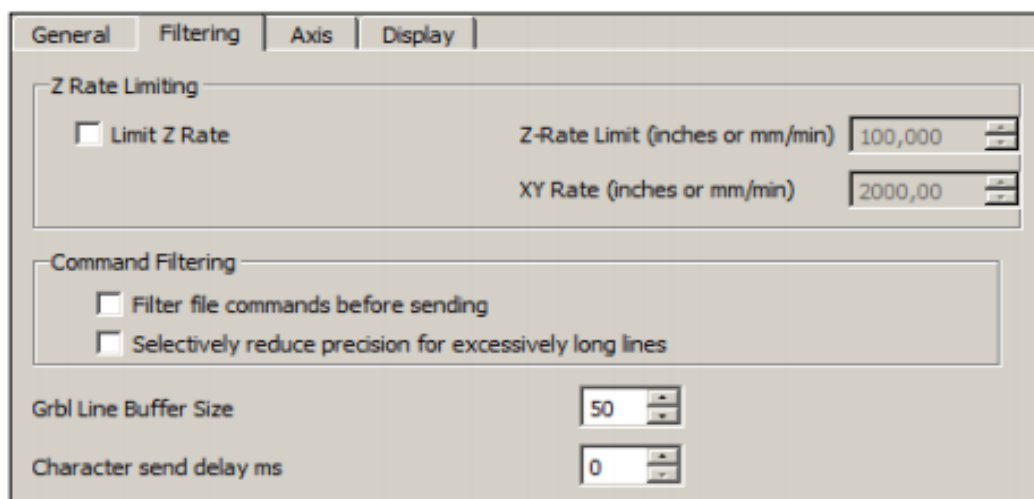


Figura 69 Configuración del eje z

Fuente: extraído de <http://zapmaker.org/projects/grbl-controller-3-0/>

Un valor muy importante es la longitud del buffer que va a almacenar cada línea de código G. Si la línea de código G contiene demasiados números decimales y el buffer se llena, entonces la máquina se va a comportar de una forma indeseable. Por ejemplo, es probable que esta línea genere problemas:

```
G01 x2.35484876666594 y1.0021848787554 z5.54545775454521
```

Figura 70 Línea de código g

Fuente: extraído de <http://zapmaker.org/projects/grbl-controller-3-0/>

Es habitual que programas de CAM generen este tipo de ficheros de control numérico por defecto. Otorgan una sensación falsa de precisión pues jamás la máquina va a ser capaz de efectuar movimientos discretos de millonésimas de milímetro. Lo único que va a hacer es llenar el buffer de GRBL y provocar problemas. Lo adecuado es aumentar el buffer de GRBL en su máximo valor (70), y restringir el programa de CAM a no generar más de cuatro cifras después del decimal.

La tercera pestaña, tiene que ver con los ejes:

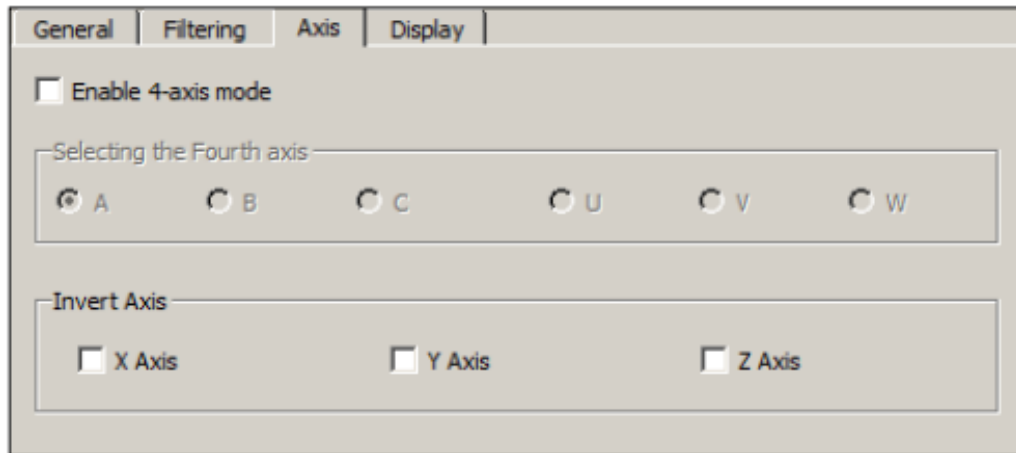
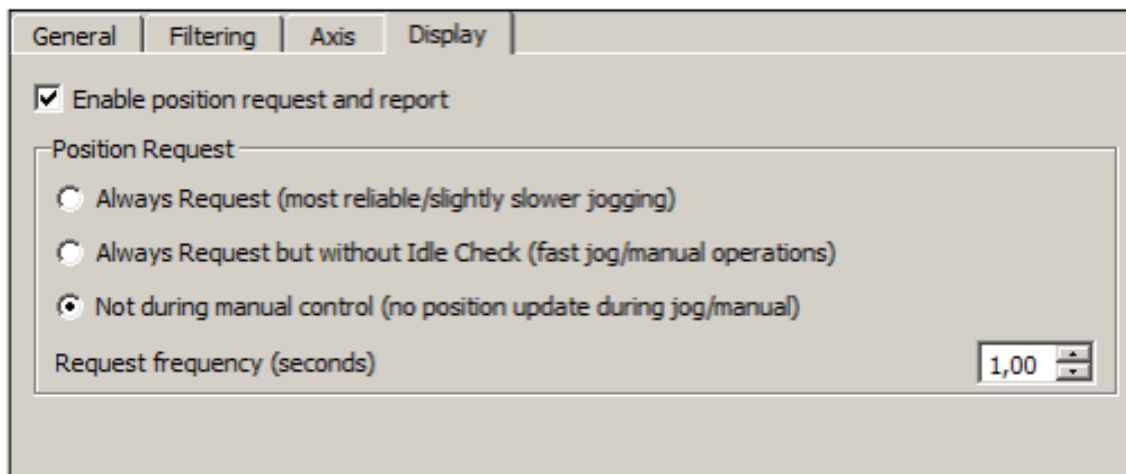


Figura 71 precisión de los ejes X, Y, Z

Fuente: extraído de <http://zapmaker.org/projects/grbl-controller-3-0/>

En este proyecto no se ha empleado el cuarto eje, por lo que realmente no se dispone de demasiada información acerca de la configuración de este eje. No obstante, como ya se ha dicho, GRBL no dispone de soporte para interpolar en más de tres ejes. La opción de invertir

ejes, además de invertir el funcionamiento del pin DIR de los drivers, invierte el sentido cuando se lee un fichero G. Si por ejemplo al presionar el botón Y+ en modo manual la máquina avanza hacia Y-, se recomienda no invertir desde esta ventana la dirección del eje Y, y hacerlo desde una máscara diseñada específicamente, para esto y que se explicará en el apartado La última pestaña es la de visualización:



*Figura 72 Ajuste de direccionamiento del eje Y*

Fuente: extraído de <http://zapmaker.org/projects/grbl-controller-3-0/>

Simplemente permite que GRBL envíe información a cerca de donde se encuentra la máquina (donde cree él que está) y se muestre en los displays de la ventana principal. Cuanto menor sea el tiempo de refresco, más carga va a tener GRBL y más lento va a ir la ejecución del programa. Con el fin de optimizar este proceso, en la última versión de GRBL Controller se ha habilitado una tercera opción que hace que no se actualice la posición al realizar un movimiento manual (se muestra activa en la captura de pantalla anterior). Esto reduce enormemente la carga de GRBL.

Existen algunos botones en el programa que al pulsarlos realizan acciones que también se pueden realizar mediante comandos. Esto simplifica y permite un flujo de trabajo más cómodo con la máquina.

### 6.4.3 Configuración de GRBL Controller.

La configuración de GRBL adaptada a esta máquina es la siguiente:



```
$0=1000 (x, step/mm)
$1=1000 (y, step/mm)
$2=1000 (z, step/mm)
$3=100 (step pulse, usec)
$4=240 (default feed, mm/min)
$5=250 (default seek, mm/min)
$6=192 (step port invert mask, int:00011100)
$7=50 (step idle delay, msec)
$8=5 (acceleration, mm/sec^2)
$9=0.050 (junction deviation, mm)
$10=0.100 (arc, mm/segment)
$11=25 (n-arc correction, int)
$12=3 (n-decimals, int)
$13=0 (report inches, bool)
$14=1 (auto start, bool)
$15=0 (invert step enable, bool)
$16=1 (hard limits, bool)
$17=0 (homing cycle, bool)
$18=0 (homing dir invert mask, int:00000000)
$19=240 (homing feed, mm/min)
$20=250.000 (homing seek, mm/min)
$21=100 (homing debounce, msec)
$22=1.000 (homing pull-off, mm)
```

*Figura 73 Configuración GRBL*

Fuente: extraído de <http://zapmaker.org/projects/grbl-controller-3-0/>

## **7. Conclusiones**

Al construir la estructura y el desplazamiento de los ejes, se pudo obtener una mayor estabilidad, lo que le permite a la CNC, un Óptimo funcionamiento

Equipando el Sistema de control de potencia, se logró suministrarle la energía suficiente a la CNC, con la cual le permite darle el Funcionamiento a los motores paso a paso, y al encendido del motor de fresado.

Se logró adaptarle un sistema de comunicaciones que pudo comunicar la CNC con el ordenador, logrando así mover los ejes de la CNC por medio de un computador.

Al lograr implementar un software, se pudo leer datos numéricos que se convirtieron en movimientos exactos, para el óptimo manejo del fresado.

Fue posible implementar la CNC con hardware y software libre, logrando una CNC funcional con bajo costo y con materiales muy económicos y software gratuito que no necesitan licencia para su funcionamiento e implantación. Se logró con el desarrollo de la CNC, equipar el laboratorio de mecatrónica, para que los estudiantes y profesores se beneficien de la CNC.

## **8. Recomendaciones**

Este proyecto a futuro puede tener mejoras en el motor de fresado, al ser cambiado por sistema de fresado, de corte o de láser. Se pueden remplazar las piezas de acrílico por aluminios para mayor resistencia y estabilidad de la CNC

## 8. Referencias bibliográficas

### Referencias

- arduino*. (22 de 05 de 2017). Obtenido de <http://arduino.cl/que-es-arduino/>
- buenastareas*. (25 de 11 de 2011). *buenas tareas*. Recuperado el 2017, de <http://www.buenastareas.com/ensayos/Torno-Cnc-Marco-Teorico/3179348.html>
- cadcamcae*. (22 de 05 de 2017). *cadcamcae*. Obtenido de <https://cadcamcae.wordpress.com/2007/06/14/el-control-numericopor-computadora-el-cnc/>
- github*. (14 de 05 de 2017). *github*. Obtenido de <https://github.com/grbl/grbl/wiki/Connecting-Grbl>
- Slideshare*. (10 de 03 de 2017). *Slideshare*. Obtenido de <https://es.slideshare.net/guest79198c/unidad-iii-1112311>
- softwarecadcam*. (s.f.). *softwarecadcam*. Recuperado el 15 de 10 de 2016, de *softwarecadcam*: [http://www.softwarecadcam.com/lpn2\\_Programacion\\_CNC.html?gclid=CIHw7Kmks9ACFYJahgodCDcD8w](http://www.softwarecadcam.com/lpn2_Programacion_CNC.html?gclid=CIHw7Kmks9ACFYJahgodCDcD8w)
- taringa*. (10 de 04 de 2017). *taringa*. Obtenido de <http://www.taringa.net/posts/info/7116000/Torno-CNC.html>
- Universitat Pompeu Fabra. (10 de 04 de 2017). *dtic*. Obtenido de <http://www.dtic.upf.edu/~jlozano/interfaces/interfaces8.html>

## **10. Bibliografía**

<http://www.libreriaproteo.com/libro-380698-TEORIA-Y-PROBLEMAS-RESUELTOS-EN-PROGRAMACION-CONTROL-NUMERICO.html>

<http://www.libreriaproteo.com/libro-350636-CONTROL-NUMERICO-Y-PROGRAMACION-SISTEMAS-DE-FABRICACION-DE-MAQUINAS-A.html>

## **11. Anexos**